IBM WebSphere eXtreme Scale
Version 8.6

Guide d'administration

Novembre 2012

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- http://www.fr.ibm.com (serveur IBM en France)
- http://www.can.ibm.com (serveur IBM au Canada)
- http://www.ibm.com (serveur IBM aux Etats-Unis)

Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex

Table des matières

Figures

Tableaux

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
(Pos1)		Home
Fin	Fin	End
(PgAr)		PgUp
(PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
(Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos du *Guide d'administration*

La documentation de WebSphere eXtreme Scale inclut trois volumes qui fournissent les informations nécessaires pour utiliser, programmer et administrer le produit WebSphere eXtreme Scale.

Bibliothèque WebSphere eXtreme Scale

La bibliothèque WebSphere eXtreme Scale contient les documents suivants :

- La *Présentation du produit* contient une vue de haut niveau des concepts de WebSphere eXtreme Scale, avec des scénarios d'utilisation et des tutoriels.
- Le *Guide d'installation* explique comment installer les topologies communes de WebSphere eXtreme Scale.
- Le *Guide d'administration* contient les informations nécessaires pour les administrateurs système et explique notamment comment planifier les déploiements d'application, planifier la capacité, installer et configurer le produit, démarrer et arrêter des serveurs, surveiller l'environnement et le sécuriser.
- Le *Guide de programmation* contient des informations destinées aux développeurs d'applications, sur la manière de développer des applications pour WebSphere eXtreme Scale à l'aide des informations d'API incluses.

Pour télécharger les documents, accédez à la page de la bibliothèque de WebSphere eXtreme Scale.

Vous pouvez également accéder aux mêmes informations dans cette bibliothèque dans le .

Utilisation hors ligne des manuels

Tous les manuels de la bibliothèque WebSphere eXtreme Scale contiennent des liens vers le centre de documentation, avec l'URL racine suivante : . Ces liens vous permettent d'accéder directement aux informations associées. Toutefois, si vous travaillez hors ligne et rencontrez l'une de ces liens, vous pouvez rechercher le titre du lien dans les autres manuels dans la bibliothèque. La documentation d'API, le glossaire et les références des messages ne sont pas disponibles dans les manuels PDF.

A qui s'adresse ce document

Ce document est principalement destiné aux administrateurs système, administrateurs de la sécurité et opérateurs système.

Obtention des mises à jour de ce document

Vous pouvez obtenir les mises à jour de ce document en téléchargeant la version la plus récente à partir de la page de la bibliothèque de WebSphere eXtreme Scale.

Comment envoyer vos commentaires

Contactez l'équipe chargée de la documentation. Avez-vous trouvé ce que vous recherchiez ? Ces informations étaient-elles précises et complètes ? Envoyez vos

commentaires sur cette documentation par courrier électronique, à l'adresse wasdoc@us.ibm.com.

Chapitre 1. Mise en route

Après avoir installé le produit, vous pouvez utiliser l'exemple de mise en route pour tester l'installation et utiliser le produit pour la première fois.

Tutoriel : Démarrer avec WebSphere eXtreme Scale

Après avoir installé WebSphere eXtreme Scale dans un environnement autonome, vous pouvez utiliser l'exemple d'application de l'initiation pour vérifier l'installation. Cet exemple présente les grilles de données en mémoire et d'entreprise. L'exemple d'application de démarrage n'est incluse que dans les installations complètes (client et serveur) de WebSphere eXtreme Scale.

Objectifs d'apprentissage

- Description du fichier XML descripteur ObjectGrid et des fichiers XML descripteurs de règles de déploiement que vous utilisez pour configurer l'environnement.
- Démarrage des serveurs de catalogue et de conteneur à l'aide des fichiers de configuration.
- Description du développement d'une application client dans les langages de programmation Java ou .NET. Apprenez à interopérer entre les langages de programmation en créant une grille de données d'entreprise.
- Exécution de l'application client pour insérer des données dans la grille de données.
- Surveillance des grilles de données avec la console Web.

Durée

60 minutes

Leçon 1.1 du tutoriel d'initialisation : Définition de grilles de données avec des fichiers de configuration

Les fichiers `objectgrid.xml` et `deployment.xml` sont nécessaires pour démarrer les serveurs de conteneur.

L'exemple utilise les fichiers `objectgrid.xml` et `deployment.xml` qui se trouvent dans le répertoire *racine_install_wxs*`/ObjectGrid/gettingstarted/server/config`. Ces fichiers sont envoyés aux commandes de démarrage pour démarrer les serveurs de conteneur et un serveur de catalogue. Le fichier `objectgrid.xml` est le fichier XML descripteur d'ObjectGrid. Le fichier `deployment.xml` est le fichier XML descripteur de la stratégie de déploiement ObjectGrid. Ensemble, ces fichiers définissent une topologie répartie.

Fichier XML descripteur d'ObjectGrid

Un fichier XML de descripteur d'ObjectGrid permet de définir la structure de la grille d'objets utilisée par l'application. Il contient la liste des configurations de mappes de sauvegarde. Ces mappes de sauvegarde stockent les données en cache. L'exemple suivant présente un fichier d'exemple `objectgrid.xml`. Les premières

lignes de ce fichier incluent l'en-tête requis de chaque fichier XML ObjectGrid. Cet exemple de fichier définit l'ObjectGrid `Grid` avec les mappes de sauvegarde `Map1` et `Map2`.

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
 <objectGrids>
  <objectGrid name="Grid" txTimeout="30">
   <backingMap name="Map1" copyMode="COPY_TO_BYTES"  lockStrategy="PESSIMISTIC"
    nullValuesSupported="false"/>
   <backingMap name="Map2" copyMode="COPY_TO_BYTES"  lockStrategy="PESSIMISTIC"
    nullValuesSupported="false"/>
  </objectGrid>
 </objectGrids>
</objectGridConfig>
```

8.6+

- La valeur **txtTmeout** `30` secondes est une valeur adaptée pour la plupart des grilles de données.
- La valeur **copyMode** `COPY_TO_BYTES` est nécessaire lorsque vous ne fournissez pas de classe d'objets de sérialisation.
- La valeur **lockStrategy** `PESSIMISTIC` est une stratégie de verrouillage adaptée lorsque vous commencez par développer l'application de grille de données. Avec cette stratégie, vous n'utilisez pas de cache local ni de plug-in de chargeur. L'application ne gère pas les problèmes de verrouillage.
- La valeur **nullValuesSupported** `false` élimine le problème qui peut apparaître lorsque vous extrayez une valeur null d'une clé. Dans ce cas, vous ne savez pas si la clé existait. Vous pouvez éliminer ce problème en n'allouant pas de valeur null dans le mappage de de sauvegarde.

Fichier XML du descripteur de la règle de déploiement

Le fichier XML de descripteur de règle de déploiement est conçu pour être couplé avec le fichier XML d'ObjectGrid correspondant, le fichier `objectgrid.xml`. Dans l'exemple suivant, les premières lignes du fichier `deployment.xml` incluent l'en-tête requis de chaque fichier XML de règle de déploiement. Le fichier définit l'élément **objectgridDeployment** de grille d'objets `Grid` défini dans le fichier `objectgrid.xml`. Les mappes de sauvegarde `Map1` et `Map2` définies dans la grille d'objets `Grid` sont incluses dans un ensemble de mappes `mapSet`.

```
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="13" minSyncReplicas="0"
     maxSyncReplicas="1" >
            <map ref="Map1"/>
            <map ref="Map2"/>
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

L'attribut **numberOfPartitions** de l'élément **mapSet** spécifie le nombre des partitions de l'ensemble de mappes. Cet attribut est facultatif. La valeur est 1. La valeur d'attribut doit être adaptée à la capacité prévue de la grille de données.

L'attribut **minSyncReplicas** de l'élément **mapSet** définit le nombre minimal de répliques synchrones pour chaque partition dans l'ensemble de mappes. Cet attribut est facultatif. La valeur par défaut est `0`. Les fragments primaires et réplique ne sont pas placés tant que le domaine de service de catalogue peut prendre en charge le nombre minimal de répliques synchrones. Pour prendre en charge la valeur **minSyncReplicas**, vous devez utiliser un ou plusieurs serveurs à la place de la valeur de l'attribut **minSyncReplicas**. Si le nombre de répliques synchrones tombe en dessous de la valeur de l'attribut **minSyncReplicas**, les transactions d'écriture ne sont plus autorisées dans la partition.

L'attribut **maxSyncReplicas** de l'élément **mapSet** définit le nombre maximal de répliques synchrones pour chaque partition dans l'ensemble de mappes. Cet attribut est facultatif ; la valeur par défaut est `0`. Aucune autre réplique synchrone n'est placée pour une partition lorsqu'un domaine de service de catalogue atteint ce nombre de répliques synchrones pour la partition. L'ajout de serveurs de conteneur qui prennent en charge cette grille d'objets peut accroître le nombre de répliques synchrones si la valeur **maxSyncReplicas** n'est pas déjà atteinte. L'exemple affecte à **maxSyncReplicas** la valeur 1, ce qui implique que le domaine de service de catalogue place au plus une réplique synchrone. Si vous démarrez plusieurs serveurs de conteneur, une seule réplique synchrone est placée dans l'une des instances de serveur de conteneur.

Point de contrôle de la leçon

Dans cette leçon, vous avez appris à :

- Définir des mappes qui stockent les données dans le fichier XML de descripteur d'ObjectGrid.
- Utiliser le fichier XML descripteur de déploiement pour définir le nombre de partitions et de répliques de la grille de données.

Module 2 du guide d'initiation : Création d'une application client

Ecrire des applications client pour insérer, mettre à jour, supprimer et extraire des données depuis la grille de données. Vous pouvez utiliser l'exemple d'application pour apprendre à créer une application pour votre environnement.

Objectifs d'apprentissage

A la fin de ce module, vous saurez :

- Java Développer une application client Java
- .NET 8.6+ Développer une application client .NET

Leçon 2.1 du tutoriel d'initiation : Création d'une application client Java

Java

Pour pouvoir insérer, supprimer, mettre à jour et extraire des données dans votre grille de données, vous devez écrire une application client. L'exemple d'initiation inclut une application client Java que vous pouvez utiliser pour en savoir plus sur la création de votre propre application client.

Le fichier `Client.java` dans le répertoire *racine_install_wxs*`/ObjectGrid/gettingstarted/client/src/` est le programme client qui montre comment se

connecter à un serveur de catalogue, obtenir l'instance ObjectGrid et utiliser l'API ObjectMap. L'API ObjectMap stocke les données comme paires clé-valeur et elle est idéale pour la mise en cache d'objets qui n'ont aucune relation. Les étapes suivantes présentent le contenu du fichier `Client.java`.

Si devez mettre en cache des objets qui ont des relations, utilisez l'API EntityManager.

1. Connectez-vous au service de catalogue en obtenant une instance ClientClusterContext.

   Pour établir la connexion au serveur de catalogues, utilisez la méthode connect de l'API ObjectGridManager. La méthode connect utilisée requiert seulement un noeud final de serveur de catalogue au format *nom_hôte*:*port*. Vous pouvez indiquer plusieurs noeuds finals de serveur de catalogue en séparant les valeurs *hostname*:*port* par une virgule. Le fragment de code suivant montre comment se connecter à un serveur de catalogue et obtenir une instance ClientClusterContext : **8.6+**

   ```
   ClientClusterContext ccc = ObjectGridManagerFactory.getObjectGridManager().connect(cep, null, null);
   ```

   Si les connexions aux serveurs de catalogue aboutissent, la méthode connect retourne une instance ClientClusterContext. L'instance ClientClusterContext est nécessaire pour obtenir la grille d'objets ObjectGrid depuis l'API ObjectGridManager.

2. Obtenez une instance ObjectGrid.

   Pour obtenir une instance ObjectGrid, utilisez la méthode getObjectGrid de l'API ObjectGridManager. La méthode getObjectGrid requiert l'instance ClientClusterContext et le nom de l'instance de grille de données. L'instance ClientClusterContext est obtenue pendant la connexion au serveur de catalogue. Le nom de l'instance ObjectGrid est `Grid` ; ce nom est spécifié dans le fichier `objectgrid.xml`. Le fragment de code suivant montre comment obtenir la grille de données en appelant la méthode getObjectGrid de l'API ObjectGridManager.

   ```
   ObjectGrid grid = ObjectGridManagerFactory.getObjectGridManager().getObjectGrid(ccc, "Grid");
   ```

3. Obtenez une instance Session.

   Vous pouvez obtenir une session de l'instance ObjectGrid obtenue. Une instance Session est indispensable pour obtenir l'instance ObjectMap et pour effectuer une démarcation de transaction. Le fragment de code suivant montre comment obtenir une instance Session en appelant la méthode getSession de l'API ObjectGrid.

   ```
   Session sess = grid.getSession();
   ```

4. Obtenez une instance ObjectMap.

   Après avoir obtenu une instance Session, vous pouvez obtenir une instance ObjectMap depuis une instance Session en appelant la méthode getMap de l'API Session. Vous devez transmettre le nom de la mappe comme paramètre à la méthode getMap pour obtenir l'instance ObjectMap. Le fragment de code suivant montre comment obtenir ObjectMap en appelant la méthode getMap de l'API Session.

   8.6+

   ```
   ObjectMap map1 = sess.getMap(mapName);
   ```

5. Utilisez les méthodes ObjectMap.

   Une fois une instance ObjectMap obtenue, vous pouvez utiliser l'API ObjectMap. N'oubliez pas que l'interface ObjectMap est une mappe transactionnelle et qu'elle requiert une démarcation de transaction à l'aide des

méthodes begin et commit de l'API Session. Faute de démarcation de transaction explicite, les opérations ObjectMap s'exécutent avec des transactions de validation automatique.

- Le fragment de code suivant montre comment utiliser l'API ObjectMap avec une transaction de validation automatique.

  8.6+

  ```
  map1.insert(key1, value1);
  ```

- **8.6+** Vous pouvez exécuter une transaction sur une seule partition à la fois ou sur plusieurs partitions. Pour exécuter une transaction sur une seule partition, utilisez une transaction de validation en une phase :

  ```
  sess.setTxCommitProtocol(TxCommitProtocol.ONEPHASE);
  sess.begin();
  map1.insert(k, v);
  sess.commit();
  ```

  Pour exécuter une transaction sur plusieurs partitions, utilisez une transaction de validation en deux phases :

  ```
  sess.setTxCommitProtocol(TxCommitProtocol.TWOPHASE);
  sess.begin();
  map1.insert(k, v);
  sess.commit();
  ```

6. Facultatif : Fermez la session. Une fois toutes les opérations Session et ObjectMap terminées, fermez la session à l'aide de la méthode Session.close(). Cette méthode renvoie les ressources qui étaient utilisées par la session.

   ```
   sess.close();
   ```

   Par conséquent, les appels suivants de la méthode getSession() sont plus rapides, et moins d'objets Session se trouvent dans le segment.

Point de contrôle de la leçon :

Dans cette leçon, vous avez appris à créer une application client simple pour effectuer des opérations de grille de données.

Leçon 2.2 du tutoriel d'initiation : Création d'une application .NET

.NET

Pour pouvoir insérer, supprimer, mettre à jour et extraire des données dans votre grille de données, vous devez écrire une application client. L'exemple du tutoriel d'initiation contient une application client .NET que vous pouvez utiliser pour apprendre à créer votre propre application client.

- Vous devez avoir installé WebSphere eXtreme Scale Client pour .NET. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale Client for .NET», à la page 202.
- Le fichier de projet de l'exemple fonctionne avec Microsoft Visual Studio 2010 et les versions suivantes. Si vous utilisez une version précédente de Microsoft Visual Studio, vous devez créer votre propre fichier de projet.

Vous pouvez utiliser l'exemple d'application .NET du tutoriel pour :

- Vérifier que vous avez installé correctement WebSphere eXtreme Scale Client for .NET.
- Apprendre à écrire des applications pour le client .NET qui communiquent avec la grille de données pour créer des applications personnalisées. L'exemple

montre comment se connecter à une grille de données sur un serveur de catalogue distant. Le mode interactif montre comment exécuter des transactions manuelles en utilisant la mappe GridMapPessimisticTx. Le mode de ligne de commande montre des transactions validées automatiquement avec la mappe GridMapPessimisticAutoTx.

- Apprendre à interagir avec l'exemple du tutoriel d'initiation iJava. Les deux exemples d'applications stockent les éléments dans la grille de données avec les paires TestKey/TestValue. L'exemple .NET contient les attributs ClassAlias et FieldAlias pour créer des identificateurs uniques pour la sérialisation et la désérialisation. Si une opération d'insertion de clé est exécutée depuis l'application client Java, le client .NET peut obtenir la valeur en exécutant une opération get sur la clé insérée.

L'exemple d'application du tutoriel d'initiation .NET a les limitations suivantes :

- Seul le verrouillage pessimiste est pris en charge.
- Les opérations de validation en deux phases sont prises en charge. Vous pouvez valider les opérations dans une seule partition. Si vous exécutez une validation qui implique plusieurs partitions, une exception MultiplePartitionWriteException est générée.
- L'exemple ne prend pas en charge les valeurs null. L'API .NET autorise les valeurs null, mais vous devez utiliser des types pouvant avoir la valeur "null".

Le fichier de projet `SimpleClient.csproj` se trouve dans le répertoire *net_client_home*`/sample/SimpleClient`. Ce fichier de projet est le programme client qui montre comment connecter un serveur de catalogue, obtenir l'instance ObjectGrid et utiliser l'API ObjectMap. L'API ObjectMap stocke les données comme paires clé-valeur et elle est idéale pour la mise en cache d'objets qui n'ont aucune relation. Les étapes suivantes contiennent des informations sur le contenu de clé du fichier `SimpleClient.csproj`. Vous pouvez également consulter le fichier de projet plus en détail dans Microsoft Visual Studio.

Le tutoriel montre comment utiliser IGridMapPessimisticTx qui est la mappe de transactions manuelles utilisée lorsque l'application est exécutée en mode interactif. Si vous utilisez l'application en mode de ligne de commande, la mappe IGridMapPessimisticAutoTx est utilisée.

1. Connectez-vous au service de catalogue en obtenant une instance IClientConnectionContext.

   Pour vous connecter au serveur de catalogue, utilisez la méthode Connect de l'API IGridManager.

   ```
   IGridManager gm = GridManagerFactory.GetGridManager( );
   ICatalogDomainInfo cdi = gm.CatalogDomainManager.CreateCatalogDomainInfo( endpoint );
   ccc = gm.Connect( cdi, "SimpleClient.properties" );
   ```

   Si la connexion au serveur de catalogue aboutit, la méthode Connect retourne une instance IClientConnectionContext. L'instance IClientConnectionContext est nécessaire pour obtenir la grille de données de l'API IGridManager.

2. Obtenez une instance ObjectGrid.

   Pour obtenir une instance ObjectGrid, utilisez la méthode GetGrid de l'API IGridManager. La méthode GetGrid nécessite l'instance IClientConnectionContext et le nom de l'instance de grille de données. L'instance IClientConnectionContext est obtenue pendant la connexion au serveur de catalogue. Le nom de l'instance de grille de commande est la grille définie dans le fichier `objectgrid.xml`.

   ```
   grid = gm.GetGrid( ccc, gridName );
   ```

3. Obtenez une instance de mappe.

Vous pouvez obtenir une instance de mappe en appelant la méthode GetGridMapPessimisticTx de l'API IGrid. Envoyez le nom de la mappe comme paramètre à la méthode GetGridMapPessimisticTx pour obtenir l'instance de mappe.

```
pessMap = grid.GetGridMapPessimisticTx<Object, Object>( mapName );
```

4. Utilisez les méthodes IGridMapPessimisticTx.

   Une fois une instance de mappe obtenue, vous pouvez utiliser l'API IGridMapPessimisticTx.

   Le fragment de code suivant montre comment utiliser l'API IGridMapPessimisticTx.

   - Pour lancer une transaction avec l'API IGridMapPessimisticTx, vous devez appeler la méthode map.Transaction.Begin(). Cette méthode lance une nouvelle transaction dans laquelle vous pouvez exécuter des opérations.

     ```
     case "begin":
        map.Transaction.Begin( );
        return 0;
     ```

   - La méthode add insère une nouvelle paire clé/valeur. Si la clé existe, une exception est émise.

     ```
      case "a":
        if( key == null ) throw new MissingParameterException( "key" );
        if( value == null ) throw new MissingParameterException( "value" );
        map.Add( key, value );
        Console.WriteLine( "SUCCESS: Added key '{0}' with value '{1}',
        partitionId={2}", key, value, partitionId );
        return 0;
     ```

   - La méthode put insère ou met à jour une paire clé/valeur.

     ```
     case "p":
       if( key == null ) throw new MissingParameterException( "key" );
       if( value == null ) throw new MissingParameterException( "value" );
       map.Put( key, value );
       Console.WriteLine( "SUCCESS: Put key '{0}' with value '{1}',
       partitionId={2}", key, value, partitionId );
       return 0;
     ```

   - La méthode replace remplace une paire clé/valeur existante. Si l'élément n'est pas présent, une exception est émise.

     ```
     case "r":
       if( key == null ) throw new MissingParameterException( "key" );
       if( value == null ) throw new MissingParameterException( "value" );
       map.Replace( key, value );
       Console.WriteLine( "SUCCESS: Replaced key '{0}' with value '{1}',
       partitionId={2}", key, value, partitionId );
       return 0;
     ```

   - La méthode remove supprime une paire clé/valeur.

     ```
     case "d":
       if( key == null ) throw new MissingParameterException( "key" );
       map.Remove( key );
       Console.WriteLine( "SUCCESS: Deleted value with key '{0}',
       partitionId={1}", key, partitionId );
        return 0;
     ```

   - La méthode get extrait la valeur de la clé.

     ```
      case "g":
       if( key == null ) throw new MissingParameterException( "key" );
       value = ( TestValue )map.Get( key );
       if ( value != null )
       {
         Console.WriteLine( "SUCCESS: Value is '{0}',
       partitionId={1}", value, partitionId );
       }
     ```

```
else
{
  Console.WriteLine( "FAILED: Key not found" );
}
return 0;
```

- Si vous voulez annuler les opérations que vous avez exécutées dans l'opération avant la validation, utilisez la méthode rollback.

```
case "rollback":
  map.Transaction.Rollback( );
  return 0;
```

- La méthode commit valide les opérations exécutées dans la transaction.

```
case "commit":
  map.Transaction.Commit( );
  return 0;
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez appris à créer une application .NET simple pour exécuter des opérations de grille de données.

Leçon 2.3 : Création d'une application de grille de données

Pour créer une application de grille de données dans laquelle les clients Java et .NET peuvent mettre à jour la même grille de données, vous devez rendre les classes compatibles. Dans les exemples d'application du tutoriel d'initiation, l'exemple d'application .NET a des alias pour correspondre aux valeurs par défaut Java.

Ajoutez des alias de classe et des attributs d'alias de zone à l'application .NET. Vous pouvez ajouter les alias de classe à l'application .NET, l'application Java ou aux deux applications. L'exemple .NET a des alias qui correspondent aux valeurs par défaut Java. Par conséquent, l'application Java n'a pas besoin d'alias. Les fichiers `TestKey.cs` et `TestValue.cs` se trouvent dans le répertoire *net_client_home*`/sample/SimpleClient`.

```
[ClassAlias( "com.ibm.websphere.xs.sample.gettingstarted.model.TestKey" )]
```

Figure 1. Attribut d'alias de classe dans le fichier TestKey.cs

```
[ClassAlias( "com.ibm.websphere.xs.sample.gettingstarted.model.TestValue" )]
```

Figure 2. Attribut d'alias de classe dans le fichier TestValue.cs

Point de contrôle de la leçon :

Vous avez ajouté des attributs à l'application d'initiation .NET. Par conséquent, vous pouvez utiliser l'application d'initiation Java en créant une grille de données d'entreprise.

Module 3 : Exécution de l'exemple d'application dans la grille de données

Pour exécuter l'exemple d'application, vous devez d'abord démarrer les serveurs de catalogue et de conteneur. Ensuite, exécutez l'exemple d'application.

Le processus de démarrage des serveurs de catalogue et de conteneur est identique, que vous exécutiez l'application .NET ou Java.

Objectifs d'apprentissage

A la fin de ce module, vous saurez :

- Démarrez les serveurs de catalogue et de conteneur
- Java Exécutez l'exemple d'application client Java de l'initiation
- .NET 8.6+ Exécutez l'exemple d'application client .NET.

8.6+ Vous pouvez exécuter les exemples d'applications Java et .NET séparément, mais vous pouvez également les exécuter simultanément dans la même grille de données. Par exemple, vous pouvez insérer une valeur dans la grille de données avec l'application .NET, puis obtenir la valeur avec l'application Java. Dans ce scénario, vous exécutez une grille de données d'entreprise.

Leçon 3.1 du tutoriel d'initiation : Démarrage des serveurs de catalogue et de conteneur

Pour exécuter l'exemple d'application client, vous devez démarrer un serveur de catalogue et un serveur de conteneur.

Le script `env.sh|bat` est appelé par les autres scripts pour la définition de variables d'environnement requises. Il n'est normalement pas nécessaire de modifier ce script.

- UNIX Linux `./env.sh`
- Windows `env.bat`

Pour exécuter l'application, vous devez d'abord démarrer le processus de service de catalogue. Le service de catalogue est le centre de contrôle de la grille de données. Il effectue le suivi des emplacements de serveurs de conteneur et contrôle le placement des données sur les serveurs de conteneur hôtes. Une fois que le service de catalogue démarre, vous pouvez démarrer les serveurs de conteneur qui stockent les données d'application de la grille de données. Pour stocker plusieurs copies des données, vous pouvez démarrer plusieurs serveurs de conteneur. Lorsque tous les serveurs sont démarrés, vous pouvez exécuter l'application client pour insérer, mettre à jour, supprimer et extraire des données de la grille de données.

1. Ouvrez une session de terminal ou une fenêtre de ligne de commande.
2. Dans la session de terminal ou une fenêtre de ligne de commande, accédez au répertoire *racine_install_wxs*`/ObjectGrid/gettingstarted` d'installation du serveur.
3. Exécutez le script suivant pour démarrer un processus de service de catalogue sur le système hôte local : 8.6+
   - UNIX Linux `./startcat.sh`
   - Windows `startcat.bat`

   Le processus du service de catalogue s'exécute dans la fenêtre du terminal en cours.

   Vous pouvez également démarrer le service de catalogue avec la commande **startXsServer**. Exécutez la commande **startXsServer** depuis le répertoire *racine_install_wxs*`/ObjectGrid/bin` :

   - UNIX Linux 8.6+
     ```
     ./startXsServer.sh cs0
     -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809
     ```

- Windows 8.6+ `startXsServer.bat cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`

4. Ouvrez une autre session terminal ou fenêtre de ligne de commande et exécutez la commande suivante pour démarrer une instance de serveur de conteneur : 8.6+
   - UNIX Linux `./startcontainer.sh server0`
   - Windows `startcontainer.bat server0`

   Le serveur de conteneur s'exécute dans la fenêtre du terminal en cours. Vous pouvez répéter cette étape avec un nom de serveur différent si vous voulez démarrer plus d'instances de serveurs de conteneur pour prendre en charge la réplication.

   Vous pouvez également démarrer les serveurs de conteneur avec la commande **startXsServer**. Exécutez la commande **startXsServer** depuis le répertoire *racine_install_wxs*/ObjectGrid/bin :
   - UNIX Linux 8.6+ `./startXsServer.sh c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted/server/config/objectgrid.xml -deploymentPolicyFile gettingstarted/server/config/deployment.xml`
   - Windows 8.6+ `startXsServer.bat c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\server\config\objectgrid.xml -deploymentPolicyFile gettingstarted\server\config\deployment.xml`

5. Java 8.6+ Facultatif : Au lieu de démarrer les serveurs de catalogue et de conteneur séparément, vous pouvez utiliser le script **runall** pour démarrer un serveur de catalogue, un serveur de conteneur et un exemple d'application client Java dans la même machine virtuelle Java. 8.6+
   - UNIX Linux `./runall.sh`
   - Windows `runall.bat`

   Restriction : Comme le script **runall** exécute les serveurs de conteneur intégrés, vous ne pouvez pas utiliser la console de surveillance pour surveiller l'environnement. Les statistiques ne sont pas collectées pour les serveurs de conteneur.

Point de contrôle de la leçon :

Dans cette leçon, vous avez appris à :

- Démarrer les serveurs de catalogue et les serveurs de conteneur

Leçon 3.2 du tutoriel d'initiation : Exécution de l'exemple d'application client Java

Java

Procédez comme suit pour exécuter un client Java pour interagir avec la grille de données. Le serveur de catalogue, le serveur de conteneur et le client s'exécutent tous sur un serveur unique dans cet exemple.

- 8.6+ Exécutez le client en mode interactif. Dans la fenêtre de ligne de commande, exécutez l'une des commandes suivantes :
  - UNIX Linux `./runclient.sh`
  - Windows `runclient.bat`

1. Démarrez une transaction. Vous pouvez utiliser une opération de validation en une phase ou deux phases pour la transaction. Avec la validation en une phase, la transaction doit être écrite dans une seule partition. Si vous insérez des clés pendant la transaction dans différentes partitions, la validation de la transaction échoue. Vous pouvez utiliser la validation en deux phases pour écrire une transaction dans plusieurs partitions.
   - Démarrez une transaction à validation en une phase.

     ```
     begin
     ```

   - Démarrez une transaction à validation en deux phases.

     ```
     begin2pc
     ```

2. Insérez une valeur.

   ```
   > i key1 helloWorld
   SUCCESS: Inserted TestValue [value=helloWorld] with key TestKey [key=key1], part
   itionId=6
   ```

3. Extrayez une valeur que vous avez insérée.

   ```
   > g key1
   Value is TestValue  [value=helloWorld], partitionId=6
   ```

4. Mettez à jour une valeur.

   ```
   > u key1 goodbyeWorld
   SUCCESS: Updated key TestKey [key=key1] with value TestValue [value=goodbyeWorld
   ], partitionId=6
   ```

5. Annulez la transaction. Lorsque vous annulez la transaction, toutes les opérations associées à la transaction sont annulées.

   ```
   > rollback
   ```

6. Pour tester l'opération d'annulation, essayez d'obtenir de nouveau la clé. Comme vous avez annulé la transaction, la clé n'existe pas :

   ```
   > g key1
   ```

7. Insérez une valeur.

   ```
   > i key1 helloWorld
   SUCCESS: Inserted TestValue [value=helloWorld] with key TestKey [key=key1], part
   itionId=6
   ```

8. Validez la valeur. Après avoir validé la transaction, vous ne pouvez pas annuler les modifications.

   ```
   > commit
   ```

9. Supprimez une valeur que vous avez insérée.

   ```
   > d key1
   SUCCESS: Deleted value with key TestKey [key=key1], partitionId=6
   ```

10. Insérez des entrées de test. Par exemple, pour insérer 1 000 clés et valeurs numérotées de 0 à 999, utilisez la commande suivante :

```
> n 1000
```

- 8.6+ Exécutez le mode de ligne de commande. Le mode de ligne de commande peut être utile si vous voulez écrire un script pour exécuter l'application client. Vous pouvez exécuter les mêmes commandes que celles que vous exécutez en mode interactif. Voici un exemple de syntaxe pour le mode de ligne de commande :
  - UNIX Linux

    ```
    ./runclient.sh i "key1" "helloWorld"
    ```

  - Windows

    ```
    runclient.bat i "key1" "helloWorld"
    ```

Point de contrôle de la leçon :

Leçons étudiées

Dans cette leçon, vous avez appris à :

- Exécuter l'exemple d'application client Java pour insérer, obtenir, mettre à jour et supprimer des données de la grille de données.

Leçon 3 du tutoriel d'initiation 3.3 : Exécution de l'exemple d'application .NET

.NET

Procédez comme suit pour exécuter une application client .NET pour interagir avec la grille de données. Le serveur de catalogue, le serveur de conteneur et le client s'exécutent tous sur un serveur unique dans cet exemple.

Le client .NET prend en charge uniquement les validations en une seule phase. Par conséquent, si vous tentez d'insérer plusieurs valeurs dans une même transaction, une exception se produit, car les valeurs sont placées dans des partitions différentes. Pour empêcher l'occurrence de ces exceptions lorsque vous exécutez l'exemple, vous pouvez changer le fichier XML descripteur de règle de déploiement pour utiliser une partition. Pour plus d'informations sur la mise à jour du nombre de partitions, voir «Leçon 1.1 du tutoriel d'initialisation : Définition de grilles de données avec des fichiers de configuration», à la page 1.

Vous pouvez exécuter l'exemple d'application en mode interactif ou de ligne de commande. En mode interactif, l'application exécute les transactions de grille de données manuelles avec l'API IGridMapPessimisticTx. Le mode de ligne de commande exécute les transactions de grille de données automatiques avec l'API IGridMapPessimisticAutoTx.

Vous pouvez exécuter l'exemple en mode interactif ou de ligne de commande :

- Exécutez l'exemple d'application client en mode interactif.
  1. Exécutez l'application client simple. Le fichier se trouve dans le répertoire *net_client_home*\gettingstarted\bin\.directory. Pour exécuter l'exemple en mode interactif, exécutez la commande suivante.

```
SimpleClient.exe -i
```

L'application se connecte à l'hôte `localhost:2809` par défaut. Pour remplacer la valeur par défaut, vous pouvez également fournir un hôte et un port distants comme paramètre à l'application :

```
SimpleClient.exe -i -h <endpoint>
```

Si vous exécutez l'application sans paramètre, l'aide de l'application s'affiche.

2. Affichez la liste des commandes disponibles.

```
Enter a command: help
This program executes simple CRUD operations on a map.
        a - Adds a value with the specified key. If the key already exists,
        DuplicatKeyException is thrown
        p - Adds a value with the specified key, replacing the entry if it
        already exists
        r - Replaces the value of the specified key. If the key does not exist,
        a CacheKeyNotFound exception is thrown
        g - Retrieve and display the value of the specified key
        d - Deletes the key
        gp - Gets the partition id for the key
        ck - Checks if the map contains the key
        h - Display help
        begin - Begin manual transaction
        commit - Commit transactions
        rollback - Rollback transactions
        exit - Exit program
```

3. Démarrez la transaction. Vous devez démarrer une transaction pour pouvoir exécuter des commandes sur la grille de données. Si vous ne démarrez pour la transaction, une exception NoActiveTransacationException se produit.

```
Enter a command: begin
```

4. Ajoutez des données à la grille.

```
Enter a command: a key1 value1
SUCCESS: Added 'TestKey [key=key1]' with value 'TestValue [value=value1]',
partitionId=6
```

5. Recherchez et affichez la valeur.

```
Enter a command: g key1
SUCCESS: Value is 'TestValue [value=value1]', partitionId=6
```

Dans cet exemple, `value1` est retourné.

6. Mettez à jour la clé. Utilisez la commande put qui ajoute une valeur avec la clé définie en remplaçant la valeur existante éventuelle.

```
Enter a command: p key1 value2
SUCCESS: Put key 'TestKey [key=key1]' with value 'TestValue [value=value2]',
 partitionId=6
Enter a command: g key1
SUCCESS: Value is 'TestValue [value=value2]', partitionId=6
```

7. Remplacez la clé. La commande remplace la valeur par la clé définie. Si la clé n'existe pas, une exception CacheKeyException est émise.

```
Enter a command: r key1 value3
SUCCESS: Replaced key 'TestKey [key=key1]' with value 'TestValue [value=value3]'
, partitionId=6
```

8. Annulez la transaction et essayez d'afficher de nouveau la clé de valeur. Vous pouvez annuler la transaction à tout moment avant la validation.

```
Enter a command: rollback
Enter a command: begin
Enter a command: g key1
FAILED: Key not found
```

   Lorsque vous exécutez la commande get, vous obtenez une exception indiquant que la clé est introuvable.

9. Validez une clé et une valeur dans la grille de données.

```
Enter a command: begin
Enter a command: a key2 value2
SUCCESS: Added 'TestKey [key=key2]' with value 'TestValue [value=value2]',
partitionId=7
Enter a command: commit
```

10. Obtenez un ID de partition pour une clé.

```
Enter a command: begin
Enter a command: gp key2
SUCCESS: partitionId=7
```

11. Recherchez les clés dans la mappe.

```
Enter a command: ck key2
SUCCESS: The map contains key 'TestKey [key=key2]'
Enter a command: ck key3
SUCCESS: The map does NOT contain key 'TestKey [key=key3]'
```

12. Supprimez la clé et quittez.

```
Enter a command: begin
Enter a command: d key2
SUCCESS: Deleted value with key 'TestKey [key=key2]', partitionId=7
Enter a command: commit
Enter a command: exit
```

- Exécutez le client en mode de ligne de commande. Le mode de ligne de commande exécute automatiquement les transactions de grille de données avec l'API IGridMapPessimisticAutoTx. Pour utiliser ce mode, envoyez l'action sur la ligne de commande. Le mode de ligne de commande peut être utile si vous voulez écrire un script pour exécuter l'application client. Vous pouvez exécuter les mêmes commandes que celles que vous avez exécutées en mode interactif. Voici un exemple de syntaxe pour le mode de ligne de commande :

```
SimpleClient [-h <host:port>] <a | p | r | g | d> <key> [<value>]
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez appris à :

- Exécuter l'exemple d'application .NET pour insérer, obtenir, mettre à jour et supprimer des objets de la grille de données.

Leçon 4 du tutoriel du guide de démarrage : Surveillance de l'environnement

Vous pouvez utiliser l'utilitaire **xscmd** et les outils de la console Web pour surveiller votre environnement de grille de données.

Surveillance à l'aide de la console Web

Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques préconfigurés pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.

Installez la console Web comme fonction facultative lorsque vous exécutez l'assistant d'installation.

1. Démarrez le serveur de la console. Le script **startConsoleServer.bat|sh** de démarrage du serveur de la console se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation.
2. Connectez-vous à la console.
   a. Dans votre navigateur Web, accédez à `https://your.console.host:7443`, en remplaçant `your.console.host` par le nom de l'hôte du serveur sur lequel vous avez installé la console.
   b. Connectez-vous à la console.
      - **ID utilisateur :** `admin`
      - **Mot de passe :** `admin`

      La page d'accueil de la console s'affiche.
3. Modifiez la configuration de la console. Cliquez sur **Paramètres** > **Configuration** pour afficher la configuration de la console. La configuration de la console comprend ce type d'informations :
   - la chaîne de trace pour le client WebSphere eXtreme Scale, comme `*=all=disabled`
   - le nom et le mot de passe de l'administrateur
   - son adresse e-mail
4. Créez et maintenez des connexions aux serveurs de catalogue que vous voulez surveiller. Répétez les étapes suivantes pour ajouter chaque serveur de catalogue à la configuration.
   a. Cliquez sur **Paramètres** > **Serveurs de catalogue eXtreme Scale**.
   b. Ajoutez un nouveau serveur de catalogue.

      1) Cliquez sur l'icône Ajouter () pour enregistrer un serveur de catalogue existant.
      2) Fournissez des informations, telles que le nom d'hôte et le port d'écoute. Voir «Planification des ports réseau», à la page 59 pour plus d'informations sur la configuration des ports et les valeurs par défaut.
      3) Cliquez sur **OK**.
      4) Vérifiez que le serveur de catalogue a bien été ajouté à l'arborescence de navigation.

5. Regroupez les serveurs de catalogue que vous avez créés dans un domaine de service de catalogue. Vous devez créer un domaine de service de catalogue lorsque la sécurité est activée dans vos serveurs de catalogue, car les paramètres de sécurité sont configurés dans le domaine de service de catalogue.
   a. Cliquez sur la page **Paramètres** > **Domaines eXtreme Scale**.
   b. Ajoutez un nouveau service de catalogue.

      1) Cliquez sur l'icône Ajouter ( ) pour enregistrer un service de catalogue existant. Entrez le nom du domaine de service de catalogue.
      2) Une fois que vous avez créé le domaine de service de catalogue, vous pouvez modifier les propriétés. Les propriétés du domaine de service de catalogue sont les suivantes :

         Nom Indique le nom d'hôte du domaine, attribué par l'administrateur.

         Serveurs de catalogue
         Liste un ou plusieurs catalogues qui appartiennent au domaine sélectionné. Vous pouvez ajouter les serveurs de catalogue que vous avez créés dans l'étape précédente.

         Classe de génération
         Indique le nom de la classe qui implémente l'interface CredentialGenerator. Cette classe utilisée pour obtenir les données d'identification des clients. Si vous définissez une valeur dans cette zone, la valeur remplace la propriété **crendentialGeneratorClass** dans le fichier `client.properties`.

         Propriétés du générateur
         Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés correspondent à l'objet avec la méthode setProperties(String). La valeur credentialGeneratorprops n'est utilisée que si la valeur de la propriété credentialGeneratorClass n'est pas null. Si vous définissez une valeur dans cette zone, la valeur remplace la propriété **credentialGeneratorProps** dans le fichier `client.properties`.

         Chemin des propriétés du client eXtreme Scale
         Indique le chemin d'accès au fichier de propriétés client que vous avez édité pour inclure les propriétés de sécurité dans une étape précédente. Par exemple, vous pouvez indiquer le fichier `c:\ObjectGridProperties\sampleclient.properties`. Si vous souhaitez empêcher la console de tenter d'utiliser les connexions sécurisées, vous pouvez supprimer la valeur dans cette zone. Après avoir défini le chemin, la console utilise une connexion non sécurisée.
      3) Cliquez sur **OK**.
      4) Vérifiez que le domaine a bien été ajouté à l'arborescence de navigation.

   Pour afficher les informations concernant un domaine de service de catalogue existant, cliquez sur le nom du domaine de service de catalogue dans l'arborescence de navigation sur la page **Paramètres** > **Domaines eXtreme Scale**.
6. Visualisez le statut de la connexion La zone **Domaine en cours** indique le nom du domaine de service de catalogue qui est actuellement utilisé pour afficher

des informations dans la console Web. L'état de la connexion s'affiche en regard du nom du domaine de service de catalogue.

7. Affichez les statistiques des grilles de données et des serveurs ou créez un rapport personnalisé.

Surveillance avec l'utilitaire xscmd

1. Facultatif : Si l'authentification de client est activée : Ouvrez une fenêtre de ligne de commande. Sur la ligne de commande, définissez les variables d'environnement appropriées.
2. Accédez au répertoire *rép_base_wxs*/bin.

   ```
   cd rép_base_wxs/bin
   ```

3. Exécutez plusieurs commandes pour afficher des informations sur votre environnement.
   - Afficher tous les serveurs de conteneur en ligne pour la grille de données de la grille et le groupe de mappes mapSet :

     ```
     xscmd -c showPlacement -g Grid -ms mapSet
     ```

   - Afficher les informations de routage de la grille de données.

     ```
     xscmd -c routetable -g Grid
     ```

   - Afficher le nombre d'entrées de mappe dans la grille de données.

     ```
     xscmd -c showMapSizes -g Grid -ms mapSet
     ```

Arrêt des serveurs

Une fois que vous avez fini d'utiliser l'application client et de surveiller l'exemple d'environnement du guide de démarrage, vous pouvez arrêter les serveurs.

- Si vous avez utilisé les fichiers script pour démarrer les serveurs, utilisez <ctrl+c> pour arrêter le processus de service de catalogue et les serveurs de conteneur dans les fenêtres correspondantes.
- Si vous avez utilisé la commande **startXsServer** pour démarrer les serveurs, utilisez la commande **stopXsServer** pour les arrêter.

  Arrêtez le serveur de conteneur :

  – UNIX Linux
  ```
  stopXsServer.sh c0 -catalogServiceEndPoints localhost:2809
  ```

  – Windows
  ```
  stopXsServer.bat c0 -catalogServiceEndPoints localhost:2809
  ```

  Arrêtez le serveur de conteneur :

  – UNIX Linux
  ```
  stopXsServer.sh cs1 -catalogServiceEndPoints localhost:2809
  ```

  – Windows
  ```
  stopXsServer.bat cs1 -catalogServiceEndPoints localhost:2809
  ```

Point de contrôle de la leçon

Dans cette leçon, vous avez appris à :

- démarrer la console Web et la connecter au serveur de catalogue ;
- surveiller les statistiques de la grille et des serveurs ;
- arrêter les serveurs.

Chapitre 2. Planification

Avant d'installer WebSphere eXtreme Scale et de déployer vos applications de grille de données, vous devez choisir votre topologie de mise en cache, planifier la capacité, vérifier les configurations matérielle et logicielle requises et les paramètres de réseau et d'optimisation, etc. Vous pouvez également utiliser la liste de contrôle opérationnelle pour vérifier que votre environnement est prêt pour le déploiement d'applications.

Vous trouverez une discussion des pratiques recommandées pour la conception d'applications WebSphere eXtreme Scale dans l'article suivant de developerWorks : Principles and best practices for building high performing and highly resilient WebSphere eXtreme Scale applications.

Présentation de la planification

Avant de commencer à utiliser WebSphere eXtreme Scale dans un environnement de production, les points suivants sont à prendre en considération afin d'optimiser le déploiement.

Remarques relatives à la topologie de mise en cache

Chaque type de topologie du cache présente des avantages et des inconvénients. La topologie de mise en cache que vous implémentez dépend de la configuration de votre environnement et de l'application. Pour plus d'informations sur les différentes topologies de cache, voir «Planification de la topologie», à la page 20.

Remarques sur la capacité de données

Points à prendre en considération :

- **Nombre de systèmes et de processeurs** : combien de machines et de processeurs physiques sont nécessaires dans l'environnement ?
- **Nombre de serveurs** : combien de serveurs eXtreme Scale pour héberger les mappes eXtreme Scale ?
- **Nombre de partitions** : la quantité de données stockées dans les mappes est l'un des facteurs déterminant le nombre de partitions nécessaires.
- **Nombre de répliques** : combien de répliques sont requises pour chacun des fragments primaires du domaine ?
- **Réplication synchrone ou asynchrone** : les données sont-elles si vitales pour nécessiter une réplication synchrone ? Ou bien, est-ce que les performances sont une priorité plus importante ? Dans ce cas, la réplication asynchrone s'impose
- **Tailles de pile** : quel volume sera stocké sur chaque serveur ?

Pour une discussion détaillée de chacune de ces considérations, voir les «Planification de la capacité de l'environnement», à la page 72..

Considérations relatives à l'installation

Vous pouvez installer WebSphere eXtreme Scale dans un environnement autonome, ou vous pouvez intégrer l'installation à WebSphere Application Server. Pour pouvoir mettre à niveau de manière transparente les serveurs, vous devez planifier l'environnement en conséquence. Pour optimiser les performances, les serveurs de

catalogue doivent s'exécuter sur des machines différentes de celles des serveurs de conteneur. Si vous devez exécuter vos serveurs de catalogue et serveurs de conteneur sur la même machine, utilisez des installations distinctes de WebSphere eXtreme Scale pour les serveurs de catalogue et les serveurs de conteneur. En utilisant deux installations, vous pouvez mettre à niveau l'installation qui exécute le serveur de catalogue en premier. Voir «Mise à jour des serveurs eXtreme Scale», à la page 255.

Planification de la topologie

Avec WebSphere eXtreme Scale, l'architecture de votre système peut utiliser la mise en cache des données locales en mémoire ou la mise en cache des données client-serveur réparties. L'architecture peut avoir des relations différentes avec vos bases de données. Vous pouvez également configurer la topologie pour l'étendre à plusieurs centres de données.

WebSphere eXtreme Scale requiert une infrastructure supplémentaire minimale pour pouvoir fonctionner. Cette infrastructure consiste en des scripts permettant d'installer, de démarrer et d'arrêter une application Java Platform, Enterprise Edition sur un serveur. Les données mises en cache sont stockées dans les serveurs de conteneur et les clients se connectent à distance au serveur.

Environnements internes

Lors du déploiement dans un environnement interne, WebSphere eXtreme Scale s'exécute dans une seule machine virtuelle Java et il n'est pas répliqué. Pour configurer un environnement local, vous pouvez utiliser un fichier XML ObjectGrid ou les API ObjectGrid.

Environnement réparti

Lorsque vous effectuez le déploiement dans un environnement réparti, WebSphere eXtreme Scale s'exécute dans un ensemble de machines virtuelles Java, ce qui améliore les performances, la disponibilité et l'évolutivité. Dans cette configuration, vous pouvez utiliser les fonctions de réplication et de partitionnement des données. Vous pouvez également ajouter d'autres serveurs sans redémarrer les serveurs eXtreme Scale existants. Comme dans le cas d'un environnement local, un fichier XML ObjectGrid ou une configuration par programmation équivalente est nécessaire dans un environnement réparti. Vous devez également fournir un fichier XML de stratégie de déploiement contenant les détails de la configuration.

Il est possible de créer des déploiements simples ou des déploiements plus vastes se chiffrant en téraoctets et comptant plusieurs milliers de serveurs.

Cache interne local

Dans le cas le plus simple, WebSphere eXtreme Scale peut être utilisé comme cache de grille de données locale (non répartie) en mémoire. Cette mise en cache locale peut s'avérer particulièrement utile pour les applications au nombre d'accès simultanés élevé où plusieurs unités d'exécution doivent accéder aux données temporaires et les modifier. Les données conservées dans une grille de données locale peuvent être indexées et extraites à l'aide de requêtes. Les requêtes permettent d'utiliser des jeux de données volumineux en mémoire. Le support fourni avec machine virtuelle Java (JVM), qui est prêt à être utilisé, dispose d'une structure de données limitées.

La topologie de cache local en mémoire de WebSphere eXtreme Scale permet d'octroyer un accès cohérent et transactionnel aux données temporaires dans une machine virtuelle Java unique.

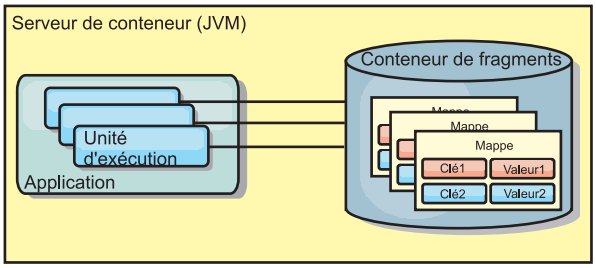


Figure 3. Scénario de cache local en mémoire

Avantages

- Configuration simple : une ObjectGrid peut être créée à l'aide d'un programme ou de manière déclarative avec le fichier XML du descripteur de déploiement ObjectGrid ou à l'aide d'une autre structure telle que Spring.
- Rapide : chaque mappe de sauvegarde peut être ajustée de façon indépendante pour optimiser l'utilisation de la mémoire et des accès simultanés.
- Configuration idéale pour les topologies de machine virtuelle Java dotées de petits jeux de données ou pour la mise en cache de données fréquemment consultées.
- Transactionnelle. Les mises à jour de mappe de sauvegarde peuvent être regroupées dans la même unité d'oeuvre et peuvent être intégrées en dernier lieu aux transactions constituées de deux phases telles que les transactions JTA (Java Transaction Architecture).

Inconvénients

- Aucune tolérance de panne.
- Les données ne sont pas répliquées. Les mémoires cache internes se prêtent aux données de référence en lecture seule.
- Non évolutive. La quantité de mémoire requise par la base de données peut dépasser la capacité de la machine virtuelle Java.
- Problèmes survenant lors de l'ajout de machines virtuelles Java :
  - Les données ne peuvent pas être facilement partitionnées ;
  - Nécessité de répliquer manuellement l'état entre les machines virtuelles Java ou chaque instance de cache peut présenter différentes versions des mêmes données.
  - L'invalidation est coûteuse.
  - Chaque cache doit être préchauffé indépendamment. Le préchauffage est la période de chargement d'un jeu de données permettant de remplir le cache avec des données valides.

Utilisation

La topologie de déploiement de la mémoire cache interne locale ne doit être utilisée que lorsque la quantité de données à mettre en cache est limitée (peut être abritée par une seule machine virtuelle Java) et est relativement stable. Cette approche doit tolérer les données obsolètes. L'utilisation d'expulseurs pour conserver les données les plus fréquemment ou récemment utilisées dans le cache

peut contribuer à réduire la taille du cache et à accroître la pertinence des données.

Cache local répliqué sur des homologues

Vous devez vous assurer que le cache est synchronisé si plusieurs processus avec des instances indépendantes de cache existent. Pour vérifier que les instances de cache sont synchronisées, activez un cache répliqué sur des homologues avec JMS (Java Message Service).

WebSphere eXtreme Scale comprend deux plug-in qui propagent automatiquement les modifications de transactions entre les instances ObjectGrid homologues. Le plug-in JMSObjectGridEventListener propage automatiquement les modifications eXtreme Scale à l'aide de JMS.

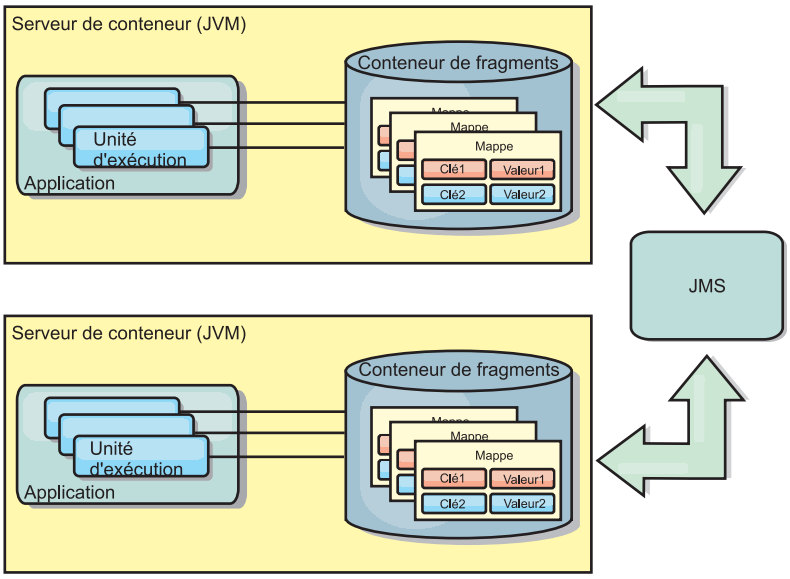


Figure 4. Cache répliqué sur des homologues avec des modifications qui sont propagées à l'aide de JMS

Si vous exécutez un environnement WebSphere Application Server, le plug-in TranPropListener est aussi disponible. Il utilise le gestion HA (high availability) pour propager les modifications à chaque instance de cache homologue.

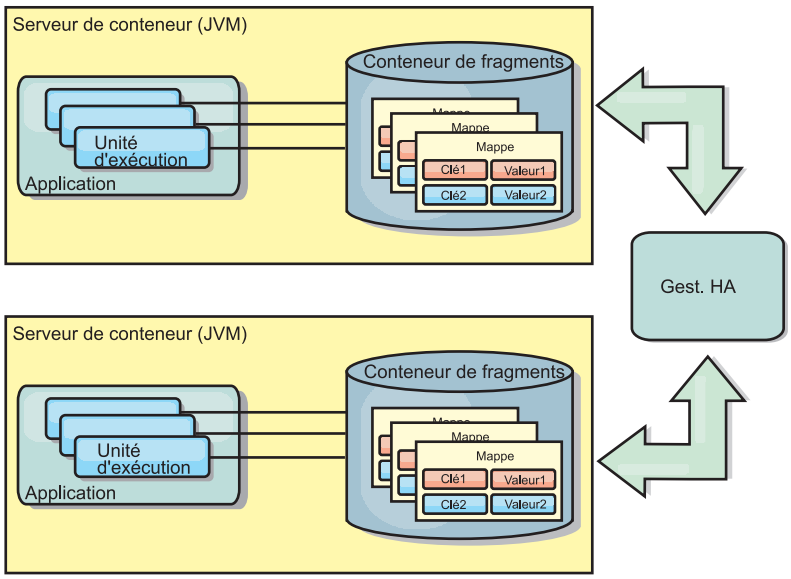


Figure 5. Cache répliqué sur des homologues avec des modifications qui sont propagées à l'aide du gestionnaire de haute disponibilité

Avantages

- Plus grande validité des données car celles-ci sont actualisées plus souvent.
- Avec le plug-in TranPropListener, tout comme avec l'environnement local, il est possible de créer la grille de données eXtreme Scale par programmation ou de manière déclarative avec le fichier XML du descripteur de déploiement d'eXtreme Scale ou avec d'autres structures de travail comme Spring. L'intégration au gestionnaire de haute disponibilité s'effectue automatiquement.
- Chaque mappe de sauvegarde peut être optimisée indépendamment en termes d'utilisation de la mémoire et de simultanéité des accès.
- Il est possible de regrouper en une seule unité d'oeuvre les mises à jour des mappes de sauvegarde qui peuvent être intégrées comme derniers participants de transactions en deux phases comme le sont les transactions Java Transaction Architecture (JTA).
- Idéal pour les topologies comprenant un nombre restreint de machines virtuelles Java avec un dataset de taille raisonnablement réduite ou pour la mise en cache des données à accès fréquent.
- Les modifications de la grille de données eXtreme Scale sont répliquées à toutes les instances eXtreme Scale homologues. Les modifications sont cohérentes tant qu'un abonnement durable est utilisé.

Inconvénients

- La configuration et la maintenance du plug-in JMSObjectGridEventListener peut s'avérer une tâche complexe. Il est possible de créer la grille de données eXtreme Scale par programmation ou de manière déclarative avec le fichier XML du descripteur de déploiement d'eXtreme Scale ou avec d'autres structures de travail comme Spring.
- Pas d'extensibilité : la quantité de mémoire requise par la base de données risque de submerger la machine virtuelle Java.
- Fonctionne de manière incorrecte lorsqu'on ajoute des machines virtuelles Java :
  - les données ne sont pas facilement partitionnées
  - l'invalidation est onéreuse

– chaque cache doit être prérempli de manière indépendante

Quand l'utiliser

Utilisez la topologie de déploiement uniquement lorsque la quantité de données à mettre en cache est faible, peut tenir sur une seule machine virtuelle Java, et relativement stable.

Cache imbriqué

Les grilles WebSphere eXtreme Scale peuvent s'exécuter dans des processus existants, tels que des serveurs eXtreme Scale intégrés ou vous pouvez les gérer comme des processus externes.

Les grilles imbriquées sont utiles lorsque l'exécution se fait dans un serveur d'applications tel que WebSphere Application Server. Vous pouvez démarrer les serveurs eXtreme Scale non imbriqués à l'aide de scripts de ligne de commande et les exécuter dans un processus Java.

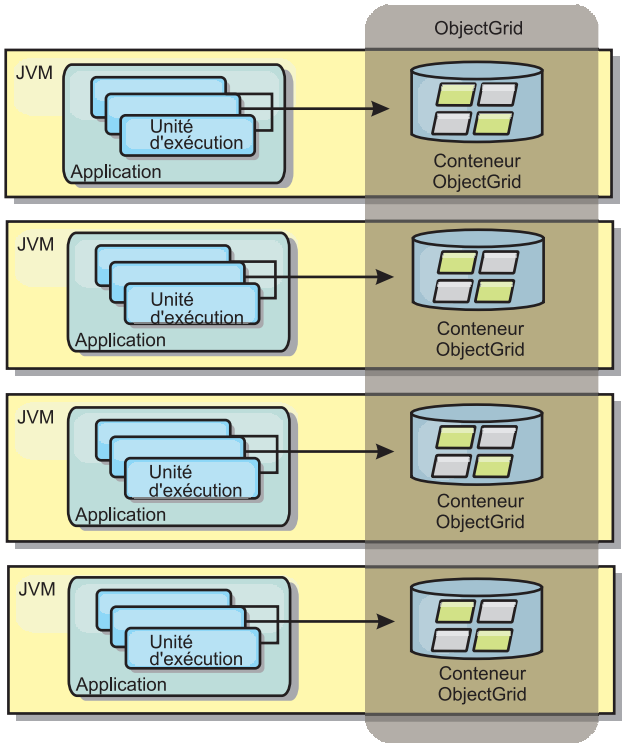


Figure 6. Cache imbriqué

Avantages

- simplification de l'administration en raison du nombre inférieur de processus à gérer
- simplification du déploiement d'application car la grille utilise le chargeur de classe de l'application client
- prise en charge du partitionnement et de la haute disponibilité.

Inconvénients

- augmentation de l'encombrement mémoire dans le processus client car toutes les données sont regroupées dans le processus
- augmentation de l'utilisation de l'unité centrale en vue de la gestion des demandes des clients
- plus grande difficulté à gérer les mises à niveau des applications car les clients utilisent les mêmes fichiers d'archive Java que les serveurs
- moindre flexibilité. Les clients et les serveurs de grille ne peuvent évoluer au même rythme. Lorsque des serveurs sont définis en externe, la gestion du nombre de processus devient plus flexible

Utilisation

Utilisez les grilles imbriquées lorsqu'une grande quantité de mémoire est disponible dans le processus client pour les données de la grille et pour les données de basculement.

Plus d'informations, voir la rubrique relative à l'activation du mécanisme d'invalidation de client dans *Guide d'administration*.

Cache réparti

La plupart du temps, WebSphere eXtreme Scale est utilisé en tant que cache partagé permettant un accès transactionnel aux données de plusieurs composants là où une base de données classique aurait été nécessaire. Avec le cache partagé, il n'est plus nécessaire de configurer une base de données.

Cohérence de la mémoire cache

Le cache est cohérent car tous les clients y voient les mêmes données. Chaque donnée est stockée dans le cache sur un seul serveur ce qui permet d'éviter la coexistence de plusieurs copies d'enregistrements risquant de contenir des versions différentes des données. Un cache cohérent contient un nombre croissant de données au fur et à mesure que l'on ajoute des serveurs à la grille et le cache évolue de manière linéaire au fur et à mesure que la taille de la grille augmente. Comme les clients accèdent aux données de cette grille de données avec des appels de procédure distante, cette mémoire est également appelée cache distant ou éloigné. Grâce au partitionnement des données, chaque processus contient un sous-ensemble unique de données. Les grandes grilles peuvent contenir davantage de données et traiter plus de demandes pour ces données. Par ailleurs la cohérence évite d'avoir à envoyer les données d'invalidation autour de la grille de données, car aucune donnée périmée n'existe. Le cache cohérent contient uniquement la copie la plus récente de chaque donnée.

Si vous exécutez un environnement WebSphere Application Server, le plug-in TranPropListener est aussi disponible. Il utilise le composant de haute disponibilité (gestionnaire HA) de WebSphere Application Server pour propager les modifications à chaque instance de cache ObjectGrid homologue.

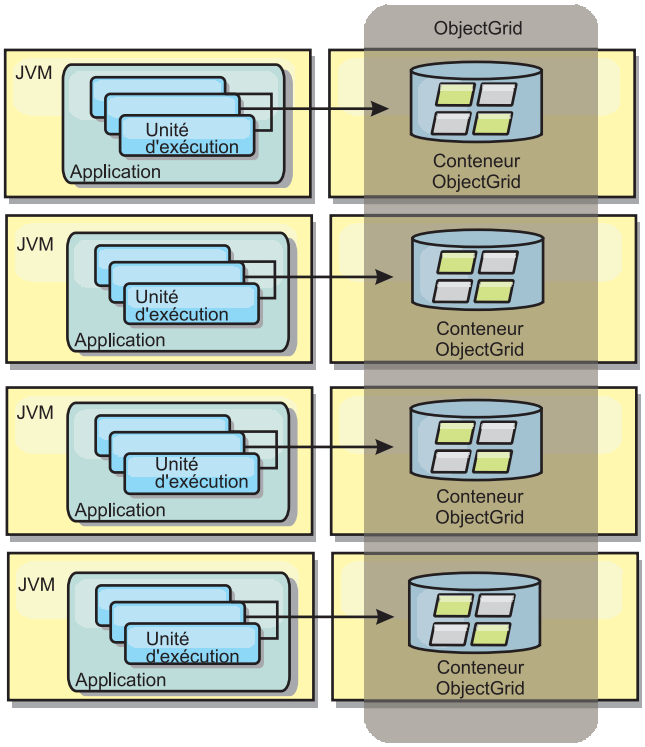


Figure 7. Cache réparti

Cache local

Lorsqu'eXtreme Scale est utilisé dans le cadre d'une topologie répartie, les clients peuvent éventuellement disposer d'un cache local en ligne. L'on appelle cache local ce cache facultatif. Il s'agit d'un ObjectGrid indépendant, présent sur chaque client et faisant office de cache du cache distant côté serveur. Il est activé par défaut lorsque le verrouillage est configuré sur OPTIMISTIC ou sur NONE. Son utilisation est impossible lorsque le verrouillage est configuré sur PESSIMISTIC.

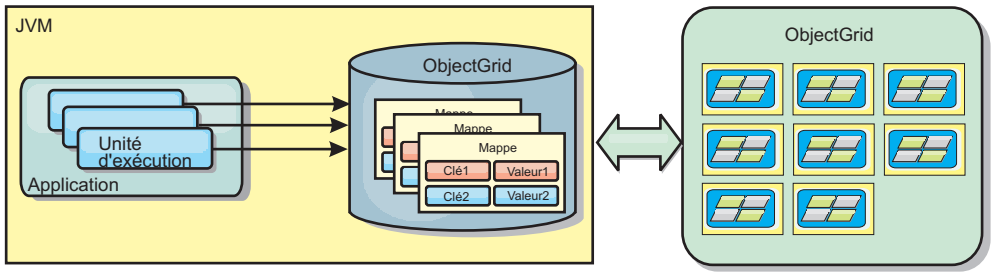


Figure 8. Cache local

Le cache local est très rapide car il offre un accès en mémoire à un sous-ensemble des données stockées à distance sur les serveurs eXtreme Scale. Il n'est pas partitionné et contient des données provenant de n'importe quelle partition eXtreme Scale distante. Jusqu'à trois groupes de caches peuvent exister dans WebSphere eXtreme Scale :

1. Le cache du groupe des transactions contient toutes les modifications apportées à une même transaction. Il contient une copie de travail des données jusqu'à ce que la transaction soit validée. Lorsqu'une transaction client demande des données à une ObjectMap, la transaction est vérifiée en priorité.

2. Le cache local du groupe des clients contient un sous-ensemble des données du groupe des serveurs. Lorsque le groupe des transactions ne contient pas les données, les données sont extraites du niveau client, si elles sont disponibles et insérées dans le cache des transactions.
3. La grille de données dans le groupe des serveurs contient la majorité des données et elle est partagée entre tous les clients. Le groupe des serveurs peut être partitionné, ce qui permet la mise en cache d'un grand nombre de données. Lorsque le cache local ne contient pas de données, celles-ci sont extraites du groupe des serveurs et insérées dans le cache du client. Le groupe des serveurs peut aussi avoir un plug-in Loader. Lorsque la grille ne contient pas les données demandées, le chargeur est appelé et les données résultantes sont insérées dans la grille à partir du magasin de données dorsal.

Pour désactiver le cache local, voir «Configuration du cache local», à la page 362.

Avantage

- Rapidité du temps de réponse, car tous les accès aux données se font localement La recherche de données dans le cache local évite de consulter la grille des serveurs et rend les données distantes accessibles localement.

Inconvénients

- Augmentation de la durée des données obsolètes, car le cache local à chaque niveau est peut-être désynchronisé avec les données en cours dans la grille de données.
- Basé sur un expulseur pour invalider les données afin d'éviter de manquer de mémoire.

Utilisation

A utiliser lorsque le temps de réponse est élevé et que la présence de données périmées est tolérée.

Intégration de la base de données : caches avec écriture différée, caches en ligne et caches secondaires

WebSphere eXtreme Scale est utilisé pour servir de frontal à une base de données classiques et ainsi éliminer l'activité de lecture qui est normalement envoyée vers la base de données. Un cache cohérent peut être utilisé avec une application soit directement, soit indirectement en passant alors par un associateur relationnel d'objets (ORM). Le cache cohérent peut décharger des tâches de lecture la base de données ou le dorsal. Dans un scénario un tout petit peu plus complexe, comme celui d'un accès transactionnel à un dataset dans lequel seules certaines données requièrent des garanties de persistance classique, il est possible d'utiliser le filtrage pour décharger même les transactions d'écriture.

Vous pouvez configurer WebSphere eXtreme Scale pour qu'il fonctionne en tant qu'espace extrêmement flexible de traitement de base de données interne. Cela dit, WebSphere eXtreme Scale n'est pas un associateur relationnel d'objets. Il ne sait pas d'où les données de la grille de données proviennent. Une application ou un associateur relationnel d'objets peuvent placer des données sur un serveur eXtreme Scale. C'est à la source de données qu'il incombe de vérifier la cohérence des données avec leur base de données d'origine. En d'autres termes, eXtreme Scale ne peut pas invalider les données qu'il a extraites automatiquement d'une base de données. C'est à l'application ou à l'associateur de fournir cette fonction et de gérer

les données stockées dans eXtreme Scale.

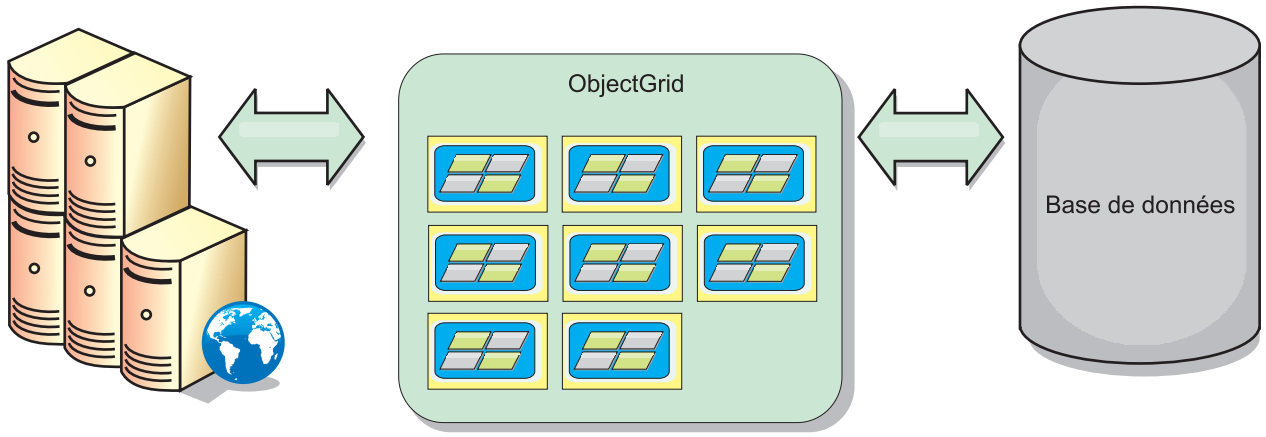


Figure 9. ObjectGrid en tant que mémoire tampon de base de données

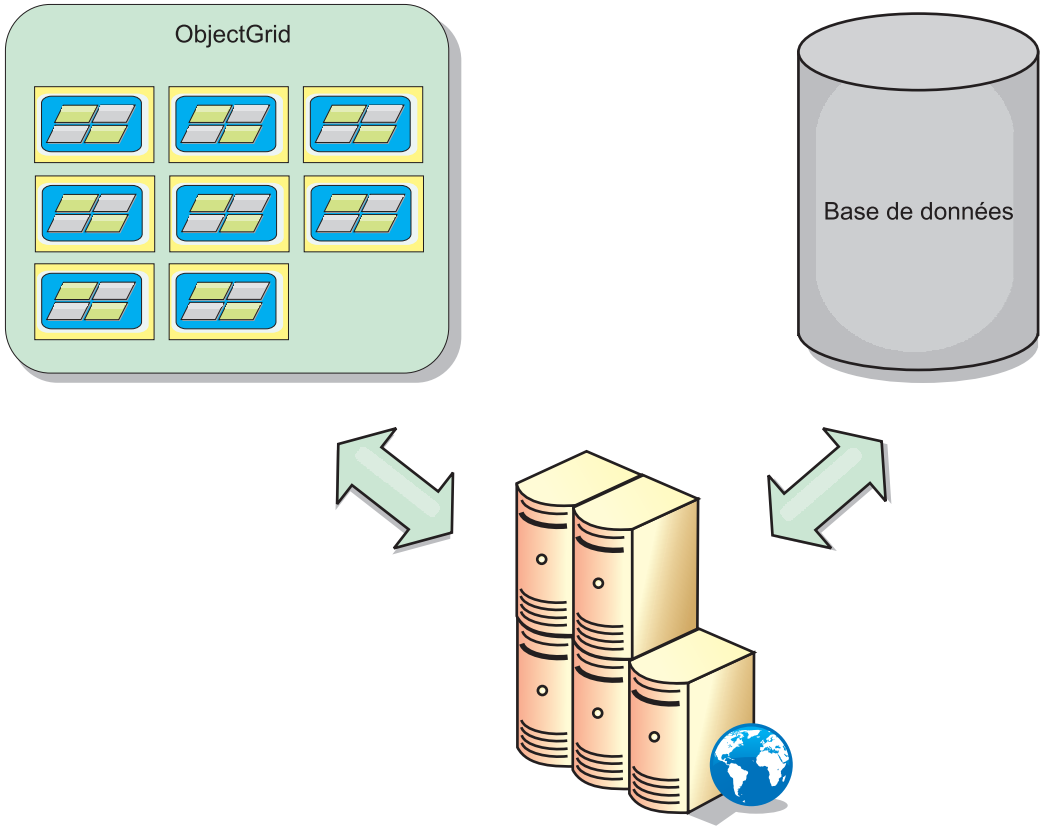


Figure 10. ObjectGrid en tant que cache secondaire

Cache partiel et cache complet

WebSphere eXtreme Scale peut s'utiliser en tant que cache partiel ou que cache complet. Un cache partiel ne conserve qu'un sous-ensemble des données totales, alors qu'un cache complet conserve toutes les données et peut être rempli en différé en fonction des besoins en données. Les caches partiels sont normalement accessibles à l'aide de clés (et non pas d'index ou de requêtes), car les données sont partiellement disponibles uniquement.

Cache partiel

Si une clé est absente dans un cache partiel ou que les données ne sont pas disponibles et qu'un échec de cache se produit, le niveau suivant est appelé. Les données sont extraites d'une base de données, par exemple, et elles sont insérées au groupe de caches de grille de données. Si vous utilisez une requête ou un index, seules les valeurs actuellement chargées sont accessibles et les requêtes ne sont pas transférées aux autres groupes.

Cache complet

Un cache complet comporte toutes les données requises et il est possible d'y accéder à l'aide d'attributs non-clés avec des index ou des requêtes. Un cache complet est préchargé avec des données de la base de données avant que l'application tente d'accéder aux données. Un cache complet peut fonctionner sous la forme d'un remplacement de base de données une fois que les données sont chargées. Etant donné que toutes les données sont disponibles, les requêtes et les index peuvent être utilisés pour rechercher et agréger les données.

Cache secondaire

Lorsque WebSphere eXtreme Scale est utilisé en tant que cache secondaire, le système dorsal est utilisé avec la grille de données.

Cache secondaire

Vous pouvez configurer le produit en tant que cache secondaire pour la couche d'accès aux données d'une application. Dans ce scénario, WebSphere eXtreme Scale permet de stocker temporairement des objets qui seraient normalement extraits d'une base de données dorsale. Les applications vérifient si la grille de données contient les données. Si les données se trouvent dans la grille de données, ces données sont renvoyées à l'appelant. Si elles n'existent pas, elles sont extraites de la base de données dorsale. Elles sont ensuite insérées dans la grille de données afin que la demande suivante puisse utiliser la copie mise en cache. Le diagramme suivant montre comment WebSphere eXtreme Scale peut être utilisé en tant que cache secondaire à l'aide d'une couche d'accès aux données arbitraire, telle qu'OpenJPA ou Hibernate.

Plug-in de cache secondaire pour Hibernate et OpenJPA

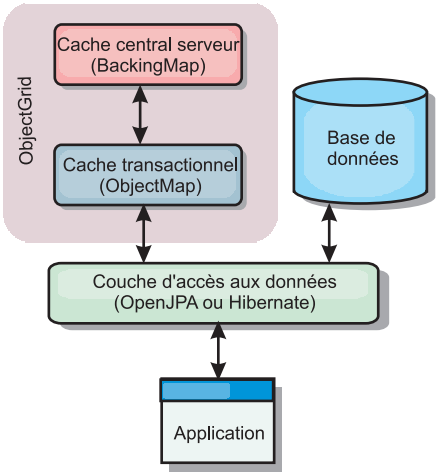


Figure 11. Cache secondaire

Les plug-in de cache pour OpenJPA et Hibernate sont inclus dans WebSphere eXtreme Scale pour que vous puissiez utiliser le produit comme cache secondaire automatique. L'utilisation d'WebSphere eXtreme Scale en tant que fournisseur de cache améliore les performances lors de la lecture et de l'interrogation des données et réduit la charge pesant sur la base de données. WebSphere eXtreme Scale présente plusieurs avantages par rapport à des implémentations de cache pré-intégrées car le cache est automatiquement répliqué entre tous les processus. Lorsqu'un client met une valeur en mémoire cache, tous les autres clients peuvent l'utiliser.

Cache en ligne

Vous pouvez configurer la mise en cache en ligne pour un système dorsal de base de données ou en tant que cache secondaire pour une base de données. La mise en cache en ligne utilise eXtreme Scale comme moyen principal pour interagir avec les données. Lorsque eXtreme Scale est utilisé en tant que cache en ligne, l'application interagit avec le système dorsal à l'aide d'un plug-in Loader.

Cache en ligne

Lorsque WebSphere eXtreme Scale est utilisé en tant que cache en ligne, il interagit avec le système dorsal à l'aide d'un plug-in Loader. Ce scénario permet de simplifier l'accès aux données car les applications peuvent accéder aux API eXtreme Scale directement. Plusieurs scénarios de cache sont pris en charge dans eXtreme Scale pour assurer la synchronisation des données dans le cache et des données dans le système dorsal. Le diagramme suivant illustre l'interaction entre le cache en ligne, l'application et le système dorsal.

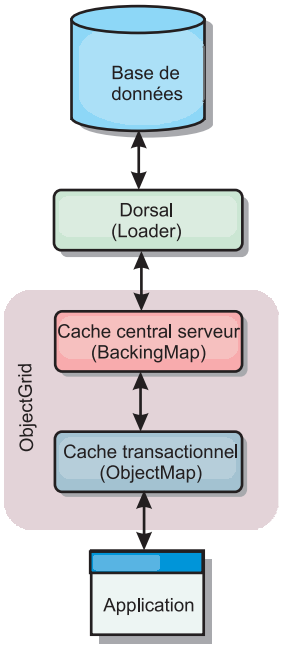


Figure 12. Cache en ligne

L'option de mise en cache en ligne simplifie l'accès aux données en permettant aux applications d'accéder directement aux API eXtreme Scale. WebSphere eXtreme Scale prend en charge plusieurs scénarios de mise en cache en ligne, comme suit.

- Sans interruption
- Ecriture immédiate
- Post-écriture

Scénario de mise en cache sans interruption

Un cache sans interruption est un cache partiel chargeant en lazy loading à partir d'une clé les entrées de données au fur et à mesure que ces entrées sont demandées. Cette opération peut se dérouler sans que l'appelant sache comment sont renseignées les entrées. Si les données sont introuvables dans le cache eXtreme Scale, eXtreme Scale récupère les données manquantes auprès du plug-in Loader qui charge les données provenant de la base de données d'arrière plan et les insère dans le cache. Les requêtes suivantes pour la même clé de données se trouveront dans le cache, jusqu'à ce qu'elles soient supprimées, invalidées ou expulsées.

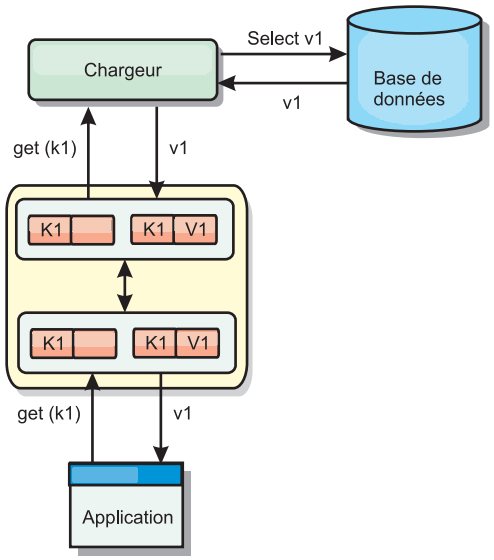


Figure 13. Mise en cache sans interruption

Scénario de mise en cache à écriture immédiate

Dans un cache à écriture immédiate, chaque écriture dans le cache est inscrite de manière synchrone dans la base de données à l'aide du chargeur. Cette méthode permet la cohérence avec le système dorsal, mais réduit les performances d'écriture étant donné que l'opération de base de données est synchrone. Le cache et la base de données étant tous deux mis à jour, les lectures suivantes à la recherche des mêmes données auront lieu dans le cache, évitant ainsi de faire appel à la base de données. Un cache à écriture immédiate est souvent utilisé conjointement à un cache sans interruption.

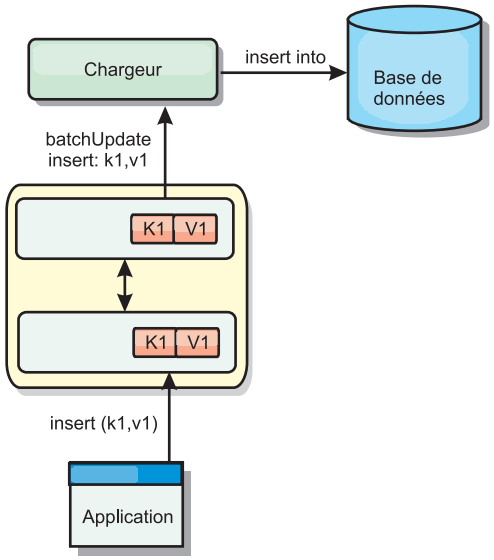


Figure 14. Mise en cache à écriture immédiate

Scénario de mise en cache en écriture différée

La synchronisation de la base de données peut être améliorée en écrivant les modifications de manière asynchrone. Cette opération est appelée mise en cache en

écriture différée. Les modifications, normalement écrites de manière synchrone dans le chargeur, sont mises en mémoire tampon dans eXtreme Scale et écrites dans la base de données à l'aide d'une unité d'exécution en arrière-plan. Les performances d'écriture sont considérablement améliorées, car l'opération de base de données est supprimée de la transaction client et les écritures de la base de données peuvent être comprimées.

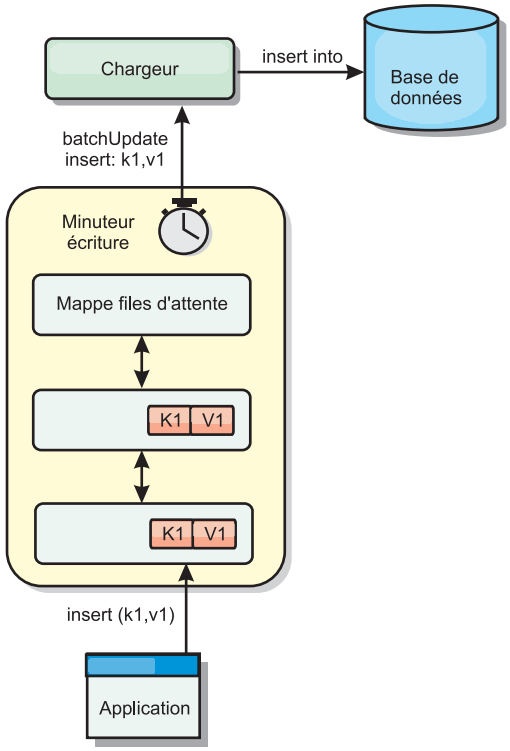


Figure 15. Mise en cache en écriture différée

Mise en cache en écriture différée

Java

Vous pouvez utiliser la mise en cache en écriture différée pour réduire le temps système supplémentaire nécessaire lors de la mise à jour d'une base de données utilisée en tant que base de données dorsale.

Présentation de la mise en cache en écriture différée

La mise en cache en écriture différée met en file d'attente de manière asynchrone les mises à jour du plug-in Loader. Vous pouvez améliorer les performances en déconnectant les mises à jour, les insertions et les suppressions au sein d'une mappe, le temps système pour la mise à jour de la base de données dorsale. La mise à jour asynchrone est effectuée après un retard (de cinq minutes, par exemple) ou après un certain nombre d'entrées (1 000 entrées).

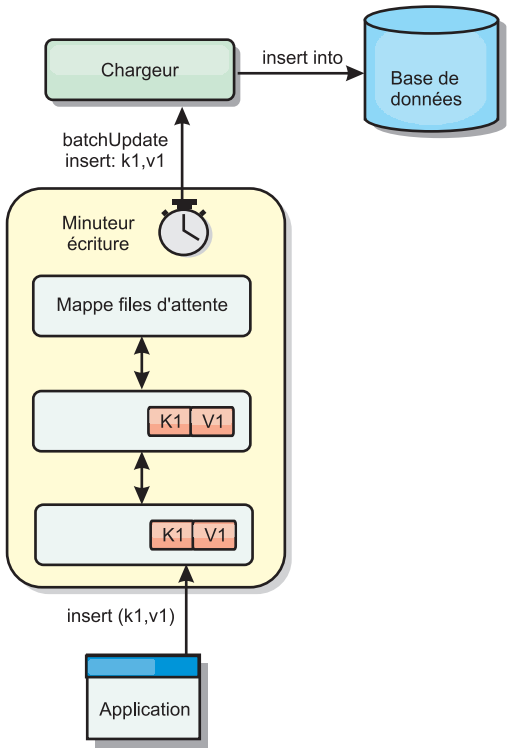


Figure 16. Mise en cache en écriture différée

La configuration à écriture différée sur une mappe de sauvegarde crée une unité d'exécution entre le chargeur et la mappe. Le chargeur délègue alors les demandes de données via l'unité d'exécution en fonction des paramètres de configuration de la méthode BackingMap.setWriteBehind. Lorsqu'une transaction eXtreme Scale insère, met à jour ou supprime une entrée dans une mappe, un objet LogElement est créé pour chacun de ces enregistrements. Ces éléments sont envoyés au chargeur à écriture différée et mis en file d'attente dans une ObjectMap spéciale appelée mappe de files d'attente. Chaque mappe de sauvegarde pour laquelle le paramètre d'écriture différée est activé a ses propres mappes de files d'attente. L'unité d'exécution à écriture différée supprime périodiquement les données mises en file d'attente des mappes correspondantes et les insère dans le chargeur dorsal.

Le chargeur à écriture différée envoie uniquement les types insertion, mise à jour et suppression des objets LogElement au chargeur réel. Tous les autres types, par exemple le type EVICT, sont ignorés.

La prise en charge de l'écriture différée *est* une extension du plug-in Loader, qui vous permet d'intégrer eXtreme Scale à la base de données. A ce sujet, vous pouvez consulter avec profit les explications «Configuration des chargeurs JPA», à la page 431 sur la configuration d 'un chargeur JPA.

Avantages

L'activation de l'écriture différée présente les avantages suivants :

- **Isolement en cas d'arrêt anormal de la base de données dorsale :** la mise en cache à écriture différée propose une couche d'isolement en cas d'arrêt anormal de la base de données dorsale. Les mises à jour sont alors placées dans la mappe de files d'attente. Les applications peuvent continuer à envoyer des transactions

vers eXtreme Scale. Lors de la reprise du système dorsal, les données contenues dans la mappe de files d'attente sont insérées dans celui-ci.

- **Réduction de la charge du système dorsal :** le chargeur à écriture différée fusionne les mises à jour en fonction des clés de façon qu'une seule mise à jour fusionnée par clé existe dans la mappe de files d'attente. Cette fusion diminue le nombre de mises à jour dans la base de données dorsale.
- **Amélioration des performances de la transaction :** la durée de chaque transaction eXtreme Scale est réduite car la transaction n'a plus à attendre que les données soient synchronisées avec le système dorsal.

Chargeurs

Java

Avec un plug-in Loader, une mappe de grille de données peut se comporter comme un cache pour les données généralement conservées dans un magasin persistant sur le même système ou un autre système. Généralement, une base de données ou un système de fichiers est utilisé comme stockage de persistance. Une machine virtuelle Java (JVM) peut également être utilisée comme source des données, ce qui permet de créer des caches basés sur un concentrateur à l'aide d'eXtreme Scale. Un chargeur peut lire et écrire des données vers un stockage persistant ou à partir de celui-ci.

Présentation

Les chargeurs sont des plug-in de mappe de sauvegarde appelés lorsque des modifications sont apportées à la mappe de sauvegarde ou lorsque cette dernière est dans l'impossibilité de répondre à une demande de données (absence dans le cache). Le chargeur est appelé lorsque le cache ne peut pas satisfaire une demande de clé, offrant ainsi une fonction de lecture et un remplissage laborieux du cache. Un chargeur permet également les mises à jour de la base de données lorsque les valeurs du cache viennent à changer. Toutes les modifications dans une transaction sont regroupées pour réduire le nombre d'interactions de base de données. Un plug-in TransactionCallback est utilisé conjointement avec le chargeur pour déclencher la démarcation de la transaction principale. L'utilisation de ce plug-in est importante lorsque plusieurs mappes sont incluses dans une seule transaction ou lorsque les données de transaction sont vidées dans le cache sans validation.

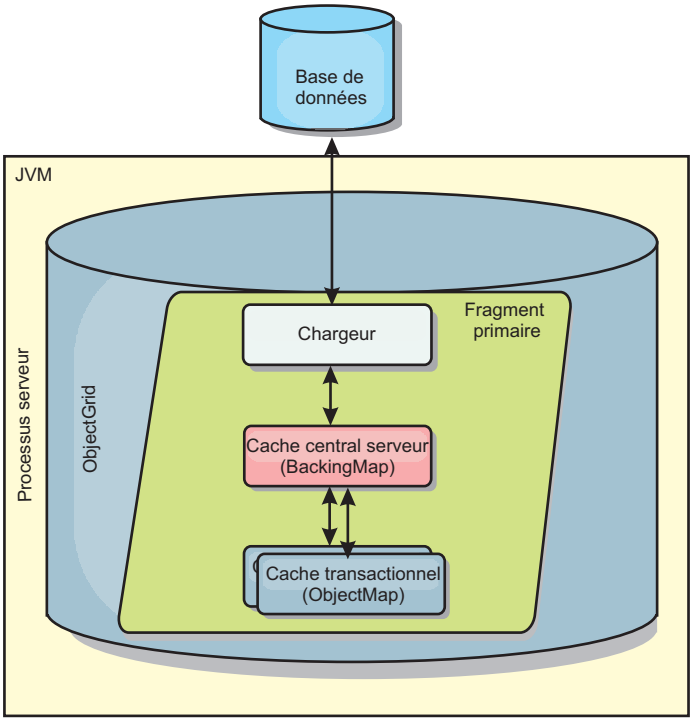


Figure 17. Chargeur

Le chargeur peut donc utiliser les mises à jour sur-qualifiées pour éviter le verrouillage intempestif de la base de données. En stockant un attribut de version dans la valeur du cache, le chargeur peut distinguer l'image de la valeur avant et après la mise à jour dans le cache. Cette valeur peut ensuite être utilisée lors de la mise à jour de la base de données ou du programme d'arrière plan pour vérifier que les données n'ont pas été mises à jour. Un chargeur peut également être configuré pour précharger la grille de données lorsqu'elle démarre. Lorsqu'elle est partitionnée, une instance de chargeur est associée à chaque partition. Si la mappe de la société comporte dix partitions, il existe dix instances de chargeur, une pour chaque partition principale. Lorsque le fragment primaire de la mappe est activé, la méthode preloadMap du chargeur est appelée de manière synchrone ou asynchrone, ce qui déclenche le chargement automatique de la partition de la mappe avec les données du programme d'arrière plan. Lorsqu'il est appelé de manière synchrone, toutes les transactions client sont bloquées, ce qui empêche tout accès incohérent à la grille de données. Sinon, un préchargeur client peut être utilisé pour charger l'intégralité de la grille de données.

Deux chargeurs pré-intégrés peuvent simplifier considérablement l'intégration aux dorsaux de bases de données relationnelles. Les chargeurs JPA utilisent les fonctions du mappage objet-relationnel(ORM) des implémentations OpenJPA et Hibernate des spécifications JPA (Java Persistence API). Pour plus d'informations, voir Chargeurs JPA.

Si vous utilisez des chargeurs dans une configuration à plusieurs centre de données, vous devez étudier la façon dont les données de révision et la cohérence de la mémoire cache est conservée entre les grilles de données. Pour plus d'informations, voir «Remarques sur les chargeurs dans une topologie multimaître» , à la page 50.

Configuration de chargeur

Pour ajouter un chargeur à la configuration BackingMap, vous pouvez utiliser la configuration à l'aide d'un programme ou la configuration XML. Un chargeur a la relation suivante avec une mappe de sauvegarde.

- Une mappe de sauvegarde peut avoir un seul chargeur.
- Une mappe de sauvegarde client (cache local) ne peut pas avoir de chargeur.
- Une définition de chargeur peut être appliquée à plusieurs mappes de sauvegarde, mais chaque mappe de sauvegarde dispose de sa propre instance de chargeur.

Préchargement et préremplissage des données

Dans la plupart des scénarios qui utilise un chargeur, vous pouvez préparer la grille de données en y préchargeant ses données.

Lorsque vous utilisez la grille de données comme un cache complet, elle doit contenir toutes les données et elle doit être chargée pour que les clients puissent s'y connecter. Lorsque vous utilisez un cache partiel, vous pouvez préparer le cache avec des données pour que les clients puissent avoir accès immédiatement à ces données dès qu'ils se connectent.

Il existe deux approches pour pré-charger des données dans la grille de données ; vous pouvez utiliser un plug-in Loader ou un chargeur client, comme décrit dans les sections suivantes.

Plug-in Loader

Le plug-in Loader est associé à chaque mappe et chargé de synchroniser un fragment primaire de partition avec la base de données. La méthode preloadMap du plug-in Loader est invoquée automatiquement lors de l'activation d'un fragment. Par exemple, vous disposez de 100 partitions, il existe 100 instances Loader, chacune chargeant les données de sa partition. En cas d'exécution synchrone, tous les clients sont bloqués jusqu'à la fin du préchargement.

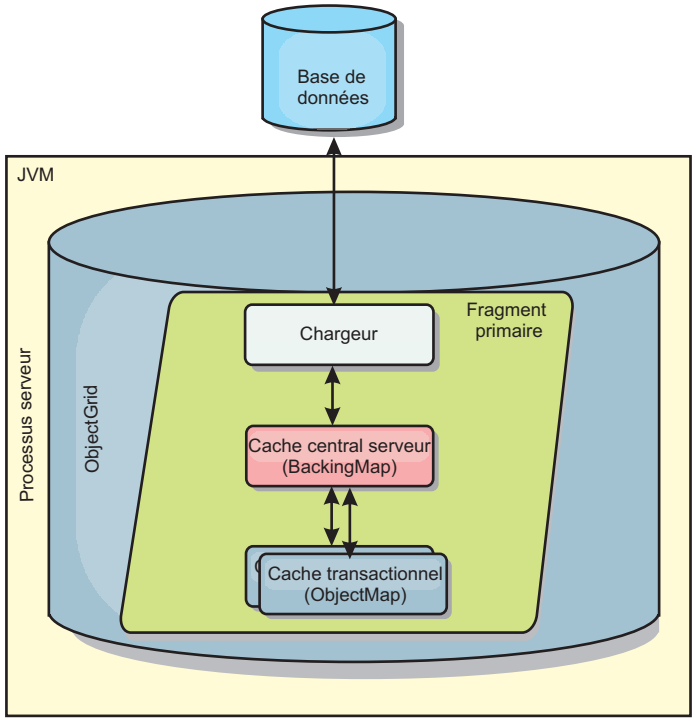


Figure 18. Plug-in Loader

Chargeur client

Un chargeur client est un pattern d'utilisation d'un ou plusieurs clients pour charger les données dans la grille. L'utilisation de plusieurs clients pour charger les données de la grille peut s'avérer efficace lorsque le schéma de partition n'est pas stocké dans la base de données. Vous pouvez appeler des chargeurs de client manuellement ou automatiquement lorsque la grille de données démarre. Ces chargeurs peuvent éventuellement utiliser StateManager pour faire passer la grille de données en mode de préchargement pour que les clients ne puissent pas accéder à la grille lorsqu'elle précharge les données. WebSphere eXtreme Scale contient un chargeur JPA (Java Persistence API) que vous pouvez utiliser pour charger automatiquement la grille de données avec le fournisseur JPA OpenJPA ou Hibernate. Pour plus d'informations sur les fournisseurs de cache, voir «Plug-in de cache niveau 2 (L2) JPA», à la page 407.

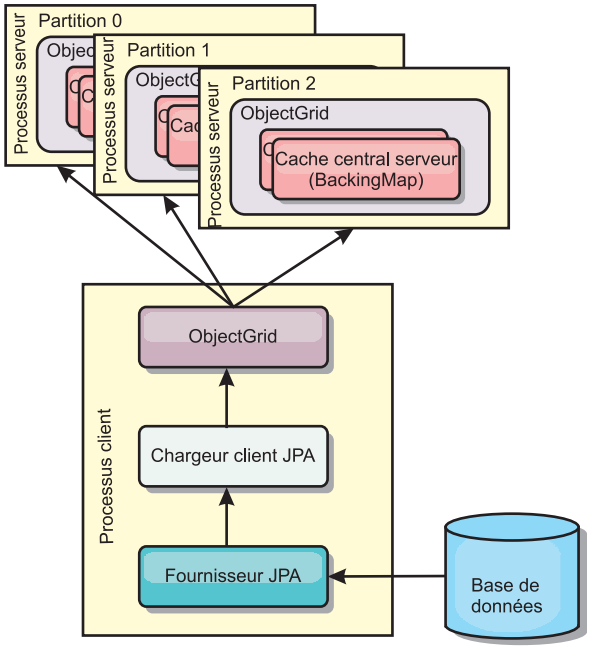


Figure 19. Chargeur client

Méthodes de synchronisation de base de données

Lorsque WebSphere eXtreme Scale est utilisé en tant que cache, les applications doivent être écrites de sorte qu'elles tolèrent les données périmées si la base de données peut être mise à jour de manière indépendante par rapport à une transaction eXtreme Scale. En tant qu'espace de traitement de base de données en mémoire synchronisé, eXtreme Scale permet d'assurer la mise à jour du cache de plusieurs manières.

Méthodes de synchronisation de base de données

Actualisation régulière

Le cache peut être régulièrement invalidé ou mis à jour de manière automatique à l'aide du programme de mise à jour temporelle de base de données JPA (Java Persistence API).Le programme de mise à jour interroge régulièrement la base de données à l'aide d'un fournisseur JPA, afin de rechercher des mises à jour ou des insertions survenues depuis la mise à jour précédente. Tous les changements détectés sont automatiquement invalidés ou mis à jour lorsqu'ils sont utilisés avec un cache incomplet. S'ils sont utilisés avec un cache complet, les entrées peuvent être détectées et insérées dans le cache. Les entrées ne sont jamais supprimées du cache.

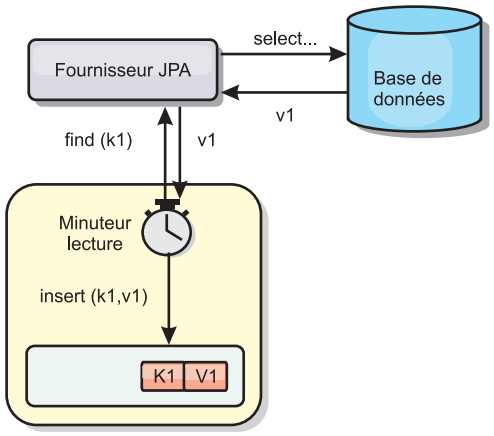


Figure 20. Actualisation régulière

Suppression

Les caches incomplets peuvent utiliser les stratégies de suppression pour supprimer automatiquement les données du cache sans que cela n'affecte la base de données. eXtreme Scale inclut trois stratégies : durée de vie, utilisation la moins récente et utilisation la moins fréquente. Si l'option de suppression en fonction de la mémoire est activée, ces trois stratégies suppriment les données de manière plus agressive à mesure que la mémoire est limitée.

Invalidation en fonction d'événements

Il est possible d'invalider les caches partiels et complets à l'aide d'un générateur d'événements comme JMS (Java Message Service). L'invalidation par le biais de JMS peut être associée de manière manuelle à tout processus qui met à jour le dorsal à l'aide d'un déclencheur de base de données. eXtreme Scale contient un plug-in JMS ObjectGridEventListener qui informe les clients des éventuelles modifications du cache du serveur. Cette procédure peut réduire la durée d'accès du client aux données périmées.

Invalidation par programme

Les API eXtreme Scale permettent l'interaction manuelle du cache local et du cache serveur à l'aide des méthodes des API Session.beginNoWriteThrough(), ObjectMap.invalidate() et EntityManager.invalidate(). Si un processus client ou serveur n'a plus besoin d'une partie des données, les méthodes d'invalidation peuvent être utilisées pour supprimer les données du serveur local ou du serveur cache. La méthode beginNoWriteThrough applique une opération ObjectMap ou EntityManager au cache local sans appeler le programme de chargement. Si l'opération est appelée à partir d'un client, elle s'applique uniquement au cache local (le programme de chargement distant n'est pas appelé). Si elle est appelée sur le serveur, l'opération s'applique uniquement au cache central du serveur sans appeler le programme de chargement.

Invalidation des données

Pour supprimer les données de mémoire cache périmées, vous pouvez utiliser des mécanismes d'invalidation.

Invalidation administrative

Vous pouvez utiliser la console Web ou l'utilitaire **xscmd** pour invalider les données en fonction de la clé. Vous pouvez filtrer les données du cache avec une expression régulière, puis invalider les données en fonction de cette expression régulière.

Invalidation basée sur les événements

Il est possible d'invalider les caches incomplets et complets à l'aide d'un générateur d'événements tel que Java Message Service (JMS). L'invalidation par le biais de JMS peut être associée de manière manuelle à tout processus qui met à jour le dorsal à l'aide d'un déclencheur de base de données. Un plug-in JMS ObjectGridEventListener est fourni dans eXtreme Scale pour permettre aux clients d'être informés des modifications dans le cache du serveur. Ce type de notification peut réduire la durée d'accès du client aux données obsolètes.

L'invalidation basée sur les événements est composée normalement des trois composants suivants.

- **File d'attente des événements :** une file d'attente d'événements stocke les événements de modification des données. Il peut s'agir d'une file d'attente JMS, d'une base de données, d'une file d'attente interne premier entré premier sorti ou de tout autre événement dans la mesure où elle peut gérer les événements de modification des données.
- **Publicateur d'événements :** un publicateur d'événements publie les événements de modification de données dans la file d'attente d'événements. Une publicateur d'événements est généralement une application que vous créez ou une implémentation de plug-in eXtreme Scale. Il sait quand les données ont été modifiées ou il modifie les données lui-même. Lorsqu'une transaction est validée, les événements sont générés pour les données modifiées et le publicateur d'événements publie ces événements dans la file d'attente d'événements.
- **Consommateur d'événements :** un consommateur d'événements consomme les événements de modification de données. Le consommateur d'événements est généralement une application permettant de vérifier la mise à jour des données de la grille cible avec les dernières modifications apportées aux autres grilles. Il interagit avec la file d'attente d'événements pour récupérer les dernières données et applique les modifications apportées aux données dans la grille cible. Les consommateurs d'événements peuvent utiliser les API eXtreme Scale pour invalider les données obsolètes ou mettre à jour la grille avec les dernières données.

Par exemple, JMSObjectGridEventListener comporte une option pour un modèle client-serveur dans lequel la file d'attente d'événements est une destination JMS désignée. Tous les processus serveur sont des publicateurs d'événements. Lorsqu'une transaction est validée, le serveur récupère les modifications apportées aux données et les publie à la destination JMS désignée. Tous les processus client sont des consommateurs d'événements. Ils reçoivent les modifications apportées aux données de la destination JMS désignée et appliquent les modifications au cache local du client.

Pour plus d'informations, voir «Configuration de la synchronisation du client JMS (Java Message Service)», à la page 366.

Invalidation par programme

Les API WebSphere eXtreme Scale autorise l'interaction manuelle du cache local et du cache serveur à l'aide des méthodes Session.beginNoWriteThrough(), ObjectMap.invalidate() et EntityManager.invalidate(). Si un processus client ou serveur n'a plus besoin d'une partie des données, les méthodes invalidate permettent de supprimer des données d'un cache local ou de serveur. La méthode beginNoWriteThrough applique toutes les opérations ObjectMap ou EntityManager au cache local sans appeler le chargeur. Si l'opération est appelée à partir d'un client, elle s'applique uniquement au cache local (le programme de chargement distant n'est pas appelé). Si elle est appelée sur le serveur, l'opération s'applique uniquement au cache central du serveur sans appeler le programme de chargement.

Vous pouvez utiliser l'invalidation par programme à l'aide d'autres techniques pour déterminer quand il convient d'invalider les données. Par exemple cette méthode d'invalidation utilise des mécanismes d'invalidation basée sur les événements pour recevoir les événements de modification de données, puis utilise les API pour invalider les données obsolètes.

8.6+
Invalidation du cache local

Si vous utilisez un cache local, vous pouvez configurer une invalidation asynchrone qui est déclenchée chaque fois qu'une mise à jour, une suppression et une invalidation est exécutée sur la grille de données. Puisque l'opération est asynchrone, la grille de données peut toujours contenir des données périmées.

Pour activer l'invalidation du cache local, définissez l'attribut **nearCacheInvalidationEnabled** sur la mappe de sauvegarde dans le fichier XML descripteur d'ObjectGrid.

Indexation

Java

Utilisez le plug-in MapIndexPlugin pour générer un ou plusieurs index dans une mappe BackingMap pour prendre en charge l'accès aux données ne correspondant pas à une clé.

Types d'indexation et configuration d'index

L'indexation est représentée par le plug-in MapIndexPlugin ou Index, en bref. Index est un plug-in BackingMap. Une mappe de sauvegarde peut avoir plusieurs index configurés, dès lors que chacun d'entre eux respecte les règles de configuration d'index.

Vous pouvez utiliser l'indexations pour générer une ou plusieurs index dans une mappe BackingMap. Un index se construit à partir d'un attribut ou d'une liste des attributs d'un objet de la mappe. L'indexation permet aux applications de trouver plus rapidement certains objets. Grâce à elle, en effet, les applications peuvent trouver les objets dont les attributs indexés ont une certaine valeur ou se situent dans une plage de valeurs.

Deux types d'indexation sont possibles : statiques et dynamiques. L'indexation statique oblige à configurer le plug-in d'indexation index dans la mappe de sauvegarde avant d'initialiser l'instance ObjectGrid. Comme pour la mappe de

sauvegarde, cela peut se faire par programmation ou via XML. L'indexation statique commence à générer l'index pendant l'initialisation de la grille d'objets. L'index est synchrone en permanence avec la mappe de sauvegarde et il est prêt à être utilisé. Après que l'indexation statique a démarré, la maintenance de l'index fait partie de la gestion des transactions par eXtreme Scale. Lorsque les transactions valident leurs modifications, ces dernières actualisent également l'index statique et les modifications apportées à l'index sont annulées en cas d'annulation de la transaction.

L'indexation dynamique permet de créer un index dans une mappe de sauvegarde avant ou après l'initialisation de l'instance ObjectGrid qui contient cette mappe. Les applications contrôlent le cycle de vie de l'indexation dynamique, ce qui permet de supprimer un index dynamique devenu inutile. Lorsqu'une application crée un index dynamique, cet index n'est pas forcément utilisable immédiatement en raison du temps que met à s'effectuer la génération complète de l'index. Comme la durée dépend de la quantité de données indexées, l'interface DynamicIndexCallback est fournie pour les applications qui souhaitent recevoir des notifications lorsque se produisent certains événements l'indexation, à savoir les événements ready, error et destroy. Les applications peuvent implémenter cette interface de rappel et s'enregistrer auprès de l'indexation dynamique.

8.6+ Si un plug-in d'indexation est configuré pour une mappe de sauvegarde, il est possible d'obtenir de la mappe d'objet correspondante l'objet proxy de l'index. L'appel de la méthode getIndex dans la mappe et la transmission du nom du plug-in Index renvoie l'objet proxy de l'index. Vous devez transtyper l'objet de proxy d'index en interface d'index d'application appropriée, telle que MapIndex, MapRangeIndex, MapGlobalIndex, ou en interface d'index personnalisée. Une fois l'objet proxy obtenu, l'on peut utiliser les méthodes définies dans l'interface d'indexation de l'application afin de trouver des objets mis en cache.

La liste qui suit récapitule la procédure à appliquer pour procéder à l'indexation :

- ajout d'index statiques ou dynamiques dans la mappe de sauvegarde
- obtention d'un objet proxy d'index grâce à la méthode getIndex de la mappe d'objet
- transtypage de l'objet proxy vers l'interface d'indexation de l'application utilisée (MapIndex, MapRangeIndex ou une interface d'indexation personnalisée, par exemple)
- utilisation des méthodes qui sont définies dans l'interface d'indexation de l'application pour rechercher les objets mis en cache

8.6+ La classe HashIndex est l'implémentation du plug-in d'indexation intégré qui peut prendre en charge les interfaces d'index d'application intégrées suivantes :

- MapIndex
- MapRangeIndex
- MapGlobalIndex

Vous pouvez également créer vos propres index. Vous pouvez ajouter HashIndex comme index statique ou dynamique dans BackingMap, obtenir l'objet de proxy d'index MapIndex, MapRangeIndex ou MapGlobalIndex et utiliser l'objet de proxy d'index pour rechercher des objets en cache.

8.6+

Index global

L'index global est une extension de la classe HashIndex intégrée qui s'exécute sur les fragments dans les environnements de grille de données répartie et partitionnée. Il suit l'emplacement des attributs indexés et fournit des méthodes efficaces pour rechercher des partitions, des clés, des valeurs ou des entrées à l'aide d'attributs dans les grands environnements de grille de données partitionnée.

Si l'index global est activé dans le plug-in HashIndex intégré, les applications peuvent transtyper un objet proxy d'index en type MapGlobalIndex et l'utiliser pour rechercher des données.

Index par défaut

Si vous souhaitez effectuer une itération dans les clés d'une mappe locale, vous pouvez utiliser l'index par défaut. Cet index ne requiert pas de configuration, mais elle doit être utilisée sur le fragment en utilisant un agent ou une instance ObjectGrid extraits de la méthode ShardEvents.shardActivated(ObjectGrid shard).

Indexation et qualité des données obtenues par une requête d'index

Il faut bien avoir présent à l'esprit que les méthodes de requêtes sur les index ne représentent qu'un cliché des données à un instant t. Les entrées de données ne sont pas verrouillées après l'envoi à l'application des résultats de la requête. L'application doit être consciente que les données peuvent très bien être actualisées après lui avoir été retournées. Supposons, par exemple, que l'application obtienne la clé d'un objet mis en cache grâce à la méthode findAll de MapIndex. Cet objet key retourné est associé dans le cache à une entrée de données. L'application doit être capable d'exécuter la méthode get sur la mappe d'objet pour trouver un objet à partir de l'objet key. Si une autre transaction supprime du cache l'objet données juste avant l'appel à la méthode get, le résultat qui sera retourné sera null.

Points à prendre en considération à propos des performances de l'indexation

L'un des objectifs primordiaux de l'indexation est d'améliorer les performances globales de la mappe de sauvegarde. Une utilisation incorrecte de l'indexation peut compromettre les performances de l'application. Avant d'utiliser l'indexation, les facteurs suivants sont à prendre en considération :

- **Le nombre de transactions simultanées en écriture :** l'indexation peut se produire chaque fois qu'une transaction écrit des données dans une mappe de sauvegarde. Les performances se dégradent si un grand nombre de transactions écrivent en même temps des données dans la mappe au moment où une application lance des requêtes sur l'index.
- **La taille des résultats retournés par une requête :** les performances de la requête déclinent d'autant plus que la taille de ses résultats augmente. Les performances tendent à se dégrader lorsque la taille des résultats atteint 15 % ou plus de la mappe de sauvegarde.
- **Le nombre d'index générés sur la même mappe de sauvegarde :** chaque index consomme des ressources système. Les performances diminuent au fur et à mesure que le nombre d'index augmente sur la mappe de sauvegarde.

Cela dit, l'indexation peut augmenter considérablement les performances des mappes de sauvegarde. C'est particulièrement vrai lorsque la mappe de sauvegarde comporte surtout des opérations de lecture. Les résultats des requêtes

représentent alors un faible pourcentage des entrées de la mappe et seul un petit nombre d'index sont générés sur la mappe.

Planification de plusieurs topologies de centre de données

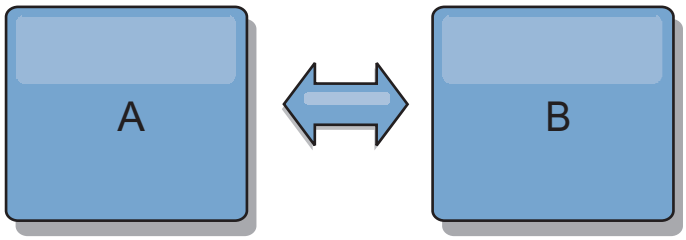
En utilisant la réplication asynchrone multimaître, au moins deux grilles de données peuvent devenir des copies exactes de l'une de l'autre. Chaque grille de données est hébergée dans un domaine de service de catalogue indépendant, avec ses propres de service de catalogue, serveurs de conteneur et un nom unique. Avec la réplication asynchrone multimaître, vous pouvez utiliser des liaisons pour connecter un ensemble de domaines de service de catalogue. Les domaines de service de catalogue sont ensuite synchronisés en utilisant la réplication via ces liaisons. Vous pouvez construire quasiment n'importe quelle topologie via la définition de liaisons entre les domaines de service de catalogue.

Topologies pour la réplication multimaître

Vous disposez de plusieurs options pour choisir la topologie de votre déploiement qui intègre la réplication multimaître.

Liaisons connectant les domaines de service de catalogue

Une infrastructure de grilles de données de réplication est un graphique de domaines de service de catalogue interconnectés avec des liaisons bidirectionnelles. Avec une liaison, deux domaines de service de catalogue peuvent communiquer les modifications de données. Par exemple, la topologie la plus simple est une paire de domaines de service de catalogue avec une liaison unique entre eux. Les domaines de service de catalogue sont nommés par ordre alphabétique: A, B, C, etc., à partir de la gauche. Une liaison peut traverser un réseau WAN (wide area network) pour couvrir une grande distance. Même si la liaison est interrompue, vous pouvez toujours modifier les données dans l'un des domaines de services de catalogue. La topologie rapproche les modifications quand la liaison reconnecte les domaines de service de catalogue. Les liens tentent automatiquement de se reconnecter si la connexion réseau est interrompue.

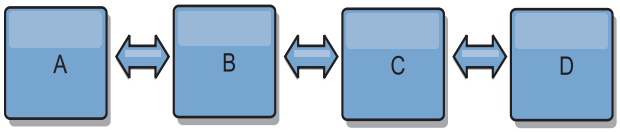


Après avoir établi les liaisons, le produit tente d'abord de rendre chaque domaine de service de catalogue identique. Ensuite, eXtreme Scale tente de maintenir identiques les conditions à mesure que des modifications se produisent dans un domaine de service de catalogue. L'objectif vise à faire de chaque domaine de service de catalogue le miroir exact d'un autre domaine de service de catalogue connecté par les liaisons. Les liaisons de réplication entre les domaines de service de catalogue permettent de copier une modification effectuée dans un domaine de service de catalogue vers les autres domaines de service de catalogue.

Topologies linéaires

Même s'il s'agit d'un déploiement simple, une topologie linéaire montre certaines qualités des liaisons. Tout d'abord, il n'est pas nécessaire qu'un domaine de service de catalogue soit directement connecté à chacun des autres domaines de services

de catalogue pour recevoir des modifications. Le domaine de service de catalogue B extrait les modifications du domaine de service de catalogue A. Le domaine de service de catalogue C reçoit les modifications du domaine de service de catalogue A via le domaine de service de catalogue B, lequel connecte les domaines A et C. De même, le domaine de service de catalogue D reçoit les modifications des autres domaines de service de catalogue via le domaine de service de catalogue C. Cette fonction répartit la charge de distribution des modifications en l'éloignant de la source des modifications.



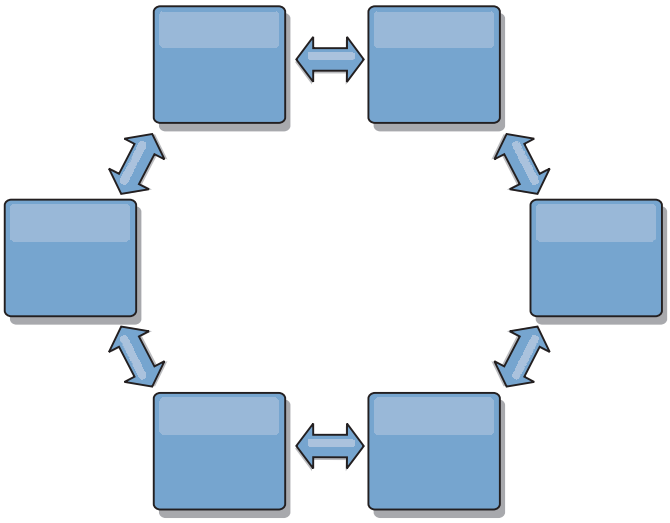
Notez que si le domaine de service de catalogue C est défaillant, les actions suivantes se produisent :

1. Le domaine de service de catalogue D serait orphelin jusqu'au redémarrage du domaine de service de catalogue C.
2. Le domaine de service de catalogue C doit se synchroniser avec le domaine de service de catalogue B, lequel est une copie du domaine de service de catalogue A.
3. Le domaine de service de catalogue D utilise le domaine de service de catalogue C pour se synchroniser avec les modifications des domaines de services de catalogues A et B. Ces modifications se sont produites initialement lorsque le domaine de service de catalogue D étaient orphelin (lorsque le domaine de service de catalogue C était arrêté).

Enfin, les domaines de service de catalogue A, B, C et D, sont de nouveau tous identiques.

Topologies en anneau

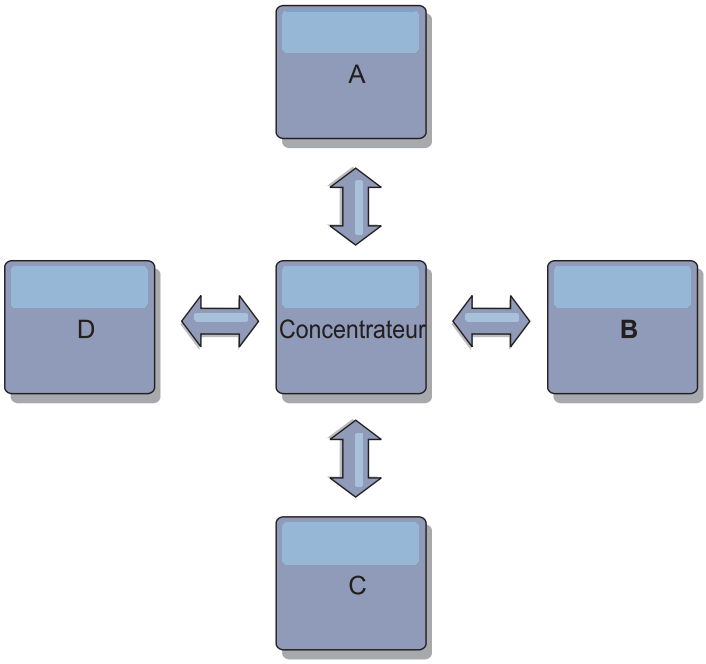
Les topologies en anneau sont un exemple de topologie encore plus résilientes. Lorsqu'un domaine de service de catalogue ou une liaison unique est défaillant, les domaines de service de catalogue restants peuvent encore obtenir des modifications. Les domaines de service de catalogue parcourent l'anneau en s'éloignant de la défaillance. Chaque domaine de service de catalogue possède au maximum deux liens vers d'autres domaines de service de catalogue, quelle que soit la taille de la topologie en anneau. Le délai de propagation des modifications peut être important. Les modifications d'un domaine de service de catalogue particulier peuvent devoir traverser plusieurs liaisons pour que tous les domaines de service de catalogue aient les modifications. Une topologie linéaire a la même caractéristique.



Vous pouvez également déployer une topologie en anneau plus sophistiquée, avec un domaine de service de catalogue racine au centre de l'anneau. Le domaine de service de catalogue racine fait office de point central de rapprochement. Les autres domaines de service de catalogue font office de points distants de rapprochement pour les modifications se produisant dans le domaine de service de catalogue racine. Le domaine de service de catalogue racine peut arbitrer les modifications entre les domaines de service de catalogue. Si une topologie en anneau contient plusieurs anneaux autour d'un domaine de service de catalogue racine, le domaine de service de catalogue ne peut pas arbitrer les modifications dans la partie interne de l'anneau. Toutefois, les résultats de l'arbitrage sont propagés dans les domaines de service de catalogue des autres anneaux.

Topologies en étoile

Avec une topologie en étoile, les modifications parcourent un domaine de service de catalogue du concentrateur. Etant donné que le concentrateur est le seul domaine de service de catalogue intermédiaire spécifié, les topologies en étoile ont une latence inférieure. Le domaine de service de catalogue du concentrateur est connecté à chaque branche de domaine de service de catalogue via une liaison. Le concentrateur répartit les modifications entre les domaines de service de catalogue. Il fait office de point de rapprochement pour les collisions. Dans un environnement soumis à une fréquence élevée de modifications, le concentrateur peut avoir besoin de s'exécuter sur plus de matériels que les branches pour rester synchronisé. WebSphere eXtreme Scale est conçu pour évoluer de manière linéaire, ce qui signifie que l'on peut, si nécessaire, étoffer le concentrateur sans difficultés. Toutefois, si le concentrateur tombe en panne, les modifications ne sont pas distribuées jusqu'à ce qu'il redémarre. Toutes les modifications sur les branches du sous-domaine de services de catalogue seront réparties après la reconnexion du concentrateur.



Vous pouvez également utiliser une stratégie avec les clients intégralement répliqués, une variante de la topologie qui utilise une paire de serveurs s'exécutant comme concentrateur. Chaque client crée une grille de données à conteneur unique autonome avec un catalogue dans la machine virtuelle Java client. Un client utilise sa grille de données pour se connecter au catalogue du concentrateur. Cette connexion provoque la synchronisation du client avec le concentrateur dès que le client obtient une connexion au concentrateur.

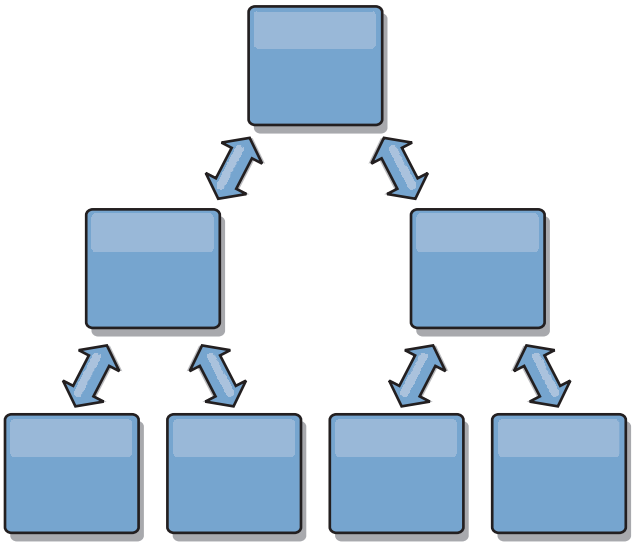
Toutes les modifications effectuées par le client sont locales pour le client et elles sont répliquées vers le concentrateur de manière asynchrone. Le concentrateur joue le rôle de domaine de service de catalogue d'arbitrage, répartissant les modifications à tous les clients connectés. La topologie de clients intégralement répliqués fournit un cache L2 fiable pour un associateur relationnel d'objets comme OpenJPA. Les modifications sont réparties rapidement via le concentrateur entre les machines virtuelles client. Si la taille du cache peut être contenue dans le segment de mémoire disponible, la topologie est une architecture fiable pour ce style de cache L2.

Si nécessaire, utilisez plusieurs partitions pour échelonner le domaine de service de catalogue concentrateur sur plusieurs machines virtuelles Java. Etant donné que toutes les données doivent toujours tenir sur une seule machine virtuelle Java client, plusieurs partitions augmentent la capacité du concentrateur à répartir et à arbitrer les modifications. Cependant, plusieurs partitions ne changent pas la capacité d'un domaine de service de catalogue unique.

Topologies en arbre

Vous pouvez également utiliser un arbre dirigé acyclique. Un arbre acyclique n'a pas de cycles ou de boucles, et une configuration dirigée limite les liaisons aux parents et enfants existants uniquement. Cette configuration est utile pour les topologies disposant d'un grand nombre de domaines de service de catalogue. Dans ces topologies, il n'est pas pratique d'avoir un concentrateur central connecté

à chaque branche. Ce type de topologie peut également être utile lorsque vous devez ajouter des domaines de service de catalogue enfant sans mettre à jour le domaine de service de catalogue racine.



Une topologie en arbre peut toujours avoir un point central de rapprochement dans le domaine de service de catalogue racine. Le deuxième niveau peut toujours fonctionner en tant que point de rapprochement distant pour les modifications se produisant dans le domaine de service de catalogue en dessous. Le domaine de service de catalogue racine peut arbitrer les modifications entre les domaines de service de catalogue sur le deuxième niveau uniquement. Vous pouvez également utiliser des arbres n-aire ayant chacun n enfants à chaque niveau. Chaque domaine de service de catalogue se connecte à *n* liaisons.

Clients intégralement répliqués

Cette variante de la topologie implique une paire de serveurs s'exécutant comme concentrateur. Chaque client crée une grille de données à conteneur unique autonome avec un catalogue dans la machine virtuelle Java client. Un client utilise sa grille de données pour se connecter au catalogue du concentrateur, ce qui provoque la synchronisation du client avec le concentrateur dès que le client obtient une connexion au concentrateur.

Toutes les modifications effectuées par le client sont locales pour le client et elles sont répliquées vers le concentrateur de manière asynchrone. Le concentrateur joue le rôle de domaine de service de catalogue d'arbitrage, répartissant les modifications à tous les clients connectés. La topologie de clients intégralement répliqués fournit un bon cache de niveau 2 pour un associateur relationnel d'objets comme OpenJPA. Les modifications sont réparties rapidement via le concentrateur entre les machines virtuelles client. Tant que la taille du cache peut être contenue dans l'espace de segment mémoire disponible des clients, cette topologie est une architecture tout à fait indiquée pour ce style de cache de niveau 2.

Si nécessaire, utilisez plusieurs partitions pour échelonner le domaine de service de catalogue concentrateur sur plusieurs machines virtuelles Java. Toutes les données devant tenir sur une seule machine virtuelle Java, l'utilisation de partitions multiples augmente la capacité du concentrateur à répartir et à arbitrer les modifications, mais elle ne change pas la capacité d'un domaine de service de catalogue unique.

Considérations de configuration pour les topologies multimaîtres

Tenez compte des points suivants lorsque vous déterminez l'opportunité et la manière d'utiliser des topologies de réplication multimaîtres.

- **Exigences de groupe de mappes**

  Les groupes de mappes doivent avoir les caractéristiques suivantes pour pouvoir répliquer les modifications dans les liaisons d'un domaine de service de catalogue :

  – Le nom ObjectGrid et le nom de groupe de mappes dans un domaine de service de catalogue doivent correspondre au nom ObjectGrid et au nom de groupe de mappes d'autres domaines de service de catalogue. Par exemple, ObjectGrid "og1" et le groupe de mappes "ms1" doivent être configurés dans les domaines de service de catalogue A et B pour pouvoir répliquer les données dans la mappe entre les domaines de service de catalogue.

  – Est une grille de données `FIXED_PARTITION`. Les grilles de données `PER_CONTAINER` ne peuvent pas être répliquées.

  – A le même nombre de partitions dans chaque domaine de service de catalogue. Le groupe de mappes peut ou peut ne pas avoir le même nombre et le même type de répliques.

  – A les mêmes types de données répliquées dans chaque domaine de service de catalogue.

  – Contient les mêmes mappes et modèles de mappes dynamiques dans chaque domaine de service de catalogue.

  – N'utilise pas le gestionnaire d'entités. Un groupe de mappes contenant une mappe d'entités n'est pas répliqué entre les domaines de service de catalogue.

  – N'utilise pas la mise en cache en écriture différée. Un groupe de mappes contenant une mappe qui est configurée avec la prise en charge de l'écriture différée n'est pas répliqué entre les domaines de service de catalogue.

  Tous les ensembles de mappes ayant les caractéristiques ci-dessus commencent à répliquer après que les domaines de service de catalogue dans la topologie ont été démarrés.

- **Chargeurs de classe avec plusieurs domaines de service de catalogue**

  Les domaines de service de catalogue doivent avoir accès à toutes les classes qui sont utilisées comme clés et valeurs Toutes les dépendances doivent être reflétées dans tous les chemins d'accès aux classes des machines virtuelles Java (JVM) de conteneur de la grille de données de tous les domaines. Si un plug-in CollisionArbiter extrait la valeur d'une entrée de cache, les classes correspondant aux valeurs doivent être présentes pour le domaine qui démarre l'arbitre.

Remarques sur les chargeurs dans une topologie multimaître

Lorsque vous utilisez des chargeurs dans une topologie multimaître, vous devez envisager les problèmes éventuels de collision et de maintenance des informations de révision. La grille de données conserve les informations de révision sur les éléments de façon à ce que les collisions puissent être détectées lorsque d'autres fragments primaires dans la configuration y écrivent des entrées. Lorsque des entrées sont ajoutées à partir d'un chargeur, ces informations de révision ne sont pas incluses et l'entrée prend une nouvelle révision. Etant donné que la révision de l'entrée semble être une nouvelle insertion, une fausse collision peut se produire si un autre fragment primaire modifie également cet état ou insère les mêmes informations à partir d'un chargeur.

Les modifications de réplication appellent la méthode get sur le chargeur avec la liste des clés qui ne sont pas déjà dans la grille de données, mais qui vont être modifiées lors de la transaction de réplication. Lorsque la réplication se produit,

ces entrées sont des entrées de collision. Lorsque les collisions sont arbitrées et que la révision est appliquée, une mise à jour par lots est appelée sur le chargeur pour appliquer les modifications à la base de données. Toutes les mappes qui ont été modifiées dans la fenêtre de révision sont mises à jour dans la même transaction.

L'énigme de préchargement

Supposons une topologie avec les deux centres de données A et B qui ont des bases de données indépendantes, mais seul le centre de données A a une grille active. Lorsque vous établissez une liaison entre les centres de données pour une configuration multimaître, les grilles de données dans le centre de données A commencent à envoyer les données aux nouvelles grilles dans le centre de données B, ce qui crée une collision avec chaque entrée. Un autre problème est l'existence de données dans la base de données du centre de données B, mais qui ne figurent pas dans la base de données du centre de données A. Ces lignes ne sont pas remplies et arbitrées, ce qui génère des incohérences qui ne sont pas résolues.

Solution de l'énigme de préchargement

Etant donné que les données qui se trouvent uniquement dans la base de données ne peuvent pas comporter des révisions, vous devez toujours précharger complètement la grille de données à partir de la base de données locale pour établir la liaison multimaître. Ensuite, les deux grilles de données peuvent réviser et arbitrer les données, pour atteindre finalement un état cohérent.

L'énigme du cache partiel

Avec un cache partiel, la première application tente de trouver des données dans la grille de données. Si les données ne sont pas dans la grille de données, elles sont recherchées dans la base de données à l'aide du chargeur. Les entrées sont supprimées de la grille de données régulièrement pour maintenir une mémoire cache de petite taille.

Ce type de mémoire cache peut être problématique dans un scénario de configuration multimaître, car les entrées dans la grille de données ont des métadonnées de révision qui permettent de détecter quand des collisions se produisent et de déterminer qui a effectué les modifications. Lorsque des liaisons entre les centres de données ne fonctionnent pas, un centre de données peut mettre à jour une entrée et ensuite éventuellement mettre à jour la base de données et invalider l'entrée dans la grille de données. Lorsque la liaison est rétablie, les centres de données tentent de synchroniser les révisions les unes par rapport aux autres. Toutefois, étant donné que la base de données a été mise à jour et que l'entrée de la grille de données a été invalidée, la modification est perdue du point de vue du centre de données qui s'est arrêté. En conséquence, les deux côtés de la grille de données sont désynchronisés et ne sont pas cohérents.

Solution de l'énigme de cache partiel

Topologie en étoile :

Vous pouvez exécuter le chargeur uniquement dans la topologie en étoile pour maintenir la cohérence des données lors de l'extension de la grille de données. Toutefois, si vous envisagez ce déploiement, notez que les chargeurs peuvent permettre à la grille de données d'être partiellement chargée, ce qui implique qu'un expulseur a été configuré. Si les rayons de la configuration sont des caches partiels, les échecs en mémoire cache n'ont aucun moyen d'extraire des données de la base

de données. En raison de cette restriction, vous devez utiliser une topologie de cache complètement remplie avec une configuration en étoile.

Invalidations et expulsion

L'invalidation crée des incohérences entre la grille de données et la base de données. Les données peuvent être supprimées de la grille de données, à l'aide d'un programme ou par l'expulsion. Lorsque vous développez votre application, sachez que le traitement des révisions ne réplique pas les modifications invalidées, ce qui provoque des incohérences entre les fragments primaires.

Les événements d'invalidation ne sont pas des modifications de l'état du cache et n'entraînent pas de réplication. Tous les expulseurs configurés s'exécutent indépendamment des autres expulseurs dans la configuration. Par exemple, vous pouvez avoir un expulseur configuré pour un seuil de mémoire dans un domaine de service de catalogue, mais un type d'expulseur différent moins agressif dans l'autre domaine de service de catalogue lié. Lorsque des entrées de grille de données sont supprimées en raison de la règle de seuil de mémoire, les entrées dans l'autre domaine de service de catalogue ne sont pas affectées.

Mises à jour de la base de données et invalidation de la grille de données

Des problèmes se produisent lorsque vous mettez à jour la base de données directement en arrière-plan lors de l'appel de l'invalidation dans la grille de données pour les entrées mises à jour dans une configuration multimaître. Ce problème se produit, car la grille de données ne peut pas répliquer la modifications dans les autres fragments primaires jusqu'à ce qu'un accès de cache transfère l'entrée vers la grille de données.

Plusieurs programmes d'écriture dans une seule base de données logique

Lorsque vous utilisez une seule base de données avec plusieurs fragments primaires qui sont connectés par l'intermédiaire d'un chargeur, des conflits transactionnels se produisent. Votre implémentation de chargeur doit gérer ces types de scénarios.

Mise en miroir des données à l'aide de la réplication multimaître

Vous pouvez configurer des bases de données indépendantes qui sont connectées à des domaines de service de catalogue indépendants. Dans cette configuration, le chargeur peut envoyer les modifications d'un centre de données vers un autre.

Considérations de conception pour la réplication multimaître

Lors de l'implémentation de la réplication multimaître, vous devez tenir compte de divers éléments dans votre conception, tels que l'arbitrage, les liaisons et les performances.

Points concernant l'arbitrage à prendre en considération dans la conception des topologies

Des collisions entre des modifications peuvent se produire s'il est possible à des enregistrements identiques d'être modifiés simultanément en deux endroits différents. Configurez chaque domaine de service de catalogue pour que les domaines aient le même nombre de processeurs, la même quantité de mémoire et le même nombre de ressources réseau. Vous remarquerez sans doute que des

domaines de service de catalogue d'exécution gérant les collisions de modifications (arbitrage) utilisent plus de ressources que d'autres domaines de service de catalogue. Les collisions sont détectées de manière automatique. Elles sont traitées avec l'un des deux mécanismes suivants :

- **Arbitre par défaut** : le protocole par défaut doit utiliser les modifications du domaine de service de catalogue occupant la position la moins basse alphabétiquement. Par exemple, si les domaines de service de catalogue A et B génèrent un conflit pour un enregistrement, la modification du domaine de service de catalogue B est ignorée. Le domaine de service de catalogue A conserve sa version et l'enregistrement dans le domaine de service de catalogue B est modifié pour qu'il corresponde à l'enregistrement du domaine de service de catalogue A. Ce comportement s'applique également aux applications où les utilisateurs ou les sessions sont normalement liés ou ont une affinité à l'une des grilles de données.
- **Arbitre personnalisé** : les applications peuvent fournir un arbitre personnalisé. Lorsqu'un domaine de service de catalogue détecte une collision, il démarre l'arbitre. Pour plus d'informations sur le développement d'un arbitre personnalisé utile, voir Développement d'arbitres personnalisés pour la réplication multi-maître.

Pour les topologies dans lesquelles les collisions sont possibles, songez à implémenter une topologie en étoile ou en arbre. Les deux topologies sont propices à éviter les collisions constantes, ce qui peut se produire dans les scénarios suivants :

1. Plusieurs domaines de service de catalogue sont affectés par une collision.
2. Chaque domaine de service de catalogue gère la collision en local, ce qui produit des révisions.
3. Les révisions entrent en collision, d'où des révisions de révisions.

Pour éviter les collisions, choisissez un domaine de service de catalogue spécifique, appelé *domaine de service de catalogue d'arbitrage* comme arbitre des collisions d'un sous-ensemble de domaines de service de catalogue. Par exemple, une topologie en étoile pourra utiliser le concentrateur comme gestionnaire de collisions. Le gestionnaire de collisions ignore toutes les collisions qui sont détectées par les sous-domaines de service de catalogue. Le domaine de service de catalogue du concentrateur crée des révisions, empêchant les révisions de collisions inattendues. Le domaine de service de catalogue qui est affecté à la gestion des collisions doit se lier à tous les domaines dont il est chargé de traiter les collisions. Dans une topologie en arbre, tous les domaines parent internes traitent les collisions pour leurs enfants immédiats. En revanche, si vous utilisez une topologie en anneau, vous ne pouvez pas désigner un domaine de service de catalogue dans le fichier comme arbitre.

Le tableau qui suit récapitule les approches en matière d'arbitrage qui sont les plus compatibles avec les diverses topologies.

Tableau 1. Approches en matière d'arbitrage. Ce tableau énonce si l'arbitrage entre applications est compatible avec les diverses topologies.

Topologie	Arbitrage d'application	Notes
Ligne de deux domaines de service de catalogue	Oui	Choisissez un domaine de service de catalogue comme arbitre.

Tableau 1. Approches en matière d'arbitrage (suite). Ce tableau énonce si l'arbitrage entre applications est compatible avec les diverses topologies.

Topologie	Arbitrage d'application	Notes
Ligne de trois domaines de service de catalogue	Oui	Le domaine de service de catalogue du milieu doit être l'arbitre. Assimilez ce domaine de service de catalogue au concentrateur dans une topologie en étoile simple.
Ligne de plus de trois domaines de service de catalogue	Non	L'arbitrage d'application n'est pas pris en charge.
Concentrateur avec n "rayons"	Oui	Le concentrateur avec des liens vers toutes les branches doit être le domaine de service de catalogue d'arbitrage.
Anneau de N domaines de service de catalogue	Non	L'arbitrage d'application n'est pas pris en charge.
Arbre dirigé acyclique (arbre n-aire)	Oui	Tous les noeuds racine doivent évaluer leurs descendants directs uniquement.

Points concernant les liens à prendre en considération dans la conception des topologies

Dans l'idéal, une topologie comprend le minimum de liens tout en optimisant les compromis entre les temps d'attente des modifications, la tolérance aux pannes et les caractéristiques de performances.

- **Temps d'attente des modifications**

  Le temps d'attente de modification est déterminé par le nombre de domaines de service de catalogue intermédiaires par lequel un changement doit passer avant d'arriver à un domaine de service de catalogue spécifique.

  Une topologie a le meilleur temps d'attente lorsqu'elle élimine les domaines de service de catalogue intermédiaires en liant chacun des domaines de service de catalogue à chacun des autres domaines de service de catalogue. Toutefois, un domaine de service de catalogue doit effectuer la réplication par rapport à son nombre de liens. Pour les topologies de grande taille, le nombre de liens à définir peut entraîner une charge administrative.

  La vitesse à laquelle une modification est copiée vers les autres domaines de service de catalogue dépend de facteurs supplémentaires, tels que :

  – Bande passante du processeur et du réseau dans le domaine de service de catalogue source
  – Nombre de domaines de service de catalogue intermédiaire et de liens entre la source et la cible du domaine de service de catalogue source et cible
  – Ressources en processeur et en réseau disponibles pour les domaines de service de catalogue source, cible et intermédiaires

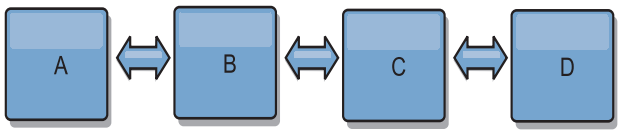
- **Tolérance aux pannes**

  La tolérance aux pannes est déterminée par le nombre de chemins existant entre deux domaines de service de catalogue pour la réplication des modifications.

  Si vous ne disposez que d'un seul lien entre une paire de domaines de service de catalogue, une défaillance de lien empêche la propagation des modifications. De même, les modifications ne sont pas propagées entre les domaines de service de catalogue si un incident de liaison se produit sur les domaines intermédiaires.

Votre topologie pourrait avoir un lien unique d'un domaine de service de catalogue vers un autre de sorte que le lien passe par des domaines intermédiaires. Dans ce cas, les modifications ne sont pas propagées si l'un des domaines de service de catalogue intermédiaires est défaillant.

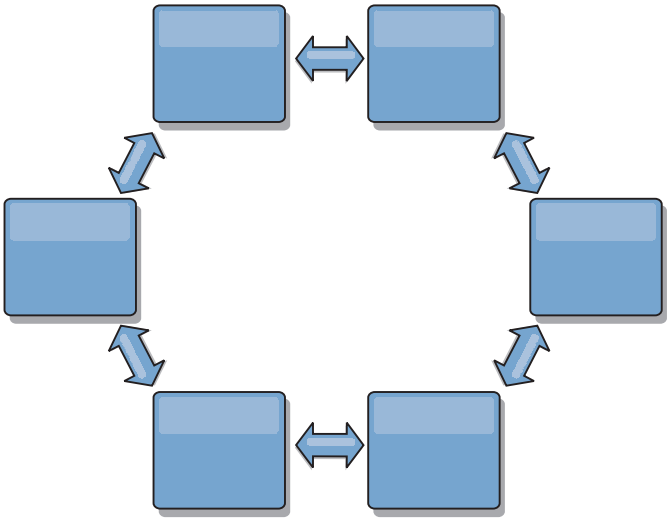
Supposons la topologie linéaire à quatre domaines de service de catalogue A, B, C et D :



Si l'une de ces conditions existe, le domaine D ne voit pas les modification de A :

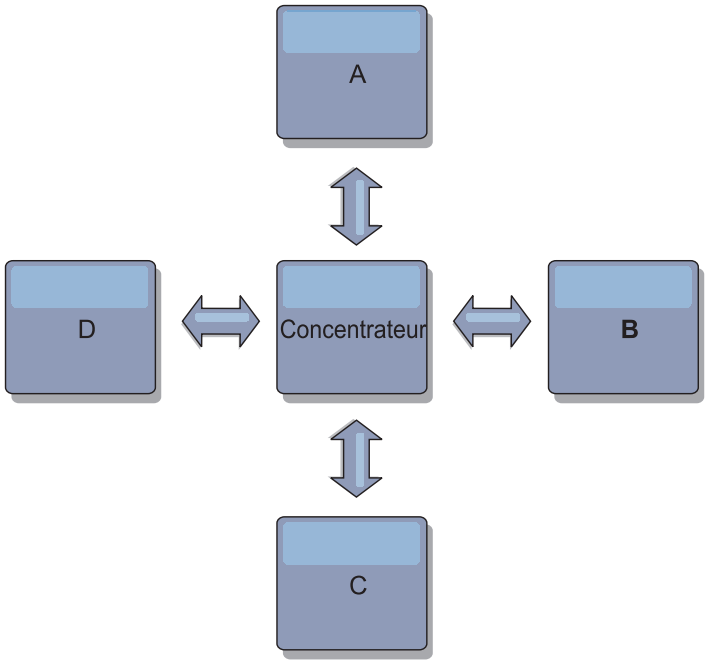
- Le domaine A est actif et B est arrêté.
- Les domaines A et B sont actifs et C est arrêté.
- Le lien entre A et B ne fonctionne pas.
- Le lien entre B et C ne fonctionne pas.
- Le lien entre C et D est arrêté.

En revanche, avec une topologie en anneau, chaque domaine de service de catalogue peut recevoir les modifications dans un sens ou dans l'autre.



Par exemple, si un service de catalogue donné de votre topologie en anneau est arrêté, les deux domaines contigus peuvent toujours extraire les modifications directement de l'autre.

Toutes les modifications sont propagées via le concentrateur. Par conséquent, contrairement aux topologies linéaires et en anneau, la conception en étoile peut tomber en panne si le concentrateur est défaillant.



Un domaine de service de catalogue unique est résilient à un certain degré de perte de service. Cependant, les incidents les plus importants, tels que les indisponibilités de réseau étendu ou les pertes de liaisons entre les centres de données physiques peuvent perturber les domaines de service de catalogue.

- **Liaison et performances**

  Le nombre de liaisons définies sur un domaine le service de catalogue affecte les performances. Un plus grand nombre de liaisons utilisent davantage de ressources et les performances de réplication peuvent baisser. La possibilité d'extraire les modifications pour un domaine A via d'autres domaines empêche le domaine A de répliquer ses transactions partout. La charge de la répartition des modifications dans un domaine est limitée par le nombre de liaisons qu'il utilise et non pas par le nombre de domaines dans la topologie. Cette propriété est synonyme d'évolutivité et les domaines de la topologie peuvent partager la charge de la répartition des modifications.

  Un domaine de service de catalogue peut extraire les modifications indirectement via d'autres domaines de service de catalogue. Supposons une topologie linéaire avec cinq domaines de service de catalogue.

  ```
  A <=> B <=> C <=> D <=> E
  ```

  – A extrait les modifications de B, C, D, et E via B
  – B extrait les modifications directement de A et de C et les modifications de D et de E via C
  – C extrait les modifications directement de B et de D et les modifications de A via B et de E via D
  – D extrait les modifications directement de C et de E et les modifications de A et de B via C
  – E extrait les modifications directement de D et les modifications de A, B et C via D

  La charge de la répartition dans les domaines de service de catalogue A et E est la plus faible, car ils ont chacun une seule liaison à un domaine de service de catalogue unique. Les domaines B, C et D ont chacun une liaison avec deux domaines. Par conséquent, la charge de la répartition dans les domaines B, C, et D est le double de celle des domaines A et E. La charge de travail dépend du

nombre de liaisons dans chaque domaine et non pas du nombre total de domaines dans la topologie. Par conséquent, la répartition de charge décrite demeurerait constante, même si la ligne contenait 1 000 domaines.

Considérations relatives aux performances de réplication multimaître

Tenez compte des limitations suivantes lorsque vous utilisez des topologies de réplication multimaître :

- **Optimisation de la répartition des modifications**, comme expliquée dans la section précédente.
- **Performances des liens de réplication** WebSphere eXtreme Scale crée un seul socket TCP/IP entre n'importe quelle paire de machines virtuelles Java. Tout le trafic entre les machines virtuelles Java passe par le socket unique, y compris le trafic de la réplication multimaître. Les domaines de service de catalogue sont hébergés dans au moins *n* machines virtuelles Java pour fournir au minimum *n* liaisons TCP aux domaines de services homologues. Ainsi, les domaines de service de catalogue avec un plus grand nombre de conteneurs offrent de meilleures performances de réplication. Un plus grand nombre de conteneurs requiert davantage de processeurs et de ressources réseau.
- **Le support de l'optimisation de la fenêtre dynamique TCP et RFC 1323** RFC 1323 à chaque extrémité d'une liaison renvoie plus de données pour un aller-retour. Ce support augmente le débit en développant la capacité de la fenêtre d'un facteur d'environ 16 000.

  Notez que les sockets TCP utilisent un mécanisme de fenêtre dynamique pour contrôler le flux des données en vrac. Ce mécanisme limite généralement le socket à 64 Ko pour un intervalle d'aller-retour. Si l'intervalle aller-retour est de 100 ms, la bande passante est limitée à 640 Ko/s sans optimisation supplémentaire. L'utilisation intégrale de la bande passante disponible sur un lien peut nécessiter une optimisation qui est spécifique au système d'exploitation. La plupart des systèmes d'exploitation comportent des paramètres d'optimisation, y compris des options RFC 1323, permettant d'améliorer le débit sur les liaisons à forte latence.

  Plusieurs facteurs peuvent affecter les performances de la réplication :

  – Vitesse d'extraction des modifications par eXtreme Scale.
  – Vitesse à laquelle eXtreme Scale peut traiter les demandes de réplication d'extraction.
  – Capacité de la fenêtre dynamique.
  – Avec l'optimisation de la mémoire tampon réseau aux deux extrémités d'une liaison, eXtreme Scale extrait les modification sur le socket de manière efficace.
- **Sérialisation des objets** Toutes les données doivent être sérialisables. Si un domaine de service de catalogue n'utilise pas `COPY_TO_BYTES`, il doit utiliser la sérialisation Java ou ObjectTransformers pour optimiser les performances de sérialisation.
- Par défaut **la compression** WebSphere eXtreme Scale compresse toutes les données envoyées entre les domaines de service de catalogue. Vous ne pouvez désactiver actuellement la compression.
- **Optimisation de la mémoire** L'utilisation de la mémoire pour une topologie de réplication multimaître est largement indépendante du nombre de domaines de service de catalogue dans la topologie.

  La réplication multimaître ajoute un temps de traitement fixe par entrée de mappe pour la gestion des versions. Chaque conteneur suit également une quantité fixe de données pour chaque domaine de service de catalogue dans la

topologie. Une topologie à deux domaines de service de catalogue utilise approximativement la même quantité de mémoire qu'une topologie à 50 domaines de service de catalogue. WebSphere eXtreme Scale n'utilise pas de journaux de relecture ou de files d'attente similaires dans son implémentation. Ainsi, aucune structure de récupération n'est prête si la liaison de réplication n'est pas disponible pendant un certain temps et redémarre ensuite.

Interopérabilité avec d'autres produits

Vous pouvez intégrer WebSphere eXtreme Scale dans d'autres produits, tels que WebSphere Application Server et WebSphere Application Server Community Edition.

WebSphere Application Server

Vous pouvez intégrer WebSphere Application Server à divers éléments de votre configuration WebSphere eXtreme Scale. Vous pouvez déployer des applications de grille de données et utiliser WebSphere Application Server pour héberger les serveurs de conteneur et de catalogue. Vous pouvez aussi utiliser un environnement mixte dans lequel WebSphere eXtreme Scale Client est installé dans l'environnement WebSphere Application Server avec un catalogue autonome et des serveurs de conteneur. Vous pouvez également utiliser la sécurité WebSphere Application Server dans votre environnement WebSphere eXtreme Scale.

Produits WebSphere Business Process Management and Connectivity

Les produits WebSphere Business Process Management and Connectivity, notamment WebSphere Integration Developer, WebSphere Enterprise Service Bus, et WebSphere Process Server, s'intègrent dans les systèmes back-end, tels que CICS, le services Web, les bases de données ou les rubriques et les files d'attente JMS. Vous pouvez ajouter WebSphere eXtreme Scale à la configuration afin de mettre en cache les sorties de ces systèmes back-end afin d'améliorer les performances globales de votre configuration.

WebSphere Commerce

WebSphere Commerce permet d'optimiser la mise en cache WebSphere eXtreme Scale en tant qu'élément de remplacement de la mise en cache dynamique. En éliminant les entrées en double dans la mémoire cache dynamique et les fréquentes invalidations nécessaires pour maintenir la mémoire cache synchronisée dans les situations de stress important, vous pouvez améliorer les performances, la mise à l'échelle et la haute disponibilité.

WebSphere Portal

Vous pouvez rendre persistantes des sessions HTTP depuis WebSphere Portal dans une grille de données dans WebSphere eXtreme Scale. En outre, IBM® Web Content Manager dans IBM WebSphere Portal peut utiliser des instances de mémoire cache dynamique pour stocker du contenu qui est extrait du gestionnaire de contenu Web lorsque la mise en cache avancée est activée. WebSphere eXtreme Scale propose une implémentation de cache dynamique qui stocke le contenu mis en cache dans une grille de données élastique au lieu d'utiliser l'implémentation de mise en cache dynamique par défaut.

WebSphere Application Server Community Edition

WebSphere Application Server Community Edition peut partager l'état des sessions, mais d'une manière peu efficace et non évolutive. WebSphere eXtreme Scale fournit une couche de persistance répartie à hautes performances qui peut servir à répliquer l'état mais sans s'intégrer facilement aux autres serveurs d'applications extérieurs à WebSphere Application Server. Vous pouvez intégrer ces deux produits pour offrir une solution de gestion de session évolutive.

WebSphere Real Time

Avec le support pour WebSphere Real Time, l'offre Java temps réel la plus efficace, WebSphere eXtreme Scale permet aux applications Extreme Transaction Processing (XTP) d'avoir des temps de réponse plus cohérents et plus prévisibles.

Contrôle

WebSphere eXtreme Scale peut être surveillé à l'aide de plusieurs solutions de surveillance d'entreprise couramment utilisées. Des agents de plug-in sont intégrés pour IBM Tivoli Monitoring et Hyperic HQ dont le rôle consiste à surveiller WebSphere eXtreme Scale à l'aide de beans de gestion accessibles publiquement. CA Wily Introscope utilise l'instrumentation de méthode Java pour capturer les statistiques.

.NET 8.6+

Environnements Microsoft Visual Studio, IIS et .NET

Pour plus d'informations sur la prise en charge des environnements Microsoft Visual Studio, IIS et .NET pris en charge, voir «Remarques relatives à Microsoft .NET», à la page 67.

Planification de la configuration

Avant de configurer le matériel ou logiciel, vous devez tenir compte des points suivants.

Planification des ports réseau

WebSphere eXtreme Scale est un cache réparti qui nécessite l'ouverture de ports pour communiquer avec les machines virtuelles Java. Planifiez et contrôlez vos ports, en particulier dans un environnement comportant un pare-feu, et lorsque vous utilisez un service de catalogue et des conteneurs sur plusieurs ports.

Important : Lorsque vous spécifiez des numéros de port, évitez de définir les ports qui se trouvent dans la plage éphémère du système d'exploitation afin d'éviter les conflits de ports.

Domaine de service de catalogue

Un domaine de service de catalogue nécessite que soient définis les ports suivants :

peerPort
Spécifie le port qui permet au gestionnaire de haute disponibilité (HA) de communiquer entre serveurs de catalogue homologues dans une pile TCP. Dans WebSphere Application Server, ce paramètre est hérité par la configuration de port du gestionnaire haut disponibilité.

clientPort
Spécifie le port qui permet aux serveurs de catalogue d'accéder aux données des services de catalogue. Dans WebSphere Application Server, ce port est défini par le biais de la configuration du domaine de service de catalogue.

listenerPort
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.

Valeur par défaut : `2809`

JMXConnectorPort
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

JMXServicePort
Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)

Valeur par défaut : `1099` pour les serveurs de catalogue

jvmArgs (facultatif)
Spécifie la liste des arguments JVM (Java virtual machine). Lorsque la sécurité est activée, vous devez utiliser l'argument suivant dans le script **startOgServer** ou **startXsServer** pour configurer le port SSL (Secure Socket Layer) : `-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>`.

Serveurs de conteneur

Les serveurs de conteneur WebSphere eXtreme Scale requièrent également que plusieurs ports soient en fonctionnement. Par défaut, le serveur de conteneur eXtreme Scale génère automatiquement son port de gestionnaire HA et son port d'écoute avec des ports dynamiques. Pour un environnement qui dispose d'un pare-feu, il est avantageux pour vous de planifier et de contrôler les ports. Pour les serveurs de conteneur à démarrer avec des ports spécifiques, vous pouvez utiliser les options suivantes dans la commande **startOgServer** ou **startXsServer**.

haManagerPort
Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, un port libre est choisi. Cette propriété est ignorée dans les environnements WebSphere Application Server.

listenerPort
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par

BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.

Valeur par défaut : `2809`

JMXConnectorPort
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

JMXServicePort
Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)

Valeur par défaut : `1099` pour les serveurs de catalogue

xioChannel.xioContainerTCPSecure.Port

Obsolète : 8.6+ La propriété est obsolète. La valeur qui est spécifiée par la propriété listenerPort est utilisée.
Indique le numéro de port SSL de eXtremeIO sur le serveur. Cette propriété est utilisée uniquement lorsque la propriété **transportType** a la valeur `SSL-Supported` ou `SSL-Required`.

xioChannel.xioContainerTCPNonSecure.Port

Obsolète : 8.6+ La propriété est obsolète. La valeur qui est spécifiée par la propriété listenerPort est utilisée.
Indique le numéro de port d'écoute non sécurisé de eXtremeIO sur le serveur. Si vous ne définissez pas de valeur, un port éphémère est utilisé. Cette propriété est utilisée uniquement lorsque la propriété **transportType** a la valeur `TCP/IP`.

jvmArgs (facultatif)
Spécifie la liste des arguments JVM (Java virtual machine). Lorsque la sécurité est activée, vous devez utiliser l'argument suivant dans le script **startOgServer** ou **startXsServer** pour configurer le port SSL (Secure Socket Layer) : `-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>`.

Une planification du port de contrôle est essentielle lorsque des centaines de machines virtuelles Java sont démarrées dans un serveur. Si un conflit de port existe, les serveurs de conteneur ne démarrent pas.

Clients

Les clients WebSphere eXtreme Scale peuvent recevoir des rappels de serveurs lorsque vous utilisez l'API DataGrid ou plusieurs autres commandes. Utilisez la propriété **listenerPort** dans le fichier de propriétés du client afin de spécifier le port sur lequel le client écoute les rappels à partir du serveur.

haManagerPort
Indique le numéro de port utilisé par le gestionnaire de haute disponibilité.

Si cette propriété n'est pas définie, un port libre est choisi. Cette propriété est ignorée dans les environnements WebSphere Application Server.

arguments JVM (facultatifs)
Spécifie la liste des arguments JVM (Java virtual machine). Lorsque la sécurité est activée, vous devez utiliser la propriété système suivante lors du démarrage du processus client : `-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>`.

listenerPort
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.

Valeur par défaut : `2809`

Ports dans WebSphere Application Server

- 8.6+ La valeur **listenerPort** est héritée. La valeur est différente selon le type de transport que vous utilisez :
  - Si vous utilisez le transport ORB, la valeur **BOOTSTRAP_ADDRESS** pour chaque serveur d'applications WebSphere Application Server est utilisée.
  - Si vous utilisez le transport IBM eXtremeIO, la valeur **XIO_ADDRESS** est utilisée.
- Les valeurs **haManagerPort** et **peerPort** sont héritées de la valeur **DCS_UNICAST_ADDRESS** pour chaque serveur d'applications WebSphere Application Server.

Vous pouvez définir un domaine de services de catalogue dans la console d'administration. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307.

Vous pouvez afficher les ports d'un serveur particulier en cliquant sur un des chemins suivants dans la console d'administration :

- WebSphere Application Server Network Deployment Version 7.0 et versions suivantes : **Servers** > **Server Types** > **Serveurs d'applications WebSphere** > *server_name* > **Ports** > ***port_name***.

Planification de l'utilisation d'IBM eXtremeMemory

En configurant eXtremeMemory, vous pouvez stocker des objets dans la mémoire native plutôt que dans le segment de mémoire Java. Lorsque vous configurez eXtremeMemory, vous pouvez autoriser la quantité par défaut de mémoire à utiliser ou calculer la quantité de mémoire que vous voulez dédier à eXtremeMemory.

Avant de commencer

- Pour plus d'informations sur eXtremeMemory, voir IBM eXtremeMemory.
- Vous devez utiliser des ensembles de mappes qui ont toutes les mappes configurées avec le mode de copie `COPY_TO_BYTES` ou `COPY_TO_BYTES_RAW`. Si aucune mappe du groupe de mappes n'est utilisée par l'un de ces modes de copie, les objets sont stockés dans le segment de mémoire Java .
- Linux Vous devez avoir installé la ressource partagée libstdc++.so.5. Utilisez le programme d'installation du module 64 bits de votre distribution Linux pour

installer le fichier de ressources requis. Pour plus d'informations, voir «Traitement des incidents IBM eXtremeMemory», à la page 684.

- Vous ne pouvez pas utiliser eXtremeMemory dans les scénarios de configuration suivants :
  - Lorsque vous utilisez des plug-in d'expulseur personnalisés.
  - Lorsque vous utilisez des index composites.
  - Lorsque vous utilisez des index dynamiques.
  - Lorsque vous utilisez des chargeurs intégrés en écriture différée.
- Vous devez disposer d'une grille de données existante à partir de laquelle vous pouvez déterminer les tailles de mappe totales.

Pourquoi et quand exécuter cette tâche

Par défaut, eXtremeMemory utilise 25 % de la mémoire totale sur le système. Vous pouvez modifier cette valeur par défaut à l'aide de la propriété **maxXMSize** qui définit le nombre de mégaoctets à consacrer pour une utilisation par eXtremeMemory.

Procédure

Facultatif : Planifiez la valeur de propriété **maxXMSize** à utiliser. Cette valeur définit la quantité maximale de mémoire, en mégaoctets, utilisée par le serveur pour le stockage eXtremeMemory.

1. Dans votre configuration existante, déterminez la taille par entrée. Exécutez la commande **xscmd -c showMapSizes** pour déterminer cette taille.
2. Calculez la valeur **maxXMSize**. Pour obtenir la taille totale maximale des entrées (*maximum_total_size*), multipliez *size_per_entry* * *maximum_number_of_entries*. N'utilisez pas plus de 60 % de **maxXMSize** pour tenir compte du traitement des métadonnées. Multipliez *maximum_total_size** 1.65 pour la valeur **maxXMSize**.

Que faire ensuite

Sécurité

WebSphere eXtreme Scale permet de sécuriser l'accès aux données et l'intégration de fournisseurs de sécurité externes.

Remarque : Dans un magasin de données non mis en cache, une base de données, par exemple, il est probable que certaines fonctions pré-intégrées de sécurité ne vous serviront à rien pour la configuration ou l'activation. Cependant, une fois vos données mises en cache avec eXtreme Scale, vous devez prendre en compte le fait que vos fonctions de sécurité du dorsal ne sont plus actives. Vous pouvez configurer la sécurité de eXtreme Scale aux niveaux nécessaires, de sorte que votre nouvelle architecture mise en cache soit également sécurisée.
Vous trouverez ci-dessous un bref récapitulatif des fonctions de sécurité de eXtreme Scale. Pour des informations plus détaillées sur la configuration de la sécurité, voir *Guide d'administration* et *Guide de programmation*.

Notions de base sur la sécurité répartie

La sécurité répartie eXtreme Scale se base sur trois concepts :

Authentification approuvée
: Possibilité de déterminer l'identité du demandeur. WebSphere eXtreme Scale prend en charge l'authentification client-serveur et serveur-serveur.

Autorisation
Possibilité d'octroyer des droits d'accès au demandeur. WebSphere eXtreme Scale prend en charge différentes autorisations pour des opérations diverses.

Transfert sécurisé
Transmission sécurisé des données sur le réseau. WebSphere eXtreme Scale prend en charge les protocoles Transport Layer Security/Secure Sockets Layer (TLS/SSL).

Authentification

WebSphere eXtreme Scale prend en charge les structures de serveurs clients répartis. Une infrastructure de sécurité du serveur client est en place pour sécuriser l'accès aux serveurs eXtreme Scale. Par exemple, lorsque l'authentification est requise par le serveur eXtreme Scale, un client eXtreme Scale doit fournir ses informations d'identification pour s'authentifier sur le serveur. Ces informations peuvent être un nom d'utilisateur et un mot de passe, un certificat client, un ticket Kerberos ou des données présentées dans un format choisi par le client et le serveur.

Autorisation

Les autorisations WebSphere eXtreme Scale sont basées sur des objets et des permissions. Vous pouvez utiliser le service JAAS (Java Authentication and Authorization Services) pour autoriser l'accès, ou vous pouvez choisir une approche personnalisée, telle que Tivoli Access Manager (TAM), pour gérer les autorisations. Les autorisations suivantes peuvent être octroyées à un client ou un groupe :

Autorisation de mappes
Effectuez des opérations d'insertion, de lecture, de mise à jour, d'expulsion ou de suppression sur les mappes.

Autorisation ObjectGrid
Lancez des requêtes sur un objet ou une entité sur les objets ObjectGrid.

Autorisation de l'agent DataGrid
Permet aux agents DataGrid d'être déployés en une base de données ObjectGrid.

Autorisation de mappes côté serveur
Répliquez une mappe de serveur côté client ou créez un index dynamique pour la mappe de serveur.

Autorisation d'administration
Effectuez des tâches d'administration.

Sécurité du transfert

Pour sécuriser la communication du serveur client, WebSphere eXtreme Scale prend en charge les protocoles TLS/SSL. Ces protocoles fournissent une sécurité de couche de transport, avec des fonctions d'authentification, d'intégrité et de confidentialité pour une connexion sécurisée entre le client eXtreme Scale et le serveur.

Sécurité de grille

Dans un environnement sécurisé, un serveur doit être capable de vérifier l'authenticité d'un autre serveur. WebSphere eXtreme Scale utilise un mécanisme de clé secrète partagée dans ce but. Ce mécanisme est similaire à un mot de passe partagé. Tous les serveurs eXtreme Scale s'accordent sur une clé secrète partagée. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si la clé secrète du serveur tentant de se joindre correspond à la clé sur serveur principal, le serveur peut se joindre à la grille. Dans le cas contraire, la requête de jointure est rejetée.

L'envoi d'une clé secrète en texte clair n'est pas sécurisé. L'infrastructure de sécurité eXtreme Scale fournit un plug-in SecureTokenManager pour permettre au serveur de sécuriser cette clé secrète avant l'envoi. Vous pouvez choisir la façon dont vous souhaitez implémenter l'opération sécurisée. Avec WebSphere eXtreme Scale, une opération sécurisée est implémentée pour chiffrer et signer la clé secrète.

Fonctions de sécurité Java Management Extensions (JMX) dans une topologie de déploiement dynamique

Les fonctions de sécurité JMX MBeans sont prises en charge dans toutes les versions de eXtreme Scale. Les clients des beans gérés de serveur de catalogue et de serveur de conteneur peuvent être authentifiés, et l'accès aux opérations MBean peut être forcé.

Sécurité eXtreme Scale locale

La sécurité eXtreme Scale locale est différente du modèle eXtreme Scale réparti car l'application s'instancie directement et utilise une instance ObjectGrid. Votre application et les instances eXtreme Scale se trouvent dans la même machine virtuelle Java (JVM). Etant donné qu'aucun concept client-serveur n'existe dans ce modèle, l'authentification n'est pas prise en charge. Vos applications doivent gérer leur propre authentification, puis transmettre l'objet authentifié à eXtreme Scale. Cependant, le mécanisme d'autorisation utilisé pour le modèle de programmation eXtreme Scale est le même que celui utilisé pour le modèle client-serveur.

Configuration et programmation

Pour plus d'informations sur la configuration et la programmation de la sécurité, voir «Intégration de la sécurité à des fournisseurs externes», à la page 638 et API de sécurité.

Planification pour l'installation

Avant d'installer le produit, vous devez prendre en compte la configuration matérielle et logicielle requise, ainsi que les paramètres d'environnement Java.

Configurations matérielle et logicielle requises

Vue d'ensemble des conditions requises en termes de matériels et de systèmes d'exploitation. Bien que vous ne soyez pas tenu d'utiliser un niveau spécifique de matériel ou de système d'exploitation pour WebSphere eXtreme Scale, nous n'en fournissons pas moins sur le site de support du produit (page Configuration requise) une liste détaillée des matériels et logiciels officiellement pris en charge. En cas de conflit entre les informations présentées par le Centre de documentation et celles figurant sur cette page, les informations fournies par le site Web

prévalent. Les conditions préalables répertoriées par le Centre de documentation sont fournies à titre informatif uniquement.

Voir la page Configuration système requise pour connaître les configurations matérielles et logicielles officielles.

Vous pouvez installer et déployer le produit dans les environnements Java EE et Java SE. Vous pouvez également regrouper le composant client avec les applications Java EE directement sans les intégrer à WebSphere Application Server.

Configuration matérielle

WebSphere eXtreme Scale ne requiert pas la présence d'un niveau spécifique de matériel. La configuration matérielle requise dépend du matériel pris en charge pour l'installation de Java Platform, Standard Edition que vous utilisez pour exécuter WebSphere eXtreme Scale. Si vous utilisez eXtreme Scale avec WebSphere Application Server ou une autre implémentation Java Platform, Enterprise Edition, la configuration matérielle requise par ces plateformes est suffisante pour WebSphere eXtreme Scale.

Configuration requise en matière de système d'exploitation

.NET 8.6+ Pour plus d'informations sur la configuration requise pour un environnement de client .NET, voir «Remarques relatives à Microsoft .NET», à la page 67.

Java Chaque implémentation Java SE et Java EE requiert un niveau différent du système d'exploitation ou des correctifs pour les problèmes identifiés lors du test de l'implémentation Java. Les niveaux nécessaires à ces implémentations sont suffisants pour eXtreme Scale.

Configuration requise pour Installation Manager

Avant de pourvoir installer WebSphere eXtreme Scale, vous devez installer Installation Manager. Vous pouvez installer Installation Manager en utilisant le support du produit, en utilisant un fichier obtenu à partir du site Passport Advantage ou en utilisant un fichier contenant la version la plus récente d'Installation Manager disponible sur le site Web de téléchargement d'IBM Installation Manager. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Navigateurs Web requis

La console Web prend en charge les navigateurs Web suivants :

- Mozilla Firefox, version 3.5.x et versions ultérieures
- Microsoft Internet Explorer, version 7 et versions ultérieures

Configuration requise pour WebSphere Application Server

8.6+

- WebSphere Application Server version 7.0.0.21 ou version suivante
- WebSphere Application Server version 8.0.0.2 ou version suivante

Pour plus d'informations, consultez la section Recommended fixes for WebSphere Application Server.

Java requis

8.6+ Les autres implémentations Java EE peuvent utiliser la phase d'exécution d'eXtreme Scale en tant qu'instance locale ou client pour les serveurs eXtreme Scale. Pour implémenter Java SE, vous devez utiliser la version 6 ou une version suivante.

Remarques relatives à Microsoft .NET

.NET

Deux environnements .NET existent dans WebSphere eXtreme Scale : l'environnement de développement et l'environnement d'exécution. Des conditions spécifiques s'appliquent à ces environnements.

Conditions de l'environnement de développement

Version Microsoft .NET

La version .NET 3.5 et les versions suivantes sont compatibles, y compris l'exécution dans un environnement .NET 4.0 uniquement.

Microsoft Visual studio

Vous pouvez utiliser les versions suivantes de Visual Studio :

- Visual Studio 2008 SP1
- Visual Studio 2010 SP1

Windows

Toute version de Windows compatible avec l'édition de Visual Studio que vous utilisez est compatible. Voir les liens suivants pour plus d'informations sur la configuration requise pour Windows pour Visual Studio :

- Configuration système requise pour Visual Studio 2008
- Configuration système requise pour Visual Studio 2010 Professional

Mémoire

- 1 Go (installation 32 bits)
- 2 Go (installation 64 bits)

Espace disque

WebSphere eXtreme Scale nécessite 50 Mo d'espace disque disponible en plus des exigences Visual Studio.

Environnement d'exécution Runtime

Microsoft .NET

.NET 3.5 et les versions suivantes sont prises en charge, y compris l'exécution dans un environnement .NET 4.0 uniquement.

Windows

- Windows Server 2003 (32 et 64 bits) Enterprise, Standard, Datacenter, Web Editions SP2
- Windows Server 2003 R2 (32 et 64 bits) Enterprise, Standard, Datacenter, Web Editions SP2
- Windows Server 2008 (32 et 64 bits) Enterprise, Standard, Datacenter, Web Editions SP2
- Windows Server 2008 R2 (32 et 64 bits) Enterprise, Standard, Datacenter, Web Editions SP2

-
- Windows Hyper-V hypervisor hébergeant les versions Windows répertoriées

Serveur IIS (Internet Information Services)

- IIS 6.0 (fourni avec Windows Server 2003)
- IIS 7.0 (fourni avec Windows Server 2008)
- IIS 7.5 (fourni avec Windows Server 2008 R2)

Mémoire

Espace disque
WebSphere eXtreme Scale nécessite 20 Mo d'espace disque disponible. Lorsque la fonction de trace est activée, un espace disque supplémentaire est nécessaire.

WebSphere eXtreme Scaleruntime
Vous devez utiliser le mécanisme de transport eXtremeIO lorsque vous utilisez des applications client .NET. Pour plus d'informations sur eXtremeIO, voir «Configuration d'IBM eXtremeIO (XIO)», à la page 351.

Java SE : points à prendre en considération

Java

WebSphere eXtreme Scale requires Java SE 6, or Java SE 7. En règle générale, les nouvelles versions de Java SE ont des fonctions plus efficaces et sont plus performantes.

Versions prises en charge

Vous pouvez utiliser WebSphere eXtreme Scale avec Java SE 6 et Java SE 7. La version que vous utilisez doit être prise en charge par le fournisseur Java Runtime Environment (JRE). Si vous voulez utiliser SSL (Secure Sockets Layer), vous devez utiliser un environnement d'exécution IBM.

IBM Runtime Environment, Java Technology Edition Version 6 et version 7 sont pris en charge pour l'utilisation avec le produit. La version 6, édition de service 9, groupe de correctifs 2, est un environnement d'exécution Java pris en charge qui est installé dans le cadre des installations autonomes de WebSphere eXtreme Scale et de WebSphere eXtreme Scale Client dans le répertoire *racine_install_wxs*`/java` et qui peut être utilisé à la fois par les clients et par les serveurs. Si vous installez WebSphere eXtreme Scale dans WebSphere Application Server, vous pouvez utiliser l'environnement JRE inclus dans l'installation WebSphere Application Server. Pour la console Web, vous devez utiliser IBM Runtime Environment, Java Technology Edition Version 6 Service Release 7 et les éditions de service ultérieures uniquement.

WebSphere eXtreme Scale tire parti de la fonctionnalité Version 6 et Version 7 lorsqu'elle devient disponible. Généralement, les nouvelles versions Java Development Kit (JDK) et Java SE sont plus performantes et ont une fonctionnalité plus efficace.

Pour plus d'informations, voir Logiciels pris en charge.

Fonctions WebSphere eXtreme Scale dépendantes de Java SE

Tableau 2. Fonctions nécessitant Java SE 6 et Java SE 7.

WebSphere eXtreme Scale utilise la fonctionnalité introduite dans Java SE 6 pour fournir les fonctions suivantes du produit.

Caractéristique	**Prise en charge dans Java SE 5 et éditions de service ultérieures** **Remarque :** Java SE 5 n'est pas pris en charge dans WebSphere eXtreme Scale Version 8.6	**Prise en charge dans Java SE version 6 , version 7 et éditions de service ultérieures**
Annotations d'API EntityManager (facultatif : vous pouvez également utiliser des fichiers XML)	X	X
Java Persistence API (JPA) : chargeur JPA, chargeur de client JPA et programme de mise à jour basé sur le temps JPA	X	X
L'expulsion basée sur la mémoire (utilise MemoryPoolMXBean)	X	X
Agents d'instrumentation : • `wxssizeagent.jar` : augmente la précision des mesures de mappe d'octets utilisées. • `ogagent.jar` : augmente la performance des entités d'accès aux zones.	X	X
Console Web de surveillance		X

Mise à niveau du kit JDK dans WebSphere eXtreme Scale

Questions courantes sur le processus de mise à niveau des éditions de WebSphere eXtreme Scale dans les environnements autonomes et WebSphere Application Server :

- Comment mettre à niveau le JDK qui est inclus avec WebSphere eXtreme Scale for WebSphere Application Server?

  Vous devez utiliser le processus de mise à niveau JDK que fournit WebSphere Application Server. Pour plus d'informations, voir http://www-304.ibm.com/support/docview.wss?uid=swg21427178.
- Quelle version JDK utiliser avec WebSphere eXtreme Scale dans un environnement WebSphere Application Server ?

  Vous pouvez utiliser n'importe quel niveau de JDK qui est pris en charge par WebSphere Application Server pour la version prise en charge de WebSphere Application Server.

Java EE : points à prendre en considération

Java

Lors de la préparation de l'intégration WebSphere eXtreme Scale dans un environnement Java Platform, Enterprise Edition, tenez compte de certains éléments, tels que les versions, les options de configuration, les conditions requises et les limitations, le déploiement et la gestion des applications.

Exécuter des applications eXtreme Scale en environnement Java EE

Une application Java EE peut se connecter à une application eXtreme Scale distante. En outre, l'environnement WebSphere Application Server permet le démarrage d'un serveur eXtreme Scale lorsqu'une application démarre dans le serveur d'applications.

Si vous utilisez un fichier XML pour créer une instance ObjectGrid et que ce fichier XML se trouve dans le module du fichier EAR, accédez à ce fichier à l'aide de la méthode getClass().getClassLoader().getResource("META-INF/objGrid.xml") afin d'obtenir un objet URL permettant de créer une instance ObjectGrid. Dans l'appel à la méthode, remplacez le nom du fichier XML utilisé.

Vous pouvez utiliser des beans de démarrage pour que, à son démarrage, une application amorce une instance ObjectGrid et supprime cette instance lorsqu'elle s'arrête. Un bean de démarrage est un bean de session sans état avec un emplacement distant com.ibm.websphere.startupservice.AppStartUpHome et une interface distante com.ibm.websphere.startupservice.AppStartUp. L'interface distante possède deux méthodes : la méthode start et la méthode stop. Utilisez la méthode start pour amorcer l'instance et la méthode stop pour détruire l'instance. L'application utilise la méthode ObjectGridManager.getObjectGrid pour maintenir la référence à cette instance. Voir les informations relatives à l'accès à un objet ObjectGrid avec ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Utiliser des chargeurs de classes

Lorsque les modules d'application qui utilisent des chargeurs de classe différents partagent une instance ObjectGrid unique dans une application Java EE, vérifiez que les objets qui sont stockés dans eXtreme Scale et que les plug-in du produit se trouvent dans un chargeur commun dans l'application.

Gérer dans un servlet le cycle de vie des instances ObjectGrid

Pour gérer le cycle de vie d'une instance ObjectGrid dans un servlet, vous pouvez utiliser la méthode init pour créer l'instance et la méthode destroy pour supprimer l'instance. Si l'instance est mise en cache, elle est extraite et manipulée dans le code du servlet. Voir les informations relatives à l'accès à un objet ObjectGrid avec l'interface ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Conventions relatives aux répertoires

Les conventions de répertoire suivantes sont utilisées dans toute la documentation pour faire référence à des répertoires spéciaux, tels que *wxs_install_root* et *wxs_home*. Vous pouvez accéder à ces répertoires pendant plusieurs scénarios différents, y compris lors de l'installation et de l'utilisation des outils de ligne de commande.

racine_install_wxs
Le répertoire *wxs_install_root* est le répertoire racine où sont installés les

fichiers du produit WebSphere eXtreme Scale. Le répertoire *wxs_install_root* peut être le répertoire dans lequel l'archive d'évaluation est extraite ou depuis lequel le produit est installé WebSphere eXtreme Scale.

- Exemple où la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`

- Exemple où WebSphere eXtreme Scale est installé dans un répertoire autonome :

  UNIX **Exemple :** `/opt/IBM/eXtremeScale`

  Windows **Exemple :** `C:\Program Files\IBM\WebSphere\eXtremeScale`

- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer`

wxs_home

Le répertoire *wxs_home* est le répertoire racine du produit, des bibliothèques, des exemples et des composants WebSphere eXtreme Scale. Ce répertoire est identique au répertoire *wxs_install_root* lorsque l'archive d'évaluation est extraite. Pour les installations autonomes, le répertoire *wxs_home* est le sous-répertoire `ObjectGrid` du répertoire *wxs_install_root*. Pour les installations qui sont intégrées à WebSphere Application Server, ce répertoire est le répertoire `optionalLibraries/ObjectGrid` du répertoire *wxs_install_root*.

- Exemple lorsque la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`

- Exemple où WebSphere eXtreme Scale est installé dans un répertoire autonome :

  UNIX **Exemple :** `/opt/IBM/eXtremeScale/ObjectGrid`

  Windows **Exemple :** *racine_install_wxs*`\ObjectGrid`

- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid`

was_root

Le répertoire *was_root* est le répertoire racine d'une installation WebSphere Application Server :

Exemple : `/opt/IBM/WebSphere/AppServer`

.NET 8.6+ **net_client_home**

Le répertoire *net_client* est le répertoire racine d'une installation client .NET.

Exemple : `C:\Program Files\IBM\WebSphere\eXtreme Scale .NET Client`

restservice_home

Le répertoire *restservice_home* est le répertoire dans lequel se trouvent les bibliothèques et les exemples du service de données REST d'WebSphere eXtreme Scale. Ce répertoire s'appelle `restservice` et il est le sous-répertoire de *wxs_home*.

- Exemple pour les déploiements autonomes :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`

  Exemple : *rép_base_wxs*`\restservice`

- Exemple pour les déploiements intégrés à WebSphere Application Server :

Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice`

tomcat_root
Le répertoire *home_tomcat* est le répertoire racine de l'installation d'Apache Tomcat.

Exemple : `/opt/tomcat5.5`

wasce_root
wasce_root est le répertoire racine de l'installation WebSphere Application Server Community Edition.

Exemple : `/opt/IBM/WebSphere/AppServerCE`

java_home
Le répertoire *java_home* est le répertoire racine d'une installation de Java Runtime Environment Kit (JRE).

UNIX **Exemple :** `/opt/IBM/WebSphere/eXtremeScale/java`

Windows **Exemple :** *racine_install_wxs*`\java`

samples_home
samples_home est le répertoire dans lequel vous extrayez les exemples de fichiers qui sont utilisés pour les tutoriels.

UNIX **Exemple:** *rép_base_wxs*`/samples`

Windows **Exemple :** *rép_base_wxs*`\samples`

dvd_root
dvd_root est le répertoire racine du DVD qui contient le produit.

Exemple : `dvd_root/docs/`

equinox_root
Le répertoire *equinox_root* est le répertoire racine de l'installation de l'infrastructure OSGi Eclipse Equinox.

Exemple :`/opt/equinox`

user_home
Le répertoire *user_home* est l'emplacement de stockage des fichiers utilisateur, tels que les profils de sécurité.

Windows `c:\Documents and Settings\`*nom_util*

UNIX `/home/`*nom_util*

Planification de la capacité de l'environnement

Si la taille initiale et la taille projetée des données ont été définies, vous pouvez planifier la capacité dont vous avez besoin pour exécuter WebSphere eXtreme Scale. En utilisant ces exercices de planification, vous pouvez déployer WebSphere eXtreme Scale de manière efficace pour les modifications futures et optimiser l'élasticité de la grille de données, ce que vous ne pourriez pas faire dans un autre scénario, par exemple avec une base de données interne ou un autre type de base de données.

Activation du dépassement de disque

Lorsque le dépassement de disque est activé, vous pouvez étendre la capacité de la grille en extrayant les entrées de cache de la mémoire et en les plaçant sur le disque. Utilisez la propriété diskOverflowEnabled dans le fichier de propriétés du serveur pour activer la fonction de dépassement de disque. Lorsqu'elle est activée, les entrées qui dépassent la capacité de mémoire disponible du serveur de conteneur sont stockées sur disque. Le stockage sur disque n'est pas un emplacement de stockage persistant. Les entrées écrites sur le disque sont détruites au cours du redémarrage des serveurs de conteneur à l'instar des entrées de cache stockées en mémoire qui sont perdues lors d'un redémarrage de conteneur.

Avant de commencer

Vous devez activer la mémoire eXtreme pour que cette option fonctionne. Pour plus d'informations, voir «Configuration d'IBM eXtremeMemory», à la page 357.

Pourquoi et quand exécuter cette tâche

Lorsqu'elle est activée, la fonction de dépassement du disque tente de conserver les dernières entrées de cache utilisées dans la mémoire. Le dépassement de disque place les entrées de cache sur le disque uniquement lorsque le nombre d'entrées en mémoire est supérieur à l'allocation de mémoire maximale définie par la propriété de serveur **maxXMSize**. S'il existe plus d'entrées que ne peut en accepter la mémoire, les entrées les moins récentes sont transférées vers le disque. Cela signifie que les temps de réponse des opérations qui accèdent aux entrées sur le disque sont inférieurs aux temps de réponse des entrées qui se trouvent en mémoire. Après l'accès initial, l'élément reste en mémoire, sauf s'il redevient un ancien élément utilisé. Dans ce cas, il est transféré vers le disque pour laisser la place à une autre entrée.

Procédure

1. Arrêtez le serveur de conteneur sur lequel vous voulez activer le dépassement de disque. Pour plus d'informations, voir «Arrêt des serveurs autonomes qui utilisent le transport IBM eXtremeIO», à la page 490.
2. Définissez les propriétés suivantes dans le fichier des propriétés :

   diskOverflowEnabled
   Active la fonction de dépassement de disque. Vous devez activer eXtreme Memory pour pouvoir utiliser cette fonction.

   Valeur par défaut : `false`

   diskOverflowCapBytes
   Indique la quantité maximale d'espace disque utilisée par ce serveur pour le dépassement de disque, en octets. La valeur par défaut indique que la quantité de données stockées sur le disque est illimitée.

   Valeur par défaut : `Long.MAX_VALUE`

   diskStoragePath
   Spécifie le chemin d'accès absolu vers un répertoire utilisé pour stocker le contenu de dépassement.

   diskOverflowMinDiskSpaceBytes
   Spécifie que les entrées ne seront pas transférées vers le disque, si l'espace disponible libre est inférieur à cette quantité dans diskStoragePath, en octets.

Valeur par défaut : `0`

3. Redémarrez les serveurs de conteneur. Pour plus d'informations, voir «Démarrage des serveurs autonomes (XIO)», à la page 479.

Définition de la taille de la mémoire et calcul du nombre de partitions

Vous pouvez calculer la quantité de mémoire et le nombre de partitions nécessaires pour votre configuration.

Avertissement : Cette rubrique s'applique lorsque vous n'utilisez **pas** le mode de copie COPY_TO_BYTES. Si vous utilisez le mode COPY_TO_BYTES, la taille de la mémoire est beaucoup plus petite et la procédure de calcul est différente. Pour plus d'informations sur ce mode, voir Optimisation du mode de copie.

WebSphere eXtreme Scale stocke les données dans l'espace adresse de machines virtuelles Java (JVM). Chaque JVM fournit un espace processeur pour traiter la création, la récupération, la mise à jour et la suppression d'appels pour les données stockées dans la JVM. En outre, chaque JVM fournit de l'espace mémoire pour les serveurs secondaires et les entrées de données. Les objets Java varient en taille. Par conséquent, vous devez effectuer une mesure afin d'estimer la quantité de mémoire nécessaire.

Pour adapter la taille de la mémoire à vos besoins, chargez les données d'application dans une seule JVM. Lorsque l'utilisation de segment de mémoire atteint 60 %, notez le nombre d'objets utilisés. Ce nombre correspond au nombre d'objets maximal recommandé pour chaque machines virtuelles Java. Pour obtenir la définition de taille la mieux adaptée, utilisez des données réalistes et introduisez tout index défini, car les index occupent également de la mémoire. La meilleure méthode pour dimensionner l'utilisation de la mémoire consiste à exécuter une sortie **verbosegc** de récupération de place, car cette sortie vous donne les valeurs après la récupération de place. Vous pouvez interroger l'utilisation du segment de mémoire à tout moment via des beans gérés ou à l'aide d'une programme, mais ces requêtes ne vous donnent qu'un cliché instantané du segment de mémoire. Ce cliché peut contenir de l'espace non récupéré. Par conséquent, cette méthode n'indique pas précisément la mémoire utilisée.

Mise à l'échelle de la configuration

Nombre de fragments par partition (valeur numShardsPerPartition)

Pour calculer le nombre de fragments par partition, ou valeur numShardsPerPartition, ajoutez 1 pour le fragment primaire plus le nombre total de fragments réplique souhaité. Pour plus d'informations sur le partitionnement, voir Partitionnement.

```
numShardsPerPartition = 1 + total_number_of_replicas
```

Nombre de machines virtuelles Java (valeur minNumJVMs)

Pour mettre à l'échelle votre configuration, décidez d'abord du nombre total maximal d'objets à stocker. Pour déterminer le nombre de machines virtuelles Java nécessaire, utilisez la formule suivante :

```
minNumJVMS=(numShardsPerPartition * numObjs) / numObjsPerJVM
```

Arrondissez cette valeur à l'entier le plus près.

Nombre de fragments (valeur numShards)

Lorsque la taille finale est atteinte, utilisez 10 fragments pour chaque JVM. Comme indiqué précédemment, chaque JVM dispose d'un fragment primaire et (N-1) fragments de réplique, soit dans ce cas, neuf répliques. Etant donné que vous disposez déjà du nombre de machines virtuelles Java pour le stockage de données, vous pouvez multiplier le nombre de machines virtuelles Java par 10 pour obtenir le nombre de fragments :

```
numShards = minNumJVMs * 10 shards/JVM
```

Nombre de partitions Si une partition dispose d'un fragment primaire et d'un fragment réplique, cette partition a donc deux fragments (primaire et réplique). Le nombre de partitions correspond au nombre de fragments divisé par 2 et arrondi au nombre premier le plus proche. Si la partition présente un fragment primaire et deux secondaires, le nombre de partitions correspond au nombre de fragments divisé par 3 et arrondi au nombre premier le plus proche.

```
numPartitions = numShards / numShardsPerPartition
```

Exemple de mise à l'échelle

Dans cet exemple, le nombre d'entrées commence à 250 millions. Chaque année, le nombre d'entrées croît d'environ 14 %. Après sept ans, le nombre total d'entrées atteint 500 millions et vous devez donc planifier la capacité en conséquence. Pour une haute disponibilité, un serveur secondaire est nécessaire. Avec un serveur secondaire, le nombre d'entrées double, soit 1 000 000 000 entrées. Dans le cadre d'un test, deux millions d'entrées peuvent être stockées dans chaque JVM. L'utilisation de calculs pour ce scénario montre le besoin de la configuration suivante :

- 500 machines virtuelles Java pour stocker le nombre final d'entrées.
- 5 000 fragments, obtenus en multipliant 500 machines virtuelles Java par 10.
- 2 500 partitions, arrondies à 2503 (nombre premier supérieur le plus proche), calculées en prenant 5 000 fragments, divisés par deux pour les fragments primaires et secondaires.

Début de la configuration

A partir des calculs précédents, démarrez avec 250 machines virtuelles Java pour atteindre 500 machines virtuelles Java en cinq ans. Avec cette configuration, vous pouvez gérer la croissance incrémentielle jusqu'à ce que vous accédiez au nombre d'entrées final.

Dans cette configuration, environ 200 000 entrées sont stockées par partition (500 millions d'entrées divisées par 2 503 partitions).

Le nombre maximal de machines virtuelles Java est atteint

Lorsque vous atteignez le nombre maximal de 500 machines virtuelles Java, vous pouvez toujours agrandir votre grille de données. Lorsque le nombre de machines virtuelles Java dépasse le nombre maximal de 500, le nombre de fragments commence à tomber en dessous de 10 pour chaque JVM, ce qui est inférieur au nombre recommandé. La taille des fragments augmente et risque d'entraîner des problèmes. Répétez le processus de dimensionnement en tenant compte de la croissance future de nouveau et redéfinissez le nombre de partitions. Cela requiert un redémarrage de la grille de données complète ou une indisponibilité de votre grille de données.

Nombre de serveurs

Avertissement : N'utilisez la pagination sur un serveur sous aucune circonstance.

Une seule JVM utilise plus de mémoire que la taille d'un segment de mémoire. Par exemple, avec 1 Go de segment de mémoire, une JVM utilise en fait 1,4 Go de mémoire réelle. Déterminez la mémoire vive disponible sur le serveur. Divisez la quantité de mémoire vive par la quantité de mémoire pour chaque JVM pour obtenir le nombre maximal de machines virtuelles Java sur le serveur.

Définition du nombre d'unités centrales par partition

Bien que l'une des fonctions principales d'eXtreme Scale soit sa capacité d'évolutivité, il est également important d'évaluer et d'adapter le nombre idéal d'unités centrales en vue d'une montée en charge.

Les coûts liés au processeur comprennent :

- Opérations de création, d'extraction, de mise à jour et de suppression depuis les clients
- Coût de la réplication à partir d'autres machines virtuelles Java
- Coût de l'invalidation
- Coût de la politique d'éviction
- Coût de la récupération de place
- Coût de la logique d'application
- Coût de la sérialisation

machines virtuelles Java par serveur

Utilisez deux serveurs et démarrez le nombre maximal de JVM par serveur. Utilisez le nombre de partitions calculé à la section précédente. Ensuite, préchargez dans ces machines virtuelles Java une quantité de données ne dépassant pas la capacité des deux ordinateurs. Utilisez un serveur distinct en tant que client. Exécutez une simulation de transaction réaliste sur cette grille de données de deux serveurs.

Pour calculer la valeur de référence, essayez de saturer l'utilisation du processeur. Si vous n'y parvenez pas, c'est probablement parce que le réseau est saturé. Dans ce cas, ajoutez des cartes réseau et procédez à une permutation circulaire de ces machines virtuelles Java.

Exécutez les ordinateurs à 60% d'utilisation du processeur et mesurez le taux de transactions de création, d'extraction, de mise à jour et de suppression. Cette mesure indique la capacité de traitement des deux serveurs. Ce nombre double avec quatre serveurs, double encore avec huit serveurs, etc. Cette progression suppose que la capacité du réseau et la capacité du client peuvent également progresser.

Les temps de réponse d'eXtreme Scale doivent donc rester stable au fur et à mesure que le nombre de serveurs évolue. Le débit des transactions doit progresser de manière linéaire à mesure que des ordinateurs sont ajoutés à la grille de données.

Définition de la taille d'unités centrales pour des transactions parallèles

Les transactions à partition unique présentent une évolution de débit linéaire à mesure que la taille de la grille augmente. Les transactions parallèles diffèrent des transactions à partition unique, car elles affectent un ensemble de serveurs (cet ensemble peut comprendre tous les serveurs).

Si une transaction affecte tous les serveurs, le débit est limité au débit du client ayant initié la transaction ou au serveur affecté le plus lent. Les grilles de données de grande taille répartissent davantage les données et fournissent plus d'espace processeur, de mémoire, de réseau, etc. Toutefois, le client doit attendre la réponse du serveur le plus lent et doit utiliser les résultats de la transaction.

Lorsqu'une transaction affecte un sous-ensemble de serveurs, M sur N serveurs reçoivent une requête. Le débit est alors "N divisé par M" fois plus vite que le débit du serveur le plus lent. Par exemple, si vous disposez de 20 serveurs et d'une transaction qui affecte 5 serveurs, le débit est 4 fois supérieur au débit du serveur le plus lent de la grille de données.

Lorsqu'une transaction parallèle se termine, les résultats sont envoyés à l'unité d'exécution du client ayant commencé la transaction. Ce client doit ensuite procéder à l'agrégation des résultats en unité d'exécution simple. Le temps d'agrégation augmente avec l'augmentation du nombre de serveurs affectés par la transaction. Toutefois, cette durée dépend de l'application, car il se peut que chaque serveur renvoie un résultat plus petit à mesure que la taille de la grille de données augmente.

Généralement, les transactions parallèles affectent tous les serveurs dans la grille de données, car les partitions y sont réparties uniformément. Dans ce cas, le débit se limite à la première hypothèse.

Récapitulatif

Avec cette définition de taille, vous disposez de trois métriques, comme suit.

- Nombre de partitions.
- Nombre de serveurs nécessaires pour la mémoire requise.
- Nombre de serveurs nécessaires pour le débit requis.

Si vous avez besoin de 10 serveurs pour la quantité de mémoire nécessaire, mais que vous obtenez uniquement 50 % du débit requis en raison d'une saturation du processeur, vous devait avoir deux fois plus de serveurs.

Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire de JVM à 60 % du chargement des segments de mémoire. Les renforts peuvent ensuite pousser l'utilisation du processeur à 80-90 %, mais n'exécutent pas régulièrement vos serveurs à un niveau supérieur à ces niveaux.

Chapitre 3. Tutoriels

Vous pouvez utiliser les tutoriels pour mieux comprendre les scénarios d'utilisation du produit, y compris le gestionnaire d'entités, les requêtes et la sécurité.

Tutoriel : interrogation d'une grille de données en mémoire locale

Java

Vous pouvez développer un ObjectGrid interne local qui peut stocker des informations de commande pour un site Web et montrer comment utiliser l'API ObjectQuery pour interroger la grille de données.

Avant de commencer

Vérifiez que le fichier `objectgrid.jar` se trouve bien dans le chemin d'accès aux classes.

Pourquoi et quand exécuter cette tâche

Chaque étape du tutoriel repose sur l'étape précédente. Suivez chacune des étapes pour générer une application Java Platform, Standard Edition Version 5 (ou ultérieure) simple qui utilise une grille de données locale interne.

Tutoriel ObjectQuery - Etape 1

Java

A l'aide de la procédure ci-après, vous pouvez continuer à développer un ObjectGrid local en mémoire qui stocke les informations sur les commandes d'un magasin en ligne à l'aide des API ObjectMap. Vous définissez un schéma pour la mappe et exécutez une requête sur cette dernière.

Procédure

1. Créez un ObjectGrid avec un schéma de mappe.

   Créez un ObjectGrid avec un schéma de mappe pour la mappe, puis insérez un objet dans le cache et extrayez-le ensuite à l'aide d'une simple requête.

   OrderBean.java

   ```
   public class OrderBean implements Serializable {
       String orderNumber;
       java.util.Date date;
       String customerName;
       String itemName;
       int quantity;
       double price;
   }
   ```

2. Définissez la clé primaire.

   Le code précédent affiche un objet OrderBean. Cet objet implémente l'interface java.io.Serializable car tous les objets du cache doivent (par défaut) être sérialisables.

L'attribut orderNumber est la clé primaire de l'objet. L'exemple de programme ci-après peut être exécuté en mode autonome. Vous devez suivre ce tutoriel dans un projet Java Eclipse dont le fichier `objectgrid.jar` est ajouté au chemin d'accès aux classes.

Application.java

```
package querytutorial.basic.step1;

import java.util.Iterator;

import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.ObjectMap;
import com.ibm.websphere.objectgrid.Session;
import com.ibm.websphere.objectgrid.config.QueryConfig;
import com.ibm.websphere.objectgrid.config.QueryMapping;
import com.ibm.websphere.objectgrid.query.ObjectQuery;

public class Application
{

    static public void main(String [] args) throws Exception
    {
        ObjectGrid og = ObjectGridManagerFactory.getObjectGridManager().createObjectGrid();
        og.defineMap("Order");

        // Définissez le schéma
        QueryConfig queryCfg = new QueryConfig();
        queryCfg.addQueryMapping(new QueryMapping("Order", OrderBean.class.getName(),
     "orderNumber", QueryMapping.FIELD_ACCESS));
        og.setQueryConfig(queryCfg);

        Session s = og.getSession();
        ObjectMap orderMap = s.getMap("Order");

        s.begin();
        OrderBean o = new OrderBean();
        o.customerName = "John Smith";
        o.date = new java.util.Date(System.currentTimeMillis());
        o.itemName = "Widget";
        o.orderNumber = "1";
        o.price = 99.99;
        o.quantity = 1;
        orderMap.put(o.orderNumber, o);
        s.commit();

        s.begin();
        ObjectQuery query = s.createObjectQuery("SELECT o FROM Order o WHERE o.itemName='Widget'");
        Iterator result = query.getResultIterator();
        o = (OrderBean) result.next();
        System.out.println("Found order for customer: " + o.customerName);
        s.commit();
 // Fermer la session (facultatif dans les versions 7.1.1 et ultérieures)
pour améliorer les performances
 s.close();
}
}
```

Cette application eXtreme Scale commence par initialiser une instance ObjectGrid locale avec un nom généré automatiquement. Ensuite, l'application crée une mappe de sauvegarde et une configuration de requête qui définit le type Java associé à la mappe, le nom de la zone qui sert de clé primaire pour la mappe et la manière d'accéder aux données dans l'objet. Vous obtenez ensuite une session pour extraire l'instance ObjectMap et insérez un objet OrderBean dans la mappe, dans une transaction.

Une fois que les données ont été validées dans le cache, vous pouvez utiliser ObjectQuery pour rechercher l'objet OrderBean à l'aide de l'une des zones persistantes de la classe. Les zones persistantes sont celles qui ne possèdent pas le modificateur transitoire. Comme vous n'avez pas défini d'index sur la mappe de sauvegarde, ObjectQuery doit analyser chaque objet de la mappe à l'aide de la réflexion Java.

Que faire ensuite

«Tutoriel ObjectQuery - Etape 2», à la page 81 montre comment un index peut être utilisé pour optimiser la requête.

Tutoriel ObjectQuery - Etape 2

Java

A l'aide de la procédure ci-après, vous pouvez continuer à créer une instance ObjectGrid avec une mappe et un index, ainsi qu'un schéma pour la mappe. Vous pouvez ensuite insérer un objet dans le cache et l'extraire ultérieurement à l'aide d'une simple requête.

Avant de commencer

Vous devez avoir effectué l'étape «Tutoriel ObjectQuery - Etape 1», à la page 79 avant de passer à cette étape du tutoriel.

Procédure

Schéma et index

Application.java

```
// Créez un index
    HashIndex idx= new HashIndex();
    idx.setName("theItemName");
    idx.setAttributeName("itemName");
    idx.setRangeIndex(true);
    idx.setFieldAccessAttribute(true);
    orderBMap.addMapIndexPlugin(idx);
}
```

L'index doit être une instance de com.ibm.websphere.objectgrid.plugins.index.HashIndex avec les paramètres suivants :

- Le nom est arbitraire, mais il doit être unique pour une mappe de sauvegarde donnée.
- Le nom d'attribut correspond au nom de la zone ou à la propriété de bean que le moteur d'indexation utilise pour introspecter la classe. En l'occurrence, il s'agit du nom de la zone pour laquelle vous créerez l'index.
- RangeIndex doit toujours avoir la valeur true.
- La valeur de FieldAccessAttribute doit correspondre à celle définie dans l'objet QueryMapping lors de la création du schéma de requête. Dans ce cas, l'objet Java est accessible directement par les zones.

Lorsqu'une requête exécute ces filtres sur la zone itemName, le moteur de requête utilise automatiquement l'index défini. Le recours à l'index permet à la requête de s'exécuter beaucoup plus vite sans qu'une analyse de la mappe soit nécessaire. L'étape suivante montre comment un index peut être utilisé pour optimiser la requête.
Etape suivante

Tutoriel ObjectQuery - Etape 3

Java

L'étape ci-après permet de créer un ObjectGrid avec deux mappes et un schéma pour les mappes possédant une relation, puis d'insérer des objets dans le cache et de les extraire ultérieurement à l'aide d'une simple requête.

Avant de commencer

Assurez-vous d'avoir bien effectué l'étape «Tutoriel ObjectQuery - Etape 2», à la page 81 avant de passer à cette étape.

Pourquoi et quand exécuter cette tâche

Cet exemple contient deux mappes, chacune mappée à un seul type Java. La mappe Order contient des objets OrderBean et la mappe Customer, des objets CustomerBean.

Procédure

Définissez les mappes avec une relation.

OrderBean.java

```
public class OrderBean implements Serializable {
    String orderNumber;
    java.util.Date date;
    String customerId;
    String itemName;
    int quantity;
    double price;
}
```

OrderBean ne contient plus customerName. A la place, il contient customerId, qui correspond à la clé primaire de l'objet CustomerBean et de la mappe Customer.

CustomerBean.java

```
public class CustomerBean implements Serializable{
    private static final long serialVersionUID = 1L;
    String id;
    String firstName;
    String surname;
    String address;
    String phoneNumber;
}
```

La relation entre les deux types ou mappes est la suivante :

Application.java

```
public class Application
{

    static public void main(String [] args)
        throws Exception
    {
        ObjectGrid og = ObjectGridManagerFactory.getObjectGridManager().createObjectGrid();
        og.defineMap("Order");
        og.defineMap("Customer");

        // Définissez le schéma
        QueryConfig queryCfg = new QueryConfig();
        queryCfg.addQueryMapping(new QueryMapping(
            "Order", OrderBean.class.getName(), "orderNumber", QueryMapping.FIELD_ACCESS));
        queryCfg.addQueryMapping(new QueryMapping(
            "Customer", CustomerBean.class.getName(), "id", QueryMapping.FIELD_ACCESS));
        queryCfg.addQueryRelationship(new QueryRelationship(
             OrderBean.class.getName(), CustomerBean.class.getName(), "customerId", null));
        og.setQueryConfig(queryCfg);

        Session s = og.getSession();
        ObjectMap orderMap = s.getMap("Order");
        ObjectMap custMap = s.getMap("Customer");

        s.begin();
        CustomerBean cust = new CustomerBean();
        cust.address = "Main Street";
```

```
        cust.firstName = "John";
        cust.surname = "Smith";
        cust.id = "C001";
        cust.phoneNumber = "5555551212";
        custMap.insert(cust.id, cust);

        OrderBean o = new OrderBean();
        o.customerId = cust.id;
        o.date = new java.util.Date();
        o.itemName = "Widget";
        o.orderNumber = "1";
        o.price = 99.99;
        o.quantity = 1;
        orderMap.insert(o.orderNumber, o);
        s.commit();

        s.begin();
        ObjectQuery query = s.createObjectQuery(
            "SELECT c FROM Order o JOIN o.customerId as c WHERE o.itemName='Widget'");
        Iterator result = query.getResultIterator();
        cust = (CustomerBean) result.next();
        System.out.println("Found order for customer: " + cust.firstName + " " + cust.surname);
 s.commit();
   // Fermer la session (facultatif dans les versions 7.1.1 et ultérieures) pour améliorer les performances
 s.close();
    }
}
```

Le XML équivalent XML dans le descripteur de déploiement ObjectGrid est le suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="CompanyGrid">
      <backingMap name="Order"/>
      <backingMap name="Customer"/>

      <querySchema>
         <mapSchemas>
           <mapSchema
             mapName="Order"
             valueClass="com.mycompany.OrderBean"
             primaryKeyField="orderNumber"
             accessType="FIELD"/>
           <mapSchema
             mapName="Customer"
             valueClass="com.mycompany.CustomerBean"
             primaryKeyField="id"
             accessType="FIELD"/>
         </mapSchemas>
         <relationships>
           <relationship
             source="com.mycompany.OrderBean"
             target="com.mycompany.CustomerBean"
             relationField="customerId"/>
         </relationships>
      </querySchema>
    </objectGrid>
  </objectGrids>
</objectGridConfig>
```

Que faire ensuite

L'étape «Tutoriel ObjectQuery - Etape 4», développe l'étape actuelle en incluant des objets d'accès par zone et par propriété ainsi que des relations supplémentaires.

Tutoriel ObjectQuery - Etape 4

Java

L'étape ci-après montre comment créer une instance ObjectGrid avec quatre mappes et un schéma pour les mappes. Certaines mappes possèdent une relation un à un (unidirectionnelle) et une relation un à plusieurs (bidirectionnelle). Une fois les mappes créées, vous pouvez exécuter l'exemple de programme `Application.java` pour insérer des objets dans le cache et exécuter des requêtes pour extraire ces objets.

Avant de commencer

Vérifiez que vous avez bien effectué l'étape «Tutoriel ObjectQuery - Etape 3», à la page 81 avant de passer à l'étape en cours.

Pourquoi et quand exécuter cette tâche

Vous devez créer quatre classes JAVA. Voici les mappes pour ObjectGrid :

- OrderBean.java
- OrderLineBean.java
- CustomerBean.java
- ItemBean.java

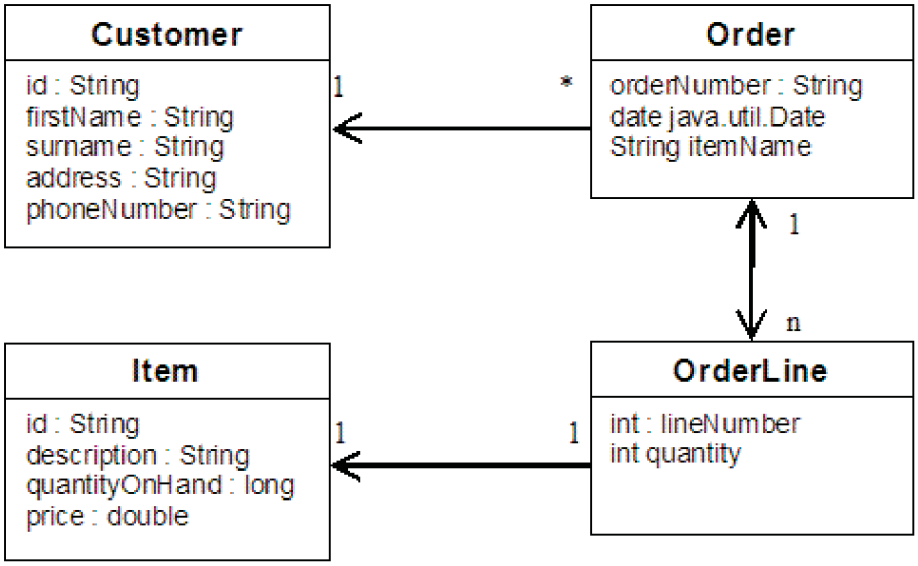


Figure 21. Schéma Order. Un schéma Order possède une relation un à un avec Customer, et une relation un à plusieurs avec OrderLine. La mappe OrderLine possède une relation un à un avec Item et inclut la quantité commandée.

Une fois ces classes JAVA créées avec ces relations, vous pouvez exécuter l'exemple de programme `Application.java`. Ce programme vous permet d'insérer des objets dans le cache et d'extraire ces objets à l'aide de différentes requêtes.

Procédure

1. Créez les classes JAVA suivantes :

   `OrderBean.java`

   ```
   public class OrderBean implements Serializable {
      String orderNumber;
      java.util.Date date;
   ```

```
   String customerId;
   String itemName;
   List<Integer> orderLines;
}
```

OrderLineBean.java

```
public class OrderLineBean implements Serializable {
  int lineNumber;
  int quantity;
  String orderNumber;
  String itemId;
}
```

CustomerBean.java

```
public class CustomerBean implements Serializable{
  String id;
  String firstName;
  String surname;
  String address;
  String phoneNumber;
}
```

ItemBean.java

```
public class ItemBean implements Serializable {
  String id;
  String description;
  long quantityOnHand;
  double price;
}
```

2. Une fois les classes créées, vous pouvez exécuter l'exemple de programme Application.java :

Application.java

```
public class Application static public void main(String [] args)throws Exception
      // Configure programatically
   objectGrid og = ObjectGridManagerFactory.getObjectGridManager().createObjectGrid();
      og.defineMap("Order");
      og.defineMap("Customer");
      og.defineMap("OrderLine");
      og.defineMap("Item");

      // Define the schema
      QueryConfig queryCfg = new QueryConfig();
      queryCfg.addQueryMapping(new QueryMapping("Order", OrderBean.class.getName(), "orderNumber", QueryMapping.FIELD_ACCESS));
      queryCfg.addQueryMapping(new QueryMapping("Customer", CustomerBean.class.getName(), "id", QueryMapping.FIELD_ACCESS));
      queryCfg.addQueryMapping(new QueryMapping("OrderLine", OrderLineBean.class.getName(), "lineNumber", QueryMapping.FIELD_ACCESS));
      queryCfg.addQueryMapping(new QueryMapping("Item", ItemBean.class.getName(), "id", QueryMapping.FIELD_ACCESS));
      queryCfg.addQueryRelationship(new QueryRelationship(OrderBean.class.getName(), CustomerBean.class.getName(), "customerId", null));
      queryCfg.addQueryRelationship(new QueryRelationship(OrderBean.class.getName(), OrderLineBean.class.getName(),
  "orderLines", "lineNumber"));
      queryCfg.addQueryRelationship(new QueryRelationship(OrderLineBean.class.getName(), ItemBean.class.getName(), "itemId", null));
      og.setQueryConfig(queryCfg);

      // Get session and maps;
      Session s = og.getSession();
      ObjectMap orderMap = s.getMap("Order");
      ObjectMap custMap = s.getMap("Customer");
      ObjectMap itemMap = s.getMap("Item");
      ObjectMap orderLineMap = s.getMap("OrderLine");

      // Add data
      s.begin();
      CustomerBean aCustomer = new CustomerBean();
      aCustomer.address = "Main Street";
      aCustomer.firstName = "John";
      aCustomer.surname = "Smith";
      aCustomer.id = "C001";
      aCustomer.phoneNumber = "5555551212";
      custMap.insert(aCustomer.id, aCustomer);

      // Insert an order with a reference to the customer, but without any OrderLines yet.
      // Because we are using CopyMode.COPY_ON_READ_AND_COMMIT, the
      // insert won't be copied into the backing map until commit time, so
      // the reference is still good.
```

```
    OrderBean anOrder = new OrderBean();
    anOrder.customerId = aCustomer.id;
    anOrder.date = new java.util.Date();
    anOrder.itemName = "Widget";
    anOrder.orderNumber = "1";
    anOrder.orderLines = new ArrayList();
    orderMap.insert(anOrder.orderNumber, anOrder);

        ItemBean anItem = new ItemBean();
        anItem.id = "AC0001";
        anItem.description = "Description of widget";
        anItem.quantityOnHand = 100;
        anItem.price = 1000.0;
        itemMap.insert(anItem.id, anItem);

    // Create the OrderLines and add the reference to the Order
        OrderLineBean anOrderLine = new OrderLineBean();
        anOrderLine.lineNumber = 99;
        anOrderLine.itemId = anItem.id;
        anOrderLine.orderNumber = anOrder.orderNumber;
        anOrderLine.quantity = 500;
        orderLineMap.insert(anOrderLine.lineNumber, anOrderLine);
        anOrder.orderLines.add(Integer.valueOf(anOrderLine.lineNumber));

    anOrderLine = new OrderLineBean();
        anOrderLine.lineNumber = 100;
        anOrderLine.itemId = anItem.id;
        anOrderLine.orderNumber = anOrder.orderNumber;
        anOrderLine.quantity = 501;
        orderLineMap.insert(anOrderLine.lineNumber, anOrderLine);
        anOrder.orderLines.add(Integer.valueOf(anOrderLine.lineNumber));
        s.commit();

    s.begin();
        // Find all customers who have ordered a specific item.
        ObjectQuery query = s.createObjectQuery("SELECT c FROM Order o JOIN o.customerId as c WHERE o.itemName='Widget'");
        Iterator result = query.getResultIterator();
        aCustomer = (CustomerBean) result.next();
        System.out.println("Found order for customer: " + aCustomer.firstName + " " + aCustomer.surname);
        s.commit();

    s.begin();
        // Find all OrderLines for customer C001.
        // The query joins are expressed on the foreign keys.
        query = s.createObjectQuery("SELECT ol FROM Order o JOIN o.customerId as c JOIN o.orderLines as ol WHERE c.id='C001'");
        result = query.getResultIterator();
        System.out.println("Found OrderLines:");
        while(result.hasNext()) {
            anOrderLine = (OrderLineBean) result.next();
            System.out.println(anOrderLine.lineNumber + ", qty=" + anOrderLine.quantity);
        }
 // Fermer la session (facultatif dans les versions 7.1.1 et ultérieures) pour améliorer les performances
 s.close();
   }
}
```

3. L'utilisation de la configuration XML ci-après (dans le descripteur de déploiement ObjectGrid) permet d'obtenir les mêmes résultats que l'approche par programme ci-dessus.

```
<?xml version="1.0" encoding="UTF-8"?><objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config
../objectGrid.xsd"xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
  <objectGrid name="CompanyGrid">
  <backingMap name="Order"/>
  <backingMap name="Customer"/>
  <backingMap name="OrderLine" />
  <backingMap name="Item" />

<querySchema>
 <mapSchemas>
  <mapSchema
  mapName="Order"
  valueClass="com.mycompany.OrderBean"
  primaryKeyField="orderNumber"
  accessType="FIELD"/>
  <mapSchema
  mapName="Customer"
  valueClass="com.mycompany.CustomerBean"
  primaryKeyField="id"
  accessType="FIELD"/>
  <mapSchema
  mapName="OrderLine"
  valueClass="com.mycompany.OrderLineBean"
  primaryKeyField="
  lineNumber"
  accessType="FIELD"/>
  <mapSchema
```

```
  mapName="Item"
  valueClass="com.mycompany.ItemBean"
  primaryKeyField="id"
  accessType="FIELD"/>
  </mapSchemas>

<relationships>
 <relationship
  source="com.mycompany.OrderBean"
  target="com.mycompany.CustomerBean"
  relationField="customerId"/>
 <relationship
  source="com.mycompany.OrderBean"
  target="com.mycompany.OrderLineBean"
  relationField="orderLines"
  invRelationField="lineNumber"/>
 <relationship
  source="com.mycompany.OrderLineBean"
  target="com.mycompany.ItemBean"
  relationField="itemId"/>
    </relationships>
   </querySchema>
  </objectGrid>
 </objectGrids>
</objectGridConfig>
```

Tutoriel : Stockage des informations de commande dans des entités

Java

Le tutoriel sur le gestionnaire d'entités présente l'utilisation de WebSphere eXtreme Scale pour stocker des informations de commande sur un site Web. Vous pouvez créer une application Java Platform, Standard Edition 5 qui utilise une version d'eXtreme Scale locale en mémoire. Les entités utilisent les annotations et les valeurs génériques de Java SE 5.

Avant de commencer

Assurez-vous de respecter les exigences suivantes avant de commencer le tutoriel :

- Vous devez disposer de Java SE 5.
- Vous devez disposer du fichier `objectgrid.jar` dans le chemin d'accès aux classes.

Tutoriel du gestionnaire d'entités : création d'une classe entité

Java

Créez un ObjectGrid local avec une entité en créant une classe entité, en enregistrant le type d'entité avec et en stockant une instance d'entité dans le cache.

Procédure

1. Créez l'objet Order. Pour identifier l'objet en tant qu'entité ObjectGrid, ajoutez l'annotation @Entity. Lorsque vous ajoutez cette annotation, tous les attributs sérialisables de l'objet sont automatiquement conservés dans eXtreme Scale, à moins que vous n'utilisiez des annotations permettant de substituer ces attributs. L'attribut **orderNumber** est annoté avec `@Id` pour indiquer que l'attribut est une clé primaire. Ci-après, un exemple d'objet Order :

   Order.java

   ```
   @Entity
   public class Order {
       @Id String orderNumber;
       Date date;
       String customerName;
   ```

```
    String itemName;
    int quantity;
    double price;
}
```

2. Exécutez l'application Hello world d'eXtreme Scale pour démontrer les opérations d'entités. L'exemple de programme suivant peut être lancé en mode autonome pour démontrer les opérations d'entités. Utilisez ce programme dans un projet Java Eclipse auquel le fichier `objectgrid.jar` a été ajouté dans le chemin d'accès aux classes. Ci-après, un exemple d'une application Hello world simple qui utilise eXtreme Scale :

Application.java

```
package emtutorial.basic.step1;

import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.Session;
import com.ibm.websphere.objectgrid.em.EntityManager;

public class Application
{

    static public void main(String [] args)
        throws Exception
    {
        ObjectGrid og =
    ObjectGridManagerFactory.getObjectGridManager().createObjectGrid();
        og.registerEntities(new Class[] {Order.class});

        Session s = og.getSession();
        EntityManager em = s.getEntityManager();

        em.getTransaction().begin();

        Order o = new Order();
        o.customerName = "John Smith";
        o.date = new java.util.Date(System.currentTimeMillis());
        o.itemName = "Widget";
        o.orderNumber = "1";
        o.price = 99.99;
        o.quantity = 1;

        em.persist(o);
        em.getTransaction().commit();

        em.getTransaction().begin();
        o = (Order)em.find(Order.class, "1");
        System.out.println("Found order for customer: " + o.customerName);
        em.getTransaction().commit();
    }
}
```

Cet exemple d'application effectue les opérations suivantes :

a. Initialise une version d'eXtreme Scale locale avec un nom généré automatiquement.
b. Enregistre les classes entité avec l'application à l'aide de l'API registerEntities, même si l'API registerEntities n'est pas toujours nécessaire.
c. Restaure une session et une référence dans le gestionnaire d'entités pour la session.
d. Associe chaque session eXtreme Scale à un seul EntityManager et un seul EntityTransaction. L'EntityManager est à présent utilisé.
e. La méthode registerEntities crée un objet BackingMap appelé Order et associe les métadonnées de cet objet à l'objet BackingMap. Ces métadonnées incluent les attributs clés et non clés, ainsi que les noms et les types d'attribut.
f. Une transaction démarre et crée une instance Order. La transaction est remplie avec des valeurs. La transaction est ensuite conservée à l'aide de la méthode EntityManager.persist, qui identifie l'entité comme étant en attente d'inclusion dans la mappe associée.
g. La transaction est ensuite validée et l'entité est incluse dans ObjectMap.

h. Une autre transaction est créée et l'objet Order est restauré à l'aide de la clé 1. Le transtypage est nécessaire dans la méthode EntityManager.find. La fonction Java SE 5 n'est pas utilisée pour vérifier que le fichier `objectgrid.jar` est compatible avec les machine virtuelle Java.

Tutoriel du gestionnaire d'entités : mise en forme de relations d'entités

Java

Création d'une relation simple entre des entités en créant deux classes entité avec une relation, en enregistrant les entités avec l'ObjectGrid et en stockant les instances d'entités dans le cache.

Procédure

1. Créez l'entité customer qui sert à pour stocker les informations sur les clients indépendamment de l'objet Order. Exemple d'entité customer :

   Customer.java

   ```
   @Entity
   public class Customer
   {
       @Id String id;
       String firstName;
       String surname;
       String address;
       String phoneNumber;
   }
   ```

   Cette classe comprend des informations sur le client, telles que le nom, l'adresse et le numéro de téléphone.

2. Créez l'objet Order, qui est similaire à l'objet Order de la rubrique «Tutoriel du gestionnaire d'entités : création d'une classe entité», à la page 87. Ci-après, un exemple d'objet Order :

   Order.java

   ```
   @Entity
   public class Order {
       @Id String orderNumber;
       Date date;
       @ManyToOne(cascade=CascadeType.PERSIST) Customer customer;
       String itemName;
       int quantity;
       double price;
   }
   ```

   Dans cet exemple, une référence à l'objet Customer remplace l'attribut customerName. La référence possède une annotation qui indique une relation plusieurs à un. Ce type de relation indique que chaque commande a un client, mais que plusieurs commandes peuvent référencer le même client. Le modificateur d'annotations en cascade indique que si EntityManager conserve l'objet Order, il doit également conserver l'objet Customer. Si vous décidez de ne pas définir l'option de conservation de la cascade (option par défaut), vous devez conserver manuellement l'objet Customer avec l'objet Order.

3. A l'aide des entités, définissez les mappes pour l'instance ObjectGrid. Chaque mappe est définie pour une entité spécifique : l'une est nommée Order, l'autre Customer. L'exemple d'application suivant illustre le stockage et la récupération d'une commande client :

Application.java

```
public class Application
{
    static public void main(String [] args)
        throws Exception
    {
        ObjectGrid og =
     ObjectGridManagerFactory.getObjectGridManager().createObjectGrid();
        og.registerEntities(new Class[] {Order.class});

        Session s = og.getSession();
        EntityManager em = s.getEntityManager();

        em.getTransaction().begin();

        Customer cust = new Customer();
        cust.address = "Main Street";
        cust.firstName = "John";
        cust.surname = "Smith";
        cust.id = "C001";
        cust.phoneNumber = "5555551212";

        Order o = new Order();
        o.customer = cust;
        o.date = new java.util.Date();
        o.itemName = "Widget";
        o.orderNumber = "1";
        o.price = 99.99;
        o.quantity = 1;

        em.persist(o);
        em.getTransaction().commit();

        em.getTransaction().begin();
        o = (Order)em.find(Order.class, "1");
        System.out.println("Found order for customer: "
     + o.customer.firstName + " " + o.customer.surname);
        em.getTransaction().commit();
 // Fermer la session (facultatif dans les versions 7.1.1 et ultérieures)
pour améliorer les performances
 s.close();
    }
}
```

Cette application est similaire à l'exemple d'application de l'étape précédente. Dans l'exemple précédent, seule une classe Order a été enregistrée. WebSphere eXtreme Scale détecte et inclut automatiquement la référence dans l'entité Customer et une instance Customer pour `John Smith` est créée et référencée depuis le nouvel objet Order. Par conséquent, le nouveau client est automatiquement conservé, car la relation entre deux commandes inclut le modificateur de cascade, qui requiert la conservation de chaque objet. Lorsque l'objet Order est détecté, le gestionnaire d'entités recherche l'objet Customer associé et insère une référence dans l'objet.

Tutoriel du gestionnaire d'entités : schéma d'entité de commande

Java

Création de quatre classes entité à l'aide de relations uniques et bidirectionnelles, de listes ordonnées et de relations de clés externes. Les API Entity sont utilisées pour conserver et rechercher les entités. Construite sur les entités Order et Customer des sections précédentes du tutoriel, cette étape ajoute deux entités supplémentaires : Item et OrderLine.

Pourquoi et quand exécuter cette tâche

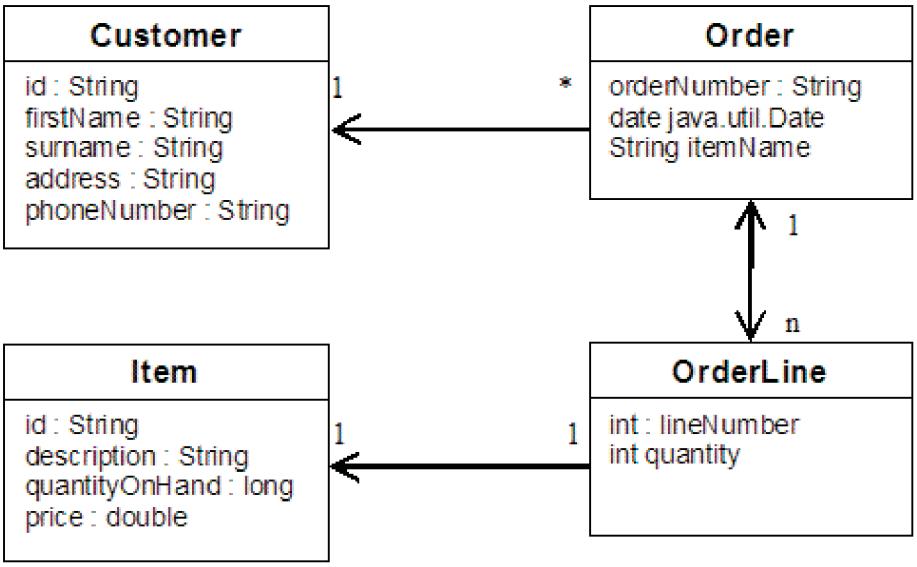


Figure 22. Schéma d'entité de commande. Une entité Order (de commande) a une référence à un client et à zéro ou plus lignes de commande (OrderLines). Chaque entité OrderLine a une référence à un seul article et inclut la quantité commandée.

Procédure

1. Créez l'entité client, similaire aux exemples précédents.

   Customer.java
   ```
   @Entity
   public class Customer
   {
       @Id String id;
       String firstName;
       String surname;
       String address;
       String phoneNumber;
   }
   ```

2. Créez l'entité Item, qui contient les informations sur un produit inclus dans le stock du magasin, telles que la description, la quantité et le prix du produit.

   Item.java
   ```
   @Entity
   public class Item
   {
       @Id String id;
       String description;
       long quantityOnHand;
       double price;
   }
   ```

3. Créez l'entité OrderLine. Chaque entité Order possède zéro ou plus entités OrderLines, qui identifient la quantité de chaque article de la commande. La clé pour OrderLine est une clé composée qui comprend l'entité Order possédée par l'entité OrderLine et un nombre entier qui affecte un numéro à la ligne de commande. Ajoutez le modificateur de conservation de cascade à chaque relation de vos entités.

   OrderLine.java
   ```
   @Entity
   public class OrderLine
   {
   ```

```
    @Id @ManyToOne(cascade=CascadeType.PERSIST) Order order;
    @Id int lineNumber;
    @OneToOne(cascade=CascadeType.PERSIST) Item item;
    int quantity;
    double price;
}
```

4. Créez l'objet Order final, doté d'une référence au client de la commande et à une collection d'objets OrderLine.

   Order.java
   ```
   @Entity
   public class Order {
       @Id String orderNumber;
       java.util.Date date;
       @ManyToOne(cascade=CascadeType.PERSIST) Customer customer;
       @OneToMany(cascade=CascadeType.ALL, mappedBy="order")
      @OrderBy("lineNumber") List<OrderLine> lines;
   }
   ```

   La cascade ALL est utilisée comme modificateur des lignes. Ce modificateur signale EntityManager pour cascader les opérations PERSIST et REMOVE. Par exemple, si l'entité Order est conservée ou supprimée, toutes les entités OrderLine sont également conservées ou supprimées.

   Si une entité OrderLine est supprimée de la liste des lignes dans l'objet Order, la référence est rompue. Toutefois, l'entité OrderLine n'est pas supprimée du cache. Vous devez utiliser l'API de suppression d'EntityManager pour supprimer les entités du cache. L'opération REMOVE n'est pas utilisée sur l'entité Customer ou Item de la ligne de commande. Ainsi, l'entité Customer est conservée même si la commande ou l'article est supprimé(e) lors de la suppression de la ligne de commande.

   Le modificateur mappedBy indique une relation inverse avec l'entité cible. Le modificateur identifie l'attribut de l'entité cible qui référence l'entité source, ainsi que le côté propriétaire de la relation un à un ou plusieurs à plusieurs. En règle générale, vous pouvez omettre le modificateur. Toutefois, une erreur s'affiche indiquant qu'il doit être spécifié si WebSphere eXtreme Scale ne parvient pas à le détecter automatiquement. Une entité OrderLine qui contient deux des attributs Order dans une relation plusieurs à un provoque généralement cette erreur.

   L'annotation @OrderBy spécifie l'ordre dans lequel les entités OrderLine doivent apparaître dans la liste des lignes. Si l'annotation n'est pas spécifiée, les lignes s'affichent dans un ordre arbitraire. Bien que les lignes soient ajoutées à l'entité Order par la soumission d'une liste ArrayList, qui conserve l'ordre, EntityManager ne reconnaît pas nécessairement cet ordre. Lorsque vous émettez la méthode de recherche pour récupérer l'objet Order depuis le cache, l'objet List n'est pas un objet ArrayList.

5. Créez l'application. L'exemple suivant illustre l'objet Order final, qui est doté d'une référence au client de la commande et à une collection d'objets OrderLine.
   a. Recherchez les articles à commander, qui deviennent ensuite des entités gérées.
   b. Créez l'entité OrderLine et liez-la à chaque article.
   c. Créez l'entité Order et associez-la à chaque ligne de commande et au client.
   d. Conservez la commande, qui conserve automatiquement chaque ligne de commande.
   e. Validez la transaction, qui détache chaque entité et synchronise l'état des entités avec le cache.

f. Imprimez les informations de la commande. Les entités OrderLine sont automatiquement triées par l'ID OrderLine.

Application.java

```
static public void main(String [] args)
        throws Exception
    {
        ...

        // Ajoutez des articles à notre stock.
        em.getTransaction().begin();
        createItems(em);
        em.getTransaction().commit();

        // Créez un client ayant des articles dans son panier.
        em.getTransaction().begin();
        Customer cust = createCustomer();
        em.persist(cust);

        // Créez une commande et ajoutez une ligne de
           commande pour chaque article.
        // Chaque article d'une ligne est automatiquement conservé,
    // car l'option Cascade=ALL est définie.
        Order order = createOrderFromItems(em, cust, "ORDER_1",
     new String[]{"1", "2"}, new int[]{1,3});
        em.persist(order);
        em.getTransaction().commit();

        // Imprimez le récapitulatif de la commande.
        em.getTransaction().begin();
        order = (Order)em.find(Order.class, "ORDER_1");
        System.out.println(printOrderSummary(order));
        em.getTransaction().commit();
    }

    public static Customer createCustomer() {
        Customer cust = new Customer();
        cust.address = "Main Street";
        cust.firstName = "John";
        cust.surname = "Smith";
        cust.id = "C001";
        cust.phoneNumber = "5555551212";
        return cust;
    }

    public static void createItems(EntityManager em) {
        Item item1 = new Item();
        item1.id = "1";
        item1.price = 9.99;
        item1.description = "Widget 1";
        item1.quantityOnHand = 4000;
        em.persist(item1);

        Item item2 = new Item();
        item2.id = "2";
        item2.price = 15.99;
        item2.description = "Widget 2";
        item2.quantityOnHand = 225;
        em.persist(item2);

    }

    public static Order createOrderFromItems(EntityManager em,
   Customer cust, String orderId, String[] itemIds, int[] qty) {

        Item[] items = getItems(em, itemIds);
```

```
        Order order = new Order();
        order.customer = cust;
        order.date = new java.util.Date();
        order.orderNumber = orderId;
        order.lines = new ArrayList<OrderLine>(items.length);
    for(int i=0;i<items.length;i++){
      OrderLine line = new OrderLine();
            line.lineNumber = i+1;
            line.item = items[i];
            line.price = line.item.price;
            line.quantity = qty[i];
            line.order = order;
            order.lines.add(line);
        }
        return order;
    }

    public static Item[] getItems(EntityManager em, String[] itemIds) {
        Item[] items = new Item[itemIds.length];
        for(int i=0;i<items.length;i++){
    items[i] = (Item) em.find(Item.class, itemIds[i]);
        }
        return items;
    }
```

L'étape suivante consiste à supprimer une entité. L'interface d'EntityManager est dotée d'une méthode de suppression qui désigne un objet comme étant supprimé. L'application doit supprimer l'entité de toutes les collections de relations avant d'appeler la méthode de suppression. Pour la dernière étape, modifiez les références et émettez la méthode de suppression, em.remove(object).

Tutoriel du gestionnaire d'entités : mise à jour d'entrées

Java

Si vous souhaitez modifier une entité, vous pouvez rechercher l'instance, mettre à jour cette instance ainsi que toute entité référencée et valider la transaction.

Procédure

Mettez à jour des entrées. L'exemple suivant explique comment rechercher une instance Order, comment modifier cette instance ainsi que toute entité référencée et comment valider la transaction.

```
public static void updateCustomerOrder(EntityManager em) {
    em.getTransaction().begin();
    Order order = (Order) em.find(Order.class, "ORDER_1");
    processDiscount(order, 10);
    Customer cust = order.customer;
    cust.phoneNumber = "5075551234";
    em.getTransaction().commit();
}

public static void processDiscount(Order order, double discountPct) {
    for(OrderLine line : order.lines) {
        line.price = line.price * ((100-discountPct)/100);
    }
}
```

Le vidage de la transaction synchronise toutes les entités gérées avec le cache. Lorsqu'une transaction est validée, un vidage se produit automatiquement. Dans ce cas, l'instance Order devient une entité gérée. Toutes les entités référencées depuis les instances Order, Customer et OrderLine deviennent également des entités

gérées. Lorsque la transaction est vidée, chaque entité est vérifiée afin de déterminer si elle a été modifiée. Les entités modifiées sont mises à jour dans le cache. Une fois la transaction terminée, après sa validation ou son annulation, les entités sont détachées et les modifications qui y sont apportées ne sont pas reflétées dans le cache.

Tutoriel du gestionnaire d'entités : mise à jour et suppression d'entrées à l'aide d'un index

Java

Vous pouvez utiliser un index pour rechercher, mettre à jour et supprimer des entités.

Procédure

Mettez à jour et supprimez des entités à l'aide d'un index. Utilisez un index pour rechercher, mettre à jour et supprimer des entités. Dans les exemples suivants, la classe de l'entité Order est mise à jour afin d'utiliser l'annotation @Index. L'annotation @Index signale à WebSphere eXtreme Scale de créer un index d'intervalles pour un attribut. Le nom de l'index est identique au nom de l'attribut et l'index est toujours de type MapRangeIndex.

Order.java

```
@Entity
public class Order {
    @Id String orderNumber;
    @Index java.util.Date date;
    @OneToOne(cascade=CascadeType.PERSIST) Customer customer;
    @OneToMany(cascade=CascadeType.ALL, mappedBy="order")
    @OrderBy("lineNumber") List<OrderLine> lines;  }
```

L'exemple suivant montre l'annulation de toutes les commandes soumises au cours de la dernière minute. Recherchez la commande à l'aide d'un index, replacez les articles de la commande en stock et supprimez la commande ainsi que les articles de la ligne associée du système.

```
public static void cancelOrdersUsingIndex(Session s)
  throws ObjectGridException {
    // Annulez toutes les commandes soumises il y a une minute.
    java.util.Date cancelTime = new
   java.util.Date(System.currentTimeMillis() - 60000);
    EntityManager em = s.getEntityManager();
    em.getTransaction().begin();
    MapRangeIndex dateIndex = (MapRangeIndex)
   s.getMap("Order").getIndex("date");
    Iterator<Tuple> orderKeys = dateIndex.findGreaterEqual(cancelTime);
  while(orderKeys.hasNext()) {
   Tuple orderKey = orderKeys.next();
   // Recherchez la commande à supprimer.
   Order curOrder = (Order) em.find(Order.class, orderKey);
   // Vérifiez que la commande n'a pas été mise à jour par une autre personne.
   if(curOrder != null && curOrder.date.getTime() >= cancelTime.getTime()) {
    for(OrderLine line : curOrder.lines) {
     // Replacez l'article en stock.
     line.item.quantityOnHand += line.quantity;
     line.quantity = 0;
    }
   em.remove(curOrder);
   }
  }
 em.getTransaction().commit();
}
```

Tutoriel du gestionnaire d'entités : mise à jour et suppression d'entrées à l'aide d'une requête

Java

Vous pouvez mettre à jour et supprimer des entités à l'aide d'une requête.

Procédure

Mettez à jour et supprimez des entités à l'aide d'une requête.

Order.java

```
@Entity
public class Order {
    @Id String orderNumber;
    @Index java.util.Date date;
    @OneToOne(cascade=CascadeType.PERSIST) Customer customer;
    é@OneToMany(cascade=CascadeType.ALL, mappedBy="order")
   @OrderBy("lineNumber") List<OrderLine> lines;
}
```

La classe de l'entité Order est identique à celle de l'exemple précédent. La classe fournit toujours l'annotation @Index, car la chaîne de requête utilise la date pour rechercher l'entité. Le moteur de requête utilise des index chaque fois que possible.

```
public static void cancelOrdersUsingQuery(Session s) {
        // Annulez toutes les commandes soumises il y a une minute.
        java.util.Date cancelTime =
     new java.util.Date(System.currentTimeMillis() - 60000);
        EntityManager em = s.getEntityManager();
        em.getTransaction().begin();

        // Créez une requête qui recherche la commande par rapport à la date.  Etant donné
        // que nous avons un index défini sur la date de commande, la requête
    // l'utilisera automatiquement.
        Query query = em.createQuery("SELECT order FROM Order order
     WHERE order.date >= ?1");
        query.setParameter(1, cancelTime);
        Iterator<Order> orderIterator = query.getResultIterator();
    while(orderIterator.hasNext()) {
     Order order = orderIterator.next();
     // Vérifiez que la commande n'a pas été mise à jour par une autre personne.
     // Etant donné que la requête a utilisé un index, il n'y avait pas de verrou sur la ligne.
     if(order != null && order.date.getTime() >= cancelTime.getTime()) {
       for(OrderLine line : order.lines) {
        // Replacez l'article en stock.
        line.item.quantityOnHand += line.quantity;
        line.quantity = 0;
       }
       em.remove(order);
     }
   }
  em.getTransaction().commit();
}
```

Comme dans l'exemple précédent, la méthode cancelOrdersUsingQuery tente d'annuler toutes les commandes soumises au cours de la dernière minute. Pour annuler la commande, recherchez la commande à l'aide d'une requête, replacez les articles de la commande en stock et supprimez la commande ainsi que les articles de la ligne associée du système.

Tutoriel : Configuration de la sécurité Java SE

Le tutoriel suivant vous permet de créer un environnement eXtreme Scale dans un environnement Java Platform, Standard Edition.

Avant de commencer

Assurez-vous que vous connaissez les principes de base d'une configuration eXtreme Scale répartie.

Pourquoi et quand exécuter cette tâche

Utilisez ce tutoriel lorsque vous avez installé eXtreme Scale dans un environnement autonome. Chaque étape du tutoriel est liée à l'étape qui la précède. Effectuez toutes les étapes afin de sécuriser un eXtreme Scale réparti et de développer une application Java SE simple pour accéder au eXtreme Scale sécurisé.

Commencer le tutoriel

Tutoriel sur la sécurité Java SE - Etape 1

Pour pouvoir utiliser le reste du tutoriel, vous devez créer et empaqueter un programme Java simple et deux fichiers XML. Ces fichiers définissent une configuration ObjectGrid simple avec une instance ObjectGrid `accounting` et une mappe `customer`. Le fichier `SimpleDP.xml` comporte une règle de déploiement d'un groupe de mappes configuré avec une partition sans répliques requises minimales.

Procédure

1. Dans une fenêtre de ligne de commande, accédez au répertoire *rép_base_wxs*.
2. Créez un répertoire appelé `applib`.
3. Vérifiez que l'environnement de développement contient le fichier `ogclient.jar` dans le chemin d'accès aux classes. Pour plus d'informations, voir *Guide de programmation*.
4. Créez et compilez la classe `SimpleApp.java` suivante :

SimpleApp.java

```
// Cet exemple de programme est fourni TEL QUEL et peut être utilisé, exécuté, copié et modifié
// gratuitement par le client
// (a) à des fins d'études,
// (b) afin de développer des applications conçues pour être exécutées
       avec un produit IBM WebSphere,
// soit pour un usage interne soit pour une redistribution par le client, en tant que partie
// de l'application, au sein des produits du client.
//   Eléments sous licence - Propriété d'IBM
// 5724-J34 (C) COPYRIGHT International Business Machines Corp. 2007-2009
package com.ibm.websphere.objectgrid.security.sample.guide;

import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.ObjectMap;
import com.ibm.websphere.objectgrid.Session;

public class SimpleApp {

    public static void main(String[] args) throws Exception {

        SimpleApp app = new SimpleApp();
        app.run(args);
    }

    /**
     * read and write the map
     * @throws Exception
     */
    protected void run(String[] args) throws Exception {
        ObjectGrid og = getObjectGrid(args);

        Session session = og.getSession();

        ObjectMap customerMap = session.getMap("customer");

        String customer = (String) customerMap.get("0001");
```

```
        if (customer == null) {
            customerMap.insert("0001", "fName lName");
        } else {
            customerMap.update("0001", "fName lName");
        }
        customer = (String) customerMap.get("0001");
 // Fermer la session (facultatif dans les versions 7.1.1
et ultérieures) pour améliorer les performances
 session.close();
        System.out.println("The customer name for ID 0001 is " + customer);
    }

    /**
     * Get the ObjectGrid
     * @return an ObjectGrid instance
     * @throws Exception
     */
    protected ObjectGrid getObjectGrid(String[] args) throws Exception {
        ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();

        // Create an ObjectGrid
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", null, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;

    }

}
```

5. Compilez le package avec ce fichier et nommez le fichier JAR `sec_sample.jar`.
6. Accédez au répertoire *rép_base_wxs* et créez un répertoire appelé `xml`
7. Dans le répertoire *rép_base_wxs*`/xml`, créez les fichiers de configuration suivants :

   SimpleApp.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="accounting">
            <backingMap name="customer" readOnly="false" copyKey="true"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

   Le fichier XML suivant permet de configurer l'environnement de déploiement.

   SimpleDP.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="accounting">
  <mapSet name="mapSet1" numberOfPartitions="1" minSyncReplicas="0" maxSyncReplicas="2"
maxAsyncReplicas="1">
   <map ref="customer"/>
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Résultats

Ces fichiers créent une configuration ObjectGrid simple avec une grille ObjectGrid, une instance `accounting` et une mappe `customer`.

Tutoriel sur la sécurité Java SE - Etape 2

Avant de pouvoir vérifier que l'exemple de fichier `SimpleApp.java` s'exécute, vous devez démarrer un serveur de catalogue et un serveur de conteneur. Une fois ces services démarrés, vous pouvez lancer le client et exécuter les exemples. Des

fonctions de sécurité supplémentaires sont ajoutées au fur et à mesure des étapes du tutoriel afin d'augmenter le niveau de sécurité intégrée disponible.

Avant de commencer

Pour terminer cette étape du tutoriel, vous devez accéder aux fichiers suivants :

- Package `sec_sample.jar` compilé. Ce package contient le programme `SimpleApp.java`.
- Fichiers de configuration `SimpleApp.xml` et `SimpleDP.xml`.

Vous devez avoir créé ces fichiers dans «Tutoriel sur la sécurité Java SE - Etape 1», à la page 97 de ce tutoriel.

Vous devez également savoir comment :

- Démarrer et arrêter les serveurs de catalogue et de conteneur. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.

  Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.
- Exécutez l'utilitaire **xscmd** pour vérifier la taille de mappe insérée dans la grille de données.

Procédure

1. Dans une fenêtre de ligne de commande, accédez au répertoire *rép_base_wxs*`/bin` et démarrez le service de catalogue.
   - UNIX Linux `./startOgServer.sh catalogServer`
   - Windows `startOgServer.bat catalogServer`
   - UNIX Linux 8.6+ `./startXsServer.sh catalogServer`
   - Windows 8.6+ `startXsServer.bat catalogServer`
2. Démarrez un service de conteneur appelé `c0` :
   - UNIX Linux
     ```
     ./startOgServer.sh c0 -objectGridFile
     ../xml/SimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml
     -catalogServiceEndPoints localhost:2809
     ```
   - Windows
     ```
     startOgServer.bat c0 -objectGridFile ..\xml\SimpleApp.xml -
     deploymentPolicyFile ..\xml\SimpleDP.xml  -catalogServiceEndPoints
     localhost:2809
     ```
   - UNIX Linux 8.6+
     ```
     ./startXsServer.sh c0 -objectGridFile
     ../xml/SimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml
     -catalogServiceEndPoints localhost:2809
     ```
   - Windows 8.6+
     ```
     startXsServer.bat c0 -objectGridFile
     ..\xml\SimpleApp.xml - deploymentPolicyFile ..\xml\SimpleDP.xml
     -catalogServiceEndPoints localhost:2809
     ```
3. Une fois le serveur de catalogue et le serveur de conteneur démarrés, exécutez l'exemple de fichier `sec_sample.jar` comme suit : `java -classpath ../lib/objectgrid.jar:../applib/sec_sample.jar com.ibm.websphere.objectgrid.security.sample.guide.SimpleApp`

java -classpath ..\lib\objectgrid.jar;..\applib\sec_sample.jar com.ibm.websphere.objectgrid.security.sample.guide.SimpleApp Sortie de l'exemple : Le nom de client pour l'ID 0001 est fName lName La métjode getObjectGrid dans cette classe obtient un grille ObjectGrid, et la méthode d'exécution lit un enregistrement depuis la mappe client et met à jour la valeur dans la grille de comptabilité.

4. Vérifiez la taille de la mappe Customer insérée dans la grille Accounting, en exécutant l'utilitaire de commande **xscmd** comme suit :
   - UNIX Linux `./xscmd.sh -c showMapSizes -g accounting -ms mapSet1`
   - Windows `xscmd.bat -c showMapSizes -g accounting -ms mapSet1`
5. Arrêtez un serveur de conteneur appelé c0 avec l'un des scripts suivants :
   - UNIX Linux `./stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809`
   - Windows `stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809`
   - 8.6+ UNIX Linux `./stopXsServer.sh c0 -catalogServiceEndPoints localhost:2809`
   - 8.6+ Windows `stopXsServer.bat c0 -catalogServiceEndPoints localhost:2809`

   Si le serveur s'est arrêté correctement, le message suivant s'affiche :

   ```
   CWOBJ2512I: ObjectGrid server c0 stopped.
   ```
6. Arrêtez le serveur de catalogue à l'aide de l'un des scripts suivants :
   - UNIX Linux `./stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809`
   - Windows `stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809`
   - 8.6+ UNIX Linux `./stopXsServer.sh catalogServer -catalogServiceEndPoints localhost:2809`
   - 8.6+ Windows `stopXsServer.bat catalogServer -catalogServiceEndPoints localhost:2809`

   Si le serveur s'est arrêté correctement, le message suivant s'affiche :

   ```
   CWOBJ2512I: ObjectGrid server catalogServer stopped.
   ```

Tutoriel sur la sécurité Java SE - Etape 3

Le reste du tutoriel explique comment activer l'authentification de client avant de se connecter à un serveur eXtreme Scale. Pour vous préparer à l'étape suivante de ce tutoriel, vous devez empaqueter le programme `SecureSimpleApp.java` en un fichier JAR et créer un ensemble de fichiers de configuration, incluant un fichier `security.xml` et deux fichiers de configuration JAAS. Le fichier `security.xml` vous permet d'écrire l'authentification dans l'environnement et les fichiers de configuration JAAS fournissent le mécanisme d'authentification lors de la connexion au serveur.

Procédure

1. Dans une fenêtre de ligne de commande, accédez au répertoire *rép_base_wxs*/applib que vous avez créé dans «Tutoriel sur la sécurité Java SE - Etape 1», à la page 97.
2. Créez et compilez la classe SecureSimpleApp.java suivante :

SecureSimpleApp.java

```
package com.ibm.websphere.objectgrid.security.sample.guide;

import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory;
import com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator;
import com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator;

public class SecureSimpleApp extends SimpleApp {

    public static void main(String[] args) throws Exception {

        SecureSimpleApp app = new SecureSimpleApp();
        app.run(args);
    }

    /**
     * Get the ObjectGrid
     * @return an ObjectGrid instance
     * @throws Exception
     */
    protected ObjectGrid getObjectGrid(String[] args) throws Exception {
        ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();
        ogManager.setTraceFileName("logs/client.log");
        ogManager.setTraceSpecification("ObjectGrid*=all=enabled:ORBRas=all=enabled");

        // crée un objet ClientSecurityConfiguration à l'aide du fichier spécifié
        ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
                .getClientSecurityConfiguration(args[0]);

        // crée un CredentialGenerator en utilisant le nom de l'utilisateur et le mot de passe fournis.
        CredentialGenerator credGen = new UserPasswordCredentialGenerator(args[1], args[2]);
        clientSC.setCredentialGenerator(credGen);

        // crée un ObjectGrid en se connectant au serveur de catalogue
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", clientSC, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;

    }

}
```

3. Vérifiez que l'environnement contient le fichier ogclient.jar dans le chemin d'accès aux classes. Pour plus d'informations, voir *Guide de programmation*.
4. Compilez le package avec ces fichiers et nommez le fichier JAR sec_sample.jar.
5. Accédez au répertoire *rép_base_wxs*.
6. Créez un répertoire appelé security.
7. Créez un fichier de configuration appelé security.xml. Les propriétés de sécurité du serveur sont spécifiées dans ce fichier. Ces propriétés sont communes pour les serveurs de catalogue et les serveurs de conteneur.

security.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true" loginSessionExpirationTime="300" >

        <authenticator
className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
```

```
            </authenticator>
        </security>

    </securityConfig>
```

Tutoriel sur la sécurité Java SE - Etape 4

Une fois les étapes précédentes effectuées, la rubrique suivante illustre l'implémentation d'une authentification client dans un environnement eXtreme Scale réparti.

Avant de commencer

Vous devez avoir effectué l'étape «Tutoriel sur la sécurité Java SE - Etape 3», à la page 100. Vous devez avoir créé et compilé l'exemple `SecureSimpleApp.java` dans un fichier `sec_sample.jar`, et avoir créé un fichier de configuration appelé `security.xml`.

Pourquoi et quand exécuter cette tâche

Une fois l'authentification client activée, un client est authentifié avant de se connecter au serveur eXtreme Scale. Cette section montre comment l'authentification du client peut être réalisée dans un environnement de serveur eXtreme Scale en utilisant l'exemple `SecureSimpleApp.java`.

Données d'identification client

L'exemple `SecureSimpleApp.java` utilise les deux implémentations de plug-in suivantes pour obtenir les données d'identification client :

`com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredential`

`com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator`

Pour plus d'informations sur ces plug-in, voir Programmation de l'authentification de client.

Authentificateur de serveur

Cet exemple utilise une implémentation pré-intégrée dans eXtreme Scale : `KeyStoreLoginAuthenticator`, qui n'est là qu'à des fins de test et qu'à titre d'exemple (un fichier de clés est un registre simple d'utilisateurs, qui ne doit pas être utilisé dans le cadre d'un environnement de production). Pour plus d'informations, consultez la rubrique relative au plug-in Authenticator Programmation de l'authentification de client.

Procédure

1. Dans une fenêtre de ligne de commande, accédez au répertoire *rép_base_wxs*.
2. Accédez au répertoire *rép_base_wxs*`/security` que vous avez créé dans «Tutoriel sur la sécurité Java SE - Etape 3», à la page 100.
3. Créez un fichier de configuration JAAS qui impose une méthode d'authentification au serveur, `og_jaas.config`. L'élément `KeyStoreLoginAuthenticator` référencé dans le fichier `security.xml` utilise un fichier de clés via le module de connexion JAAS "KeyStoreLogin". Le fichier de clés peut être configuré comme une option de la classe KeyStoreLoginModule.

```
og_jaas.config
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
     keyStoreFile="../security/sampleKS.jks" debug = true;
};
```

4. Accédez au répertoire *rép_base_java*/bin et exécutez l'outil de clé.
5. Accédez au répertoire *rép_base_wxs* /security et créez deux utilisateurs, "manager" et "cashier", possédant leurs propres mots de passe.
   a. Utilisez l'outil de clé pour créer un utilisateur "manager" avec le mot de passe "manager1" dans le fichier de clés sampleKS.jks.
      - UNIX Linux

        ```
        keytool -genkey -v -keystore sampleKS.jks -storepass sampleKS1 \
        -alias manager -keypass manager1 \
        -dname CN=manager,O=acme,OU=OGSample -validity 10000
        ```

      - Windows

        ```
        keytool -genkey -v -keystore sampleKS.jks -storepass sampleKS1 ^
        -alias manager -keypass manager1 ^
        -dname CN=manager,O=acme,OU=OGSample -validity 10000
        ```

   b. Utilisez l'outil de clé pour créer un utilisateur "cashier" avec le mot de passe "cashier1" dans le fichier de clés sampleKS.jks.
      - UNIX Linux

        ```
        keytool -genkey -v -keystore sampleKS.jks -storepass sampleKS1 \
        -alias cashier -keypass cashier1 \
        -dname CN=cashier,O=acme,OU=OGSample -validity 10000
        ```

      - Windows

        ```
        keytool -genkey -v -keystore sampleKS.jks -storepass sampleKS1 ^
        -alias cashier -keypass cashier1 ^
        -dname CN=cashier,O=acme,OU=OGSample -validity 10000
        ```

6. Copiez le fichier sampleClient.properties dans le répertoire wxs_home/properties directory vers wxs_home/security/client.properties
   - UNIX Linux

     ```
     cp ../properties/sampleClient.properties client.properties
     ```

   - Windows

     ```
     copy ..\properties\sampleClient.properties client.properties
     ```

7. Dans le répertoire *rép_base_wxs*/security, enregistrez ce fichier sous le nom client.properties.

   Effectuez les modifications suivantes dans le fichier client.properties :
   a. **securityEnabled :** définissez **securityEnabled** sur true (valeur par défaut) ; cela active la sécurité client, ce qui inclut l'authentification.
   b. **credentialAuthentication :** définissez **credentialAuthentication** sur Supported (valeur par défaut) ; cela signifie que le client prend en charge l'authentification des données d'identification.
   c. **transportType :** définissez **transportType** sur TCP/IP, ce qui signifie qu'aucune couche Secure Sockets Layer (SSL) ne sera utilisée.
8. Copiez le fichier sampleServer.properties dans le répertoire *rép_base_wxs*/security et enregistrez-le sous server.properties.
   - UNIX Linux

     ```
     cp ../properties/sampleServer.properties server.properties
     ```

   - Windows

     ```
     copy ..\properties\sampleServer.properties server.properties
     ```

   Effectuez les modifications suivantes dans le fichier server.properties :

a. **securityEnabled :** définissez l'attribut **securityEnabled** sur true.

b. **transportType :** définissez l'attribut **transportType** sur TCP/IP, ce qui signifie qu'aucune couche Secure Sockets Layer ne sera utilisée.

c. **secureTokenManagerType :** définissez l'attribut **secureTokenManagerType**sur sur none pour ne pas configurer le gestionnaire des jetons sécurisés.

9. Accédez au répertoire *rép_base_wxs*/bin et, selon la plateforme utilisée, exécutez l'une des commandes suivantes pour démarrer un serveur de catalogue. Vous devez exécuter les options de ligne de commande **-clusterFile** et **-serverProps** pour indiquer les propriétés de sécurité :

   - UNIX Linux

     ```
     ./startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties -jvmArgs
     -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - Windows

     ```
     startOgServer.bat catalogServer -clusterSecurityFile ..\security\security.xml
     -serverProps ..\security\server.properties -jvmArgs
     -Djava.security.auth.login.config="..\security\og_jaas.config"
     ```

   - UNIX Linux 8.6+

     ```
     ./startXsServer.sh catalogServer -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties -jvmArgs
     -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - Windows 8.6+

     ```
     startXsServer.bat catalogServer -clusterSecurityFile ..\security\security.xml
     -serverProps ..\security\server.properties -jvmArgs
     -Djava.security.auth.login.config="..\security\og_jaas.config"
     ```

10. Démarrez un serveur de conteneur appelé c0 avec l'un des scripts ci-dessous. Le fichier de propriétés du serveur est indiqué par le biais de l'option **-serverProps**.

    a.

    - UNIX Linux

      ```
      ./startOgServer.sh c0 -objectgridFile ../xml/SimpleApp.xml
      -deploymentPolicyFile ../xml/SimpleDP.xml
      -catalogServiceEndPoints localhost:2809
      -serverProps ../security/server.properties
      -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
      ```

    - Windows

      ```
      startOgServer.bat c0 -objectgridFile ..\xml\SimpleApp.xml
      -deploymentPolicyFile ..\xml\SimpleDP.xml
      -catalogServiceEndPoints localhost:2809
      -serverProps ..\security\server.properties
      -jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
      ```

    - UNIX Linux 8.6+

      ```
      ./startXsServer.sh c0 -objectgridFile ../xml/SimpleApp.xml
      -deploymentPolicyFile ../xml/SimpleDP.xml
      -catalogServiceEndPoints localhost:2809
      -serverProps ../security/server.properties
      -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
      ```

    - Windows 8.6+

      ```
      startXsServer.bat c0 -objectgridFile ..\xml\SimpleApp.xml
      -deploymentPolicyFile ..\xml\SimpleDP.xml
      -catalogServiceEndPoints localhost:2809
      -serverProps ..\security\server.properties
      -jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
      ```

11. Une fois le serveur de catalogue et le serveur de conteneur démarrés, exécutez l'exemple de fichier sec_sample.jar comme suit :

    - UNIX Linux

      ```
      java -classpath ../lib/objectgrid.jar:../applib/sec_sample.jar
        com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
        ../security/client.properties manager manager1
      ```

    - Windows

      ```
      java -classpath ..\lib\objectgrid.jar;..\applib\sec_sample.jar
        com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
        ..\security\client.properties manager manager1
      ```

    Linux Utilisez le signe deux-points (:) pour le séparateur de chemins d'accès aux classes au lieu du point-virgule (;) comme dans l'exemple précédent.

    Une fois la classe publiée, la sortie est la suivante :

    Le nom du client pour ID 0001 est fName lName.

12. Vérifiez la taille de la mappe Customer insérée dans la grille Accounting, en exécutant l'utilitaire de commande **xscmd** comme suit :

    - UNIX Linux `./xscmd.sh -c showMapSizes -g accounting -m customer -username manager -password manager1`
    - Windows `xscmd.bat -c showMapSizes -g accounting -m customer -username manager -password manager1`

13. Facultatif : pour arrêter les serveurs de conteneur ou de catalogue, vous pouvez utiliser la commande **stopOgServer** ou **stopXsServer**. Il est néanmoins nécessaire de fournir un fichier de configuration des paramètres de sécurité. L'exemple de fichier de propriétés du client définit les deux propriétés suivantes pour générer un ID d'utilisateur et un mot de passe (manager/manager1).

```
credentialGeneratorClass=com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
credentialGeneratorProps=manager manager1
```

    Arrêtez le conteneur c0 à l'aide de la commande suivante :

    - UNIX Linux `./stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ../security/client.properties`
    - Windows `stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
    - UNIX Linux 8.6+ `./stopXsServer.sh c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ../security/client.properties`
    - Windows 8.6+ `stopXsServer.bat c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

    Si vous ne définissez pas l'option **-clientSecurityFile**, l'exception suivante se produit avec le message suivant.

    ```
    >> SERVER (id=39132c79, host=9.10.86.47) TRACE START:
    >> org.omg.CORBA.NO_PERMISSION : le serveur requiert une
    authentification par données d'identification mais aucun contexte de
    sécurité n'est fourni par le client. Cela est généralement dû au fait
    que le client ne fournit pas de données d'identification au serveur.
    vmcid: 0x0
    code mineur : 0
    terminé : non
    ```

Vous pouvez également arrêter le serveur de catalogue à l'aide de la commande suivante. Cependant, si vous souhaitez effectuer la prochaine étape du tutoriel, vous pouvez maintenir l'exécution du serveur de catalogue.

- UNIX Linux `./stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ../security/client.properties`
- Windows `stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- UNIX Linux 8.6+ `./stopXsServer.sh -catalogServiceEndPoints localhost:2809 -clientSecurityFile ../security/client.properties`
- Windows 8.6+ `stopXsServer.bat -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Si vous arrêtez le serveur de catalogue, la sortie suivante s'affiche.

```
CWOBJ2512I: ObjectGrid server catalogServer stopped
```

Votre système est à présent partiellement sécurisé grâce à l'activation de l'authentification. Vous avez configuré le serveur pour activer le registre utilisateur, configuré le client pour fournir des données d'identification client et modifié le fichier de propriétés du client et le fichier XML du cluster pour activer l'authentification.

Si vous avez fourni un mot de passe non valide, une exception s'affiche et indique que le nom d'utilisateur ou le mot de passe est incorrect.

Pour plus d'informations sur l'authentification du client, voir «Authentification des clients d'application», à la page 622.

Etape suivante du tutoriel

Tutoriel sur la sécurité Java SE - Etape 5

Après avoir authentifié un client, comme dans l'étape précédente, vous pouvez attribuer des privilèges de sécurité par le biais des mécanismes d'autorisation eXtreme Scale.

Avant de commencer

Assurez-vous d'avoir terminé l'étape «Tutoriel sur la sécurité Java SE - Etape 4», à la page 102 avant de commencer cette tâche.

Pourquoi et quand exécuter cette tâche

Au cours de l'étape précédente, vous avez appris à activer l'authentification dans une grille eXtreme Scale. Par conséquent, aucun client non authentifié ne peut se connecter à votre serveur ni soumettre des requêtes à votre système. Toutefois, tous les clients authentifiés possèdent les mêmes permissions ou privilèges liés au serveur, tels que la lecture, l'écriture ou la suppression des données stockées dans les mappes ObjectGrid. Les clients peuvent également soumettre tout type de requête. Cette section explique comment utiliser l'autorisation eXtreme Scale pour attribuer différents privilèges aux utilisateurs authentifiés.

A l'instar de nombreux autres systèmes, eXtreme Scale adopte un mécanisme d'autorisation basé sur les permissions. WebSphere eXtreme Scale permet d'utiliser plusieurs catégories de permission, chacune étant représentée par une classe distincte. Cette rubrique décrit MapPermission. Pour la liste des catégories d'autorisations, voir Programmation d'autorisations client.

Dans WebSphere eXtreme Scale, la classe com.ibm.websphere.objectgrid.security.MapPermission représente les permissions liées aux ressources eXtreme Scale, notamment les méthodes des interfaces ObjectMap ou JavaMap. WebSphere eXtreme Scale définit les chaînes de permission suivantes pour accéder aux méthodes des interfaces ObjectMap et JavaMap :

- read : accorde la permission de lire les données de la mappe.
- write : accorde la permission de mettre à jour les données de la mappe.
- insert : accorde la permission d'insérer les données dans la mappe.
- remove : accorde la permission de supprimer les données de la mappe.
- invalidate : accorde la permission d'invalider les données de la mappe.
- all : accorde les permissions de lire, d'écrire, d'insérer, de supprimer et d'invalider.

L'autorisation est accordée lorsque le client invoque une méthode de l'interface ObjectMap ou JavaMap. L'environnement d'exécution eXtreme Scale vérifie différentes autorisations de mappe pour différentes méthodes. Si les permissions nécessaires ne sont pas accordées au client, une AccessControlException se produit.

Ce tutoriel montre comment utiliser l'autorisation JAAS (Java Authentication and Authorization Service) afin d'autoriser plusieurs utilisateurs à accéder à la mappe.

Procédure

1. **Activation de l'autorisation eXtreme Scale**. Pour activer l'autorisation sur l'ObjectGrid, vous devez définir l'attribut securityEnabled sur `true` pour cet ObjectGrid spécifique dans le fichier XML. L'activation de la sécurité sur cet ObjectGrid revient à activer l'autorisation. Utilisez les commandes suivantes pour créer un fichier XML ObjectGrid en activant la sécurité.
   a. Accédez au répertoire `xml`.
      ```
      cd objectgridRoot/xml
      ```
   b. Copiez le fichier `SimpleApp.xml` dans le fichier `SecureSimpleApp.xml`.
      - UNIX Linux
        ```
        cp SimpleApp.xml SecureSimpleApp.xml
        ```
      - Windows
        ```
        copy SimpleApp.xml SecureSimpleApp.xml
        ```
   c. Ouvrez le fichier `SecureSimpleApp.xml` et ajoutez `securityEnabled="true"` au niveau de l'ObjectGrid comme indiqué dans la syntaxe XML ci-dessous :
      ```
      <?xml version="1.0" encoding="UTF-8"?>
      <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
         xmlns="http://ibm.com/ws/objectgrid/config">
          <objectGrids>
              <objectGrid name="accounting" securityEnabled="true">
                  <backingMap name="customer" readOnly="false" copyKey="true"/>
              </objectGrid>
          </objectGrids>
      </objectGridConfig>
      ```
2. **Définition de la politique d'autorisation.** Dans la rubrique d'authentification de client précédente, vous avez créé les utilisateurs, le "cashier" (caissier) et le manager (gestionnaire), dans le fichier de clés. Dans cet exemple, l'utilisateur "cashier" dispose uniquement de la permission de lecture sur toutes les mappes. L'utilisateur "manager", quant à lui, dispose de toutes les permissions. L'autorisation JAAS est utilisée dans cet exemple. Vous devez créer un fichier

de règles d'autorisation JAAS =our accorder des autorisations aux principaux. Créez le fichier suivant `og_auth.policy` dans le répertoire `objectgridRoot/security` :

```
og_auth.policy
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    principal javax.security.auth.x500.X500Principal "CN=cashier,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "accounting.*", "read ";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "accounting.*", "all";
};
```

Remarque :

- Le `codebase "http://www.ibm.com/com/ibm/ws/objectgridRoot/security/PrivilegedAction"` est une URL réservée spécialement à ObjectGrid. Toutes les permissions ObjectGrid accordées aux principaux utilisent cette base de code spécifique.
- La première déclaration d'attribution accorde la permission de lecture ("read") de mappe à l'utilisateur principal `"CN=cashier,O=acme,OU=OGSample"`, de sorte que le caissier dispose uniquement de la permission de lecture sur toutes les mappes de l'ObjectGrid accounting.
- La deuxième déclaration d'attribution accorde toutes ("all") les permissions à l'utilisateur principal `"CN=manager,O=acme,OU=OGSample"`, de sorte que le gestionnaire dispose de toutes les permissions sur toutes les mappes de l'ObjectGrid accounting.

Vous pouvez désormais démarrer un serveur avec une politique d'autorisation. Le fichier de règles d'autorisation JAAS peut être défini en utilisant la propriété standard -D : `-Djava.security.policy=../security/og_auth.policy`

3. **Exécutez l'application.**

   Après avoir créé les fichiers mentionnés ci-dessus, vous pouvez exécuter l'application.

   Démarrez le serveur de catalogue à l'aide des commandes suivantes. Pour plus d'informations sur le démarrage du service de catalogue, voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494.

   a. Accédez au répertoire bin : `cd objectgridRoot/bin`

   b. Démarrez le serveur de catalogue.

   - UNIX Linux

     ```
     ./startOgServer.sh catalogServer
     -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties
     -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - Windows

     ```
     startOgServer.bat catalogServer
     -clusterSecurityFile ..\security\security.xml
     -serverProps ..\security\server.properties
     -jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
     ```

   - 8.6+ UNIX Linux

     ```
     ./startXsServer.sh catalogServer
     -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties
     -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - 8.6+ Windows

```
startXsServer.bat catalogServer
-clusterSecurityFile ..\security\security.xml
-serverProps ..\security\server.properties
-jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
```

Vous avez créé les fichiers `security.xml` et `server.properties` au cours de l'étape précédente de ce tutoriel.

c. Vous pouvez démarrer un serveur de conteneur sécurisé à l'aide du script suivant. Exécutez le script suivant à partir du répertoire bin :

- UNIX Linux

```
./startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml
-catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties
-jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.policy="../security/og_auth.policy"
```

- Windows

```
startOgServer.bat c0 -objectGridFile ..\xml\SecureSimpleApp.xml
-deploymentPolicyFile ..\xml\SimpleDP.xml
-catalogServiceEndPoints localhost:2809
-serverProps ..\security\server.properties
-jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
-Djava.security.policy="..\security\og_auth.policy"
```

- 8.6+ UNIX Linux

```
./startXsServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml
-catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties
-jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.policy="../security/og_auth.policy"
```

- 8.6+ Windows

```
startXsServer.bat c0 -objectGridFile ..\xml\SecureSimpleApp.xml
-deploymentPolicyFile ..\xml\SimpleDP.xml
-catalogServiceEndPoints localhost:2809
-serverProps ..\security\server.properties
-jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
-Djava.security.policy="..\security\og_auth.policy"
```

Veuillez noter les différences suivantes concernant la commande de démarrage de serveur de conteneur précédente :

- Utilisez le fichier `SecureSimpleApp.xml` au lieu du fichier `SimpleApp.xml`.
- Ajoutez un autre argument `-Djava.security.policy` pour définir le fichier de règles d'autorisation JAAS dans le processus serveur de conteneur.

Utilisez la même commande que dans l'étape précédente du tutoriel :

a. Accédez au répertoire bin.

- UNIX Linux

```
java -classpath ../lib/objectgrid.jar:../applib/sec_sample.jar com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties manager manager1
```

- Windows

```
java -classpath ..\lib\objectgrid.jar;..\applib\sec_sample.jar com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
 ..\security\client.properties manager manager1
```

b. L'application s'exécute correctement car l'utilisateur "manager" dispose de toutes les permissions sur les mappes de l'ObjectGrid accounting.

Désormais, utilisez l'utilisateur "cashier" et non "manager" pour lancer l'application client.

c. Accédez au répertoire bin.

- UNIX Linux

```
java -classpath ../lib/objectgrid.jar:../applib/sec_sample.jar com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties cashier cashier1
```

- Windows

```
java -classpath ..\lib\objectgrid.jar;..\applib\sec_sample.jar com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp
..\security\client.properties cashier cashier1
```

Vous obtenez l'exception suivante :

```
Exception in thread "P=387313:O=0:CT" com.ibm.websphere.objectgrid.TransactionException:
rolling back transaction, see caused by exception
 at com.ibm.ws.objectgrid.SessionImpl.rollbackPMapChanges(SessionImpl.java:1422)
  at com.ibm.ws.objectgrid.SessionImpl.commit(SessionImpl.java:1149)
  at com.ibm.ws.objectgrid.SessionImpl.mapPostInvoke(SessionImpl.java:2260)
  at com.ibm.ws.objectgrid.ObjectMapImpl.update(ObjectMapImpl.java:1062)
  at com.ibm.ws.objectgrid.security.sample.guide.SimpleApp.run(SimpleApp.java:42)
 at com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp.main(SecureSimpleApp.java:27)
Caused by: com.ibm.websphere.objectgrid.ClientServerTransactionCallbackException:
   Client Services - received exception from remote server:
     com.ibm.websphere.objectgrid.TransactionException: transaction rolled back,
   see caused by Throwable
        at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.processReadWriteResponse(
            RemoteTransactionCallbackImpl.java:1399)
        at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.processReadWriteRequestAndResponse(
            RemoteTransactionCallbackImpl.java:2333)
        at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.commit(RemoteTransactionCallbackImpl.java:557)
        at com.ibm.ws.objectgrid.SessionImpl.commit(SessionImpl.java:1079)
        ... 4 more
Caused by: com.ibm.websphere.objectgrid.TransactionException: transaction rolled back, see caused by Throwable
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processLogSequence(ServerCoreEventProcessor.java:1133)
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processReadWriteTransactionRequest
     (ServerCoreEventProcessor.java:910)
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processClientServerRequest(ServerCoreEventProcessor.java:1285)

        at com.ibm.ws.objectgrid.ShardImpl.processMessage(ShardImpl.java:515)
        at com.ibm.ws.objectgrid.partition.IDLShardPOA._invoke(IDLShardPOA.java:154)
        at com.ibm.CORBA.poa.POAServerDelegate.dispatchToServant(POAServerDelegate.java:396)
        at com.ibm.CORBA.poa.POAServerDelegate.internalDispatch(POAServerDelegate.java:331)
        at com.ibm.CORBA.poa.POAServerDelegate.dispatch(POAServerDelegate.java:253)
        at com.ibm.rmi.iiop.ORB.process(ORB.java:503)
        at com.ibm.CORBA.iiop.ORB.process(ORB.java:1553)
        at com.ibm.rmi.iiop.Connection.respondTo(Connection.java:2680)
        at com.ibm.rmi.iiop.Connection.doWork(Connection.java:2554)
        at com.ibm.rmi.iiop.WorkUnitImpl.doWork(WorkUnitImpl.java:62)
        at com.ibm.rmi.iiop.WorkerThread.run(ThreadPoolImpl.java:202)
        at java.lang.Thread.run(Thread.java:803)
Caused by: java.security.AccessControlException: Access denied (
   com.ibm.websphere.objectgrid.security.MapPermission accounting.customer write)
        at java.security.AccessControlContext.checkPermission(AccessControlContext.java:155)
        at com.ibm.ws.objectgrid.security.MapPermissionCheckAction.run(MapPermissionCheckAction.java:141)
        at java.security.AccessController.doPrivileged(AccessController.java:275)
        at javax.security.auth.Subject.doAsPrivileged(Subject.java:727)
        at com.ibm.ws.objectgrid.security.MapAuthorizer$1.run(MapAuthorizer.java:76)
        at java.security.AccessController.doPrivileged(AccessController.java:242)
        at com.ibm.ws.objectgrid.security.MapAuthorizer.check(MapAuthorizer.java:66)
        at com.ibm.ws.objectgrid.security.SecuredObjectMapImpl.checkMapAuthorization(SecuredObjectMapImpl.java:429)
        at com.ibm.ws.objectgrid.security.SecuredObjectMapImpl.update(SecuredObjectMapImpl.java:490)
        at com.ibm.ws.objectgrid.SessionImpl.processLogSequence(SessionImpl.java:1913)
        at com.ibm.ws.objectgrid.SessionImpl.processLogSequence(SessionImpl.java:1805)
        at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processLogSequence(ServerCoreEventProcessor.java:1011)
        ... 14 more
```

Cette exception se produit car l'utilisateur "cashier" ne dispose pas de permission d'écriture et ne peut donc mettre à jour le client de mappe.

Votre système prend désormais en charge l'autorisation. Vous pouvez définir des politiques d'autorisation pour accorder différentes permissions à différents utilisateurs. Pour plus d'informations sur l'autorisation, voir «Autorisation des clients d'application», à la page 624.

Que faire ensuite

Effectuez l'étape suivante du tutoriel. Voir «Tutoriel sur la sécurité Java SE - Etape 6», à la page 111.

Tutoriel sur la sécurité Java SE - Etape 6

L'étape suivante vous explique comment activer une couche de sécurité pour la communication entre les noeuds finals de votre environnement.

Avant de commencer

Assurez-vous d'avoir terminé le «Tutoriel sur la sécurité Java SE - Etape 5», à la page 106 avant de commencer cette étape.

Pourquoi et quand exécuter cette tâche

La topologie eXtreme Scale prend en charge les protocoles TLS (Transport Layer Security) et SSL (Secure Sockets Layer) pour sécuriser la communication entre les noeuds finals de l'ObjectGrid (client, serveurs de conteneur et serveurs de catalogue). Cette étape du tutoriel se base sur les étapes précédentes pour activer la sécurité du transport.

Procédure

1. **Créez des clés et un fichier de clés TLS/SSL**

   Pour activer la sécurité du transport, vous devez créer un fichier de clés et un fichier de clés certifiées. Cet exercice crée une seule clé et une seule paire fichier de clés-fichier de clés certifiées. Ces fichiers sont utilisés pour les clients, les serveurs de conteneur et les serveurs de catalogue ObjectGrid et sont créés par le biais de la commande keytool du Java Development Kit.

   - *Créez une clé privée dans le fichier de clés*

     ```
     keytool -genkey -alias ogsample -keystore key.jks -storetype JKS
     -keyalg rsa -dname "CN=ogsample, OU=OGSample, O=acme, L=Your City,
     S=Your State, C=Your Country" -storepass ogpass -keypass ogpass
     -validity 3650
     ```

     Avec cette commande, le fichier de clé key.jks est créé avec la clé "ogsample". Cette clé de fichier de clés est utilisée comme fichier de clés SSL.

   - *Exporter le certificat public*

     ```
     keytool -export -alias ogsample -keystore key.jks -file temp.key
     -storepass ogpass
     ```

     Cette commande permet d'extraire et de stocker le certificat public de la clé "ogsample" dans le fichier temp.key.

   - *Importer le certificat public du client dans le fichier de clés certifiées*

     ```
     keytool -import -noprompt -alias ogsamplepublic -keystore trust.jks
     -file temp.key -storepass ogpass
     ```

     Cette commande permet d'ajouter le certificat public au fichier de clés trust.jks. Ce fichier trust.jks est utilisé en tant que fichier de clés certifiées SSL.

2. **Configuration des fichiers de propriétés d'ObjectGrid**

   Au cours de cette étape, vous devez configurer le fichier de propriétés ObjectGrid pour activer la sécurité du transport.

   Tout d'abord, copiez les fichiers key.jks et trust.jks dans le répertoire objectgridRoot/security.

   Définissez les propriétés suivantes dans les fichiers `client.properties` et `server.properties`.

   ```
   transportType=SSL-Required

   alias=ogsample
   ```

```
contextProvider=IBMJSSE2
protocol=SSL
keyStoreType=JKS
keyStore=../security/key.jks
keyStorePassword=ogpass
trustStoreType=JKS
trustStore=../security/trust.jks
trustStorePassword=ogpass
```

transportType : la valeur de transportType est définie sur "SSL-Required", ce qui signifie que le transport requiert la couche Secure Sockets Layer. La configuration SSL doit être définie pour tous les points de contact ObjectGrid (clients, serveurs de catalogue et serveurs de conteneur). Tous les transports seront chiffrés.

Les autres propriétés sont utilisées pour définir les configurations SSL. Pour plus d'informations, voir «Protocole TLS et couche de connexion sécurisée», à la page 634. Veillez à suivre les instructions de cette rubrique pour mettre à jour votre fichier `orb.properties`.

Assurez-vous de suivre les instructions de cette page pour mettre à jour le fichier `orb.properties`.

Dans le fichier `server.properties`, vous devez ajouter une propriété supplémentaire clientAuthentication et la définir sur false. Du côté serveur, vous n'avez pas besoin de certifier le client.

```
clientAuthentication=false
```

3. **Exécution de l'application**

   Ces commandes sont identiques aux commandes de la section «Tutoriel sur la sécurité Java SE - Etape 3», à la page 100.

   Utilisez les commandes suivantes pour démarrer un serveur de catalogue.

   a. Accédez au répertoire bin : `cd objectgridRoot/bin`

   b. Démarrez le serveur de catalogue :

   - Linux UNIX

     ```
     ./startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties -JMXServicePort 11001
     -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - Windows

     ```
     startOgServer.bat catalogServer -clusterSecurityFile ..\security\security.xml
     -serverProps ..\security\server.properties -JMXServicePort 11001 -jvmArgs
     -Djava.security.auth.login.config="..\security\og_jaas.config"
     ```

   - Linux UNIX 8.6+

     ```
     ./startXsServer.sh catalogServer -clusterSecurityFile ../security/security.xml
     -serverProps ../security/server.properties -JMXServicePort 11001
     -jvmArgs -Djava.security.auth.login.config="../security/og_jaas.config"
     ```

   - Windows 8.6+

     ```
     startXsServer.bat catalogServer -clusterSecurityFile ..\security\security.xml
     -serverProps ..\security\server.properties -JMXServicePort 11001 -jvmArgs
     -Djava.security.auth.login.config="..\security\og_jaas.config"
     ```

     Les fichiers `security.xml` et `server.properties` ont été créés au cours de la procédure de la section «Tutoriel sur la sécurité Java SE - Etape 2», à la page 98.

     Utilisez l'option **`-JMXServicePort`** pour spécifier explicitement le port JMX pour le serveur. Cette option est nécessaire pour pouvoir utiliser l'utilitaire **`xscmd`**.

     Démarrez le serveur de conteneur ObjectGrid sécurisé :

   c. Placez-vous de nouveau dans le répertoire bin : `cd objectgridRoot/bin`

   d.

- Linux UNIX

```
./startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints
localhost:2809 -serverProps ../security/server.properties
-JMXServicePort 11002 -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.policy="../security/og_auth.policy"
```

- Windows

```
startOgServer.bat c0 -objectGridFile ..\xml\SecureSimpleApp.xml
-deploymentPolicyFile ..\xml\SimpleDP.xml -catalogServiceEndPoints localhost:2809
-serverProps ..\security\server.properties -JMXServicePort 11002
-jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
-Djava.security.policy="..\security\og_auth.policy"
```

- Linux UNIX **8.6+**

```
./startXsServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints
localhost:2809 -serverProps ../security/server.properties
-JMXServicePort 11002 -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.policy="../security/og_auth.policy"
```

- Windows **8.6+**

```
startXsServer.bat c0 -objectGridFile ..\xml\SecureSimpleApp.xml
-deploymentPolicyFile ..\xml\SimpleDP.xml -catalogServiceEndPoints localhost:2809
-serverProps ..\security\server.properties -JMXServicePort 11002
-jvmArgs -Djava.security.auth.login.config="..\security\og_jaas.config"
-Djava.security.policy="..\security\og_auth.policy"
```

Veuillez noter les différences suivantes concernant la commande de démarrage de serveur de conteneur précédente :

- Utilisez `SecureSimpleApp.xml` à la place des fichiers `SimpleApp.xml`.
- Ajoutez un autre argument -Djava.security.auth.policy pour définir le fichier de règles de l'autorisation JAAS dans le processus du serveur de conteneur.

Exécutez la commande suivante pour l'authentification de client :

a. `cd objectgridRoot/bin`

- UNIX Linux

```
javaHome/java -classpath ../lib/objectgrid.jar:../applib/sec_sample.jar
com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties manager manager1
```

- Windows

```
javaHome\java -classpath ..\lib\objectgrid.jar;..\applib\sec_sample.jar
com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
..\security\client.properties manager manager1
```

b. Etant donné que l'utilisateur "manager" bénéficie de droits d'accès à toutes les mappes de l'ObjectGrid accounting, l'application s'exécute correctement.

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher les tailles de mappes de la grille "accounting".

- Placez-vous dans le répertoire `objectgridRoot/bin`.
- Utilisez la commande **xscmd** pour afficher les tailles de mappes :
  - UNIX Linux

```
./xscmd.sh -c showMapsizes -g accounting -m customer -prot SSL
-ts ../security/trust.jks -tsp ogpass -tst jks
-user manager -pwd manager1 -ks ../security/key.jks -ksp ogpass -kst JKS
-cxpv IBMJSSE2 -tt SSL-Required
```

  - Windows

```
xscmd.bat -c showMapsizes -g accounting -m customer -prot SSL
-ts ..\security\trust.jks -tsp ogpass -tst jks
-user manager -pwd manager1 -ks ..\security\key.jks -ksp ogpass -kst JKS
-cxpv IBMJSSE2 -tt SSL-Required
```

  Notez que le port JMX du service de catalogue est ici spécifié par le biais de -p 11001.

La sortie suivante s'affiche.

```
Cet utilitaire administratif est fourni à titre d'exemple uniquement
et ne doit pas être considéré en tant que composant pris en charge par WebSphere eXtreme Scale.
Connexion au service de catalogue au localhost:1099
********** Résultats pour la grille - accounting, MapSet - client **********
*** Liste des mappes pour c0 ***
Nom de la mappe : client N° de partition #: 0 Taille de la mappe : 1 Type de fragment : principal
Nombre de serveurs : 1
Nombre de domaines : 1
```

Exécution de l'application avec un fichier de clés incorrect

Si le fichier de clés certifiées ne contient pas le certificat public de la clé privée dans le fichier de clés, une exception s'affiche pour indiquer que la clé n'est pas certifiée.

Pour afficher cette exception, créez un autre fichier de clés, `key2.jks`.

```
keytool -genkey -alias ogsample -keystore key2.jks -storetype JKS
-keyalg rsa -dname "CN=ogsample, OU=Your Organizational Unit, O=Your
Organization, L=Your City, S=Your State, C=Your Country" -storepass
ogpass -keypass ogpass -validity 3650
```

Modifiez le fichier `server.properties` pour que keyStore pointe vers le nouveau fichier de clés `key2.jks` :

```
keyStore=../security/key2.jks
```

Exécutez la commande suivante pour démarrer le serveur de catalogue :

a. Placez-vous dans le répertoire bin : `cd objectgridRoot/bin`

b. Démarrez le serveur de catalogue :

- Linux UNIX

```
./startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.policy="../security/og_auth.policy"
```

- Windows

```
startOgServer.bat c0 -objectGridFile ..\xml\SecureSimpleApp.xml
-deploymentPolicyFile ..\xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
 -serverProps ..\security\server.properties -jvmArgs
-Djava.security.auth.login.config="..\security\og_jaas.config"
-Djava.security.policy="..\security\og_auth.policy"
```

- 8.6+

  Linux UNIX

```
./startXsServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.policy="../security/og_auth.policy"
```

- 8.6+

  Windows

```
startXsServer.bat c0 -objectGridFile ..\xml\SecureSimpleApp.xml
-deploymentPolicyFile ..\xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
 -serverProps ..\security\server.properties -jvmArgs
-Djava.security.auth.login.config="..\security\og_jaas.config"
-Djava.security.policy="..\security\og_auth.policy"
```

L'exception suivante s'affiche :

```
Caused by: com.ibm.websphere.objectgrid.ObjectGridRPCException:
    com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
        SSL connection fails and plain socket cannot be used.
```

Enfin, modifiez le fichier `server.properties` pour utiliser de nouveau le fichier `key.jks`.

Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server

Ce tutoriel explique comment sécuriser un déploiement de serveur WebSphere eXtreme Scale dans un environnement WebSphere Application Server.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configurer WebSphere eXtreme Scale pour utiliser les plug-in d'authentification WebSphere Application Server
- Configurer la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2
- Utiliser l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utiliser un module de connexion personnalisé pour l'autorisation JAAS de groupe
- Utiliser l'utilitaire WebSphere eXtreme Scale **xscmd** dans l'environnement WebSphere Application Server

Durée

Ce tutoriel dure environ 4 heures.

Présentation : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server en utilisant des plug-in WebSphere Application Server Authentication

Dans ce tutoriel, vous intégrez la sécurité WebSphere eXtreme Scale à WebSphere Application Server. Tout d'abord, vous devez configurer l'authentification avec une application Web simple qui utilise les données d'identification de l'utilisateur authentifié à partir de l'unité d'exécution en cours pour se connecter à ObjectGrid. Ensuite, vous examinez le chiffrement des données qui sont transférées entre le client et le serveur avec la sécurité de la couche de transport. Pour accorder aux utilisateurs différents niveaux des autorisations, vous pouvez configurer Java Authentication and Authorization Service (JAAS). Une fois la configuration effectuée, vous pouvez utiliser l'utilitaire **xscmd** pour contrôler les grilles de données et les mappes.

Ce tutoriel suppose que l'ensemble de vos clients, serveurs de conteneur et serveurs de catalogue WebSphere eXtreme Scale sont déployés dans l'environnement WebSphere Application Server.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configurer WebSphere eXtreme Scale pour utiliser les plug-in d'authentification WebSphere Application Server
- Configurer la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2
- Utiliser l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server

- Utiliser un module de connexion personnalisé pour l'autorisation JAAS de groupe
- Utiliser l'utilitaire WebSphere eXtreme Scale **xscmd** dans l'environnement WebSphere Application Server

Durée

Ce tutoriel prend 4 heures environ.

Niveau de compétence

Intermédiaire.

Public ciblé

Développeurs et les administrateurs qui sont intéressés par l'intégration de la sécurité entre WebSphere eXtreme Scale et WebSphere Application Server.

Configuration système requise et topologie

- WebSphere Application Server Version 7.0.0.11 ou version suivante
- Mettez à jour l'environnement d'exécution Java pour appliquer le correctif suivant : IZ79819: IBMJDK FAILS TO READ PRINCIPAL STATEMENT WITH WHITESPACE FROM SECURITY FILE

Ce tutoriel utilise dans l'exemple quatre serveurs d'applications WebSphere Application Server et un gestionnaire de déploiement.

Prérequis

Une connaissance de base des éléments suivants est utile avant de démarrer ce tutoriel :

- Modèle de programmation WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere Application Server

Pour plus d'informations sur l'intégration de la sécurité WebSphere eXtreme Scale et WebSphere Application Server, voir «Intégration de la sécurité dans WebSphere Application Server», à la page 643.

Module 1 : Préparation de WebSphere Application Server

Avant de commencer le tutoriel d'intégration à WebSphere eXtreme Scale, vous devez créer une configuration de sécurité de base dans WebSphere Application Server.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer la sécurité WebSphere Application Server pour utiliser un fichier interne basé sur un référentiel fédéré sous la forme d'un registre de comptes utilisateur ;
- créer des groupes d'utilisateurs et des utilisateurs ;
- créer des clusters pour l'application et les serveurs WebSphere eXtreme Scale.

Durée

Ce module prend 60 minutes environ.

Leçon 1.1 : Compréhension de la topologie et obtention des fichiers du tutoriel

Pour préparer votre environnement pour le tutoriel, vous devez configurer la sécurité WebSphere Application Server. Vous configurez la sécurité d'administration et d'application en utilisant des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur.

Cette leçon vous guide dans l'exemple de topologie et les applications qui sont utilisés dans le tutoriel. Pour commencer à exécuter le tutoriel, vous devez télécharger les applications et placer les fichiers de configuration dans les emplacements propres à votre environnement. Vous pouvez télécharger l'exemple d'application depuis le wiki WebSphere eXtreme Scale.

Exemple de topologie WebSphere Application Server : Ce tutoriel vous guide tout au long de la création de quatre serveurs d'applications WebSphere Application Server pour montrer l'utilisation des exemples d'applications avec la sécurité activée. Ces serveurs d'applications sont regroupés dans deux clusters contenant deux serveurs :

- **Cluster appCluster** : héberge l'exemple d'application d'entreprise EmployeeManagement. Ce cluster contient les deux serveurs d'applications s1 et s2.
- **Clster xsCluster** : héberge les serveurs de conteneur eXtreme Scale. Ce cluster contient les deux serveurs d'applications xs1 et xs2.

Dans cette topologie de déploiement, les serveurs d'applications s1 et s2 sont les serveurs client qui accèdent aux données qui sont stockées dans la grille de données. Les serveurs xs1 et xs2 sont les serveurs de conteneur qui hébergent la grille de données.

Le serveur de catalogue est déployé dans le processus du gestionnaire de déploiement par défaut. Ce tutoriel utilise le comportement par défaut. L'hébergement du serveur de catalogue dans le gestionnaire de déploiement n'est pas une pratique recommandée dans un environnement de production. Dans un environnement de production, vous devez créer un domaine de service de catalogue pour définir où les serveurs de catalogue démarrent. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307.

Autre configuration : vous pouvez héberger tous les serveurs d'applications dans un seul cluster, tel que appCluster. Avec cette configuration, tous les serveurs du cluster sont les clients et les serveurs de conteneur. Ce tutoriel utilise deux clusters pour distinguer les serveurs d'applications qui hébergent les clients et les serveurs de conteneur.

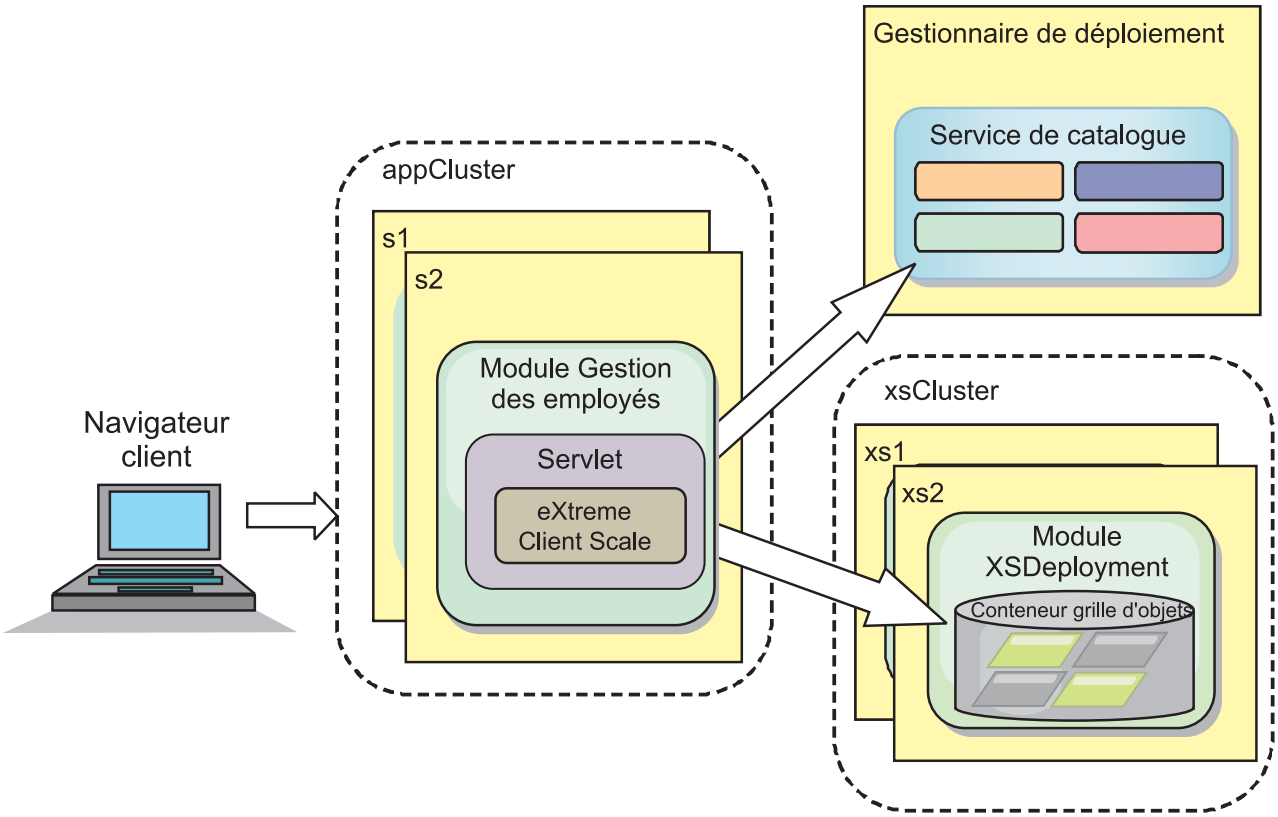


Figure 23. Topologie du tutoriel

Applications : Dans ce tutoriel, vous utilisez deux applications et un fichier de bibliothèque partagée :

- **`EmployeeManagement.ear`** : l'application `EmployeeManagement.ear` est une application d'entreprise simplifiée Java 2 Platform, Enterprise Edition (J2EE). Elle contient un module Web pour gérer les profils des employés. Le module Web contient le fichier `management.jsp` pour afficher, insérer, mettre à jour et supprimer les profils d'employés qui sont stockés dans les serveurs de conteneur.
- **`XSDeployment.ear`** : cette application contient un module d'application d'entreprise, sans artefacts d'application. Les objets cache sont regroupés dans le fichier `EmployeeData.jar`. Le fichier `EmployeeData.jar` est déployé comme bibliothèque partagée pour le fichier `XSDeployment.ear` pour que le fichier `XSDeployment.ear` puisse accéder aux classes. Cette application à pour fonction de modulariser le fichier de configuration eXtreme Scale. Lorsque cette application d'entreprise est démarrée, les fichiers de configuration eXtreme Scale sont automatiquement détectés par l'environnement d'exécution eXtreme Scale pour créer les serveurs de conteneur. Ces fichiers de configuration incluent les fichiers `objectGrid.xml` et `objectGridDeployment.xml`.
- **`EmployeeData.jar`** : ce fichier JAR contient une classe, com.ibm.websphere.sample.xs.data.EmployeeData. Cette classe représente les données de l'employé qui sont stockés dans la grille. Ce fichier d'archive Java (JAR) est déployé avec les fichiers `EmployeeManagement.ear` et `XSDeployment.ear` comme bibliothèque partagée.

Obtention des fichiers du tutoriel :

1. Téléchargez les fichiers `WASSecurity.zip` et `security.zip`. Vous pouvez télécharger l'exemple d'application depuis le wiki WebSphere eXtreme Scale.

2. Extrayez le ficher WASSecurity.zip dans un répertoire pour afficher les données binaires et les artefacts source (par exemple un répertoire wxs_samples/). Ce répertoire est *samples_home* pour le reste du tutoriel. Pour la description du contenu du fichier WASSecurity.zip et savoir comment charger le source dans votre espace de travail Eclipse, voir le fichier README.txt dans le package.
3. Extrayez le fichier security.zip vers le répertoire *samples_home* Le fichier security.zip contient les fichiers de configuration de sécurité suivants qui sont utilisés dans ce tutoriel :
   - catServer2.props
   - server2.props
   - client2.props
   - securityWAS2.xml
   - xsAuth2.props

A propos des fichiers de configuration :

Les fichiers objectGrid.xml et objectGridDeployment.xml créent les grilles de données et les mappes qui stockent les données d'application.

Ces fichiers de configuration doivent s'appeler objectGrid.xml et objectGridDeployment.xml. Lorsque le serveur d'applications démarre, eXtreme Scale détecte ces fichiers dans le répertoire META-INF de l'EJB et des modules Web. Si ces fichiers sont trouvés, il suppose que la machine JVM (Java virtual machine) fait office de serveur de conteneur pour les grilles de données définie dans les fichiers de configuration.

Fichier objectGrid.xml

Le fichier objectGrid.xml défini un ObjectGrid nommé Grid. La grille de données Grid a une mappe, la mappe Map1, qui stocke le profil d'employé pour l'application.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
        <objectGrid name="Grid" txTimeout="15">
            <backingMap name="Map1" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

Fichier objectGridDeployment.xml

Le fichier objectGridDeployment.xml indique comment déployer la grille de données Grid. Lorsque la grille est déployée, elle dispose de cinq partitions et d'une réplique synchrone.

```
<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="1" >
            <map ref="Map1"/>
```

```
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez découvert la topologie du tutoriel et ajouté les fichiers de configuration et les exemples d'applications à votre environnement.

Si vous souhaitez en savoir plus sur le démarrage automatique des serveurs de conteneur, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur», à la page 333.

Leçon 1.2 : Configuration de l'environnement WebSphere Application Server

Pour préparer votre environnement pour le tutoriel, vous devez configurer la sécurité WebSphere Application Server. Activez la sécurité d'administration et d'application en utilisant des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur. Ensuite, vous pouvez créer des clusters de serveurs pour héberger les serveurs d'applications client et les serveurs de conteneur.

Les étapes suivantes s'appliquent à WebSphere Application Server Version 7.0. Toutefois, vous pouvez appliquez les concepts aux versions antérieures de WebSphere Application Server.

Configuration de la sécurité WebSphere Application Server :

1. Configurez la sécurité WebSphere Application Server.
   a. Dans la console d'administration WebSphere Application Server, cliquez sur **Sécurité** > **Sécurité globale**.
   b. Sélectionnez **Référentiels fédérés** comme **définition de domaine disponible**. Cliquez sur **Définir comme actif**.
   c. Cliquez sur **Configurer** pour accéder au panneau des référentiels fédérés.
   d. Entrez le **nom de l'administrateur principal**, tel que `admin`. Cliquez sur **Appliquer.**
   e. Lorsque vous y êtes invité, entrez le mot de passe de l'administrateur et cliquez sur **OK**. Sauvegardez vos modifications.
   f. Dans la page **Sécurité globale**, vérifiez que le paramètre **Référentiels fédérés** est affecté du registre de comptes utilisateur en cours.
   g. Sélectionnez **Activer la sécurité administrative**, **Activer la sécurité de l'application** et **Utiliser la sécurité Java 2 pour limiter l'accès de l'application aux ressources locales**. Cliquez sur **Appliquer** et enregistrez les modifications.
   h. Redémarrez le gestionnaire de déploiement et les serveurs d'applications actifs.

   La sécurité administrative WebSphere Application Server est activée à l'aide des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur.
2. Créez deux groupes d'utilisateurs : adminGroup et operatorGroup.
   a. Cliquez sur **Utilisateurs et groupes** > **Gérer les groupes** > **Créer...**
   b. Entrez `adminGroup` comme nom de groupe. Entrez `Administration de groupe` comme description. Cliquez sur **Créer**.

c. Cliquez sur **Créer à l'identique**. Entrez `operatorGroup` comme nom de groupe. Entrez `Groupe d'opérateurs` comme description. Cliquez sur **Créer**.
d. Cliquez sur **Fermer**.

3. Créez les utilisateurs `admin1` et `operator1`.
   a. Cliquez sur **Utilisateurs et groupes** > **Gérer les utilisateurs** > **Créer...**
   b. Créez l'utilisateur `admin1` avec le prénom `Jean` et le nom `Doe` avec le mot de passe `admin1`. Cliquez sur **Créer**.
   c. Créer un second utilisateur. Cliquez sur **Créer à l'identique** pour créer l'utilisateur `operator1` avec le prénom `Jane` et le nom `Doe` avec le mot de passe `operator1`. Cliquez sur **Créer**. Cliquez sur **Fermer**.
4. Ajouter des utilisateurs aux groupes d'utilisateurs. Ajoutez l'utilisateur `admin1` à `adminGroup` et l'utilisateur `operator1` à `operatorGroup`.
   a. Cliquez sur **Utilisateurs et groupes** > **Gérer les utilisateurs**.
   b. Recherchez des utilisateurs à ajouter aux groupes. Cliquez sur **Rechercher..** et définissez un astérisque (*) comme valeur de recherche pour afficher tous les utilisateurs.
   c. Dans le résultat de la recherche, sélectionnez l'utilisateur `admin1` et cliquez sur l'onglet **Groupes**. Cliquez sur **Ajouter** pour ajouter le groupe.
   d. Recherchez les groupes pour identifier les groupes disponibles. Cliquez sur `adminGroup` et sur **Ajouter**.
   e. Répétez ces étapes pour ajouter l'utilisateur `operator1` au groupe d'utilisateurs `operatorGroup`.
5. Sauvegardez vos modifications, déconnectez-vous de la console d'administration, puis redémarrez le gestionnaire de déploiement et l'agent de noeud pour activer les paramètres de sécurité.

Vous avez activé la sécurité et créé des utilisateurs et les groupes d'utilisateurs ont un accès administrateur et opérateur à la configuration WebSphere Application Server.

Création de clusters de serveurs :

Créez deux clusters de serveurs dans votre configuration WebSphere Application Server : `appCluster` pour héberger l'exemple d'application du tutoriel et `xsCluster` pour héberger la grille de données.

1. Dans la console d'administration WebSphere Application Server, ouvrez le panneau des clusters. Cliquez sur **Serveurs** > **Clusters** > **Clusters de serveurs d'applications WebSphere** > **Nouveau**.
2. Entrez `appCluster` comme nom de cluster, ne renseignez pas l'option **Environnement local préféré** et cliquez sur **Suivant**.
3. Créez des serveurs dans le cluster. Créez le serveur `s1` en conservant les options par défaut. Ajoutez le membre `s2` au cluster.
4. Exécutez les étapes restantes dans l'assistant pour créer le cluster. Sauvegardez les modifications.
5. Répétez ces étapes pour créer le cluster `xsCluster`. Ce cluster contient les deux serveurs `xs1` et `xs2`.

Point de contrôle de la leçon :

Vous avez activé la sécurité globale de la cellule WebSphere Application Server, créé des utilisateurs et des groupes d'utilisateurs et créé des clusters pour héberger l'application et la grille de données.

Module 2 : Configuration de WebSphere eXtreme Scale pour utiliser les plug-in WebSphere Application Server Authentication

Une fois que vous avez créé la configuration WebSphere Application Server, vous pouvez intégrer l'authentification WebSphere eXtreme Scale à WebSphere Application Server.

Lorsqu'un client WebSphere eXtreme Scale se connecte à un serveur de conteneur qui exige une authentification, le client doit fournir un générateur de données d'identification représenté par l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. Un générateur de données d'identification est une fabrique pour créer des données d'identification de client. Les données d'identification de client peuvent être une paire nom-mot de passe, un ticket Kerberos, un certificat client ou des données d'identification de client dans n'importe quel format convenu entre le client et le serveur. Voir la documentation sur l'API Credential pour plus d'informations. Dans cet exemple, le client WebSphere eXtreme Scale est l'application Web EmployeeManagment qui est déployée dans le cluster appCluster. Les données d'identification de client sont un jeton de sécurité WebSphere qui représente l'identité de l'utilisateur Web.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer la sécurité du serveur du client ;
- configurer la sécurité du serveur de catalogue ;
- configurer la sécurité du serveur de conteneur ;
- installer et exécuter l'exemple d'application.

Durée

Ce module prend 60 minutes environ.

Leçon 2.1 : Configuration de la sécurité du serveur de conteneur

Le fichier des propriétés du client indique la classe d'implémentation CredentialGenerator à utiliser.

Configurez le fichier de propriétés du client avec la propriété JVM **-Dobjectgrid.client.props**. Le nom de fichier spécifié pour cette propriété est un chemin absolu, tel que *samples_home*/security/client2.props. Voir Fichier de propriétés du client pour plus d'informations sur le fichier des propriétés du client.

Contenu du fichier des propriétés du client :
Cet exemple utilise des jetons de sécurité WebSphere Application Server comme données d'identification. Le fichier client2.props se trouve dans le répertoire *samples_home*/security. Le fichier client2.props contient les paramètres suivants :

securityEnabled
Lorsque la valeur est true, elle indique que le client doit envoyer les informations de sécurité disponibles au serveur.

credentialAuthentication
Lorsque la valeur est Supported, elle indique que le client prend en charge l'authentification des données d'identification.

credentialGeneratorClass

Indique la classe d'implémentation de com.ibm.websphere.objectgrid.security.plugins.builtins. Classe WSTokenCredentialGenerator pour que le client puisse extraire les jetons de sécurité de l'unité d'exécution. Voir «Intégration de la sécurité dans WebSphere Application Server», à la page 643 pour plus d'informations sur les jetons de sécurité extraits.

Définition du fichier des propriétés du client en utilisant des propriétés JVM (Java virtual machine) :

Dans la console d'administration, procédez comme suit pour les serveurs `s1` et `s2` dans le cluster `appCluster`. Si vous utilisez une topologie différente, procédez comme suit pour tous les serveurs d'applications sur lesquels l'application `EmployeeManagement` est déployée.

1. **Serveurs** > **Serveurs d'applications WebSphere** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
2. Créez la propriété JVM générique suivante pour définir l'emplacement du fichier de propriétés du client :

   ```
   -Dobjectgrid.client.props=samples_home/security/client2.props
   ```
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez édité le fichier de propriétés du client et configuré les serveurs dans le cluster `appCluster` pour utiliser le fichier des propriétés du client. Le fichier des propriétés indique la classe d'implémentation CredentialGenerator à utiliser.

Leçon 2.2 : Configuration de la sécurité du serveur de catalogue

Propriétés de sécurité communes à tous les serveurs WebSphere eXtreme Scale, y compris les serveurs de service de catalogue et de conteneur et les propriétés de sécurité du serveur de catalogue.

Les propriétés de sécurité qui sont communes aux serveurs de catalogue et aux serveurs de conteneur sont configurées dans le fichier descripteur XML de sécurité. La configuration de l'authentificateur, qui représente le registre d'utilisateurs et le mécanisme d'authentification, est un exemple des propriétés communes. Voir Fichier XML du descripteur de sécurité pour plus d'informations sur les propriétés de sécurité.

Pour configurer le fichier descripteur XML de sécurité, créez une propriété -Dobjectgrid.cluster.security.xml.url dans l'argument de machine virtuelle Java. Le nom de fichier spécifié pour cette propriété doit avoir le format URL, tel que `file:///samples_home/security/securityWAS2.xml`.

Fichier `securityWAS2.xml` :

Dans ce tutoriel, le fichier `securityWAS2.xml` se trouve dans le répertoire *samples_home*`/security`. Contenu du fichier `securityWAS2.xml` avec les commentaires supprimés :

```
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true">
  <authenticator
```

```
 className="com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenAuthenticator">
  </authenticator>
 </security>
</securityConfig>
```

Les propriétés suivantes sont définies dans le fichier `securityWAS2.xml` :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer au serveur de catalogue que la sécurité globale WebSphere eXtreme Scale est activée.

authenticator
: L'authentificateur est configuré comme suit : com.ibm.websphere.objectgrid.security.plugins.builtins. Classe WSTokenAuthenticator. Avec cette implémentation intégrées dans le plug-in Authenticator, le serveur WebSphere eXtreme Scale peut convertir les jetons de sécurité en objet Subject. Voir «Intégration de la sécurité dans WebSphere Application Server», à la page 643 pour plus d'informations sur la conversion des jetons de sécurité.

Fichier `catServer2.props` :

Le fichier de propriétés du serveur stocke les propriétés du serveur qui incluent ses propriétés du serveur. Pour plus d'informations, voir Fichier de propriétés du serveur. Vous pouvez configurer le fichier de propriétés du serveur avec la propriété `-Dobjectgrid.server.props` dans l'argument JVM. Définissez la valeur de nom de fichier de cette propriété dans un chemin absolu, tel que *samples_home*`/security/catServer2.props`. Pour ce tutoriel, un fichier `catServer2.props` est inclus dans le répertoire *samples_home*`/security`. Contenu du fichier `catServer2.props` avec les commentaires supprimés :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer que le serveur est un serveur sécurisé.

credentialAuthentication
: La propriété credentialAuthentication a la valeur `Requis`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification.

secureTokenManagerType
: La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors du regroupement avec les serveurs existants.

authenticationSecret
: La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

transportType
: La propriété transportType a la valeur `TCP/IP` initialement. Plus loin dans le tutoriel, la sécurité du transport est activée.

Définition du fichier des propriétés du serveur avec des propriétés JVM :

Définissez le fichier des propriétés de serveur sur le serveur du gestionnaire de déploiement. Si vous utilisez une topologie différente de celle du tutoriel, définissez le fichier de propriétés sur tous les serveurs d'applications que vous utilisez pour héberger les serveurs de conteneur.

1. Ouvrez la configuration de la machine virtuelle Java du serveur. Dans la console d'administration, cliquez sur **Administration du système** > **Gestionnaire de déploiement** > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
2. Ajoutez les arguments JVM génériques suivants :

   ```
   -Dobjectgrid.cluster.security.xml.url=file:///samples_home/security/securityWAS2.xml
   -Dobjectgrid.server.props=samples_home/security/catServer2.props
   ```

3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez configuré la sécurité du serveur de catalogue en associant les fichiers `securityWAS2.xml` et `catServer2.props` au gestionnaire de déploiement qui héberge le processus serveur de catalogue dans la configuration WebSphere Application Server.

Leçon 2.3 : Configuration de la sécurité du serveur de conteneur

Lorsqu'un serveur de conteneur se connecte au service de catalogue, le serveur de conteneur obtient toutes les configurations de sécurité qui sont configurées dans le fichier XML Object Grid Security XML, telles que la configuration de l'authentificateur, la valeur de temporisation de la session de connexion et d'autres informations de configuration. Un serveur de conteneur dispose également de ses propres propriétés de sécurité dans le fichier de propriétés du serveur.

Configurez le fichier de propriétés du serveur avec la propriété JVM `-Dobjectgrid.server.props` (Java virtual machine). Le nom de fichier spécifié pour cette propriété est un chemin absolu, tel que *samples_home*`/security/server2.props`.

Dans ce tutoriel, les serveurs de conteneur sont hébergés dans les serveurs `xs1` et `xs2` du cluster `xsCluster`.

Fichier `server2.props` :

Le fichier `server2.props` se trouve dans le répertoire *samples_home*`/security` du répertoire `WASSecurity`. Les propriétés suivantes sont définies dans le fichier `server2.props` :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer que ce serveur de conteneur est un serveur sécurisé.

credentialAuthentication
: La propriété credentialAuthentication a la valeur `Required`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification.

secureTokenManagerType
: La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors de la connexion aux serveurs existants.

authenticationSecret
: La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`.

Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur devient membre de la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

Définition du fichier de propriétés du serveur avec des propriétés JVM :

Définition du fichier de propriétés des serveurs xs1 et xs2. Si vous n'utilisez pas la topologie du tutoriel, définissez le fichier de propriétés de serveur sur tous les serveurs d'applications que vous utilisez pour héberger les serveurs de conteneur.

1. Ouvrez la page de la machine virtuelle Java du serveur. **Serveurs** > **Serveurs d'applications** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**
2. Ajoutez les arguments JVM génériques suivants :
   `-Dobjectgrid.server.props=`*samples_home*`/security/server2.props`
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

A présent, l'authentification du serveur WebSphere eXtreme Scale est sécurisée. En configurant cette sécurité, toutes les applications qui tentent de se connecter aux serveurs WebSphere eXtreme Scale doivent fournir des données d'identification. Dans ce tutoriel, WSTokenAuthenticator est l'authentificateur. En conséquence, le client doit fournir un jeton de sécurité WebSphere Application Server.

Leçon 2.4 : Installation et exécution de l'exemple

Une fois l'authentification configurée, vous pouvez installer et exécuter l'exemple d'application.

Création d'une bibliothèque partagée pour le fichier `EmployeeData.jar` :

1. Dans la console d'administration de WebSphere Application Server, ouvrez la page **Bibliothèques partagées**. Cliquez sur **Environnement** > **Bibliothèques partagées**.
2. Choisissez la portée **cellule**.
3. Créez la bibliothèque partagée. Cliquez sur **Nouveau**. Entrez `EmployeeManagementLIB` pour le **nom**. Entrez le chemin d'accès au fichier `EmployeeData.jar` dans le chemin de classes, par exemple, *samples_home*`/WASSecurity/EmployeeData.jar`.
4. Cliquez sur **Appliquer**.

Installation de l'exemple :

1. Installez le fichier `EmployeeManagement.ear`.
   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé d'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster `appCluster` pour installer le module EmployeeManagementWeb.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module `EmployeeManagementWeb`.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque `EmployeeManagementLIB`.

e. Associez le rôle webUser à **Tous authentifiés dans le domaine de l'application**.

f. Cliquez sur **OK**.

Les clients s'exécutent dans les serveurs s1 et s2 du cluster.

2. Installez l'exemple de fichier XSDeployment.ear.
   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé pour l'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster xsCluster pour installer le module Web XSDeploymentWeb.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module XSDeploymentWeb.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque EmployeeManagementLIB.
   e. Cliquez sur **OK**.

   Les serveurs xs1 et xs2 de ce cluster hébergent les serveurs de conteneur.
3. Redémarrez le gestionnaire de déploiement. Lorsque le gestionnaire de déploiement démarre, le serveur de catalogue démarre également. Si vous examinez le fichier SystemOut.log du gestionnaire de déploiement, vous pouvez voir le message suivant qui indique que le fichier des propriétés du serveur eXtreme Scale est chargé.

   ```
   CWOBJ0913I: Server property files have been loaded:
   /wxs_samples/security/catServer2.props.
   ```
4. Redémarrez le cluster xsCluster. Lorsque xsCluster démarre, l'application XSDeployment démarre et un serveur de conteneur est démarré sur les serveurs xs1 et xs2 respectivement. Si vous examinez le fichier SystemOut.log du gestionnaire de déploiement, vous pouvez voir le message suivant qui indique que le fichier des propriétés du serveur est chargé :

   ```
   CWOBJ0913I: Server property files have been loaded:
   /wxs_samples/security/server2.props.
   ```
5. Redémarrez le cluster appClusters. Lorsque le cluster appCluster démarre, l'application EmployeeManagement démarre également. Si vous examinez le fichier SystemOut.log des serveurs s1 et s2, le message suivant indique que le fichiers des propriétés est chargé.

   ```
   CWOBJ0924I: The client property file {0} has been loaded.
   ```

   Vous pouvez ignorer les messages d'avertissement concernant les propriétés authenticationRetryCount, transportType et clientCertificateAuthentication. Les valeurs par défaut doivent être utilisées, car les valeurs n'ont pas été indiquées dans le fichier des propriétés.Si vous utilisez WebSphere eXtreme Scale Version 7.0, le message CWOBJ9000I s'affiche en anglais et indique que le fichier des propriétés du client a été chargé. Si vous ne voyez pas le message attendu, vérifiez que vous avez configuré la propriété -Dobjectgrid.server.props ou -Dobjectgrid.client.props dans l'argument JVM. Si vous l'avez configurée, vérifiez que le tiret (-) est un caractère UTF.

Exécution de l'exemple d'application :

1. Exécutez le fichier management.jsp.. Dans un navigateur Web, accédez à http://<*your_servername*>:<*port*>/EmployeeManagementWeb/management.jsp. Par exemple, vous pouvez utiliser l'URL http://localhost:9080/EmployeeManagementWeb/management.jsp.
2. Fournissez les informations d'authentification à l'application. Entrez les données d'identification de l'utilisateur que vous avez associé au rôle webUser. Par

défaut, ce rôle utilisateur est associé à tous les utilisateurs authentifiés. Tapez `admin1` comme ID utilisateur et `admin1` comme mot de passe. Une page pour afficher, ajouter, mettre à jour et supprimer des employés apparaît.

3. Affichez les employés. Cliquez sur **Ajouter un employé**. Entrez `emp1@acme.com` comme adresse électronique et cliquez sur **Soumettre**. Un message indique que l'utilisateur est introuvable.
4. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Entrez `emp1@acme.com` comme adresse électronique, `Joe` comme prénom et `Doe` comme nom. Cliquez sur **Soumettre**. Un message s'affiche pour indiquer qu'un employé avec l'adresse `emp1@acme.com` a été ajouté.
5. Affichez le nouvel employé. Cliquez sur **Afficher un employé**. Entrez `emp1@acme.com` comme adresse électronique avec des zones vides pour les nom et prénom, et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été trouvé et les noms corrects figurent dans les zones du prénom et du nom.
6. Supprimez l'employé. Cliquez sur **Supprimer un employé**. Entrez `emp1@acme.com` et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été supprimé.

Point de contrôle de la leçon :

Vous avez installé et exécuté l'exemple d'application. Comme ce tutoriel utilise l'intégration WebSphere Application Server, vous ne pouvez pas voir le scénario lorsqu'un client ne parvient pas à s'authentifier sur le serveur eXtreme Scale. Si l'utilisateur s'authentifie auprès d' WebSphere Application Server correctement, eXtreme Scale est également correctement authentifié.

Module 3 : Configuration de la sécurité du transport

Configuration de la sécurité du transport pour protéger le transfert des données entre les clients et les serveurs dans la configuration.

Dans le module précédent dans le tutoriel, vous avez activé l'authentification WebSphere eXtreme Scale. Avec l'authentification, une application qui tente de se connecter au serveur WebSphere eXtreme Scale doit fournir des données d'identification. Par conséquent, un client non authentifié peut se connecter au serveur WebSphere eXtreme Scale. Les clients doivent être une application authentifiée qui s'exécute dans une cellule WebSphere Application Server.

Avec la configuration jusqu'à ce module, le transfert de données entre les clients dans le cluster appCluster et les serveurs du cluster xsCluster n'est pas chiffré. Cette configuration peut être acceptable si vos clusters WebSphere Application Server sont installés sur les serveurs derrière un pare-feu. Toutefois, dans certains scénarios, le trafic non chiffré n'est pas accepté pour certaines raisons, même si la topologie est protégée par un pare-feu. Par exemple, une politique gouvernementale pourrait imposer de chiffrer le trafic. WebSphere eXtreme Scale prend en charge Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour sécuriser la communication entre les noeuds finals ObjectGrid, qui incluent des serveurs client, des serveurs de conteneur et des serveurs de catalogue.

Dans cet exemple de déploiement, les clients et les servers de conteneur eXtreme Scale s'exécutent tous dans l'environnement WebSphere Application Server. Les propriétés client ou serveur ne sont pas nécessaires pour configurer les paramètres SSL, car la sécurité du transport eXtreme Scale est gérée par les paramètres de transport CSIV2 (Application Server Common Secure Interoperability Protocol Version 2). Les serveurs WebSphere eXtreme Scale utilisent la même instance ORB

(Object Request Broker) que les serveurs d'applications où ils sont exécutés. Définissez tous les paramètres SSL des serveurs client et de conteneur dans la configuration WebSphere Application Server en utilisant ces paramètres de transport CSIv2. Le serveur de catalogue dispose de ses propres chemins de transport propriétaires qui n'utilisent pas IIOP (Internet Inter-ORB Protocol) ni RMI (Remote Method Invocation). En raison de ces chemins de transport propriétaires, le serveur de catalogue ne peuvent pas être gérés par les paramètres de transport WebSphere Application Server CSIV2. Vous devez donc définir les propriétés SSL dans le fichier des propriétés du serveur de catalogue.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer le transport entrant et sortant CSIv2 ;
- ajouter des propriétés SSL dans le fichier de propriétés du serveur de catalogue ;
- vérifier le fichier des propriétés ORB ;
- exécuter l'exemple.

Durée

Ce module prend 60 minutes environ.

Prérequis

Cette étape du tutoriel repose sur les modules précédents. Etudiez les modules précédents du présent tutoriel avant de configurer la sécurité du transport.

Leçon 3.1 : Configuration du transport entrant et sortant CSIv2

Pour configurer Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour le transport du serveur, affectez aux propriétés de transport entrant et sortant Common Secure Interoperability Protocol Version 2 (CSIv2) la valeur `SSL requis` pour tous les serveurs WebSphere Application Server qui hébergent des clients, des serveurs de catalogue et des serveurs de conteneur.

Dans l'exemple de topologie du tutoriel, vous devez définir ces propriétés pour les serveurs d'applications s1, s2, xs1 et xs2. Procédez comme suit pout définir les transports entrant et sortant de tous les serveurs de la configuration.

Définissez les transports entrant et sortant dans la console d'administration. Vérifiez que la sécurité administrative est activée.

- **WebSphere Application Server Version 7.0** : Cliquez sur **Sécurité** > **Sécurité globale** > **Sécurité RMI/IIOP** > **Communications entrants CSIv2**. Dans la couche de transport CSIv2 remplacez le type de transport par **SSL requis**. Répétez cette étape pour configurer les communications sortantes CSIv2.

Vous pouvez utiliser les paramètres de sécurité de noeud final géré de manière centralisée, ou configurer des référentiels SSL. Voir Paramètres de transport entrant CSIV2 (Common Secure Interoperability Version 2) pour plus d'informations.

Leçon 3.2 : Ajout de propriétés SSL au fichier des propriétés du serveur de catalogue

Le serveur de catalogue dispose de ses propres chemins de transport propriétaires qui ne peuvent pas être gérés par les paramètres de transport WebSphere

Application Server Common Secure Interoperability Protocol Version 2 (CSIV2). Vous devez donc définir les propriétés SSL (Secure Sockets Layer) dans le fichier des propriétés du serveur de catalogue.

Pour configurer la sécurité du serveur de catalogue, des étapes supplémentaires sont nécessaires, car le serveur de catalogue dispose de ses propres chemins de transport propriétaires. Ces chemins de transport ne peut pas être gérés par les paramètres de transport du serveur d'applications CSIV2.

1. Editez les propriétés SSL dans le fichier `catServer2.props`. Pour configurer la sécurité du serveur de catalogue, supprimez les propriétés SSL suivantes dans le fichier de propriétés du serveur de catalogue. Pour ce tutoriel, les propriétés du serveur de catalogue se trouvent dans le fichier `catServer2.props`. Mettez à jour les propriétés keyStore et trustStore pour faire référence à l'emplacement approprié dans votre environnement.

   ```
   #alias=default
   #contextProvider=IBMJSSE2
   #protocol=SSL
   #keyStoreType=PKCS12
   #keyStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR_NAME>/config/
   cells/<CELL_NAME>/nodes/<NODE_NAME>/key.p12
   #keyStorePassword=WebAS
   #trustStoreType=PKCS12
   #trustStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR_NAME>/config/
   cells/<CELL_NAME>/nodes/<NODE_NAME>/trust.p12
   #trustStorePassword=WebAS
   #clientAuthentication=false
   ```

   Le fichier `catServer2.props` utilise le fichier de clés et le fichier de clés certifiées au niveau du noeud WebSphere Application Server par défaut. Si vous déployez un environnement de déploiement plus complexe, vous devez choisir le fichier de clés et le fichier de clés certifiées corrects. Dans certains cas, vous devez créer un fichier de clés et un fichier de clés certifiées et importez les clés depuis les fichiers de clés des autres serveurs. Notez que la chaîne `WebAS` est le mot de passe par défaut des fichiers de clés et de clés certifiées WebSphere Application Server. Voir Configuration des certificats autosignés pour plus d'informations.
2. Dans le fichier `catServer2.props`, mettez à jour la valeur de la propriété transportType. Pour les étapes précédentes du tutoriel, la valeur est `TCP/IP`. Remplacez la valeur `SSL requis`.
3. Redémarrez le gestionnaire de déploiement pour activer les modifications apportées aux paramètres de sécurité du serveur de catalogue.

Point de contrôle de la leçon :

Vous avez défini les propriétés SSL pour le serveur de catalogue.

Leçon 3.3 : Exécution de l'exemple

Redémarrez tous les serveurs et exécutez de nouveau le modèle d'application. Vous devriez être en mesure d'exécuter les étapes sans aucun problème.

Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 126 pour plus d'informations sur l'exécution et l'installation de l'exemple d'application.

Point de contrôle de la leçon :

Vous avez exécuté l'exemple d'application avec la sécurité du transport activée.

Module 4 : Utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server

Maintenant que vous avez configuré l'authentification pour les clients, vous pouvez configurer les autorisations de manière plus précise pour accorder aux utilisateurs des autorisations différentes. Par exemple, un "opérateur" peut être autorisé uniquement à afficher les données, alors qu'un "gestionnaire" peut exécuter toutes les opérations.

Après avoir authentifié un client, comme dans le module précédent dans ce tutoriel, vous pouvez attribuer des privilèges de sécurité par le biais des mécanismes d'autorisation eXtreme Scale. Le module précédent de ce tutoriel vous a montré comment activer l'authentification pour une grille de données à l'aide de l'intégration à WebSphere Application Server. Par conséquent, aucun client non authentifié ne peut se connecter aux serveurs eXtreme Scale ni envoyer des demandes au système. Toutefois, tous les clients authentifiés possèdent les mêmes permissions ou privilèges liés au serveur, tels que la lecture, l'écriture ou la suppression des données stockées dans les mappes ObjectGrid. Les clients peuvent également soumettre tout type de requête.

Cette partie du tutoriel explique comment utiliser l'autorisation eXtreme Scale pour attribuer différents privilèges aux utilisateurs authentifiés. WebSphere eXtreme Scale utilise un mécanisme d'autorisation basé sur l'autorisation. Vous pouvez affecter des catégories d'autorisations différentes qui sont représentées par des classes d'autorisation différentes. Ce module utilise la classe MapPermission. Pour la liste de toutes les propriétés possibles, voir Programmation d'autorisations client.

Dans WebSphere eXtreme Scale, la classe `com.ibm.websphere.objectgrid.security.MapPermission` représente les autorisations d'accès aux ressources eXtreme Scale , notamment les méthodes des interfaces ObjectMap ou JavaMap. WebSphere eXtreme Scale définit les chaînes de permission suivantes pour accéder aux méthodes des interfaces ObjectMap et JavaMap :

- **read** : accorde l'autorisation de lire les données de la mappe.
- **write** : accorde l'autorisation de mettre à jour les données de la mappe.
- **insert** : accorde l'autorisation d'insérer les données dans la mappe.
- **remove**: accorde l'autorisation de supprimer les données de la mappe.
- **invalidate** : accorde l'autorisation d'invalider les données dans la mappe.
- **all** : accorde toutes les autorisations ci-dessus.

L'autorisation se produit lorsqu'un client eXtreme Scale utilise une API d'accès aux données, telles que ObjectMap ,JavaMap, ou les API EntityManager. L'environnement d'exécution vérifie les autorisations d'exécution de la mappe correspondante lorsque la méthode est appelée. Si les autorisations d'accès requises ne sont pas accordées au client, une exception AccessControlException est générée. Ce module explique comment utiliser l'autorisation JAAS (Java Authentication and Authorization Service) pour accorder des autorisations d'accès à la mappe pour différents utilisateurs.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- activer l'autorisation WebSphere eXtreme Scale ;

- activer les autorisations utilisateur ;
- configure les autorisations de groupe.

Durée

Ce module prend 60 minutes environ.

Prérequis

Vous devez effectuer les modules précédents de ce tutoriel avant de configurer l'authentification.

Leçon 4.1 : Activation de l'autorisation WebSphere eXtreme Scale

Pour activer l'autorisation dans WebSphere eXtreme Scale, vous devez activer la sécurité sur un ObjectGrid spécifique.

Pour activer l'autorisation sur l'ObjectGrid, vous devez affecter à l'attribut **securityEnabled** la valeur true pour cet ObjectGrid spécifique dans le fichier XML. Pour ce tutoriel, vous pouvez utiliser le fichier `XSDeployment_sec.ear` dans le répertoire *samples_home*/`WASSecurity` dont la sécurité est déjà définie dans le fichier `objectGrid.xml` ou vous pouvez modifier le fichier existant `objectGrid.xml` pour activer la sécurité. Cette leçon explique comment modifier le fichier pour activer la sécurité.

1. Extrayez les fichiers dans le fichier `XSDeployment.ear`, puis décompressez le fichier `XSDeploymentWeb.war`.
2. Ouvrez le fichier `objectGrid.xml` et affectez à l'attribut la valeur `true` sur le niveau ObjectGrid. Voir un exemple de cet attribut ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>

<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" securityEnabled="true">
            <backingMap name="Map1" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

   Si vous avez plusieurs ObjectGrids définis, vous devez définir cet attribut dans chaque grille.
3. Remodularisez les fichiers `XSDeploymentWeb.war` et `XSDeployment.ear` pour inclure vos modifications. Nommez le fichier `XSDeployment_sec.ear` pour ne pas remplacer le package d'origine.
4. Désinstallez l'application existante `XSDeployment` et installez le fichier `XSDeployment_sec.ear`. Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 126 pour plus d'informations sur le déploiement des applications.

Point de contrôle de la leçon :

Vous avez activé la sécurité sur l'ObjectGrid, ce qui permet également d'activer l'autorisation dans la grille de données.

Leçon 4.2 : Activation des autorisations basées sur l'utilisateur

Dans le module d'authentification de ce tutoriel, vous avez créé deux utilisateurs : `operator1` et `admin1`. Vous pouvez affecter des droits différents à ces utilisateurs avec l'autorisation JASS (Java Authentication and Authorization Service).

Définition de la règle d'autorisation JAAS (Java Authentication and Authorization Service) en utilisant des principaux d'utilisateur :
Vous pouvez affecter des autorisations aux utilisateurs que vous avez créés. Affectez les autorisations de lecture `operator1` uniquement à toutes les mappes. Affectez à l'utilisateur `admin1` toutes les autorisations. Utilisez le fichier de règle d'autorisation JAAS pour accorder des autorisations aux principaux.

Editez le fichier d'autorisation JAAS. Le fichier `xsAuth2.policy` se trouve dans le répertoire *samples_home*`/security` :

```
grant codebase http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction
Principal com.ibm.ws.security.common.auth.WSPrincipalImpl "defaultWIMFileBasedRealm/operator1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction
Principal com.ibm.ws.security.common.auth.WSPrincipalImpl "defaultWIMFileBasedRealm/admin1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Dans ce fichier, le codebase `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` est une URL réservée à ObjectGrid. Toutes les autorisations ObjectGrid accordées aux principaux doivent utiliser ce codebase spécial. Les autorisations suivantes sont affectées dans ce fichier :

- Le première instruction accorde l'autorisation de mappe `read` au principal `operator1`. L'utilisateur `operator1` dispose uniquement de l'autorisation de lecture sur la mappe Map1 dans l'instance Grid ObjectGrid.
- La deuxième instruction accorde toutes les autorisations de mappe au principal admin1. L'utilisateur `admin1` dispose de toutes les autorisations sur la mappe Map1 dans l'instance Grid ObjectGrid.
- Le principal s'appelle `defaultWIMFileBasedRealm/operator1` et non pas `Operator1`. WebSphere Application Server ajoute automatiquement le nom de domaine au nom du principal lorsque des référentiels fédérés sont utilisés comme registre de comptes utilisateur. Modifiez cette valeur si nécessaire.

Définition du fichier de règle d'autorisation JAAS à l'aide des propriétés JVM :
Procédez comme suit pour définir les propriétés JVM des serveurs xs1 et xs2qui qui se trouvent dans le cluster xsCluster. Si vous utilisez une topologie qui est différente de l'exemple de topologie qui est utilisé dans ce tutoriel, définissez le fichier sur tous vos serveurs de conteneur.

1. Dans la console d'administration, cliquez sur **Serveurs** > **Serveurs d'applications** > *server_name* > **Java et gestion des processus** > **Définition de processus** > **Java Virtual Machine.**
2. Ajoutez les arguments JVM génériques suivants :
   `-Djava.security.policy=`*samples_home*`/security/xsAuth2.policy`
3. Cliquez sur **OK** et enregistrez les modifications.

Exécution de l'exemple d'application pour tester les autorisations :
Vous pouvez utiliser l'exemple d'application pour tester les paramètres d'autorisation. L'administrateur continue d'avoir tous les autorisations dans dans la mappe Map1, y compris l'affichage et l'ajout d'employés. L'opérateur doit pouvoir afficher uniquement les employés, car seule l'autorisation Lecture lui a été affectée.

1. Redémarrez tous les serveurs d'applications qui exécutent des serveurs de conteneur.
2. Ouvrez l'application `EmployeeManagementWeb`. Dans un navigateur Web, ouvrez `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`.
3. Connectez-vous à l'application en tant qu'administrateur. Utilisez le nom d'utilisateur `admin1` et le mot de passe `admin1`.
4. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. Un message indique que l'utilisateur est introuvable.
5. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp1@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Un message indique que l'employé a été ajouté.
6. Connectez-vous en tant qu'opérateur. Ouvrez une seconde fenêtre de navigateur Web et `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`. Utilisez le nom d'utilisateur `operator1` et le mot de passe `operator1`.
7. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. L'employé est affiché.
8. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp2@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Le message suivant s'affiche :

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   L'exception suivante se trouve dans la chaîne d'exception :

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Ce message s'affiche, car l'utilisateur `operator1` n'est pas autorisé à insérer des données dans la mappe Map1.

Si vous utilisez une version WebSphere Application Server antérieure à la version 7.0.0.11, une erreur java.lang.StackOverflowError peut s'afficher sur le serveur de conteneur. Elle est provoquée par l'IBM Developer Kit. Le problème est résolu dans l'IBM Developer Kit fourni avec WebSphere Application Server Version 7.0.0.11 et les versions suivantes.

Point de contrôle de la leçon :

Dans cette leçon, vous avez configuré l'autorisation en attribuant des autorisations à des utilisateurs spécifiques.

Leçon 4.4 : Configuration des autorisations de groupe

Dans la leçon précédente, vous avez affecté des autorisations utilisateur individuelles aux principaux utilisateur dans la règle d'autorisation JAAS (Java Authentication and Authorization CService). Cependant, lorsque vous avez des centaines ou des milliers d'utilisateurs, utilisez l'autorisation de groupe, qui autorise l'accès en fonction des groupes plutôt que des utilisateurs individuels.

Malheureusement, l'objet Subject qui est authentifié depuis de WebSphere Application Server contient uniquement un principal utilisateur. Cet objet ne contient pas de principal. Vous pouvez ajouter un module de connexion personnalisé pour alimenter le principal de groupe dans l'objet Subject.

Pour ce tutoriel, le module de connexion personnalisé s'appelle com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule. Le module se trouve dans le fichier `groupLM.jar`. Placez le fichier JAR dans le répertoire `WAS-INSTALL/lib/ext`.

WASAddGroupLoginModule extrait les données d'identification de groupe public depuis le sujet WebSphere Application Server et crée un principal Groupe, com.ibm.websphere.samples.objectgrid.security.WSGroupPrincipal, pour représenter le groupe. Ce principal de groupe peut ensuite être utilisé pour l'autorisation de groupe. Les groupes sont définis dans le fichier `xsAuthGroup2.policy` :

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
  "defaultWIMFileBasedRealm/cn=operatorGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
 "defaultWIMFileBasedRealm/cn=adminGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Le nom du principal est `WSGroupPrincipal` qui représente le groupe.

Ajout du module de connexion personnalisé :
Le module de connexion personnalisé doit être ajouté à chacune des entrées de module de connexion système suivantes : si vous utilisez l'authentification LTPA (Lightweight Third Party Authentication), ajoutez l'entrée aux modules de connexion RMI_INBOUND. LTPA est le mécanisme d'authentification par défaut pour WebSphere Application Server Version 7.0. Pour une configuration WebSphere Application Server Network Deployment, il suffit de configurer les entrées de configuration du mécanisme d'authentification LTPA.

Procédez comme suit pour configurer le module de connexion com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule :

1. Dans la console d'administration, cliquez sur **Sécurité** > **Sécurité globale** > **Java Authentication and Authorization Service** > **Connexions système** > ***login_module_name*** > **Modules de connexion JAAS** > **Nouveau**.
2. Entrez le nom de classe `com.ibm.websphere.sample.xs.security.lm.WASAddGroupLoginModule`.
3. Facultatif : Ajoutez une propriété `debug` et définissez la valeur `true`.
4. Cliquez sur **Appliquer** pour ajouter le nouveau module à la liste des modules de connexion.

Définition du fichier de règle d'autorisation JAAS à l'aide des propriétés JVM :

Dans la console d'administration, effectuez les étapes suivantes pour les serveurs xs1 et xs2 dans xsCluster. Si une topologie de déploiement différente est utilisée, effectuez les étapes suivantes pour les serveurs d'applications qui hébergent les serveurs de conteneur.

1. Dans la console d'administration, cliquez sur **Serveurs** > **Serveurs d'applications** > ***server_name*** > **Java et gestion des processus** > **Définition de processus** > **Java virtual machine**
2. Entrez les arguments JVM Generic suivants ou remplacez l'entrée `-Djava.security.policy` par le texte suivant :

   `-Djava.security.policy=`*`samples_home`*`/security/xsAuthGroup2.policy`

3. Cliquez sur **OK** et enregistrez les modifications.

Test d'autorisation de groupe avec l'exemple d'application :

Vous pouvez vérifier que l'autorisation de groupe est configurée par le module de connexion avec l'exemple d'application.

1. Redémarrez les serveurs de conteneur. Pour ce tutoriel, les serveurs de conteneur sont les serveurs xs1 et xs2.
2. Connectez-vous à l'exemple d'application. Dans un navigateur Web, ouvrez `http://<host>:<port>/EmployeeManagementWeb/management.jsp` et connectez-vous avec le nom d'utilisateur `admin1` et le mot de passe `admin1`.
3. Affichez un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp2@acme.com`. Un message indique que l'utilisateur est introuvable.
4. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp2@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Un message indique que l'employé a été ajouté.
5. Connectez-vous en tant qu'opérateur. Ouvrez une seconde fenêtre de navigateur Web et l'URL `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`. Utilisez le nom d'utilisateur `operator1` et le mot de passe `operator1`.
6. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp2@acme.com`. L'employé est affiché.
7. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp3@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Le message suivant s'affiche :

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   L'exception suivante se trouve dans la chaîne d'exception :

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Ce message s'affiche, car l'utilisateur operator n'est pas autorisé à insérer des données dans la mappe Map1.

Point de contrôle de la leçon :

Vous avez configuré des groupes pour simplifier l'attribution des droits aux utilisateurs de votre application.

Module 5 : Utilisation de l'utilitaire `xscmd` pour surveiller les grilles de données et les mappes

Vous pouvez utiliser l'utilitaire **`xscmd`** pour afficher les grilles de données principales et les tailles de mappe de la grille de données `Grid`. L'outil **`xscmd`** utilise le bean géré pour interroger tous les artefacts de grille de données, telles que les fragments primaires, les fragments de réplique, les serveurs de conteneur, les tailles de mappe.

Dans ce tutoriel, les serveurs de conteneur et de catalogue s'exécutent sur des serveurs d'applications WebSphere Application Server. L'environnement d'exécution WebSphere eXtreme Scale enregistre les beans gérés (MBean) avec le serveur de bean créé par l'environnement d'exécution WebSphere Application Server. La sécurité qui est utilisée par l'outil **`xscmd`** est fournie par la sécurité

WebSphere Application Server MBean. Par conséquent, une configuration de sécurité WebSphere eXtreme Scale n'est pas nécessaire.

1. A l'aide d'un outil de ligne de commande, ouvrez le répertoire *DMGR_PROFILE*/bin.
2. Exécutez l'outil **xscmd**.

   Utilisez la commande **-c showPlacement -sf P** pour lister le placement des fragments primaires. Linux UNIX

   ```
   xscmd.sh -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Windows

   ```
   xscmd.bat -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Pour pouvoir afficher la sortie, le système vous demande de vous connecter avec votre ID et votre mot de passe WebSphere Application Server.

Point de contrôle de la leçon

Vous avez utilisé l'outil **xscmd** dans WebSphere Application Server.

Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale dans un environnement mixte avec un authentificateur externe

Ce tutoriel explique comment sécuriser les serveurs WebSphere eXtreme Scale partiellement déployés dans un environnement WebSphere Application Server.

Dans le déploiement de ce tutoriel, les serveurs de conteneur sont déployés dans WebSphere Application Server. Le serveur de catalogue est déployé en tant que serveur autonome lancé dans un environnement Java SE (Java Standard Edition).

Comme le serveur de catalogue n'est pas déployé dans WebSphere Application Server, vous ne pouvez pas utiliser les plug-in WebSphere Application Server Authentication. Pour plus d'informations sur le processus de configuration des plug-in WebSphere Application Server Authentication, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 115. Dans ce tutoriel, un authentificateur différent est requis pour l'authentification du serveur de catalogue. Vous pouvez configurer un authentificateur de fichier de clés pour authentifier les clients.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configuration de WebSphere eXtreme Scale pour utiliser le plug-in KeyStoreLoginAuthenticator
- Configuration de la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2 et le fichier de propriétés WebSphere eXtreme Scale
- Utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utilisation de l'utilitaire **xscmd** pour surveiller les grilles de données et les mappes que vous avez créés dans le tutoriel.

Durée

Ce tutoriel dure environ 4 heures.

Introduction : Sécurité dans un environnement mixte

Dans ce tutoriel, vous intégrez la sécurité WebSphere eXtreme Scale dans un environnement mixte. Les serveurs de conteneur s'exécutent dans WebSphere Application Server et le service de catalogue s'exécute en mode autonome. Etant donné que le serveur de catalogue est en mode autonome, vous devez configurer un authentificateur externe.

Important : Si le serveur de conteneur et le serveur de catalogue fonctionnent dans WebSphere Application Server, vous pouvez utiliser des plug-in WebSphere Application Server Authentication ou un authentificateur externe. Pour plus d'informations sur l'utilisation des plug-in WebSphere Application Server Authentication, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 115.

Objectifs d'apprentissage

Les objectifs d'apprentissage de ce tutoriel sont les suivants :

- Configuration de WebSphere eXtreme Scale pour utiliser le plug-in KeyStoreLoginAuthenticator
- Configuration de la sécurité du transport WebSphere eXtreme Scale pour utiliser la configuration WebSphere Application Server CSIv2 et le fichier de propriétés WebSphere eXtreme Scale
- Utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server
- Utilisation de l'utilitaire **xscmd** pour surveiller les grilles de données et les mappes que vous avez créés dans le tutoriel.

Durée

Ce tutoriel prend 4 heures environ.

Niveau de compétence

Intermédiaire

Audience

Les développeurs et les administrateurs qui veulent intégrer la sécurité entre WebSphere eXtreme Scale et WebSphere Application Server et configurer des authentificateurs externes.

Configuration requise

- WebSphere Application Server Version 7.0.0.11 ou version suivante avec les correctifs suivants appliqués : correctif temporaire PM20613 et correctif temporaire PM15818.
- Le serveur de catalogue doit être en cours d'exécution sur une installation autonome et non pas une installation qui est intégrée à WebSphere Application Server.
- Mettez à jour l'environnement d'exécution Java pour appliquer le correctif IZ79819: IBMJDK FAILS TO READ PRINCIPAL STATEMENT WITH WHITESPACE FROM SECURITY FILE
- Le noeud autonome qui exécute le service de catalogue doit utiliser IBM Software Development Kit Version 1.6 J9. Ce kit de développement de logiciels

est inclus dans l'installation de WebSphere Application Server. Le noeud de serveur de catalogue doit être une installation autonome, car vous ne pouvez pas exécuter la commande **startOgServer** dans une installation de WebSphere eXtreme Scale sur WebSphere Application Server.

Ce tutoriel utilise dans l'exemple quatre serveurs d'applications WebSphere Application Server et un gestionnaire de déploiement.

Prérequis

Une connaissance de base des éléments suivants est utile avant de démarrer ce tutoriel :

- Modèle de programmation WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere eXtreme Scale
- Concepts de sécurité de base WebSphere Application Server

Pour plus d'informations sur l'intégration de la sécurité WebSphere eXtreme Scale et WebSphere Application Server, voir «Intégration de la sécurité dans WebSphere Application Server», à la page 643.

Module 1 : Préparation de l'environnement WebSphere Application Server et autonome

Avant de commencer le tutoriel, vous devez créer une topologie de base qui inclut des serveurs de conteneur qui s'exécutent dans WebSphere Application Server. Dans ce tutoriel, les serveurs de catalogue s'exécutent en mode autonome.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous :

- comprendrez la topologie mixte et les fichiers qui sont nécessaires pour le tutoriel ;
- saurez configurer WebSphere Application Server pour exécuter les serveurs de conteneur.

Durée

Ce module prend 60 minutes environ.

Leçon 1.1 : Compréhension de la topologie et obtention des fichiers du tutoriel

Pour préparer votre environnement pour le tutoriel, vous devez configurer les serveurs de catalogue et de conteneur de la topologie.

Cette leçon vous guide dans l'exemple de topologie et les applications qui sont utilisés dans le tutoriel. Pour commencer à exécuter le tutoriel, vous devez télécharger les applications et placer les fichiers de configuration dans les emplacements propres à votre environnement. Vous pouvez télécharger l'exemple d'application depuis le wiki WebSphere eXtreme Scale.

Topologie : Dans ce tutoriel, vous créez les clusters suivants dans la cellule WebSphere Application Server :

- **Cluster appCluster** : héberge l'exemple d'application d'entreprise EmployeeManagement. Ce cluster contient les deux serveurs d'applications s1 et s2.

- **Clster xsCluster** : héberge les serveurs de conteneur eXtreme Scale. Ce cluster contient les deux serveurs d'applications xs1 et xs2.

Dans cette topologie de déploiement, les serveurs d'applications s1 et s2 sont les serveurs client qui accèdent aux données qui sont stockées dans la grille de données. Les serveurs xs1 et xs2 sont les serveurs de conteneur qui hébergent la grille de données.

Autre configuration : vous pouvez héberger tous les serveurs d'applications dans un seul cluster, tel que appCluster. Avec cette configuration, tous les serveurs du cluster sont les clients et les serveurs de conteneur. Ce tutoriel utilise deux clusters pour distinguer les serveurs d'applications qui hébergent les clients et les serveurs de conteneur.

Dans ce tutoriel, vous configurez un domaine de service de catalogue qui se compose d'un serveur distant qui ne se trouve pas dans la cellule WebSphere Application Server. Cette configuration n'étant pas la valeur par défaut, les serveurs de catalogue sont exécutés dans le gestionnaire de déploiement et les autres processus le sont dans la cellule WebSphere Application Server. Voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307 pour plus d'informations sur la création d'un domaine de service de catalogue constitué de serveurs distants.

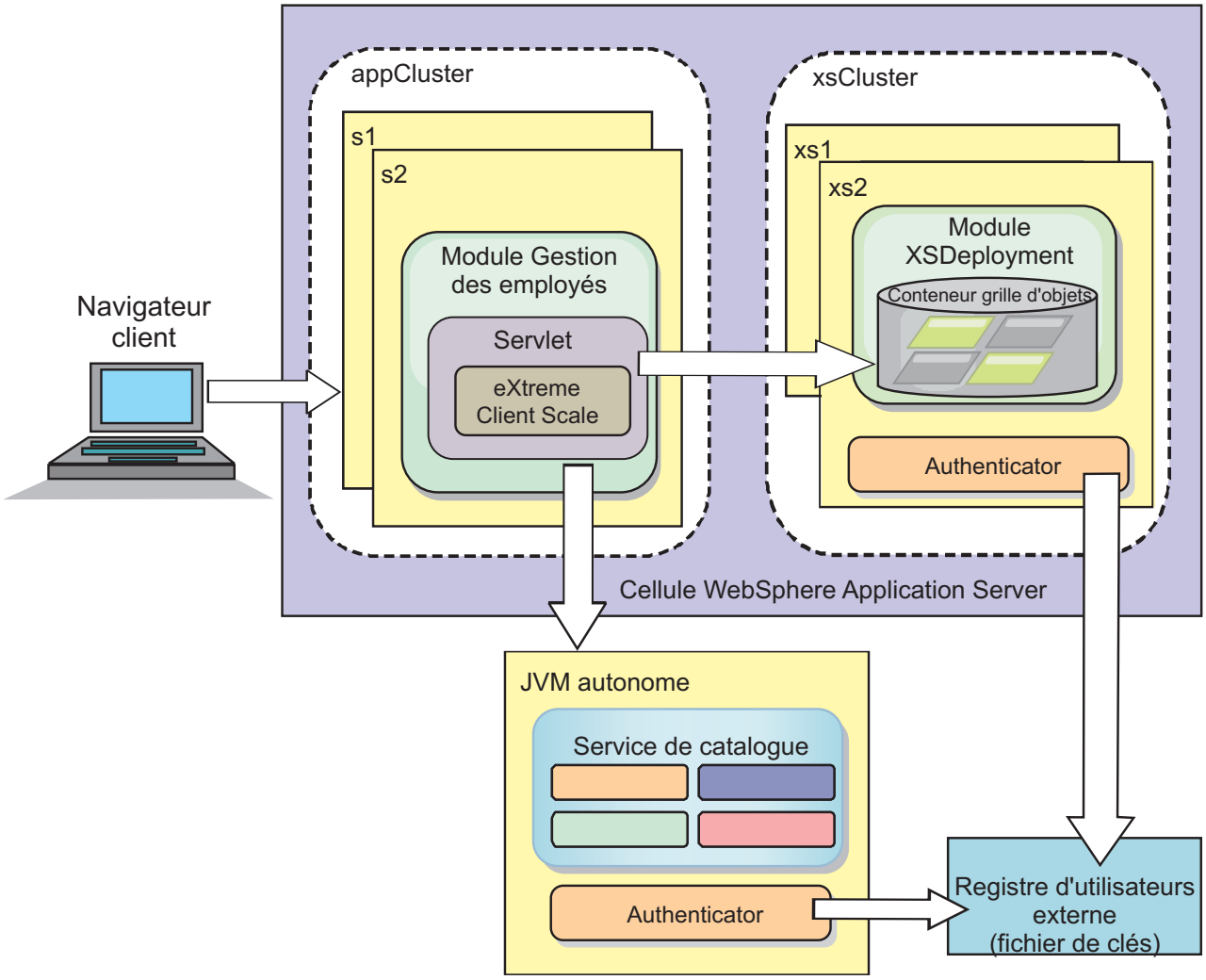


Figure 24. Topologie du tutoriel

Applications : Dans ce tutoriel, vous utilisez deux applications et un fichier de bibliothèque partagée :

- **`EmployeeManagement.ear`** : l'application `EmployeeManagement.ear` est une application d'entreprise simplifiée Java 2 Platform, Enterprise Edition (J2EE). Elle contient un module Web pour gérer les profils des employés. Le module Web contient le fichier `management.jsp` pour afficher, insérer, mettre à jour et supprimer les profils d'employés qui sont stockés dans les serveurs de conteneur.
- **`XSDeployment.ear`** : cette application contient un module d'application d'entreprise, sans artefacts d'application. Les objets cache sont regroupés dans le fichier `EmployeeData.jar`. Le fichier `EmployeeData.jar` est déployé comme bibliothèque partagée pour le fichier `XSDeployment.ear` pour que le fichier `XSDeployment.ear` puisse accéder aux classes. Le but de cette application consiste à modulariser le fichier de configuration eXtreme Scale et le fichier de propriétés. Lorsque cette application d'entreprise est démarrée, les fichiers de configuration eXtreme Scale sont automatiquement détectés par l'environnement d'exécution eXtreme Scale pour créer les serveurs de conteneur. Ces fichiers de configuration incluent les fichiers `objectGrid.xml` et `objectGridDeployment.xml`.
- **`EmployeeData.jar`** : ce fichier jar contient une classe, com.ibm.websphere.sample.xs.data.EmployeeData. Cette classe représente les

données de l'employé qui est stocké dans la grille. Ce fichier d'archive Java (JAR) est déployé avec les fichiers `EmployeeManagement.ear` et `XSDeployment.ear` comme bibliothèque partagée.

Obtention des fichiers du tutoriel :

1. Téléchargez les fichiers `WASSecurity.zip` et `security_extauth.zip` depuis le wiki WebSphere eXtreme Scale.
2. Extrayez le ficher `WASSecurity.zip` dans un répertoire pour afficher les données binaires et les artefacts source (par exemple un répertoire `wxs_samples/`). Ce répertoire est *samples_home* pour le reste du tutoriel. Consultez le fichier `README.txt` dans le package pour la description de son contenu et savoir comment charger le code source dans votre espace de travail Eclipse. Les fichiers de configuration ObjectGrid suivants se trouvent dans le répertoire `META-INF` :
   - `objectGrid.xml`
   - `objectGridDeployment.xml`
3. Créez un répertoire pour stocker les fichiers de propriétés qui sont utilisés pour sécuriser cet environnement. Par exemple, vous pouvez créer le répertoire `/opt/wxs/security`.
4. Extrayez le fichier `security_extauth.zip` vers *samples_home*. Le fichier `security_extauth.zip` contient les fichiers de configuration de sécurité suivants qui sont utilisés dans ce tutoriel : Fichiers de configuration :
   - `catServer3.props`
   - `server3.props`
   - `client3.props`
   - `security3.xml`
   - `xsAuth3.props`
   - `xsjaas3.config`
   - `sampleKS3.jks`

A propos des fichiers de configuration :

Les fichiers `objectGrid.xml` et `objectGridDeployment.xml` créent les grilles de données et les mappes qui stockent les données d'application.

Ces fichiers de configuration doivent s'appeler `objectGrid.xml` et `objectGridDeployment.xml`. Lorsque le serveur d'applications démarre, eXtreme Scale détecte ces fichiers dans le répertoire `META-INF` de l'EJB et des modules Web. Si ces fichiers sont trouvés, il suppose que la machine JVM (Java virtual machine) fait office de serveur de conteneur pour les grilles de données définie dans les fichiers de configuration.

Fichier `objectGrid.xml`

Le fichier `objectGrid.xml` défini un ObjectGrid nommé `Grid`. La grille de données `Grid` a une mappe, la mappe `Map1`, qui stocke le profil d'employé pour l'application.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
        <objectGrid name="Grid" txTimeout="15">
            <backingMap name="Map1" />
```

```
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

Fichier `objectGridDeployment.xml`

Le fichier `objectGridDeployment.xml` indique comment déployer la grille de données `Grid`. Lorsque la grille est déployée, elle dispose de cinq partitions et d'une réplique synchrone.

```
<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="1" >
            <map ref="Map1"/>
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez découvert la topologie du tutoriel et ajouté les fichiers de configuration et les exemples d'applications à votre environnement.

Leçon 1.2 : Configuration de l'environnement WebSphere Application Server

Pour préparer votre environnement pour le tutoriel, vous devez configurer la sécurité WebSphere Application Server. Activez la sécurité d'administration et d'application en utilisant des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur. Ensuite, vous pouvez créer des clusters de serveurs pour héberger les serveurs d'applications client et les serveurs de conteneur. Vous devez créer et démarrer le serveurs de catalogue également.

Les étapes suivantes s'appliquent à WebSphere Application Server Version 7.0. Toutefois, vous pouvez appliquez les concepts aux versions antérieures de WebSphere Application Server.

Configuration de la sécurité WebSphere Application Server :
Créez et étendez des profils pour le gestionnaire de déploiement et les noeuds avec WebSphere eXtreme Scale. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226.

Configurez la sécurité WebSphere Application Server.

1. Dans la console d'administration WebSphere Application Server, cliquez sur **Sécurité** > **Sécurité globale**.
2. Sélectionnez **Référentiels fédérés** comme **définition de domaine disponible**. Cliquez sur **Définir comme actif**.
3. Cliquez sur **Configurer** pour accéder au panneau des référentiels fédérés.
4. Entrez le **nom de l'administrateur principal**, tel que `admin`. Cliquez sur **Appliquer.**
5. Lorsque vous y êtes invité, entrez le mot de passe de l'administrateur et cliquez sur **OK**. Sauvegardez vos modifications.

6. Dans la page **Sécurité globale**, vérifiez que le paramètre **Référentiels fédérés** est affecté du registre de comptes utilisateur en cours.
7. Sélectionnez **Activer la sécurité administrative**, **Activer la sécurité de l'application** et **Utiliser la sécurité Java 2 pour limiter l'accès de l'application aux ressources locales**. Cliquez sur **Appliquer** et enregistrez les modifications.
8. Redémarrez le gestionnaire de déploiement et les serveurs d'applications actifs.

La sécurité administrative WebSphere Application Server est activée à l'aide des référentiels fédérés basés sur un fichier interne comme registre de comptes utilisateur.

Création de clusters de serveurs :

Créez deux clusters de serveurs dans votre configuration WebSphere Application Server : `appCluster` pour héberger l'exemple d'application du tutoriel et `xsCluster` pour héberger la grille de données.

1. Dans la console d'administration WebSphere Application Server, ouvrez le panneau des clusters. Cliquez sur **Serveurs** > **Clusters** > **Clusters de serveurs d'applications WebSphere** > **Nouveau**.
2. Entrez `appCluster` comme nom de cluster, ne renseignez pas l'option **Environnement local préféré** et cliquez sur **Suivant**.
3. Créez des serveurs dans le cluster. Créez le serveur `s1` en conservant les options par défaut. Ajoutez le membre `s2` au cluster.
4. Exécutez les étapes restantes dans l'assistant pour créer le cluster. Sauvegardez les modifications.
5. Répétez ces étapes pour créer le cluster `xsCluster`. Ce cluster contient les deux serveurs `xs1` et `xs2`.

Créer un domaine de service de catalogue :

Après avoir configuré le cluster de serveurs et la sécurité, vous devez indiquer où les serveurs de catalogue démarrent.

Définissez un domaine de service de catalogue dans WebSphere eXtreme Scale

1. Dans la console d'administration de WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de service de catalogue**.
2. Créez le domaine de service de catalogue. Cliquez sur **Nouveau**. Créez le domaine de service de catalogue avec le nom `catalogService1` et activez le domaine de service de catalogue comme valeur par défaut.
3. Ajoutez les serveurs distants au domaine du service de catalogue. Sélectionnez **Serveur distant**. Indiquez le nom d'hôte sur lequel le serveur de catalogue est en cours d'exécution. Utilisez la valeur de port d'écoute de `16809` pour cet exemple.
4. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez activé la sécurité dans WebSphere Application Server et créé une topologie de serveur pour WebSphere eXtreme Scale.

Module 2 : Configuration de l'authentification WebSphere eXtreme Scale dans un environnement mixte

En configurant l'authentification, vous pouvez déterminer l'identité du demandeur. WebSphere eXtreme Scale prend en charge l'authentification client-serveur et serveur-serveur.

Flux d'authentification

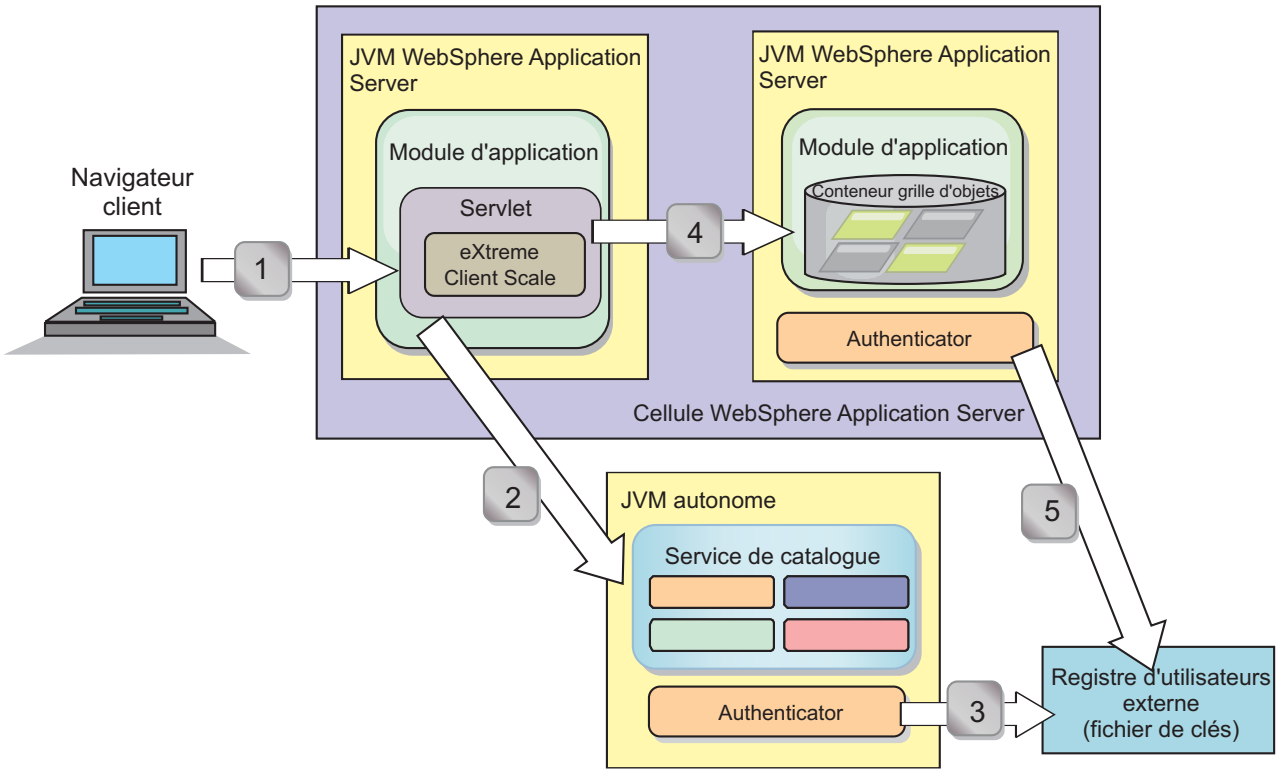


Figure 25. Flux d'authentification

Le diagramme suivant montre deux serveurs d'applications. Le premier serveur d'applications héberge l'application Web, qui est également un client WebSphere eXtreme Scale. Le second serveur d'applications héberge un serveur de conteneur. Le serveur de catalogue s'exécute dans une machine JVM (Java Virtual Machine) et non pas WebSphere Application Server.

Les flèches numérotées dans le diagramme indiquent le flux d'authentification :

1. Un utilisateur d'application d'entreprise accède au navigateur Web et se connecte au premier serveur d'applications avec un nom d'utilisateur et un mot de passe. Le premier serveur d'applications envoie le nom d'utilisateur et le mot de passe du client à l'infrastructure de sécurité pour s'authentifier auprès du registre des utilisateurs. Le registre d'utilisateurs est un fichier de clés. En conséquence, les informations de sécurité sont stockées sur l'unité d'exécution WebSphere Application Server.
2. Le fichier JSP (JavaServer Pages) fait office de client WebSphere eXtreme Scale pour extraire les informations de sécurité à partir du fichier des propriétés du client. L'application JSP qui fait office de client WebSphere eXtreme Scale envoie les données d'identification de sécurité du client WebSphere eXtreme Scale avec la demande au serveur de catalogue. L'envoi des données d'identification de sécurité avec la demande est un modèle *runAs*. Dans ce

modèle, le client du navigateur Web s'exécute en tant que client WebSphere eXtreme Scale pour accéder aux données stockées dans le serveur de conteneur. Le client utilise les données d'identification de client JVM (Java virtual machine) pour se connecter aux serveurs WebSphere eXtreme Scale. Le modèle runAs revient à se connecter à une base de données avec un ID utilisateur et un mot de passe au niveau de la source de données.

3. Le serveur de catalogue reçoit les données d'identification du client WebSphere eXtreme Scale, qui contiennent les jetons de sécurité WebSphere Application Server. Ensuite, le serveur de catalogue appelle le plug-in Authentificateur pour authentifier les données d'identification du client. L'authentificateur se connecte au registre d'utilisateurs externe et envoie les données d'identification du client vers le registre d'utilisateurs pour les authentifier.
4. Le client envoie l'ID utilisateur et le mot de passe au serveur de conteneur qui est hébergé sur le serveur d'applications.
5. Le service de conteneur, hébergé sur le serveur d'applications, reçoit les données d'identification du client WebSphere eXtreme Scale, à savoir la paire ID utilisateur-mot de passe. Ensuite, le serveur de catalogue appelle le plug-in Authentificator pour authentifier les données d'identification du client. L'authentificateur se connecte au registre d'utilisateurs du fichier de clés et envoie les données d'identification du client au registre d'utilisateurs pour les authentifier

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer la sécurité du client WebSphere eXtreme Scale ;
- configurer la sécurité du serveur de catalogue WebSphere eXtreme Scale ;
- configurer la sécurité du serveur de conteneur WebSphere eXtreme Scale ;
- installer et exécuter l'exemple d'application.

Durée

Ce module prend 60 minutes environ.

Leçon 2.1 : Configuration de la sécurité du client WebSphere eXtreme Scale

Vous définissez les propriétés du client avec un fichier de propriétés. Ce fichier indique la classe d'implémentation CredentialGenerator à utiliser.

Contenu du fichier des propriétés du client :

Ce tutoriel utilise des jetons de sécurité WebSphere Application Server pour les données d'identification du client. Le répertoire *samples_home*/`security_extauth` contient le fichier `client3.props`.

Le fichier `client3.props` contient les paramètres suivants :

securityEnabled
Active la sécurité du client WebSphere eXtreme Scale. La valeur est `true` pour indiquer que le client doit envoyer les informations de sécurité disponibles au serveur.

credentialAuthentication
Spécifie la prise en charge de l'authentification des données d'identification

du client. La valeur est `Supported` pour indiquer que le client prend en charge l'authentification des données d'identification du client.

credentialGeneratorClass
Indique le nom de la classe qui implémente l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. La valeur correspond à la classe com.ibm.websphere.objectgrid.security.plugins.builtins. UserPasswordCredentialGenerator pour que le client extrait les informations de sécurité de la classe UserPasswordCredentialGenerator.

credentialGeneratorProps
Indique le nom et le mot de passe : `manager manager1`. Le nom d'utilisateur est `manager` et le mot de passe est `manager1`. Vous pouvez également utiliser la commande **FilePasswordEncoder.bat|sh** pour coder cette propriété à l'aide d'un algorithme or exclusif (xor).

Définition du fichier des propriétés du client en utilisant des propriétés JVM (Java virtual machine) :
Dans la console d'administration, procédez comme suit pour les serveurs `s1` et `s2` dans le cluster `appCluster`. Si vous utilisez une topologie différente, procédez comme suit pour tous les serveurs d'applications sur lesquels l'application `EmployeeManagement` est déployée.

1. **Serveurs** > **Serveurs d'applications WebSphere** > *server_name* > **Java et gestion de processus** > **Définition de processus** > **Java Virtual Machine**.
2. Créez la propriété JVM générique suivante pour définir l'emplacement du fichier de propriétés du client :
   `-Dobjectgrid.client.props=`*samples_home*`/security_extauth/client3.props`
3. Cliquez sur **OK** et enregistrez les modifications.

Point de contrôle de la leçon :

Vous avez édité le fichier de propriétés du client et configuré les serveurs dans le cluster `appCluster` pour utiliser le fichier de propriétés du client. Le fichier des propriétés indique la classe d'implémentation CredentialGenerator à utiliser.

Leçon 2.2 : Configuration de la sécurité du serveur de catalogue

Un serveur de catalogue contient deux niveaux d'informations de sécurité : le premier niveau contient les propriétés de sécurité communes à tous les serveurs WebSphere eXtreme Scale, y compris les serveurs de service de catalogue et de conteneur. Le deuxième niveau contient les propriétés de sécurité qui sont spécifiques du serveur de catalogue.

Les propriétés de sécurité qui sont communes au serveur de catalogue et au serveur de conteneur sont configurées dans le fichier XML du descripteur de sécurité. La configuration de l'authentificateur, qui représente le registre d'utilisateurs et le mécanisme d'authentification, est un exemple des propriétés communes. Voir Fichier XML du descripteur de sécurité pour plus d'informations sur les propriétés de sécurité.

Pour configurer le fichier descripteur XML de sécurité dans un environnement Java SE, utilisez une option **-clusterSecurityFile** lorsque vous exécutez la commande **startOgServer** ou **startXsServer**. Indiquez une valeur dans un format de fichier, tel que *samples_home*`/security_extauth/security3.xml`.

Fichier `security3.xml` :

Dans ce tutoriel, le fichier `security3.xml` se trouve dans le répertoire *samples_home*/`security_extauth`. Contenu du fichier `security3.xml` avec les commentaires supprimés :

```
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true">
  <authenticator
className="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
  </authenticator>
 </security>
</securityConfig>
```

Les propriétés suivantes sont définies dans le fichier `security3.xml` :

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer au serveur de catalogue que la sécurité globale WebSphere eXtreme Scale est activée.

authenticator
: L'authentificateur est configuré com.ibm.websphere.objectgrid.security.plugins.builtins. KeyStoreLoginAuthenticator class. Avec cette implémentation intégrée dans le plug-in Authenticator, l'ID utilisateur et le mot de passe sont transmis pour vérifier qu'il est configuré dans le fichier de clés. La classe KeyStoreLoginAuthenticator utilise un alias de module de connexion `KeyStoreLogin` et une configuration de connexion Java Authentication and Authorization Service (JAAS) est donc nécessaire.

`catServer3.props` file :

Le fichier de propriétés du serveur stocke les propriétés du serveur qui incluent ses propriétés de sécurité. Pour plus d'informations, voir Fichier de propriétés du serveur. Vous pouvez utiliser l'option **`-serverProps`** pour spécifier la propriété de serveur de catalogue lorsque vous exécutez la commande **`startOgServer`** ou **`startXsServer`**. Pour ce tutoriel, un fichier `catServer3.props` se trouve dans le répertoire c. Contenu du fichier `catServer3.props` avec les commentaires supprimés :

```
securityEnabled=true
credentialAuthentication=Required
transportType=TCP/IP
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
: La propriété securityEnabled a la valeur `true` pour indiquer que ce serveur de catalogue est un serveur sécurisé.

credentialAuthentication
: La propriété credentialAuthentication a la valeur `Requis`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification. Dans le fichier de propriétés du client, la propriété credentialAuthentication a la valeur `Pris en charge` et le serveur reçoit donc les données d'identification envoyées par le client.

secureTokenManagerType
: La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors du regroupement avec les serveurs existants.

authenticationSecret

La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

transportType

La propriété transportType a la valeur `TCP/IP` initialement. Plus loin dans le tutoriel, la sécurité du transport est activée.

Fichier `xsjaas3.config` :

Etant donné que l'implémentation KeyStoreLoginAuthenticator utilise un module de connexion, vous devez configurer le modèle de connexion avec un fichier de configuration de connexion d'authentification JAAS. Contenu du fichier `xsjaas3.config` :

```
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
     keyStoreFile="samples_home/security_extauth/sampleKS3.jks" debug = true;
};
```

Si vous avez utilisé un emplacement pour *samples_home* autre que `/wxs_samples/`, vous devez mettre à jour l'emplacement de keyStoreFile. Cette configuration de connexion indique que le module com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule est utilisé en tant que module de connexion. Le fichier de clés est le fichier `sampleKS3.jks`.

L'exemple de fichier de clés `sampleKS3.jks` stocke deux ID utilisateur et les mots de passe : manager/manager1 et cashier/cashier1.

Vous pouvez utiliser les commandes suivantes **`keytool`** pour créer ce fichier de clés :

- `keytool -genkey -v -keystore ./sampleKS3.jks -storepass sampleKS1 -alias manager -keypass manager1 -dname CN=manager,O=acme,OU=OGSample -validity 10000`
- `keytool -genkey -v -keystore ./sampleKS3.jks -storepass sampleKS1 -alias operator -keypass operator1 -dname CN=operator,O=acme,OU=OGSample -validity 10000`

Démarrage du serveur de catalogue avec la sécurité activée :

Pour démarrer le serveur de catalogue, exécutez la commande **`startOgServer`** ou **`startXsServer`** avec les paramètres **`-clusterFile`** et **`-serverProps`** pour transmettre les propriétés de sécurité.

Utilisez une installation autonome de WebSphere eXtreme Scale pour exécuter le serveur de catalogue. Lors de l'utilisation de l'image d'installation autonome, vous devez utiliser le SDK IBM. Vous pouvez utiliser le logiciel SDK qui est inclus avec WebSphere Application Server en définissant la variable *JAVA_HOME* pour qu'elle pointe vers le SDK IBM. Par exemple, `set JAVA_HOME=racine_was/IBM/WebSphere/AppServer/java/`

1. Accédez au répertoire `bin`.

   ```
   cd rép_base_wxs/bin
   ```

2. Exécutez la commande **startOgServer** ou **startXsServer**.

Linux UNIX

```
./startOgServer.sh cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Windows

```
startOgServer.bat cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Linux UNIX 8.6+

```
./startXsServer.sh cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Windows 8.6+

```
startXsServer.bat cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Une fois que vous avez exécuté la commande **startOgServer** ou **startXsServer**, un serveur sécurisé démarre le port d'écoute 16809, le port client 16601, le port homologue 16602 et le port JMX 16099. Si un conflit de port existe, remplacez le numéro de port par un numéro de port non utilisé.

Arrêt d'un serveur de catalogue dont la sécurité est activée :

Vous pouvez utiliser la commande **stopOgServer** ou **stopXsServer** pour arrêter le serveur de catalogue.

1. Accédez au répertoire `bin`.

   ```
   cd rép_base_wxs/bin
   ```

2. Exécutez la commande **stopOgServer** ou **stopXsServer**. Linux UNIX

   ```
   stopOgServer.sh cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   samples_home/security_extauth/client3.props
   ```

   Windows

   ```
   stopOgServer.bat cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   samples_home/security_extauth/client3.props
   ```

   Linux UNIX 8.6+

   ```
   stopXsServer.sh cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   samples_home/security_extauth/client3.props
   ```

   Windows 8.6+

   ```
   stopXsServer.bat cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   samples_home/security_extauth/client3.props
   ```

Point de contrôle de la leçon :

Vous avez configuré la sécurité du serveur de catalogue en associant les fichiers `security3.xml`, `catServer3.props`, `xsjaas3.config` au service de catalogue.

Leçon 2.3 : Configuration de la sécurité du serveur de conteneur

Lorsqu'un serveur de conteneur se connecte au service de catalogue, le serveur de conteneur obtient toutes les configurations de sécurité qui sont configurées dans le fichier ObjectGrid XML de sécurité. Le fichier XML de sécurité ObjectGrid définit la configuration de l'authentificateur, la valeur de temporisation de la session de connexion et d'autres informations de configuration. Un serveur de conteneur dispose également de ses propres propriétés de sécurité dans le fichier de propriétés de serveur.

Configurez le fichier de propriétés du serveur avec la propriété JVM (Java virtual machine)-Dobjectgrid.server.props. Le nom de fichier spécifié pour cette propriété correspond à un chemin absolu, tel que *samples_home*/`security_extauth/server3.props`.

Dans ce tutoriel, les serveurs de conteneur sont hébergés dans les serveurs `xs1` et `xs2` du cluster `xsCluster`.

Fichier `server3.props` :

Le fichier `server3.props` se trouve dans le répertoire *samples_home*/`security_extauth/`. Contenu du fichier `server3.props` :

```
securityEnabled=true
credentialAuthentication=Required
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
La propriété securityEnabled a la valeur `true` pour indiquer que ce serveur de conteneur est un serveur sécurisé.

credentialAuthentication
La propriété credentialAuthentication a la valeur `Required`. Par conséquent, un client qui se connecte au serveur doit fournir des données d'identification. Dans le fichier de propriétés du client, la propriété credentialAuthentication a la valeur `Supported` et le serveur reçoit donc les données d'identification envoyées par le client.

secureTokenManagerType
La propriété secureTokenManagerType a la valeur `none` pour indiquer que la valeur secrète d'authentification n'est pas chiffrée lors de la connexion aux serveurs existants.

authenticationSecret
La propriété authenticationSecret a la valeur `ObjectGridDefaultSecret`. Cette chaîne secrète est utilisée pour devenir membre du cluster de serveurs eXtreme Scale. Lorsqu'un serveur demande à rejoindre la grille de données, il est invité à présenter la chaîne secrète. Si la chaîne secrète du serveur qui rejoint le cluster correspond à la chaîne dans le serveur de catalogue, le serveur devient membre du cluster. Dans le cas contraire, la demande de jointure est rejetée.

Définition du fichier de propriétés du serveur avec des propriétés JVM :

Définition du fichier de propriétés des serveurs xs1 et xs2. Si vous n'utilisez pas la topologie du tutoriel, définissez le fichier de propriétés de serveur sur tous les serveurs d'applications que vous utilisez pour héberger les serveurs de conteneur.

1. Ouvrez la page de la machine virtuelle Java du serveur. **Serveurs** > **Serveurs d'applications WebSphere** > *server_name* > **Java et gestion des processus** > **Définition de processus** > **Java Virtual Machine**.
2. Ajoutez l'argument JVM générique :
   `-Dobjectgrid.server.props=samples_home/security_extauth/server3.props`
3. Cliquez sur **OK** et enregistrez les modifications.

Ajout du module de connexion personnalisé :

Le serveur de conteneur utilise la même implémentation KeyStoreAuthenticator que le serveur de catalogue. L'implémentation KeyStoreAuthenticator utilise un alias de module de connexion **KeyStoreLogin** pour que vous puissiez ajouter un module de connexion aux entrées de modèle de connexion des applications.

1. Dans la console d'administration WebSphere Application Server, cliquez sur **Sécurité** > **Sécurité globale** > **Java Authentication and Authorization Service**.
2. Cliquez sur **Connexions des applications**.
3. Cliquez sur **Nouveau**, ajoutez un alias `KeyStoreLogin`. Cliquez sur **Appliquer**.
4. Sous **Module de connexion JAAS**, cliquez sur **Nouveau**.
5. Entrez `com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule` comme nom de classe de module et choisissez **SUFFICIENT** comme stratégie d'authentification. Cliquez sur **Appliquer**.
6. Ajoutez la propriété personnalisée `keyStoreFile` avec la valeur *samples_home*`/security_extauth/sampleKS.jks`.
7. Facultatif : Ajoutez la propriété personnalisée `debug` avec la valeur `true`.
8. Enregistrez le fichier.

Point de contrôle de la leçon :

A présent, le serveur d'authentification est sécurisé WebSphere eXtreme Scale. En configurant cette sécurité, toutes les applications qui tentent de se connecter aux serveurs WebSphere eXtreme Scale doivent fournir des données d'identification. Dans ce tutoriel, KeyStoreLoginAuthenticator est l'authentificateur. En conséquence, le client doit fournir un nom d'utilisateur et un mot de passe.

Leçon 2.4 : Installation et exécution de l'exemple

Une fois l'authentification configurée, vous pouvez installer et exécuter l'exemple d'application.

Création d'une bibliothèque partagée pour le fichier `EmployeeData.jar` :

1. Dans la console d'administration de WebSphere Application Server, ouvrez la page **Bibliothèques partagées**. Cliquez sur **Environnement** > **Bibliothèques partagées**.
2. Choisissez la portée **cellule**.
3. Créez la bibliothèque partagée. Cliquez sur **Nouveau**. Entrez `EmployeeManagementLIB` pour le **nom**. Entrez le chemin d'accès au fichier `EmployeeData.jar` dans le chemin de classes, par exemple, *samples_home*`/WASSecurity/EmployeeData.jar`.
4. Cliquez sur **Appliquer**.

Installation de l'exemple :

1. Installez le fichier `EmployeeManagement_extauth.ear` sous le répertoire *samples_home*`/security_extauth`.

Important : Le fichier `EmployeeManagement_extauth.ear` est différent du fichier *samples_home*`/WASSecurity/EmployeeManagement.ear`. La manière dont la session ObjectGrid est extraite a été mise à jour pour utiliser les données d'identification mises en cache dans le fichier des propriétés du client dans l'application `EmployeeManagement_extauth.ear`. Voir les commentaires dans la classe com.ibm.websphere.sample.xs.DataAccessor du projet *samples_home*`/WASSecurity/EmployeeManagementWeb` pour identifier le code mis à jour pour cette modification.

a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé pour l'installation de l'application.
b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster `appCluster` pour installer le module EmployeeManagementWeb module.
c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module `EmployeeManagementWeb`.
d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque `EmployeeManagementLIB`.
e. Associez le rôle `webUser` à **Tous authentifiés dans le domaine de l'application**.
f. Cliquez sur **OK**.

Les clients s'exécutent dans les serveurs `s1` et `s2` du cluster.

2. Installez l'exemple de fichier `XSDeployment.ear` qui se trouve dans le répertoire *samples_home*`/WASSecurity`.
   a. Pour commencer l'installation, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Choisissez le chemin détaillé pour l'installation de l'application.
   b. Dans l'étape d'**association des modules aux serveurs**, définissez le cluster `xsCluster` pour installer le module Web `XSDeploymentWeb`.
   c. Dans l'étape d'**association des bibliothèques partagées**, sélectionnez le module `XSDeploymentWeb`.
   d. Cliquez sur **Bibliothèques partagées de référence**. Sélectionnez la bibliothèque `EmployeeManagementLIB`.
   e. Cliquez sur **OK**.

   Les serveurs `xs1` et `xs2` de ce cluster hébergent les serveurs de conteneur.
3. Vérifiez que le serveur de catalogue est démarré. Pour plus d'informations sur le démarrage d'un serveur de catalogue pour ce tutoriel, voir «Démarrage du serveur de catalogue avec la sécurité activée», à la page 149.
4. Redémarrez le cluster xsCluster. Lorsque xsCluster démarre, l'application `XSDeployment` démarre et un serveur de conteneur est démarré sur les serveurs xs1 et xs2 respectivement. Si vous examinez le fichier `SystemOut.log` des serveurs xs1 et xs2, le message suivant indique que le fichier des propriétés du serveur est chargé :

   ```
   CWOBJ0913I: Server property files have been loaded:
   samples_home/security_extauth/server3.props.
   ```
5. Redémarrez le cluster appClusters. Lorsque appCluster démarre, l'application EmployeeManagement démarre également. Si vous examinez le fichier `SystemOut.log` des serveurs s1 et s2, le message suivant indique que le fichier des propriétés du client est chargé.

   ```
   CWOBJ0924I: The client property file {0} has been loaded.
   ```

Si vous utilisez WebSphere eXtreme Scale Version 7.0, le message CWOBJ9000I s'affiche en anglais et indique que le fichier des propriétés du client a été chargé. Si vous ne voyez pas le message attendu, vérifiez que vous avez configuré la propriété -Dobjectgrid.server.props ou -Dobjectgrid.client.props dans l'argument JVM. Si vous l'avez configurée, vérifiez que le tiret (-) est un caractère UTF.

Exécution de l'exemple d'application :

1. Exécutez le fichier `management.jsp`. Dans un navigateur Web, accédez à `http://<your_servername>:<port>/EmployeeManagementWeb/management.jsp`. Par exemple, vous pouvez utiliser l'URL `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Fournissez les informations d'authentification à l'application. Entrez les données d'identification de l'utilisateur que vous avez associé au rôle webUser. Par défaut, ce rôle utilisateur est associé à tous les utilisateurs authentifiés. Entrez un nom d'utilisateur et un mot de passe valides, tels que le nom d'utilisateur et le mot de passe d'administrateur. Une page pour afficher, ajouter, mettre à jour et supprimer des employés apparaît.
3. Affichez les employés. Cliquez sur **Afficher un employé**. Entrez `emp1@acme.com` comme adresse électronique et cliquez sur **Soumettre**. Un message indique que l'utilisateur est introuvable.
4. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Entrez `emp1@acme.com` comme adresse électronique, `Joe` comme prénom et `Doe` comme nom. Cliquez sur **Soumettre**. Un message s'affiche pour indiquer qu'un employé avec l'adresse `emp1@acme.com` a été ajouté.
5. Affichez le nouvel employé. Cliquez sur **Afficher un employé**. Entrez `emp1@acme.com` comme adresse électronique avec des zones vides pour les nom et prénom, et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été trouvé et que les noms corrects figurent dans les zones du prénom et du nom.
6. Supprimez l'employé. Cliquez sur **Supprimer un employé**. Entrez `emp1@acme.com` et cliquez sur **Soumettre**. Un message s'affiche pour indiquer que l'employé a été supprimé.

Etant donné que le type de transport du serveur de catalogue est `TCP/IP`, vérifiez que le paramètre de transport sortant des serveurs s1 et s2 n'a pas la valeur `SSL requis`. Sinon, une exception se produit. Si vous examinez le fichier de sortie système du serveur de catalogue, `logs/cs1/SystemOut.log`, la sortie de débogage suivante indique l'authentification de fichier de clés :

```
SystemOut     0 [KeyStoreLoginModule] initialize: Successfully loaded key store
SystemOut     0 [KeyStoreLoginModule] login: entry
SystemOut     0 [KeyStoreLoginModule] login: user entered user name: manager
SystemOut     0   Print out the certificates:
...
```

Point de contrôle de la leçon :

Vous avez installé et exécuté l'exemple d'application.

Module 3 : Configuration de la sécurité du transport

Configuration de la sécurité du transport pour protéger le transfert des données entre les clients et les serveurs dans la configuration.

Dans le module précédent du tutoriel, vous avez activé l'authentification WebSphere eXtreme Scale. Avec l'authentification, une application qui tente de se

connecter au serveur WebSphere eXtreme Scale doit fournir des données d'identification. Par conséquent, aucun client non authentifié ne peut se connecter au serveur WebSphere eXtreme Scale. Les clients doivent être une application authentifiée qui s'exécute dans une cellule WebSphere Application Server.

Avec la configuration jusqu'à ce module, le transfert de données entre les clients dans le cluster appCluster et les serveurs du cluster xsCluster n'est pas chiffré. Cette configuration peut être acceptable si vos clusters WebSphere Application Server sont installés sur les serveurs derrière un pare-feu. Toutefois, dans certains scénarios, le trafic non chiffré n'est pas accepté pour certaines raisons, même si la topologie est protégée par un pare-feu. Par exemple, une politique gouvernementale pourrait imposer de chiffrer le trafic. WebSphere eXtreme Scale prend en charge Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour sécuriser la communication entre les noeuds finals ObjectGrid, qui incluent des serveurs client, des serveurs de conteneur et des serveurs de catalogue.

Dans cet exemple de déploiement, les clients et les servers de conteneur eXtreme Scale s'exécutent tous dans l'environnement WebSphere Application Server. Les propriétés client ou serveur ne sont pas nécessaires pour configurer les paramètres SSL, car la sécurité du transport eXtreme Scale est gérée par les paramètres de transport CSIV2 (Application Server Common Secure Interoperability Protocol Version 2). Les serveurs WebSphere eXtreme Scale utilisent la même instance ORB (Object Request Broker) que les serveurs d'applications où ils sont exécutés. Définissez tous les paramètres SSL des services client et de conteneur dans la configuration WebSphere Application Server en utilisant ces paramètres de transport CSIv2. Vous devez configurer les propriétés SSL dans le fichier des propriétés du serveur de catalogue.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- configurer le transport entrant et sortant CSIv2 ;
- ajouter des propriétés SSL dans le fichier de propriétés du serveur de catalogue ;
- vérifier le fichier des propriétés ORB :
- exécuter l'exemple.

Durée

Ce module prend 60 minutes environ.

Prérequis

Cette étape du tutoriel repose sur les modules précédents. Etudiez les modules précédents du présent tutoriel avant de configurer la sécurité du transport.

Leçon 3.1 : Configuration du transport entrant et sortant CSIv2

Pour configurer Transport Layer Security/Secure Sockets Layer (TLS/SSL) pour le transport du serveur, affectez aux propriétés de transport entrant et sortant Common Secure Interoperability Protocol Version 2 (CSIv2) la valeur `SSL requis` pour tous les serveurs WebSphere Application Server qui hébergent des clients, des serveurs de catalogue et des serveurs de conteneur.

Dans l'exemple de topologie du tutoriel, vous devez définir ces propriétés pour les serveurs d'applications s1, s2, xs1 et xs2. Procédez comme suit pout définir les transports entrant et sortant de tous les serveurs de la configuration.

Définissez les transports entrant et sortant dans la console d'administration. Vérifiez que la sécurité administrative est activée.

- **WebSphere Application Server Version 7.0** : Cliquez sur **Sécurité** > **Sécurité globale** > **Sécurité RMI/IIOP** > **Communications entrants CSIv2**. Dans la couche de transport CSIv2 remplacez le type de transport par **SSL requis**. Répétez cette étape pour configurer les communications sortantes CSIv2.

Vous pouvez utiliser les paramètres de sécurité de noeud final géré de manière centralisée, ou configurer des référentiels SSL. Voir Paramètres de transport entrant CSIV2 (Common Secure Interoperability Version 2) pour plus d'informations.

Leçon 3.2 : Ajout de propriétés SSL au fichier des propriétés du serveur de catalogue

Le serveur de catalogue fonctionne en dehors de WebSphere Application Server. Vous devez donc configurer les propriétés SSL dans le fichier des propriétés du serveur.

Vous configurez les propriétés SSL dans le fichier des propriétés du serveur aussi parce que le serveur de catalogue ne peut pas être géré par les paramètres de transport WebSphere Application Server Common Secure Interoperability Protocol Version 2 (CSIV2). Vous devez donc définir les propriétés SSL (Secure Sockets Layer) dans le fichier des propriétés du serveur de catalogue.

Propriétés SSL dans le fichier `catServer3.props` :

```
alias=default
contextProvider=IBMJSSE2
protocol=SSL
keyStoreType=PKCS12
keyStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/key.p12
keyStorePassword=WebAS
trustStoreType=PKCS12
trustStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/trust.p12
trustStorePassword=WebAS
clientAuthentication=false
```

Le fichier `catServer3.props` utilise le fichier de clés et le fichier de clés certifiées au niveau noeud par défaut WebSphere Application Server. Si vous déployez un environnement de déploiement plus complexe, vous devez choisir le fichier de clés et le fichier de clés certifiées corrects. Dans certains cas, vous devez créer un fichier de clés et un fichier de clés certifiées et importez les clés depuis les fichiers de clés des autres serveurs. Notez que la chaîne `WebAS` est le mot de passe par défaut des fichiers de clés et de clés certifiées WebSphere Application Server. Voir Configuration des certificats autosignés par défaut pour plus d'informations.

Ces entrées sont déjà incluses dans le fichier *samples_home*`/security_extauth/catServer3.props` sous forme de commentaires. Vous pouvez supprimer la mise en commentaire des entrées et effectuer les mises à jour correspondant à l'installation dans les variables *racine_was*, *<deployment_manager_name>*, *<cell_name>*, et *<node_name>*.

Après avoir défini les propriétés SSL, remplacez la valeur de la propriété transportType `TCP/IP` par `SSL-Required`.

Propriétés SSL dans le fichier `client3.props` :

Vous devez également configurer les propriétés SSL dans le fichier `client3.props`, car ce fichier est utilisé lorsque vous arrêtez le serveur de catalogue qui est en cours d'exécution en dehors de WebSphere Application Server.

Ces propriétés n'ont aucune incidence sur les serveurs client qui sont en cours d'exécution dans WebSphere Application Server, car ils utilisent les paramètres de transport WebSphere Application Server Common Security Interoperability Protocol Version 2 (CSIV2). Toutefois, lorsque vous arrêtez le serveur de catalogue, vous devez fournir un fichier de propriétés de client dans la commande **stopOgServer**. Définissez les propriétés suivantes dans le fichier <SAMPLES_HOME>/security_extauth/client3.props pour qu'elles correspondent aux valeurs définies ci-dessus dans le fichier catServer3.props :

```
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/racine_was/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/trust.p12
#trustStorePassword=WebAS
```

Comme avec le fichier `catServer3.props`, vous pouvez utiliser les commentaires déjà fournis dans le fichier *samples_home*`/security_extauth/client3.props` avec les mises à jour correspondant aux variables *racine_was*, <*deployment_manager_name*>, <*cell_name*> et <*node_name*> pour qu'elles correspondent à votre environnement.

Point de contrôle de la leçon :

Vous avez configuré les propriétés SSL pour le serveur de catalogue.

Leçon 3.3 : Exécution de l'exemple

Redémarrez tous les serveurs et exécutez de nouveau le modèle d'application. Vous devriez être en mesure d'exécuter les étapes sans aucun problème.

Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 152 pour plus d'informations sur l'exécution et l'installation de l'exemple d'application.

Module 4 : utilisation de l'autorisation JAAS (Java Authentication and Authorization Service) dans WebSphere Application Server

Maintenant que vous avez configuré l'authentification pour les clients, vous pouvez configurer l'autorisation de manière plus précise pour accorder aux utilisateurs des autorisations différentes. Par exemple, un "opérateur" peut être autorisé uniquement à afficher les données, alors qu'un "gestionnaire" peut exécuter toutes les opérations.

Après avoir authentifié un client, comme dans le module précédent dans ce tutoriel, vous pouvez attribuer des privilèges de sécurité par le biais des mécanismes d'autorisation eXtreme Scale. Le module précédent de ce tutoriel vous a montré comment activer l'authentification pour une grille de données à l'aide de l'intégration à WebSphere Application Server. Par conséquent, aucun client non authentifié ne peut se connecter aux serveurs eXtreme Scale ni envoyer des demandes au système. Toutefois, tous les clients authentifiés possèdent les mêmes permissions ou privilèges liés au serveur, tels que la lecture, l'écriture ou la

suppression des données stockées dans les mappes ObjectGrid. Les clients peuvent également soumettre tout type de requête.

Cette partie du tutoriel explique comment utiliser l'autorisation eXtreme Scale pour attribuer différents privilèges aux utilisateurs authentifiés. WebSphere eXtreme Scale utilise un mécanisme d'autorisation basé sur l'autorisation. Vous pouvez affecter des catégories d'autorisations différentes qui sont représentées par des classes d'autorisation différentes. Ce module utilise la classe MapPermission. Pour la liste de toutes les propriétés possibles, voir Programmation d'autorisations client.

Dans WebSphere eXtreme Scale, la classe `com.ibm.websphere.objectgrid.security.MapPermission` représente les autorisations d'accès aux ressources eXtreme Scale , notamment les méthodes des interfaces ObjectMap ou JavaMap. WebSphere eXtreme Scale définit les chaînes de permission suivantes pour accéder aux méthodes des interfaces ObjectMap et JavaMap :

- **read** : accorde l'autorisation de lire les données de la mappe.
- **write** : accorde l'autorisation de mettre à jour les données de la mappe.
- **insert** : accorde l'autorisation d'insérer les données dans la mappe.
- **remove**: accorde l'autorisation de supprimer les données de la mappe.
- **invalidate** : accorde l'autorisation d'invalider les données dans la mappe.
- **all** : accorde toutes les autorisations ci-dessus.

L'autorisation se produit lorsqu'un client eXtreme Scale utilise une API d'accès aux données, telles que ObjectMap ,JavaMap, ou les API EntityManager. L'environnement d'exécution vérifie les autorisations d'exécution de la mappe correspondante lorsque la méthode est appelée. Si les autorisations d'accès requises ne sont pas accordées au client, une exception AccessControlException est générée. Ce module explique comment utiliser l'autorisation JAAS (Java Authentication and Authorization Service) pour accorder des autorisations d'accès à la mappe pour différents utilisateurs.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- activer les autorisations pour WebSphere eXtreme Scale ;
- activer les autorisations utilisateur.

Durée

Ce module prend 60 minutes environ.

Leçon 4.1 : Activation de l'autorisation WebSphere eXtreme Scale

Pour activer l'autorisation dans WebSphere eXtreme Scale, vous devez activer la sécurité sur un ObjectGrid spécifique.

Pour activer l'autorisation sur l'ObjectGrid, vous devez affecter à l'attribut **`securityEnabled`** la valeur true pour cet ObjectGrid spécifique dans le fichier XML. Pour ce tutoriel, vous pouvez utiliser le fichier `XSDeployment_sec.ear` dans le répertoire *samples_home*`/WASSecurity` dont la sécurité est déjà définie dans le fichier `objectGrid.xml` ou vous pouvez modifier le fichier existant `objectGrid.xml` pour activer la sécurité. Cette leçon explique comment modifier le fichier pour activer la sécurité.

1. Facultatif : Extrayez les fichiers dans le fichier XSDeployment.ear, puis décompressez le fichier XSDeploymentWeb.war.
2. Facultatif : Ouvrez le fichier objectGrid.xml et affectez à l'attribut **securityEnabled** sur true sur le niveau ObjectGrid. Voir un exemple de cet attribut ci-dessous :

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15" securityEnabled="true">
               <backingMap name="Map1" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

   Si vous avez plusieurs ObjectGrids définis, vous devez définir cet attribut dans chaque grille.
3. Facultatif : Remodularisez les fichiers XSDeploymentWeb.war et XSDeployment.ear pour inclure vos modifications.
4. Obligatoire : Désinstallez le fichier XSDeployment.ear, puis installez la mise à jour XSDeployment.ear. Vous pouvez utiliser le fichier que vous avez modifié dans les étapes précédentes ou vous pouvez installer le fichier XSDeployment_sec.ear qui est fourni dans le répertoire *samples_home*/WASSecurity. Voir «Leçon 2.4 : Installation et exécution de l'exemple», à la page 152 pour plus d'informations sur l'installation de l'application.
5. Redémarrez tous les serveurs d'applications pour activer l'autorisation WebSphere eXtreme Scale.

Point de contrôle de la leçon :

Vous avez activé la sécurité sur l'ObjectGrid, ce qui permet également d'activer l'autorisation dans la grille de données.

Leçon 4.2 : Activation des autorisations utilisateur

Dans le module d'authentification de ce tutoriel, vous avez créé les deux utilisateurs operator et manager. Vous pouvez affecter des autorisations différentes à ces utilisateurs avec l'autorisation JASS (Java Authentication and Authorization Service).

Définition de la règle d'autorisation JAAS (Java Authentication and Authorization Service) en utilisant des principaux utilisateur :
Vous pouvez affecter des autorisations aux utilisateurs que vous avez créés. Affectez les autorisations de lecture operator uniquement à toutes les mappes. Affectez à l'utilisateur manager toutes les autorisations. Utilisez le fichier de règle d'autorisation JAAS pour accorder des autorisations aux principaux.

Editez le fichier d'autorisation JAAS. Le fichier xsAuth3.policy se trouve dans le répertoire *samples_home*/security_extauth.

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    principal javax.security.auth.x500.X500Principal
    "CN=operator,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
```

```
    principal javax.security.auth.x500.X500Principal
    "CN=manager,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Dans ce fichier, le codebase `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` est une URL réservée à ObjectGrid. Toutes les autorisations ObjectGrid accordées aux principaux doivent utiliser ce codebase spécial. Les autorisations suivantes sont affectées dans ce fichier :

- Le première instruction accorde l'autorisation de mappe `read` au principal `"CN=operator,O=acme,OU=OGSample"`. L'utilisateur `"CN=operator,O=acme,OU=OGSample"` dispose uniquement de l'autorisation de lecture sur la mappe Map1 dans l'instance Grid ObjectGrid.
- La seconde instruction accorde toutes les autorisations de mappe au principal `"CN=manager,O=acme,OU=OGSample"`. L'utilisateur `"CN=manager,O=acme,OU=OGSample"` dispose uniquement de l'autorisation de lecture sur la mappe Map1 dans l'instance Grid ObjectGrid.

Définition du fichier de règle d'autorisation JAAS à l'aide de propriétés JVM :
Procédez comme suit pour définir les propriétés JVM pour les serveurs xs1 et xs2qui se trouvent dans le cluster xsCluster. Si vous utilisez une topologie qui est différente de l'exemple de topologie qui est utilisé dans ce tutoriel, définissez le fichier sur tous vos serveurs de conteneur.

1. Dans la console d'administration, cliquez sur **Serveurs** > **Serveurs d'applications** > *server_name* > **Java t gestion de processus** > **Définition de processus** > **Java virtual machine.**
2. Ajoutez les arguments JVM génériques suivants :
   `-Djava.security.policy=`*samples_home*`/security_extauth/xsAuth3.policy`
3. Cliquez sur **OK** et enregistrez les modifications.

Exécution de l'exemple d'application pour tester les autorisations :
Vous pouvez utiliser l'exemple d'application pour tester les paramètres d'autorisation. Le gestionnaire continue de disposer de toutes les autorisations dans la mappe Map1, y compris des autorisations d'affichage et d'ajout d'employés. L'opérateur doit pouvoir afficher uniquement les employés, car seule l'autorisation de lecture lui a été affectée.

1. Redémarrez tous les serveurs d'applications qui exécutent des serveurs de conteneur. Pour ce tutoriel, redémarrez les serveurs `xs1` et `xs2`.
2. Ouvrez l'application `EmployeeManagementWeb`. Dans un navigateur Web, ouvrez `http://<host>:<port>/EmployeeManagermentWeb/management.jsp`.
3. Connectez-vous à l'application en utilisant un nom et un mot de passe utilisateur.
4. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique `authemp1@acme.com`. Un message indique que l'utilisateur est introuvable.
5. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique `authemp1@acme.com`, le prénom `Joe` et le nom `Doe`. Cliquez sur **Soumettre**. Un message indique que l'employé a été ajouté.
6. Editez *samples_home*`/security_extauth/client3.props`. Remplacez la valeur `manager manager1` de la propriété credentialGeneratorProps par `operator operator1`. Après avoir modifié le fichier, le servlet utilise le nom d'utilisateur "operator" et le mot de passe "operator1" pour s'authentifier sur les serveurs WebSphere eXtreme Scale.

7. Redémarrez le cluster appCluster pour appliquer les modifications dans le fichier *samples_home*/security_extauth/client3.props.
8. Essayez d'afficher un employé. Cliquez sur **Afficher un employé** et recherchez l'adresse électronique authemp1@acme.com. L'employé s'affiche.
9. Ajoutez un employé. Cliquez sur **Ajouter un employé**. Ajoutez l'adresse électronique authemp2@acme.com, le prénom Joe et le nom Doe. Cliquez sur **Soumettre**. Le message suivant s'affiche :

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   Texte de l'exception détaillée :

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Ce message s'affiche, car l'opérateur n'est pas autorisé à insérer des données dans la mappe Map1.

Si vous utilisez une version de WebSphere Application Server antérieure à la version 7.0.0.11, une erreur java.lang.StackOverflowError peut s'afficher sur le serveur de conteneur. Elle est provoquée par IBM Developer Kit. Le problème est résolu dans IBM Developer Kit fourni avec WebSphere Application Server Version 7.0.0.11 et les versions suivantes.

Point de contrôle de la leçon :

Dans cette leçon, vous avez configuré l'autorisation en attribuant des autorisations à des utilisateurs spécifiques.

Module 5 : Utilisation de l'utilitaire xscmd pour surveiller les grilles de données et les mappes

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher les grilles de données primaires et les tailles de mappe de la grille de données Gride. L'outil **xscmd** utilise le bean géré pour interroger tous les artefacts de grille de données, tels que les fragments primaires, les fragments de réplique, des serveurs de conteneur, les tailles de mappe et d'autres données.

Dans ce tutoriel, les serveurs de catalogue s'exécutent comme serveurs Java SE autonomes. Les serveurs de conteneur s'exécutent sur des serveurs d'applications WebSphere Application Server.

Pour le serveur de catalogue, un serveur MBean est créé dans la machine JVM (Java virtual machine). Lorsque vous utilisez l'outil **xscmd** sur le serveur de catalogue, la sécurité WebSphere eXtreme Scale est utilisée.

Pour les serveurs de conteneur, l'exécution WebSphere eXtreme Scale enregistre les beans gérés (MBean) avec le serveur MBean créé par l'exécution WebSphere Application Server. La sécurité est utilisée par l'outil **xscmd** est fournie par la sécurité MBean WebSphere Application Server.

1. A l'aide d'un outil de ligne de commande, ouvrez le répertoire *DMGR_PROFILE*/bin.
2. Exécutez l'outil **xscmd**. Utilisez les paramètres **-c showPlacement -st P** comme dans les exemples suivants :

   Linux UNIX

   ```
   xscmd.sh -c showPlacement -cep localhost:16099 -g Grid -ms mapSet -sf P
   -user manager -pwd manager1
   ```

Windows

```
xscmd.bat -c showPlacement -cep localhost:16099 -g Grid -m mapSet -sf P
-user manager -pwd manager1
```

Avertissement :

Si vous utilisez la commande suivante pour accéder à la grille de données, vous pouvez être également autorisé à exécuter des actions d'administration, telles qu'exécuter la commande listAllJMXAddresses :

```
./xscmd.sh -user <user> -password <password> <other_parameters>
```

Si cette opération fonctionne pour cet utilisateur, une opération **xscmd** peut également être exécutée par le même utilisateur. Pour plus d'informations, voir «Traitement des problèmes de sécurité», à la page 695
Le nom d'utilisateur et le mot de passe sont transmis au serveur de catalogue pour l'authentification.

3. Affichez les résultats de la commande.

```
*** Showing all primaries for grid - Grid & mapset - mapSet
Partition Container Host Server
0 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
1 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
2 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
3 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
4 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
```

4. Exécutez l'outil **xscmd**. Utilisez le paramètre **-c showMapSizes** comme dans les exemples suivants :

Linux UNIX

```
xscmd.sh -c showMapSizes -cep localhost:16099 -g Grid -ms mapSet  -user manager -pwd manager1
```

Windows

```
xscmd.bat -c showMapSizes -cep localhost:16099 -g Grid -ms mapSet -user manager -pwd manager1
```

Le nom d'utilisateur et le mot de passe sont transmis au serveur de catalogue pour être authentifiés. Après avoir exécuté la commande, vous êtes invité à indiquer l'ID utilisateur et le mot de passe WebSphere Application Server pour l'authentification dans WebSphere Application Server. Vous devez fournir ces informations de connexion, car l'option **-c showMapSizes** obtient la taille de mappe de chaque serveur de conteneur qui nécessite la sécurité WebSphere Application Server.

5. Facultatif : Vous pouvez modifier le fichier `PROFILE/properties/sas.client.props` pour exécuter la commande sans que l'ID utilisateur et le mot de passe soient nécessaires. Remplacez la propriété com.ibm.CORBA.loginSource `prompt` par `properties`, puis indiquez l'ID utilisateur et le mot de passe. Voici un exemple des propriétés dans le fichier `PROFILE/properties/sas.client.props` :

```
com.ibm.CORBA.loginSource=properties
# RMI/IIOP user identity
com.ibm.CORBA.loginUserid=Admin
com.ibm.CORBA.loginPassword=xxxxxx
```

6. Facultatif : Si vous utilisez la commande **xscmd** sur une installation autonome WebSphere eXtreme Scale, vous devez ajouter les options suivantes :
   - Si vous utilisez la sécurité WebSphere eXtreme Scale :

     ```
     -user
     -pwd
     ```

- Si vous utilisez la sécurité WebSphere eXtreme Scale avec la génération de données d'identification personnalisée :

  ```
  -user
  -pwd
  -cgc
  -cgp
  ```

- Si SSL est activé :

  ```
  -tt
  -cxpv
  -prot
  -ks
  -ksp
  -kst
  -ts
  -tsp
  -tst
  ```

  Si la sécurité WebSphere eXtreme Scale et SSL sont tous les deux activés, les deux groupes de paramètres sont requis.

Point de contrôle de la leçon

Vous avez utilisé l'outil **xscmd** pour surveiller les grilles de données et les mappes dans la configuration.

Tutoriel : Exécution des ensembles eXtreme Scale dans la structure OSGi

L'exemple OSGi repose sur les exemples de sérialiseurs Google Protocol Buffers. A la fin de ce groupe de leçons, vous aurez exécuter les exemples de plug-in du sérialiseur dans l'infrastructure OSGi.

Objectifs d'apprentissage

Cet exemple montre les ensembles OSGi. Le plug-in de sérialiseur est secondaire et il n'est pas requis. L'exemple OSGi est disponible dans la galerie des exemples WebSphere eXtreme Scale. Vous devez télécharger l'exemple et l'extraire dans le répertoire *wxs_home*/samples. Le répertoire racine de l'exemple OSGi est wxs_home/samples/OSGiProto.

Les exemples de commande dans ce tutoriel suppose que vous exécutez le système d'exploitation UNIX. Vous devez ajuster l'exemple de commande pour l'exécuter sur un système d'exploitation Windows.

A la fin des leçons de ce tutoriel, vous comprendrez les concepts des exemples OSGi et saurez :

- installer l'ensemble de serveur WebSphere eXtreme Scale dans le conteneur OSGi pour démarrer le serveur eXtreme Scale ;
- configurer votre environnement de développement eXtreme Scale pour exécuter l'exemple de client ;
- utiliser la commande xscmd pour interroger le classement de l'exemple d'ensemble, le mettre à niveau vers un nouveau classement de services et vérifier le nouveau classement de services.

Durée

Ce module prend 60 minutes environ.

Prérequis

Dans ce tutoriel, vous devez télécharger et extraire les exemples de sérialiseurs et :

- Installer et extraire le produit eXtreme Scale
- Configurer l'environnement Eclipse Equinox

Introduction : Démarrage et configuration du serveur eXtreme Scale et du conteneur pour exécuter les plug-in dans la structure OSGi

Dans ce tutoriel, vous allez démarrer un serveur eXtreme Scale dans l'infrastructure OSGi, démarrer un conteneur eXtreme Scale et connecter les exemples de plug-in avec l'environnement d'exécution eXtreme Scale.

Objectifs d'apprentissage

A la fin des leçons de ce tutoriel, vous comprendrez les concepts des exemples OSGi et saurez :

- installer l'ensemble de serveur WebSphere eXtreme Scale dans le conteneur OSGi pour démarrer le serveur eXtreme Scale ;
- configurer l'environnement de développement eXtreme Scale pour exécuter l'exemple de client ;
- utiliser la commande xscmd pour interroger le classement de l'exemple d'ensemble, le mettre à niveau vers un nouveau classement de services et vérifier le nouveau classement de services.

Durée

Ce tutorial prend 60 minutes environ. Si vous explorez d'autres concepts liés à ce tutoriel, il peut prendre plus de temps.

Niveau de compétence

Intermédiaire

Audience

Les développeurs et les administrateurs qui veulent créer, installer et exécuter des ensembles eXtreme Scale dans l'infrastructure OSGi.

Configuration requise

- Client de ligne de commande Luminis OSGi Configuration Admin, version 0.2.5
- Apache Felix File Install, version 3.0.2
- Lorsque vous utilisez Eclipse Gemini en tant que fournisseur de conteneur Blueprint, les éléments suivants sont requis :
  - Eclipse Gemini Blueprint, version 1.0.0
  - Spring Framework, version 3.0.5
  - SpringSource AOP Alliance API, version 1.0.0
  - SpringSource Apache Commons Logging, version 1.1.1
- Lorsque vous utilisez Apache Aries en tant que fournisseur du conteneur Blueprint, vous devez disposer de la configuration suivante :
  - Dernière image instantanée Aries

– Bibliothèque ASM
– Consignation PAX

Prerequis

Pour pouvoir exécuter ce tutoriel, vous devez télécharger l'exemple et l'extraire dans le répertoire `wxs_home/samples`. Le répertoire racine de l'exemple OSGi est `wxs_home/samples/OSGiProto`.

Résultats attendus

A la fin de ce tutoriel, vous aurez installé les exemples d'ensembles et exécuté un client eXtreme Scale pour insérer des données dans la grille. Vous serez également amené à interroger et mettre à jour ces exemples d'ensembles en utilisant les fonctions dynamiques que fournit le conteneur OSGi.

Module 1 : Préparation de l'installation et de la configuration des ensembles de serveur eXtreme Scale

Effectuez ce module pour explorer les exemples d'ensembles OSGi et examiner les fichiers de configuration que vous utilisez pour configurer le serveur eXtreme Scale.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous comprendrez les concepts et saurez :

- localiser et explorer les ensembles qui sont inclus dans le modèle OSG ;
- examiner les fichiers de configuration utilisés pour configurer la grille et le serveur eXtreme Scale.

Leçon 1.1 : Explication des exemples d'ensembles OSGi

Suivez cette leçon pour localiser et explorer les ensembles fournis dans l'exemple OSGi.

Exemples d'ensembles OSGi :
Hormis les ensembles qui sont configurés dans le fichier `config.ini`, qui est indiqué dans la rubrique sur la configuration de l'environnement Eclipse Equinox, les ensembles supplémentaires suivants sont utilisés dans le modèle OSGi :

`objectgrid.jar`
Ensemble d'exécution de serveur WebSphere eXtreme Scale. Cet ensemble se trouve dans le répertoire *`wxs_home`*`/lib`.

`com.google.protobuf_2.4.0a.jar`
Ensemble Google Protocol Buffers, version 2.4.0a. Cet ensemble se trouve dans le répertoire *`wxs_sample_osgi_root`*`/lib`.

`ProtoBufSamplePlugins-1.0.0.jar`
Version 1.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *`wxs_sample_osgi_root`*`/lib`. Les services sont configurés avec le classement de service 1.

Cette version utilise le XML Blueprint standard pour configurer les services de plug-in eXtreme Scale. La classe de service est une classe implémentée par l'utilisateur pour l'interface WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. La classe

implémentée par l'utilisateur crée un bean pour chaque demande et fonctionne de la même manière qu'un bean de portée prototype.

ProtoBufSamplePlugins-2.0.0.jar
Version 2.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 2.

Cette version utilise le XML Blueprint standard pour configurer les services de plug-in eXtreme Scale. La classe de service utilise une classe intégrée WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, qui utilise le service BlueprintContainer. En utilisant la configuration XML Blueprint standard, les beans peuvent être configurés en tant que portée singleton ou portée prototype. Le bean n'est pas configuré en tant que portée de fragment.

ProtoBufSamplePlugins-Gemini-3.0.0.jar
Version 3.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 3.

Cette version utilise le XML Blueprint XML d'Eclipse Gemini pour configurer les services de plug-in eXtreme Scale. La classe de service utilise une classe de service intégrée, WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, qui utilise le service BlueprintContainer. Pour configurer un bean de portée de fragment, utilisez une approche Gemini. Cette version configure le bean myShardListener comme bean de portée de fragment en fournissant {http://www.ibm.com/schema/objectgrid}shard comme valeur de portée et en configurant un attribut factice pour que Gemini reconnaisse la portée personnalisée. Le problème Eclipse est généré par : https://bugs.eclipse.org/bugs/show_bug.cgi?id=348776

ProtoBufSamplePlugins-Aries-4.0.0.jar
Version 4.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 4.

Cette version utilise le XML Blueprint standard pur configurer les services de plug-in eXtreme Scale. La classe de service utilise une classe intégrée WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, qui utilise le service BlueprintContainer. En utilisant la configuration XML Blueprint standard, les beans peuvent être configurés en utilisant une porté personnalisée. Cette version configure myShardListenerbean comme bean à portée de fragment en fournissant {http://www.ibm.com/schema/objectgrid}shard comme valeur de portée.

ProtoBufSamplePlugins-Activator-5.0.0.jar
Version 5.0.0 de l'ensemble de plug-in utilisateur avec l'exemple ObjectGridEventListener et les implémentations de plug-in

MapSerializerPlugin. Cet ensemble se trouve dans le répertoire *wxs_sample_osgi_root*/lib. Les services sont configurés avec le classement de service 5.

Cette version n'utilise pas du tout le conteneur Blueprint. Dans cette version, les services sont enregistrés à l'aide de l'enregistrement de service OSGi. La classe de service est une classe implémentée par l'utilisateur pour l'interface WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. La classe implémentée par l'utilisateur crée un bean pour chaque demande. Elle fonctionne d'une manière similaire à un bean à portée prototype.

Point de contrôle de la leçon :

En explorant les ensembles qui sont fournis avec le modèle OSGi, vous pouvez mieux comprendre la procédure de développement de vos propres implémentations qui s'exécutent dans le conteneur OSGi.

Vous avez appris :

- les ensembles inclus avec l'exemple OSGi ;
- l'emplacement de ces ensembles ;
- l'élément utilisé pour configurer le classement de service de chaque ensemble.

Leçon 1.2 : Description des fichiers de configuration OSGi

L'exemple OSGi inclut des fichiers de configuration que vous pouvez utiliser pour démarrer et configurer le serveur et la grille WebSphere eXtreme Scale.

Fichiers de configuration OSGi :
Dans cette leçon, vous allez explorer les fichiers de configuration suivants fournis dans l'exemple OSGI :

- `collocated.server.properties`
- `protoBufObjectGrid.xml`
- `protoBufDeployment.xml`
- `blueprint.xml`

collocated.server.properties

Une configuration de serveur est nécessaire pour démarrer un serveur. Lorsque l'ensemble de serveur eXtreme Scale est démarré, il ne démarre pas un serveur. Il attend la création du PID de configuration `com.ibm.websphere.xs.server` avec un fichier de propriétés de serveur. Ce fichier de propriétés du serveur indique le nom du serveur, le numéro de port et d'autres propriétés du serveur.

Dans la plupart des cas, vous créez une configuration pour définir le fichier des propriétés du serveur. Dans de rares cas, vous pouvez vouloir uniquement démarrer un serveur avec chaque propriété affectée d'une valeur par défaut. Dans ce cas, vous pouvez créer une configuration appelée `com.ibm.websphere.xs.server` avec la valeur `default`.

Pour plus d'informations sur le fichier des propriétés du serveur, voir la rubrique Fichier de propriétés du serveur.

L'exemple de fichier de propriétés du serveur OSGi démarre un catalogue unique. Ce fichier de propriétés démarre un service de catalogue unique et un serveur de

conteneur dans le processus d'infrastructure OSGi. Les clients eXtreme Scale se connectent sur le port 2809 et les clients JMX, sur le port 1099. Contenu du fichier de propriétés de serveur :

```
serverName=collocatedServer
isCatalog=true
catalogClusterEndPoints=collocatedServer:localhost:6601:6602
traceSpec=ObjectGridOSGi=all=enabled
traceFile=logs/trace.log
listenerPort=2809
JMXServicePort=1099
```

protoBufObjectGrid.xml

L'exemple de fichier XML descripteur `protoBufObjectGrid.xml` ObjectGrid contient les éléments suivants avec les commentaires supprimés.

```
<objectGridConfig
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15">

            <bean id="ObjectGridEventListener"
                osgiService="myShardListener"/>

            <backingMap name="Map" readOnly="false"
                lockStrategy="PESSIMISTIC" lockTimeout="5"
                copyMode="COPY_TO_BYTES"
                pluginCollectionRef="serializer"/>

        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="serializer">
            <bean id="MapSerializerPlugin"
                osgiService="myProtoBufSerializer"/>"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
 </objectGridConfig>
```

Il existe deux plug-in configurés dans ce fichier descripteur XML ObjectGrid :

ObjectGridEventListener
Plug-in au niveau du fragment. Pour chaque instance ObjectGrid, il existe une instance de ObjectGridEventListener. Elle est configurée pour utiliser le service OSGi myShardListener. Cela signifie que lorsque la grille est créée, le plug-in ObjectGridEventListener utilise le service OSGi myShardListener avec le classement de service le plus élevé disponible.

MapSerializerPlugin
Plug-in au niveau de la mappe. Pour la mappe de sauvegarde nommée `Map`, il existe un plug-in MapSerializerPlugin configuré. Il est configuré pour utiliser le service OSGI myProtoBufSerializer. Cela signifie que lorsque la grille est créée, le plug-in MapSerializerPlugin utilise le service, myProtoBufSerializer, avec le classement de service le plus élevé disponible.

protoBufDeployment.xml

Le fichier descripteur XML de déploiement décrit la stratégie de déploiement pour la grille `Grid` qui utilise cinq partitions. Voir l'exemple de code de fichier XML suivant :

```
<deploymentPolicy
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="Grid">
    <mapSet name="MapSet" numberOfPartitions="5">
      <map ref="Map"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

blueprint.xml

Comme alternative à l'utilisation du fichier `collocated.server.properties` en association avec le PID de configuration, `com.ibm.websphere.xs.server`, vous pouvez inclure le fichier XML ObjectGrid et des fichiers XML de déploiement dans un ensemble OSGi, avec un fichier XML Blueprint, comme dans l'exemple suivant :

```
<blueprint
   xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
       xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
       default-activation="lazy">

    <objectgrid:server id="server" isCatalog="true"
                name="server"
                tracespec="ObjectGridOSGi=all=enabled"
                tracefile="C:/Temp/logs/trace.log"
                workingDirectory="C:/Temp/working"
                jmxport="1099">
        <objectgrid:catalog host="localhost" port="2809"/>
    </objectgrid:server>

    <objectgrid:container id="container"
  objectgridxml="/META-INF/objectgrid.xml"
       deploymentxml="/META-INF/deployment.xml"
  server="server"/>
</blueprint>
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez découvert les fichiers de configuration qui sont utilisées dans l'exemple OSGi. Maintenant, lorsque vous démarrez et configurez la grille eXtreme Scale et le serveur, vous savez quels fichiers sont utilisés dans ces processus et comment ces fichiers interagissent avec vos plug-in dans l'infrastructure OSGi.

Module 2 : Installation et démarrage des ensembles eXtreme Scale dans l'infrastructure OSGi

Utilisez les leçons de ce module leçon pour installer l'ensemble de serveurs eXtreme Scale dans le conteneur OSGi et démarrer le serveur WebSphere eXtreme Scale.

Le démarrage du serveur dans l'infrastructure OSGi n'implique pas que les ensembles OSGi sont prêts à être exécutés. Vous devez configurer les propriétés du serveur et les conteneurs de sorte que les ensembles OSGi que vous installez soient reconnus et puissent s'exécuter correctement.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous comprendrez les concepts et saurez :

- installer les ensembles eXtreme Scale en utilisant la console OSGi Equinox ;
- configurer le serveur eXtreme Scale ;

- configurer le conteneur eXtreme Scale ;
- installer et démarrer les exemples d'ensemble eXtreme Scale.

Prérequis

Pour pouvoir exécuter ce module, vous devez effectuer préalablement les tâches suivantes :

- Installer et extraire le produit eXtreme Scale
- Définir l'environnement Eclipse Equinox

Vous devez également préparer l'accès aux fichiers suivants pour suivre les leçons de ce module :

- Ensemble `objectgrid.jar`. Vous installez cet ensemble eXtreme Scale.
- `Fichier collocated.server.properties`. Vous ajoutez les propriétés du serveur à ce fichier de configuration.

Vous pouvez envisager d'installer et de démarrer les ensembles suivants :

- `protobuf-java-2.4.0a-bundle.jar`
- `ProtoBufSamplePlugins-1.0.0.jar`

Leçon 2.1 : Démarrage de la console et installation de l'ensemble de serveur eXtreme Scale

Dans cette leçon, vous utilisez la console Equinox OSGi pour installer l'ensemble de serveur WebSphere eXtreme Scale.

1. Utilisez la commande suivante pour démarrer la console OSGi Equinox :

   ```
   cd equinox_root
   java -jar plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```

2. Une fois la console OSGi démarrée, exécutez la commande `ss` dans la console ; les ensembles suivants sont démarrés :

   Avertissement : Si vous avez terminé d'exécuter la tâche Installation des ensembles eXtreme Scale, l'ensemble a déjà été activé. Si l'ensemble est démarré, arrêtez-le avant de terminer cette tâche.

   Sortie Eclipse Gemini :

   ```
   osgi> ss
   Framework is launched.
   id State Bundle
   0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
   1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
   2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
   3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
   4 ACTIVE com.springsource.org.apache.commons.logging_1.1.1
   5 ACTIVE com.springsource.org.aopalliance_1.0.0
   6 ACTIVE org.springframework.aop_3.0.5.RELEASE
   7 ACTIVE org.springframework.asm_3.0.5.RELEASE
   8 ACTIVE org.springframework.beans_3.0.5.RELEASE
   9 ACTIVE org.springframework.context_3.0.5.RELEASE
   10 ACTIVE org.springframework.core_3.0.5.RELEASE
   11 ACTIVE org.springframework.expression_3.0.5.RELEASE
   12 ACTIVE org.apache.felix.fileinstall_3.0.2
   13 ACTIVE net.luminis.cmc_0.2.5
   14 ACTIVE org.eclipse.gemini.blueprint.core_1.0.0.RELEASE
   15 ACTIVE org.eclipse.gemini.blueprint.extender_1.0.0.RELEASE
   16 ACTIVE org.eclipse.gemini.blueprint.io_1.0.0.RELEASE
   ```

   Sortie Apache Aries :

   ```
   osgi> ss
   Framework is launched.
   id State Bundle
   ```

```
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE org.ops4j.pax.logging.pax-logging-api_1.6.3
5 ACTIVE org.ops4j.pax.logging.pax-logging-service_1.6.3
6 ACTIVE org.objectweb.asm.all_3.3.0
7 ACTIVE org.apache.aries.blueprint_0.3.2.SNAPSHOT
8 ACTIVE org.apache.aries.util_0.4.0.SNAPSHOT
9 ACTIVE org.apache.aries.proxy_0.4.0.SNAPSHOT
10 ACTIVE org.apache.felix.fileinstall_3.0.2
11 ACTIVE net.luminis.cmc_0.2.5
```

3. Installez l'ensemble `objectgrid.jar`. Pour démarrer un serveur dans la machine JVM (Java virtual machine), vous devez installer un ensemble de serveur eXtreme Scale. Cet ensemble de serveur eXtreme Scale peut démarrer un serveur et créer des conteneurs. Utilisez la commande suivante pour installer le fichier `objectgrid.jar` :

   ```
   osgi> install file:///wxs_home/lib/objectgrid.jar
   ```

   Reportez-vous à l'exemple suivant :

   ```
   osgi> install file:///opt/wxs/ObjectGrid/lib/objectgrid.jar
   ```

   Equinox affiche son ID d'ensemble, par exemple :

   ```
   Bundle id is 19
   ```

   A faire : Votre ID d'ensemble peut être différent. Le chemin de ficher doit être une adresse URL absolue dans le chemin de l'ensemble. Les chemins relatifs ne sont pas pris en charge.

Point de contrôle de la leçon :

Dans cette leçon, vous avez utilisé la console OSGi Equinox pour installer l'ensemble `objectgrid.jar` que vous allez utiliser pour démarrer et créer ensuite un conteneur dans ce tutoriel.

Leçon 2.2 : Personnalisation et configuration du serveur eXtreme Scale

Suivez cette leçon pour personnaliser et ajouter les propriétés au serveur WebSphere eXtreme Scale.

1. Editez le fichier `wxs_sample_osgi_root/projects/server/properties/collocated.server.properties`.
   a. Remplacez la propriété traceFile par `equinox_root/logs/trace.log`.
2. Enregistrez le fichier.
3. Entrez les lignes de code suivantes dans la console OSGI pour créer la configuration du serveur à partir du fichier. L'exemple suivant est présenté sur plusieurs lignes en raison des contraintes liées à la publication.

   ```
   osgi> cm create com.ibm.websphere.xs.server
   osgi> cm put com.ibm.websphere.xs.server objectgrid.server.props
   wxs_sample_osgi_root/projects/server/properties/collocated.server.properties
   ```

4. Pour afficher la configuration, exécutez la commande suivante :

   ```
   osgi> cm get com.ibm.websphere.xs.server
   Configuration for service (pid) "com.ibm.websphere.xs.server"
   (bundle location = null)
   key value
   ----                    ----
   ```

```
objectgrid.server.props  wxs_sample_osgi_root/projects/server/properties
                         /collocated.server.properties
service.pid              com.ibm.websphere.xs.server
```

Point de contrôle de la leçon :

Au cours de cette leçon, vous avez édité le fichier `wxs_sample_osgi_root/projects/server/properties/collocated.server.properties` pour spécifier les paramètres du serveur, tels que le répertoire de travail et l'emplacement des fichiers journaux de trace.

Leçon 2.3 : Configuration du conteneur eXtreme Scale

Suivez cette leçon pour configurer un conteneur qui inclut le fichier descripteur XML d'ObjectGrid et le fichier XML de déploiement d'ObjectGrid WebSphere eXtreme Scale. Ces fichiers incluent la configuration de la grille et sa topologie.

Pour créer un conteneur, commencez par créer un service de configuration à l'aide du PID (process identification number) de la fabrique de services gérés, com.ibm.websphere.xs.container. La configuration de service est une fabrique de services gérés qui permet de créer plusieurs PID de service depuis le PID de la fabrique. Pour démarrer le service de conteneur, affectez aux PID objectgridFile et deploymentPolicyFile chaque PID de service.

Procédez comme suit pour personnaliser et ajouter les propriétés du serveur à l'infrastructure OSGi :

1. Dans la console OSGI, entrez la commande suivante pour créer le conteneur depuis le fichier :

   ```
   osgi> cm createf com.ibm.websphere.xs.container
   PID: com.ibm.websphere.xs.container-1291179621421-0
   ```

2. Entrez les commandes suivantes pour lier le PID nouvellement créé aux fichiers XML ObjectGrid.

   A faire : Le numéro PID sera différent de celui de cet exemple.

```
osgi> cm put com.ibm.websphere.xs.container-1291179621421-0 objectgridFile wxs_sample_osgi_root/projects/server/META-INF/protoBufObjectgrid.xml

osgi> cm put com.ibm.websphere.xs.container-1291179621421-0 deploymentPolicyFile wxs_sample_osgi_root/projects/server/META-INF
/protoBufDeployment.xml
```

3. Utilisez la commande suivante pour afficher la configuration :

```
osgi> cm get com.ibm.websphere.xs.container-1291760127968-0
Configuration for service (pid) "com.ibm.websphere.xs.container-1291760127968-0"
(bundle location = null)

key value
------                  ------
deploymentPolicyFile    /opt/wxs/ObjectGrid/samples/OSGiProto/server/META-INF/protoBufDeployment.xml
objectgridFile          /opt/wxs/ObjectGrid/samples/OSGiProto/server/META-INF/protoBufObjectgrid.xml
service.factoryPid    com.ibm.websphere.xs.container
service.pid        com.ibm.websphere.xs.container-1291760127968-0
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez créé un service de configuration qui vous a permis de créer un conteneur eXtreme Scale. Comme les fichiers XML ObjectGrid contiennent la configuration de la grille et sa topologie, vous devez lier le conteneur que vous avez créé pour ces fichiers XML ObjectGrid. Avec cette configuration, le conteneur eXtreme Scale peut reconnaître les ensembles OSGi que vous allez exécuter ultérieurement dans ce tutoriel.

Leçon 2.4 : Installation de Google Protocol Buffers et des ensembles de plug-in

Suivez ce tutoriel pour installer l'ensemble `protobuf-java-2.4.0a-bundle.jar` et l'ensemble de plug-in `ProtoBufSamplePlugins-1.0.0.jar` en utilisant la console OSGi Equinox OSGi.

Installation du plug-in Google Protocol Buffers :
Suivez la procédure ci-après pour installer le plug-in Google Protocol Buffers.

Dans la console OSGI, entrez la commande suivante pour installer le plug-in :

```
osgi> install file:///wxs_sample_osgi_root/lib/com.google.protobuf_2.4.0a.jar
```

La sortie suivante s'affiche :

```
Bundle ID is 21
```

Présentation des exemples d'ensembles de plug-in :
L'exemple OSGi inclut cinq exemples d'ensembles qui contiennent les plug-in eXtreme Scale, notamment un plug-in personnalisé ObjectGridEventListener et un plug-in MapSerializerPlugin. Le plug-in MapSerializerPlugin utilise l'exemple Google Protocol Buffers et les messages fournis par l'exemple MapSerializerPlugin.

Les ensembles suivants se trouvent dans le répertoire *wxs_sample_osgi_root*`/lib` : `ProtoBufSamplePlugins-1.0.0.jar` et `ProtoBufSamplePlugins-2.0.0.jar`.

Le fichier `blueprint.xml` contient ce qui suit sans les commentaires :

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
 <bean id="myShardListener" class="com.ibm.websphere.samples.xs.proto.osgi.MyShardListenerFactory"/>
 <service ref="myShardListener" interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory" ranking="1">
 </service>

 <bean id="myProtoBufSerializer" class="com.ibm.websphere.samples.xs.proto.osgi.ProtoMapSerializerFactory">
  <property name="keyType" value="com.ibm.websphere.samples.xs.serializer.app.proto.DataObjects1$OrderKey" />
  <property name="valueType" value="com.ibm.websphere.samples.xs.serializer.app.proto.DataObjects1$Order" />
 </bean>

 <service ref="myProtoBufSerializer" interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
  ranking="1">
 </service>
</blueprint>
```

Le fichier XML Blueprint exporte les deux services myShardListener et myProtoBufSerializer. Ces deux services sont référencés dans le fichier `protoBufObjectgrid.xml`.

Installation de l'ensemble de plug-in :

Procédez comme suit pour installer l'ensemble `ProtoBufSamplePlugins-1.0.0.jar`.

Exécutez la commande suivante dans la console OSGi Equinox pour installer l'ensemble de plug-in `ProtoBufSamplePlugins-1.0.0.jar` :

```
osgi> install file:///wxs_sample_osgi_root/lib/ProtoBufSamplePlugins-1.0.0.jar
```

La sortie suivante s'affiche :

```
Bundle ID is 22
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez installé l'ensemble `protobuf-java-2.4.0a-bundle.jar` et l'ensemble de plug-in `ProtoBufSamplePlugins-1.0.0.jar`.

Leçon 2.5 : Démarrage des ensembles OSGi

Le serveur WebSphere eXtreme Scale est modularisé comme ensemble de serveur OSGi. Suivez cette leçon pour installer l'ensemble de serveur eXtreme Scale et d'autres ensembles OSGi que vous avez installés.

1. Exécutez la commande **ss** pour afficher les ID de chaque ensemble.

```
osgi> ss

Framework is launched.

id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE com.springsource.org.apache.commons.logging_1.1.1
5 ACTIVE com.springsource.org.aopalliance_1.0.0
6 ACTIVE org.springframework.aop_3.0.5.RELEASE
7 ACTIVE org.springframework.asm_3.0.5.RELEASE
8 ACTIVE org.springframework.beans_3.0.5.RELEASE
9 ACTIVE org.springframework.context_3.0.5.RELEASE
10 ACTIVE org.springframework.core_3.0.5.RELEASE
11 ACTIVE org.springframework.expression_3.0.5.RELEASE
12 ACTIVE org.apache.felix.fileinstall_3.0.2
13 ACTIVE net.luminis.cmc_0.2.5
15 ACTIVE org.eclipse.gemini.blueprint.core_1.0.0.RELEASE
16 ACTIVE org.eclipse.gemini.blueprint.extender_1.0.0.RELEASE
17 ACTIVE org.eclipse.gemini.blueprint.io_1.0.0.RELEASE
19 RESOLVED com.ibm.websphere.xs.server_7.1.1
21 RESOLVED Google_ProtoBuf_2.4.0
22 RESOLVED ProtoBufPlugins_1.0.0
```

2. Démarrez chaque ensemble que vous avez installé. Vous devez démarrer les ensembles en suivant un ordre précis. Consultez l'exemple précédent pour connaître l'ordre des ID des ensembles.
   a. Démarrez l'ensemble des exemples de plug-in, `ProtoBufPlugins_1.0.0`. Exécutez la commande suivante dans la console OSGi Equinox pour démarrer l'ensemble. Dans cet exemple, l'ID d'ensemble de l'exemple de plug-in est 22.

      ```
      osgi> start 22
      ```

   b. Démarrez l'ensemble Google Protocol Buffers, `Google_ProtoBuf_2.4.0`. Exécutez la commande suivante dans la console OSGi Equinox pour démarrer l'ensemble. Dans cet exemple, l'ID d'ensemble du plug-in Google Protocol Buffers est 21.

      ```
      osgi> start 21
      ```

   c. Démarrez l'ensemble des serveurs, `com.ibm.websphere.xs.server_7.1.1`. Exécutez la commande suivante dans la console OSGi Equinox pour démarrer le serveur. Dans cet exemple, l'ID de l'ensemble de serveur eXtreme Scale est 19.

      ```
      osgi> start 19
      ```

Après avoir démarré le serveur, le programme d'écoute d'événement MyShardListener démarre et il est prêt à insérer ou mettre à jour les enregistrements. Vous pouvez visualiser la sortie suivante sur la console OSGi pour vérifier que l'ensemble de plug-in a démarré correctement :

```
SystemOut O MyShardListener@1253853884(version=1.0.0) order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Order$Builder
@1aba1aba(22) inserted
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez démarré deux ensembles de plug-in et l'ensemble de serveur dans le conteneur eXtreme Scale que vous avez configuré pour l'infrastructure OSGi.

Module 3 : Exécution de l'exemple de client eXtreme Scale

Le serveur WebSphere eXtreme Scale fonctionne maintenant dans un environnement OSGi. Suivez les étapes de ce module pour exécuter un client WebSphere eXtreme Scale qui insère des données dans la grille.

Objectifs d'apprentissage

A la fin des leçons de ce module, vous saurez :

- exécuter une application client qui se connecte à la grille et y insère et en extrait des données ;
- démarrer une commande en utilisant l'application client non-OSGi.

Prerequis

Exécutez le Module 2 : Installation et démarrage des ensembles eXtreme Scale dans l'infrastructure OSGi.

Leçon 3.1 : Configuration d'Eclipse pour exécuter le client et créer les exemples

Effectuez cette leçon pour importer le projet Eclipse que vous utiliserez pour exécuter le client et générer les exemples de plug-in.

L'exemple inclut un programme client Java SE qui se connecte à la grille, insère des données et en extrait. Il contient également des projets que vous pouvez utiliser pour générer et redéployer les ensembles OSGi.

Le projet fourni a été testé avec Eclipse 3.x et les versions suivantes et ne nécessite que la perspective du projet de développement Java. Procédez comme suit pour configurer votre environnement de développement WebSphere eXtreme Scale.

1. Ouvrez Eclipse dans un espace de travail nouveau ou existant.
2. Dans le menu Fichier, sélectionnez **Importer**.
3. Développez le dossier Général. Sélectionnez **Projets existants dans l'espace de travail** et cliquez sur **Suivant**.
4. Dans la zone **Sélectionner le répertoire racine**, tapez le répertoire *wxs_sample_osgi_root* ou accédez-y. Cliquez sur **Terminer**. Plusieurs nouveaux projets sont affichés dans votre espace de travail. Les erreurs de compilation seront corrigées par la définition de deux bibliothèques utilisateur. Procédez comme suit pour définir les bibliothèques utilisateur.
5. Dans le menu Fenêtre, sélectionnez **Préférences**.
6. Développez la branche **Java** > **Chemin de génération** et sélectionnez **Bibliothèques utilisateur**.
7. Définissez la bibliothèque utilisateur eXtreme Scale.
   a. Cliquez sur **Nouveau**.
   b. Tapez `eXtremeScale` dans la zone du **nom de bibliothèque utilisateur** et cliquez sur **OK**.
   c. Sélectionnez la nouvelle bibliothèque utilisateur et cliquez **Ajouter des fichiers JAR**.

1) Recherchez et sélectionnez le fichier `objectgrid.jar` dans le répertoire *wxs_install_root*`/lib`. Cliquez sur **OK**.
2) Pour inclure la documentation d'API ObjectGrid, sélectionnez l'emplacement de la documentation d'API pour le fichier `objectgrid.jar` que vous avez ajouté à l'étape précédente. Cliquez sur **Editer**.
3) Dans la zone Chemin d'emplacement de la documentation d'API, sélectionnez le fichier `Javadoc.zip` qui se trouve dans le répertoire *wxs_install_root*`/docs/javadoc.zip`.

8. Définissez la bibliothèque utilisateur Google Protocol Buffers.
   a. Cliquez sur **Nouveau**.
   b. Tapez `com.google.protobuf` dans la zone du **nom de bibliothèque utilisateur**, et cliquez sur **OK**.
   c. Sélectionnez la nouvelle bibliothèque utilisateur et cliquez **Ajouter des fichiers JAR**.
      1) Recherchez et sélectionnez le fichier `com.google.protobuf_2.4.0.a.jar` dans le répertoire *wxs_sample_osgi_root*`/lib`. Cliquez sur **OK**.

Point de contrôle de la leçon :

Dans cette leçon, vous avez importé l'exemple de projet Eclipse et défini les bibliothèques utilisateur qui corrigeaient les erreurs de compilation.

Leçon 3.2 : Démarrage d'un client et insertion de données dans la grille

Etudiez cette leçon pour démarrer un client et exécuter une application client.

L'application client Java est `com.ibm.websphere.samples.xs.proto.client.Client`. Le projet Eclipse, wxs.sample.osgi.protobuf.client, contient l'application client Java. Le principal fichier classe est `com.ibm.websphere.samples.xs.proto.client.Client`.

Ce client utilise une substitution de client, le fichier descripteur XML ObjectGrid pour remplacer la configuration OSGi, afin que le client puisse s'exécuter dans un environnement non-OSGi. Voir le contenu suivant du fichier avec les commentaires et les en-têtes supprimés. Certaines lignes de code sont réparties sur plusieurs lignes à des fins de formatage.

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15">
            <bean id="ObjectGridEventListener" className="" osgiService=""/>
            <backingMap name="Map" readOnly="false"
                lockStrategy="PESSIMISTIC" lockTimeout="5"
                copyMode="COPY_TO_BYTES" pluginCollectionRef="serializer"/>

        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="serializer">

   <bean id="MapSerializer"
   className="com.ibm.websphere.samples.xs.serializer.proto.ProtoMapSerializer"
       osgiService="">
    <property name="keyType" type="java.lang.String"
        value="com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$OrderKey" />
          <property name="valueType" type="java.lang.String"
              value="com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order" />
```

```
    </bean>
   </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Cliquez sur **Exécuter en tant que** > **application Java** pour exécuter l'application client.

Lorsque vous exécutez l'application, le message suivant est affiché. Le message indique qu'une commande a été insérée :

```
order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Order$Builder@5d165d16(5000000) inserted
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez démarré l'application `com.ibm.websphere.samples.xs.proto.client.Client` qui a généré une commande.

Module 4: Interrogation et mise à niveau de l'exemple d'ensemble

Suivez les leçons de ce module pour utiliser la commande **xscmd** pour interroger le classement de services de l'exemple d'ensemble, mettez-le à niveau vers un nouveau classement de services et vérifiez ce dernier.

Objectifs d'apprentissage

Après avoir suivi les leçons de ce module, vous saurez :

- interroger le classement de services en cours d'un service ;
- interroger le classement en cours de tous les services ;
- interroger tous les classements disponibles d'un service ;
- interroger tous les classements de services disponibles ;
- utiliser l'outil xscmd pour déterminer les classements de services disponibles ;
- mettre à jour les classements des exemples de services OSGi.

Prérequis

Exécutez le Module 3 : Exécution de l'exemple de client eXtreme Scale.

Leçon 4.1 : recherche des classements de services

Etudiez cette leçon pour identifier les classements de services en cours ainsi que les classements de services qui sont disponibles pour la mise à niveau.

- Identifiez le classement de service en cours d'un service. Entrez la commande suivante pour interroger le classement de service en cours utilisé pour le service myShardListener utilisée par la grille ObjectGrid `Grid` et le groupe de mappes `MapSet`.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour interroger le classement en cours du service, myShardListener.
     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet -sn myShardListener
     ```

     La sortie suivante s'affiche :

```
OSGi Service Name: myShardListener
ObjectGrid Name MapSet Name Server Name        Current Ranking
--------------- ----------- -----------        ---------------
Grid            MapSet      collocatedServer   1

CWXSI0040I: The command osgiCurrent has completed successfully.
```

- Identifiez le classement en cours de tous les services. Entrez la commande suivante pour interroger le classement de service en cours utilisé pour tous les services utilisé par la grille ObjectGrid `Grid` et le groupe de mappes `MapSet`.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour interroger les classements de tous les services.
     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
     ```

     La sortie suivante s'affiche :
     ```
     OSGi Service Name    Current Ranking ObjectGrid Name MapSet Name Server Name
     -----------------    --------------- --------------- ----------- -----------
     myProtoBufSerializer 1               Grid            MapSet      collocatedServer
     myShardListener      1               Grid            MapSet      collocatedServer

     CWXSI0040I: The command osgiCurrent has completed successfully.
     ```
- Identifiez tous les classements disponibles d'un service. Entrez la commande suivante pour interroger les classements du service myShardListener.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour interroger tous les classements d'un service.
     ```
     ./xscmd.sh -c osgiAll -sn myShardListener
     ```

     La sortie suivante s'affiche :
     ```
     Server: collocatedServer
        OSGi Service Name Available Rankings
        ----------------- ------------------
        myShardListener 1

        Summary - All servers have the same service rankings.

     CWXSI0040I: The command osgiAll has completed successfully.
     ```

     La sortie est regroupée en fonction du serveur. Dans cet exemple, seul le serveur collocatedServer existe.
- Identifiez tous les classements de services disponibles. Entrez la commande suivante pour rechercher tous les classements de tous les services.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour rechercher tous les classements de services disponibles.
     ```
     ./xscmd.sh -c osgiAll
     ```

     La sortie suivante s'affiche :
     ```
     Server: collocatedServer
        OSGi Service Name    Available Rankings
        -----------------    ------------------
        myProtoBufSerializer 1
        myShardListener 1

     Summary - All servers have the same service rankings.
     ```

- Installez et démarrez la version 2 de l'ensemble de plug-in. Dans la console OSGi du serveur, installez un nouvel ensemble contenant une nouvelle version de la classe Order et le plug-in MapSerializerPlugin. Voir Leçon 2.4 : Installation de Google Protocol et exemples d'ensembles de plug-in pour plus d'informations sur l'installation de l'ensemble `ProtoBufSamplePlugins-2.0.0.jar`.
  1. Après l'installation, démarrez le nouvel ensemble. Les services du nouvel ensemble sont disponibles, mais ils ne sont pas encore utilisés par le serveur eXtreme Scale. Vous devez exécuter une demande de mise à jour de service pour pouvoir utiliser un service avec une version donnée.
- Maintenant, lorsque vous interrogez de nouveau tous les classements de services disponibles, le classement de service 2 est ajouté dans la sortie.
  1. Accédez au répertoire suivant :
     ```
     cd wxs_home/bin
     ```
  2. Entrez la commande suivante pour rechercher tous les classements de services disponibles.
     ```
     ./xscmd.sh -c osgiAll
     ```

     La sortie suivante s'affiche :
     ```
     Server: collocatedServer
        OSGi Service Name    Available Rankings
        -----------------    ------------------
        myProtoBufSerializer 1, 2
        myShardListener      1, 2

     Summary - All servers have the same service rankings.
     ```

Point de contrôle de la leçon :

Dans ce tutoriel, vous avez interrogé des classements de services spécifiques et les classements de services disponibles. Vous avez également affiché le classement de service d'un nouvel ensemble que vous avez installé et démarré.

Leçon 4.2 : Déterminer si des classements de services spécifiques sont disponibles

Suivez cette leçon pour déterminer si des classements de services spécifiques sont disponibles pour les noms de service que vous spécifiez.

1. Entrez la commande suivante pour déterminer si le service myShardListener, avec le classement de service 2 et le service myProtoBufSerializer avec le classement de service 2 sont disponibles. La liste des classements de service est transmise à l'aide de l'option `-sr`.
   a. Accédez au répertoire suivant :
      ```
      cd wxs_home/bin
      ```
   b. Entrez la commande suivante pour déterminer si les services sont disponibles :
      ```
      ./xscmd.sh -c osgiCheck -sr "myShardListener;2,myProtoBufSerializer;2"
      ```

      La sortie suivante s'affiche :
      ```
      CWXSI0040I: The command osgiCheck has completed successfully.
      ```
2. Entrez la commande suivante pour déterminer si le service myShardListener, avec le classement de service 2 et le service myProtoBufSerializer avec le classement de service 3 sont disponibles.
   a. Accédez au répertoire suivant :
      ```
      cd wxs_home/bin
      ```

b. Entrez la commande suivante pour déterminer si les services sont disponibles :

```
./xscmd.sh -c osgiCheck -sr "myShardListener;2,myProtoBufSerializer;3"
```

La sortie suivante s'affiche :

```
Server OSGi Service Unavailable Rankings
------          ------------        --------------------
collocatedServer myProtoBufSerializer 3
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez spécifié les services myShardListener et myProtoBufSerializer, ainsi que des classements de services spécifiques pour déterminer si ces classements étaient disponibles.

Leçon 4.3 : Mise à jour des classements de services

Suivez cette leçon pour mettre à jour les classements de services en cours que vous avez interrogés.

1. Mettez à jour les classements de services myShardListener et myProtoBufSerializer vers le classement de services 2. La liste des classements de services est envoyée en utilisant l'option `-sr`.
   a. Accédez au répertoire suivant :

   ```
   cd wxs_home/bin
   ```

   b. Entrez la commande suivante pour mettre à jour les classements de services :

   ```
   ./xscmd.sh -c osgiUpdate -g Grid -ms MapSet -sr "myShardListener;2,myProtoBufSerializer;2"
   ```

   La sortie suivante s'affiche :

   ```
   Update succeeded for the following service rankings:
   Service Ranking
   -------             -------
   myProtoBufSerializer 2
   myShardListener 2

   CWXSI0040I: The command osgiUpdate has completed successfully.
   ```

   La sortie suivante s'affiche sur la console OSGi :

   ```
   SystemOut O MyShardListener@326505334(version=2.0.0) order
   com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order$Builder@
   22342234(34) updated
   ```

   Notez que le service MyShardListener est maintenant au niveau de version 2.0.0 qui a le classement de service 2.
2. Exécutez la commande **xscmd** pour interroger le classement de services en cours utilisé pour tous les services utilisés par la grille ObjectGrid `Grid` et le groupe de mappes `MapSet`.
   a. Accédez au répertoire suivant :

   ```
   cd wxs_home/bin
   ```

   b. Entrez la commande suivante pour interroger les classements de tous les services qui sont utilisés par `Grid` et `MapSet`:

   ```
   ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
   ```

   La sortie suivante s'affiche :

```
OSGi Service Name    Current Ranking ObjectGrid Name MapSet Name Server Name
-----------------    --------------- --------------- ----------- -----------
myProtoBufSerializer 2 Grid MapSet collocatedServer
myShardListener 2 Grid MapSet collocatedServer

CWXSI0040I: The command osgiCurrent has completed successfully.
```

Point de contrôle de la leçon :

Dans cette leçon, vous avez mis à jour les classements des services myShardListener et myProtoBufSerializer.

.

Chapitre 4. Installation

WebSphere eXtreme Scale est une grille de données en mémoire que vous pouvez utiliser pour partitionner, répliquer et gérer dynamiquement des données d'application et une logique métier entre plusieurs serveurs. Une fois que vous avez déterminé les rôles et exigences de votre déploiement, installez eXtreme Scale sur votre système.

Avant de commencer

- Avant de commencer l'installation, vous devez connaître les architectures de mise en cache, l'intégration de cache et de base de données, la sérialisation, l'évolutivité et la disponibilité WebSphere eXtreme Scale. Voir Présentation du produit pour plus d'informations.
- Planifiez le déploiement WebSphere eXtreme Scale. Pour lus d'informations sur les différentes topologies de cache, le dimensionnement, etc., voir Chapitre 2, «Planification», à la page 19.
- Vérifiez que votre environnement remplit les conditions requises pour installer eXtreme Scale. Pour plus d'informations, voir «Configurations matérielle et logicielle requises», à la page 65.
- Pour plus d'informations sur les environnements et d'autres conditions requises, voir «Planification pour l'installation», à la page 187.
- Si vous installez une mise à niveau sur une version précédente de WebSphere eXtreme Scale, suivez les étapes décrites dans «Mise à jour des serveurs eXtreme Scale», à la page 255.

Présentation de l'installation

Vous pouvez installer WebSphere eXtreme Scale dans un environnement autonome ou WebSphere Application Server. Pour pouvoir installer WebSphere eXtreme Scale, vous devez d'abord installer IBM Installation Manager et vous procurer les fichiers produit. Après avoir installé Installation Manager et défini l'accès aux référentiels de produit appropriés, vous pouvez choisir de procéder à l'installation complète ou à l'installation client de WebSphere eXtreme Scale dans un environnement autonome ou d'installer le produit dans un environnement WebSphere Application Server. Vous pouvez également installer WebSphere eXtreme Scale comme grille de données en mémoire pour une utilisation avec des applications .NET et Java avec des fonctions XTP (extreme transaction processing). Toutefois, seule une installation client dans un environnement autonome est prise en charge pour le moment. Pour plus d'informations, voir «A propos de l'installation de WebSphere eXtreme Scale dans un environnement .NET», à la page 201.

Java

IBM Installation Manager

Installation Manager est un programme d'installation autonome qui peut exploiter des référentiels de fichiers plats de logiciels distants ou locaux pour installer, modifier ou mettre à jour de nouveaux produits pour WebSphere eXtreme Scale. Il détecte et affiche les packages disponibles (produits, groupes de correctifs, correctifs provisoires, etc.), vérifie les prérequis et interdépendances et installe les packages sélectionnés. Installation Manager permet également de désinstaller en toute simplicité les modules installés.

Présentation d'IBM Installation Manager : IBM Installation Manager est un outil d'installation et de mise à jour de logiciels polyvalente qui fonctionne sur divers systèmes informatiques. Installation Manager peut être appelé via une interface graphique ou une interface de ligne de commande. Vous pouvez aussi créer des fichiers de réponses au format XML et les utiliser pour piloter l'exécution de tâches Installation Manager en mode silencieux.

Pour plus d'informations sur l'utilisation d'Installation Manager, voir le centre de documentation IBM Installation Manager Version 1.5.

Packages et groupes de packages : chaque produit logiciel qui peut être installé avec Installation Manager s'appelle un package. Un package installé a un niveau de produit et un emplacement d'installation. Un groupe de packages est constitué de tous les produits qui sont installés à un même emplacement.

Modes Installation Manager : IBM Installation Manager peut être installé dans l'un des trois modes suivants :

- En mode administrateur, Installation Manager est installé à partir d'un ID d'administrateur ou d'utilisateur root et peut être appelé par tout administrateur ou utilisateur root.
- En mode nonAdmin (appelé également Mode utilisateur), Installation Manager peut être appelé uniquement par l'utilisateur qui l'a installé.
- UNIX Linux En mode groupe, Installation Manager peut être appelé par tout ID utilisateur connecté au groupe par défaut de l'utilisateur qui l'a installé.

  Cela ne signifie pas que deux personnes peuvent utiliser la même instance d'IBM Installation Manager en même temps.

Nombre d'instances d'Installation Manager dont vous avez besoin : Vous devez exécuter Installation Manager seulement sur les systèmes où vous installez ou où vous mettez à jour le code de produits. Vous n'avez normalement besoin que d'un seul gestionnaire d'installation Installation Manager sur un système car il peut faire le suivi d'un nombre quelconque d'installations de produit.

Installation d'Installation Manager : Lorsque le kit d'installation est disponible sur votre système, vous pouvez installer Installation Manager. Installation Manager se compose d'un ensemble de fichiers binaires qui sont copiés depuis le kit d'installation et d'un ensemble de données d'exécution qui décrivent les produits qui ont été installés avec cette instance particulière d'Installation Manager. Avant d'installer Installation Manager, vous devez déterminer le mode de fonctionnement d'Installation Manager et l'emplacement des fichiers binaires et des données d'exécution, appelées données d'agent ou appdata. Ensuite, vous exécutez la commande d'installation d'Installation Manager depuis l'ID utilisateur approprié pour installer Installation Manager.

Accès aux référentiels du produit : tous les éléments qui seront installés avec IBM Installation Manager sont stockés dans des référentiels de fichiers à plat. Chaque référentiel contient des objets de programme et des métadonnées pour un ou plusieurs packages, c'est-à-dire des produits logiciels à un niveau particulier. Les référentiels peuvent aussi contenir des éléments relatifs à la maintenance du produit, tels que des groupes de correctifs et des correctifs temporaires. Lorsque vous installez un nouveau produit, vous pouvez choisir parmi tous les niveaux de produits disponibles dans un des référentiels accessibles.

Installation du produit : après avoir installé Installation Manager et accédé à tous les référentiels du produit nécessaires, vous pouvez utiliser l'interface graphique,

les lignes de commande ou les fichiers de réponses Installation Manager pour exécuter des installations du produit. Lorsque vous installez un produit, vous fournissez le nom du package, éventuellement le niveau du produit à installer, l'emplacement du produit et d'autres propriétés facultatives. Par exemple, certains produits ont des fonctions facultatives que vous pouvez sélectionner au moment de l'installation ou une liste des modules linguistiques pris en charge que vous pouvez choisir.

Utilisation des produits installés : Vous pouvez utiliser des commandes d'Installation Manager pour obtenir la liste des produits installés et des niveaux des produits. Vous pouvez aussi obtenir ces informations pour les copies installées des produits WebSphere eXtreme Scale en lançant la commande **versionInfo** à partir du système de fichiers du produit. Vous pouvez utiliser les commandes d'Installation Manager ou des fichiers de réponses pour installer un nouveau niveau de produit, pour rétrograder vers un niveau antérieur ou pour modifier le produit en ajoutant ou en supprimant des fonctions facultatives ou des modules linguistiques.

Utilisation d'IBM Packaging Utility : avec Packaging Utility, vous pouvez créer et gérer les packages des référentiels d'installation. Vous pouvez copier plusieurs packages vers un référentiel ou plusieurs disques d'un produit vers un référentiel. Vous pouvez copier les packages depuisPassport Advantage vers un référentiel, par exemple. Pour plus d'informations sur Packaging Utility, voir le centre de documentation IBM Installation Manager Version 1.5.

Restrictions :

- Windows Si un utilisateur non-administrateur installe WebSphere eXtreme Scale version 8.5 sur un système d'exploitation Windows Vista, Windows 7 ou Windows Server 2008 dans le répertoire `Program Files` ou `Program Files (x86)` avec le contrôle de compte utilisateur activé, WebSphere eXtreme Scale ne fonctionnera pas correctement.

  Le contrôle de compte d'utilisateur est un mécanisme de contrôle d'accès qui permet aux utilisateurs non-administrateurs d'installer des produits logiciels dans le répertoire `Program Files` ou `Program Files (x86)` ; il interdit cependant tout accès en écriture à ce répertoire lorsque l'installation est terminée. WebSphere eXtreme Scale requiert un accès en écriture dans le répertoire *racine_serveur_applications* pour fonctionner correctement.

  Pour résoudre cet incident, effectuez l'une des opérations suivantes :

  – Installez WebSphere eXtreme Scale dans un répertoire autre que `Program Files` ou `Program Files (x86)`.

    Par exemple :

    ```
    C:\IBM\WebSphere\AppServer
    ```

  – Désactivez le contrôle de compte d'utilisateur.
- Lorsque vous installez un produit en utilisant Installation Manager avec des référentiels locaux, l'installation dure longtemps si vous utilisez le fichier de référentiel compressé directement sans l'extraire.

  Avant d'installer un produit avec des référentiels locaux, extrayez le fichier de référentiel compressé dans un emplacement du système local avant d'utiliser Installation Manager pour y accéder.
- Le mode Console Installation Manager, qui est inclus dans Installation Manager version 1.4.3 et les versions ultérieures, ne fonctionne pas avec les offres WebSphere eXtreme Scale version 8.5 sur les systèmes non z/OS.

Important : Ne transférez pas le contenu d'un référentiel en mode non binaire et ne convertissez aucun contenu à l'extraction.

Conseil : Bien que la plupart de toutes les instructions de cette section de centre de documentation s'appliquent aux versions antérieures d'IBM Installation Manager,les informations dans ce document sont optimisées pour les utilisateurs qui ont installé Installation Manager Version 1.5 ou une version suivante ou effectuer une mise à niveau vers cette version ou une version suivante.

Java

Offres de produit disponibles dans Installation Manager

Après avoir installé Installation Manager et accédé à tous les référentiels du produit nécessaires, Installation Manager répertorie les offres de produit disponibles. Lorsque vous pointez sur le référentiel de produit, vous devez voir apparaître les offres suivantes :

- WebSphere eXtreme Scale en environnement autonome
- WebSphere eXtreme Scale Client en environnement autonome
- WebSphere eXtreme Scale for WebSphere Application Server version 7.0
- WebSphere eXtreme Scale for WebSphere Application Server version 8.0
- WebSphere eXtreme Scale Client for WebSphere Application Server version 7.0
- WebSphere eXtreme Scale Client for WebSphere Application Server version 8.0

Important : Vous devez pointer sur le référentiel de produit WebSphere Application Server si vous souhaitez installer l'offre de produit WebSphere eXtreme Scale pour WebSphere Application Server. Ces produits sont installés comme deux produits distincts, mais ils peuvent être installés simultanément dans Installation Manager.

Installation de WebSphere eXtreme Scale dans un environnement autonome ou WebSphere Application Server

Si vous choisissez d'installer WebSphere eXtreme Scale dans un environnement autonome, vous pouvez exécuter à la fois des serveurs de catalogue et des serveurs de conteneur. Sur les serveurs qui exécutent des applications client qui accèdent à la grille de données, vous pouvez utiliser une installation client uniquement. Utilisez l'installation serveur ou l'installation serveur et client sur des noeuds qui exécutent des serveurs de catalogue ou des serveurs de conteneurs. Si vous installez WebSphere eXtreme Scale pour l'utiliser avec des applications .NET avec des fonctions XTP (extreme transaction processing), seule une installation client dans un environnement autonome est prise en charge actuellement. Pour plus d'informations, voir «A propos de l'installation de WebSphere eXtreme Scale dans un environnement .NET», à la page 201.

- Java **Installation complète (client et serveur) :**
  - Lorsque vous effectuez l'installation sur WebSphere Application Server, vous pouvez choisir d'installer le client uniquement ou le serveur et le client.
  - Lorsque vous effectuez l'installation dans un environnement autonome, vous pouvez installer à la fois le client et le serveur.
- Java .NET **Installation client :**

Vous pouvez utiliser l'installation client uniquement sur les noeuds qui exécutent les applications client. Si vous installez WebSphere eXtreme Scale pour une utilisation avec des applications .NET, seule une installation autonome est prise en charge pour le moment.

- Java Environnement **WebSphere Application Server :**

  En installant WebSphere eXtreme Scale sur les noeuds de votre environnement WebSphere Application Server, vous pouvez démarrer automatiquement les serveurs de catalogue et les serveurs de conteneur dans la même cellule que votre gestionnaire de déploiement et d'autres serveurs d'applications.

- Java **Environnement autonome :**

  Dans une installation autonome, vous installez WebSphere eXtreme Scale dans un environnement ne disposant pas de WebSphere Application Server. Avec un environnement autonome, vous configurez et démarrez manuellement le serveur de catalogue et les processus serveur de conteneur.

Planification pour l'installation

Avant d'installer le produit, vous devez tenir compte de votre environnement.

Topologies d'installation

Avec WebSphere eXtreme Scale, vous pouvez créer des topologies d'installation qui contiennent des serveurs autonomes, WebSphere Application Server ou les deux.

Noeud de développement

Le scénario d'installation le plus simple consiste à créer un noeud de développement. Dans ce scénario, vous installez le client et le serveur WebSphere eXtreme Scale une fois sur le noeud sur lequel vous voulez développer l'application.

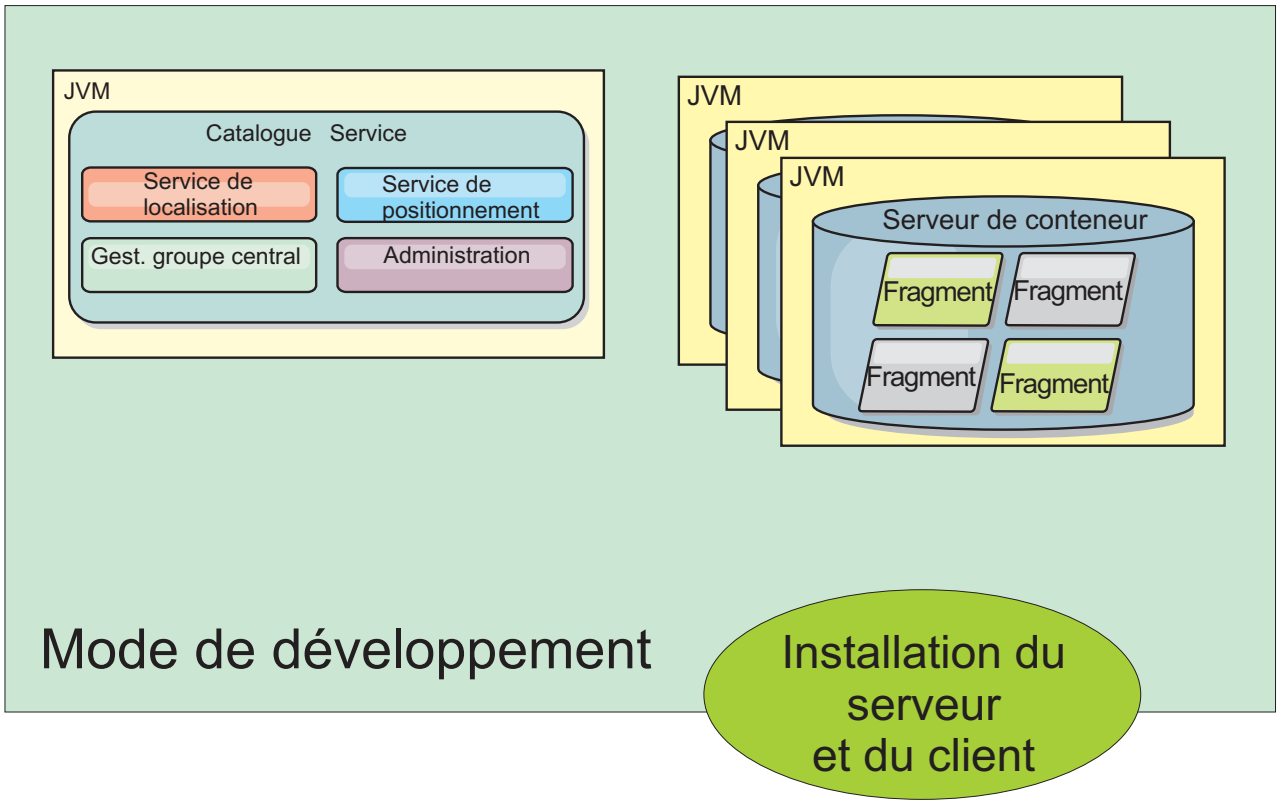


Figure 26. Noeud de développement

Après avoir effectué l'installation sur le noeud de développement, vous pouvez configurer l'environnement de développement et écrire vos applications.

Topologie autonome

Une topologie autonome est constituée de serveurs que vous n'exécutez pas sur WebSphere Application Server. Vous pouvez créez un grand nombre de topologies autonomes différentes, mais la topologie suivante est incluse comme exemple. Dans cette topologie, deux centres de données sont présents. Dans chaque centre de données, les installations complètes WebSphere eXtreme Scale (client et serveur) et les installations uniquement client sont installées sur les serveurs physiques. Les installations uniquement client se trouvent sur les noeuds qui exécutent les applications Web qui utilisent la grille de données. Ces noeuds n'exécutant pas de catalogue ou de serveurs de conteneur, il n'est pas nécessaire d'installer le serveur. Une liaison multimaître connecte deux domaines de service de catalogue dans la configuration. La liaison multimaître permet d'exécuter la réplication entre les fragments dans les serveurs de conteneur dans les différents centres de données.

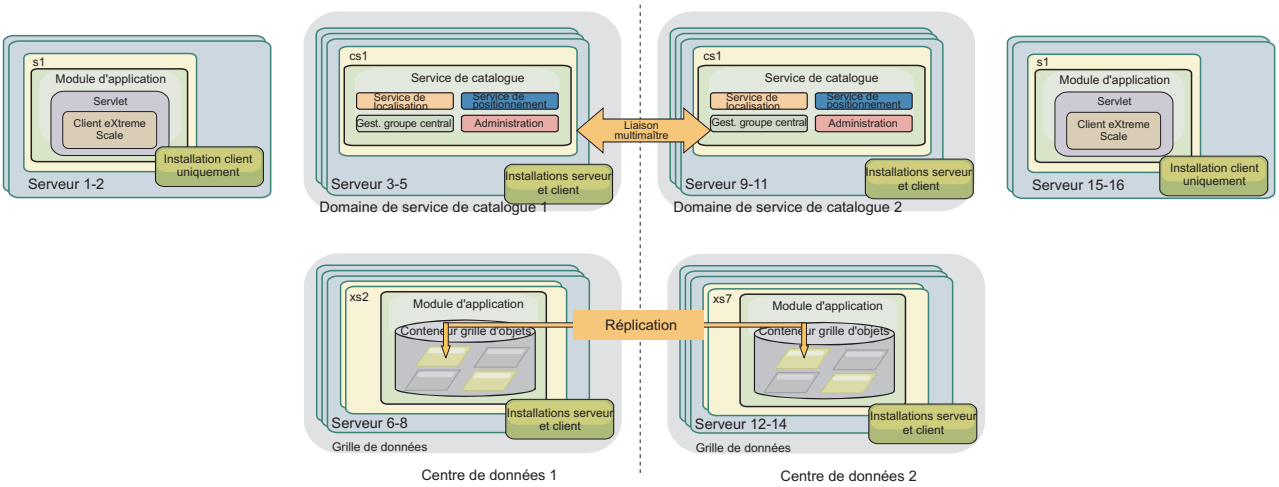


Figure 27. Topologie autonome avec deux centres de données

Avantages de l'utilisation d'une topologie autonome :

- Options d'intégration souple qui peuvent être intégrées aux infrastructure et bibliothèques des fournisseurs.
- Moindre encombrement qu'une topologie WebSphere Application Server.
- Moins d'exigences de licence qu'une topologie WebSphere Application Server.
- Options JRE (Expanded Java Runtime Environment).

Topologie WebSphere Application Server

Vous pouvez aussi créer une installation qui s'exécute entièrement dans une cellule WebSphere Application Server. Les clients, les serveurs de catalogue et les serveurs de conteneur ont chacun un cluster associé. Les noeuds qui exécutent l'application ont l'installation uniquement client. Les autres noeuds ont l'installation client et serveur.

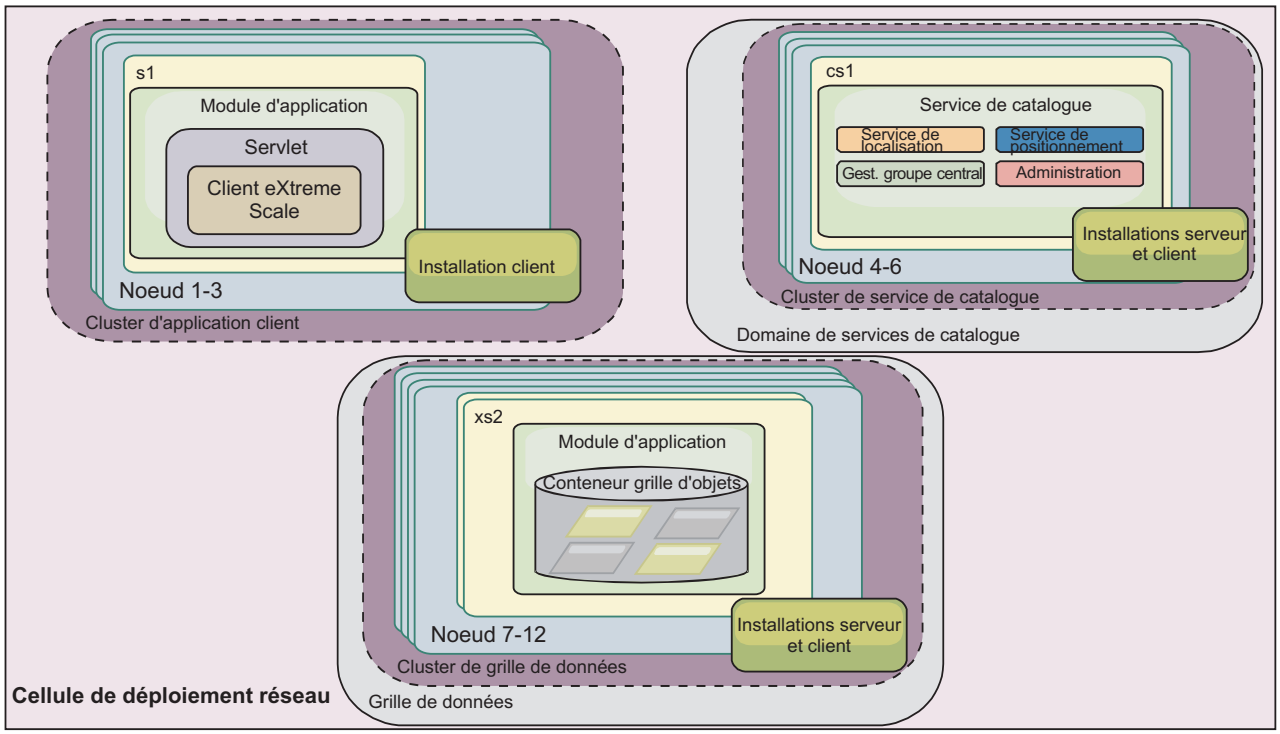


Figure 28. Exemple de topologie WebSphere Application Server

Avantages de la topologie WebSphere Application Server.

- Administration et configuration centralisées et cohérentes.
- Intégration de la sécurité.
- Intégration de l'application Java EE.
- Intégration PMI (Performance monitoring infrastructure).
- Intégration aux composants suivants WebSphere Application Server : cache OpenJPA L2, cache dynamique et persistance de session HTTP.

Topologie mixte

Vous pouvez créer une topologie mixte qui contient des serveurs WebSphere Application Server et autonomes. Dans l'exemple suivant, les applications client s'exécutent dans la cellule WebSphere Application Server, alors que les serveurs de catalogue et de conteneur s'exécutent en mode autonome.

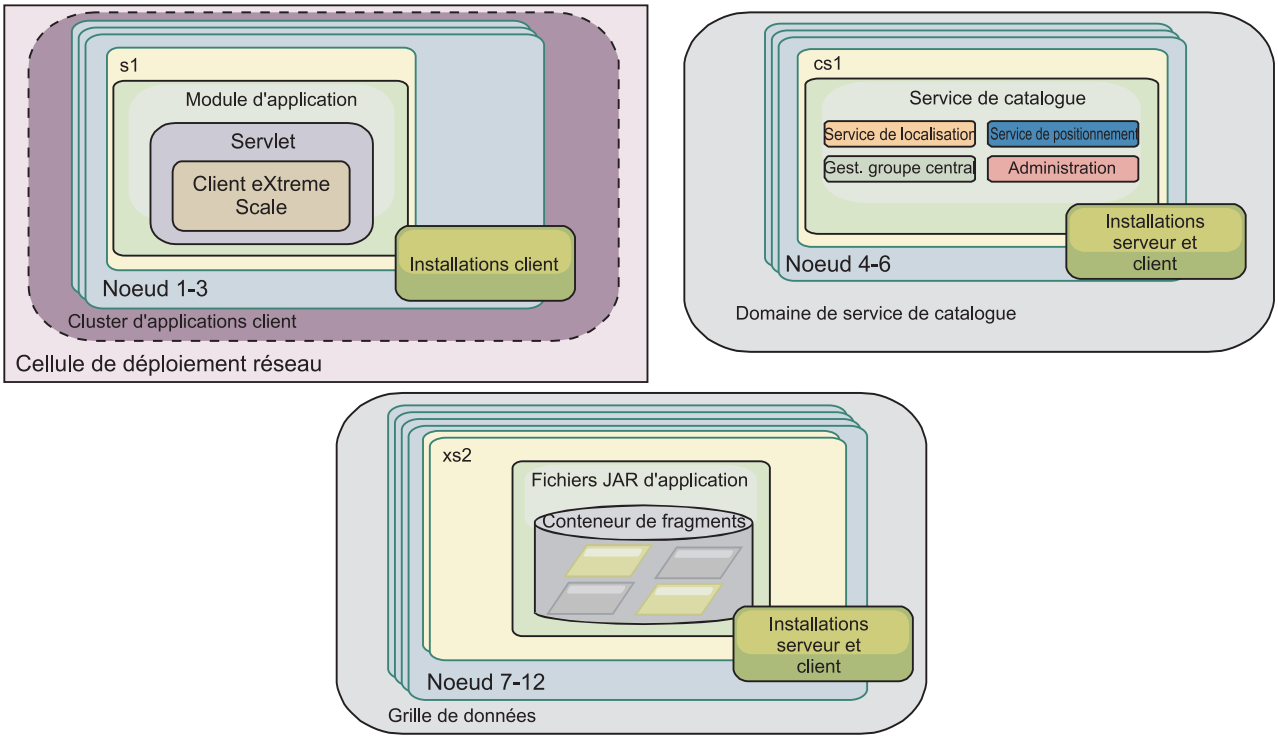


Figure 29. Exemple de topologie mixtes

Configurations matérielle et logicielle requises

Vue d'ensemble des conditions requises en termes de matériels et de systèmes d'exploitation. Bien que vous ne soyez pas tenu d'utiliser un niveau spécifique de matériel ou de système d'exploitation pour WebSphere eXtreme Scale, nous n'en fournissons pas moins sur le site de support du produit (page Configuration requise) une liste détaillée des matériels et logiciels officiellement pris en charge. En cas de conflit entre les informations présentées par le Centre de documentation et celles figurant sur cette page, les informations fournies par le site Web prévalent. Les conditions préalables répertoriées par le Centre de documentation sont fournies à titre informatif uniquement.

Voir la page Configuration système requise pour connaître les configurations matérielles et logicielles officielles.

Vous pouvez installer et déployer le produit dans les environnements Java EE et Java SE. Vous pouvez également regrouper le composant client avec les applications Java EE directement sans les intégrer à WebSphere Application Server.

Configuration matérielle

WebSphere eXtreme Scale ne requiert pas la présence d'un niveau spécifique de matériel. La configuration matérielle requise dépend du matériel pris en charge pour l'installation de Java Platform, Standard Edition que vous utilisez pour exécuter WebSphere eXtreme Scale. Si vous utilisez eXtreme Scale avec WebSphere Application Server ou une autre implémentation Java Platform, Enterprise Edition, la configuration matérielle requise par ces plateformes est suffisante pour WebSphere eXtreme Scale.

Configuration requise en matière de système d'exploitation

.NET 8.6+ Pour plus d'informations sur la configuration requise pour un environnement de client .NET, voir «Remarques relatives à Microsoft .NET», à la page 67.

Java Chaque implémentation Java SE et Java EE requiert un niveau différent du système d'exploitation ou des correctifs pour les problèmes identifiés lors du test de l'implémentation Java. Les niveaux nécessaires à ces implémentations sont suffisants pour eXtreme Scale.

Configuration requise pour Installation Manager

Avant de pourvoir installer WebSphere eXtreme Scale, vous devez installer Installation Manager. Vous pouvez installer Installation Manager en utilisant le support du produit, en utilisant un fichier obtenu à partir du site Passport Advantage ou en utilisant un fichier contenant la version la plus récente d'Installation Manager disponible sur le site Web de téléchargement d'IBM Installation Manager. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Navigateurs Web requis

La console Web prend en charge les navigateurs Web suivants :

- Mozilla Firefox, version 3.5.x et versions ultérieures
- Microsoft Internet Explorer, version 7 et versions ultérieures

Configuration requise pour WebSphere Application Server

8.6+

- WebSphere Application Server version 7.0.0.21 ou version suivante
- WebSphere Application Server version 8.0.0.2 ou version suivante

Pour plus d'informations, consultez la section Recommended fixes for WebSphere Application Server.

Java requis

8.6+ Les autres implémentations Java EE peuvent utiliser la phase d'exécution d'eXtreme Scale en tant qu'instance locale ou client pour les serveurs eXtreme Scale. Pour implémenter Java SE, vous devez utiliser la version 6 ou une version suivante.

ID des offres de produit WebSphere eXtreme Scale

Java

Lorsque vous installez des mises à jour de produit ou que vous annulez des correctifs, vous devez spécifier l'ID d'offre à partir de la ligne de commande. Utilisez le tableau ci-dessous pour identifier l'offre de produit.

Tableau 3. ID d'offre pour les produits WebSphere eXtreme Scale

Nom du produit	ID d'offre
WebSphere eXtreme Scale dans un environnement autonome	`com.ibm.websphere.WXS.v86`
WebSphere eXtreme Scale Client dans un environnement autonome	`com.ibm.websphere.WXSCLIENT.v86`

Tableau 3. ID d'offre pour les produits WebSphere eXtreme Scale (suite)

Nom du produit	ID d'offre
WebSphere eXtreme Scale for WebSphere Application Server version 7	`com.ibm.websphere.WXS.was7.v86`
WebSphere eXtreme Scale for WebSphere Application Server version 8	`com.ibm.websphere.WXS.was8.v86`
WebSphere eXtreme Scale Client for WebSphere Application Server version 7	`com.ibm.websphere.WXSCLIENT.was7.v86`
WebSphere eXtreme Scale Client for WebSphere Application Server version 8	`com.ibm.websphere.WXSCLIENT.was8.v86`

Java SE : points à prendre en considération

Java

WebSphere eXtreme Scale requires Java SE 6, or Java SE 7. En règle générale, les nouvelles versions de Java SE ont des fonctions plus efficaces et sont plus performantes.

Versions prises en charge

Vous pouvez utiliser WebSphere eXtreme Scale avec Java SE 6 et Java SE 7. La version que vous utilisez doit être prise en charge par le fournisseur Java Runtime Environment (JRE). Si vous voulez utiliser SSL (Secure Sockets Layer), vous devez utiliser un environnement d'exécution IBM.

IBM Runtime Environment, Java Technology Edition Version 6 et version 7 sont pris en charge pour l'utilisation avec le produit. La version 6, édition de service 9, groupe de correctifs 2, est un environnement d'exécution Java pris en charge qui est installé dans le cadre des installations autonomes de WebSphere eXtreme Scale et de WebSphere eXtreme Scale Client dans le répertoire *racine_install_wxs*`/java` et qui peut être utilisé à la fois par les clients et par les serveurs. Si vous installez WebSphere eXtreme Scale dans WebSphere Application Server, vous pouvez utiliser l'environnement JRE inclus dans l'installation WebSphere Application Server. Pour la console Web, vous devez utiliser IBM Runtime Environment, Java Technology Edition Version 6 Service Release 7 et les éditions de service ultérieures uniquement.

WebSphere eXtreme Scale tire parti de la fonctionnalité Version 6 et Version 7 lorsqu'elle devient disponible. Généralement, les nouvelles versions Java Development Kit (JDK) et Java SE sont plus performantes et ont une fonctionnalité plus efficace.

Pour plus d'informations, voir Logiciels pris en charge.

Fonctions WebSphere eXtreme Scale dépendantes de Java SE

Tableau 4. Fonctions nécessitant Java SE 6 et Java SE 7.

WebSphere eXtreme Scale utilise la fonctionnalité introduite dans Java SE 6 pour fournir les fonctions suivantes du produit.

Caractéristique	**Prise en charge dans Java SE 5 et éditions de service ultérieures** **Remarque :** Java SE 5 n'est pas pris en charge dans WebSphere eXtreme Scale Version 8.6	**Prise en charge dans Java SE version 6 , version 7 et éditions de service ultérieures**
Annotations d'API EntityManager (facultatif : vous pouvez également utiliser des fichiers XML)	X	X
Java Persistence API (JPA) : chargeur JPA, chargeur de client JPA et programme de mise à jour basé sur le temps JPA	X	X
L'expulsion basée sur la mémoire (utilise MemoryPoolMXBean)	X	X
Agents d'instrumentation : • `wxssizeagent.jar` : augmente la précision des mesures de mappe d'octets utilisées. • `ogagent.jar` : augmente la performance des entités d'accès aux zones.	X	X
Console Web de surveillance		X

Mise à niveau du kit JDK dans WebSphere eXtreme Scale

Questions courantes sur le processus de mise à niveau des éditions de WebSphere eXtreme Scale dans les environnements autonomes et WebSphere Application Server :

- Comment mettre à niveau le JDK qui est inclus avec WebSphere eXtreme Scale for WebSphere Application Server?

  Vous devez utiliser le processus de mise à niveau JDK que fournit WebSphere Application Server. Pour plus d'informations, voir http://www-304.ibm.com/support/docview.wss?uid=swg21427178.
- Quelle version JDK utiliser avec WebSphere eXtreme Scale dans un environnement WebSphere Application Server ?

  Vous pouvez utiliser n'importe quel niveau de JDK qui est pris en charge par WebSphere Application Server pour la version prise en charge de WebSphere Application Server.

Java EE : points à prendre en considération

Java

Lors de la préparation de l'intégration WebSphere eXtreme Scale dans un environnement Java Platform, Enterprise Edition, tenez compte de certains éléments, tels que les versions, les options de configuration, les conditions requises et les limitations, le déploiement et la gestion des applications.

Exécuter des applications eXtreme Scale en environnement Java EE

Une application Java EE peut se connecter à une application eXtreme Scale distante. En outre, l'environnement WebSphere Application Server permet le démarrage d'un serveur eXtreme Scale lorsqu'une application démarre dans le serveur d'applications.

Si vous utilisez un fichier XML pour créer une instance ObjectGrid et que ce fichier XML se trouve dans le module du fichier EAR, accédez à ce fichier à l'aide de la méthode getClass().getClassLoader().getResource("META-INF/objGrid.xml") afin d'obtenir un objet URL permettant de créer une instance ObjectGrid. Dans l'appel à la méthode, remplacez le nom du fichier XML utilisé.

Vous pouvez utiliser des beans de démarrage pour que, à son démarrage, une application amorce une instance ObjectGrid et supprime cette instance lorsqu'elle s'arrête. Un bean de démarrage est un bean de session sans état avec un emplacement distant com.ibm.websphere.startupservice.AppStartUpHome et une interface distante com.ibm.websphere.startupservice.AppStartUp. L'interface distante possède deux méthodes : la méthode start et la méthode stop. Utilisez la méthode start pour amorcer l'instance et la méthode stop pour détruire l'instance. L'application utilise la méthode ObjectGridManager.getObjectGrid pour maintenir la référence à cette instance. Voir les informations relatives à l'accès à un objet ObjectGrid avec ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Utiliser des chargeurs de classes

Lorsque les modules d'application qui utilisent des chargeurs de classe différents partagent une instance ObjectGrid unique dans une application Java EE, vérifiez que les objets qui sont stockés dans eXtreme Scale et que les plug-in du produit se trouvent dans un chargeur commun dans l'application.

Gérer dans un servlet le cycle de vie des instances ObjectGrid

Pour gérer le cycle de vie d'une instance ObjectGrid dans un servlet, vous pouvez utiliser la méthode init pour créer l'instance et la méthode destroy pour supprimer l'instance. Si l'instance est mise en cache, elle est extraite et manipulée dans le code du servlet. Voir les informations relatives à l'accès à un objet ObjectGrid avec l'interface ObjectGridManager dans *Guide de programmation* pour plus d'informations.

Conventions relatives aux répertoires

Les conventions de répertoire suivantes sont utilisées dans toute la documentation pour faire référence à des répertoires spéciaux, tels que *wxs_install_root* et *wxs_home*. Vous pouvez accéder à ces répertoires pendant plusieurs scénarios différents, y compris lors de l'installation et de l'utilisation des outils de ligne de commande.

racine_install_wxs
Le répertoire *wxs_install_root* est le répertoire racine où sont installés les

fichiers du produit WebSphere eXtreme Scale. Le répertoire *wxs_install_root* peut être le répertoire dans lequel l'archive d'évaluation est extraite ou depuis lequel le produit est installé WebSphere eXtreme Scale.

- Exemple où la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`

- Exemple où WebSphere eXtreme Scale est installé dans un répertoire autonome :

  UNIX **Exemple :** `/opt/IBM/eXtremeScale`

  Windows **Exemple :** `C:\Program Files\IBM\WebSphere\eXtremeScale`

- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer`

wxs_home

Le répertoire *wxs_home* est le répertoire racine du produit, des bibliothèques, des exemples et des composants WebSphere eXtreme Scale. Ce répertoire est identique au répertoire *wxs_install_root* lorsque l'archive d'évaluation est extraite. Pour les installations autonomes, le répertoire *wxs_home* est le sous-répertoire `ObjectGrid` du répertoire *wxs_install_root*. Pour les installations qui sont intégrées à WebSphere Application Server, ce répertoire est le répertoire `optionalLibraries/ObjectGrid` du répertoire *wxs_install_root*.

- Exemple lorsque la version d'essai a été extraite :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale`

- Exemple où WebSphere eXtreme Scale est installé dans un répertoire autonome :

  UNIX **Exemple :** `/opt/IBM/eXtremeScale/ObjectGrid`

  Windows **Exemple :** *racine_install_wxs*`\ObjectGrid`

- Exemple lorsque WebSphere eXtreme Scale est intégré à WebSphere Application Server :

  Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid`

was_root

Le répertoire *was_root* est le répertoire racine d'une installation WebSphere Application Server :

Exemple : `/opt/IBM/WebSphere/AppServer`

.NET 8.6+ **net_client_home**

Le répertoire *net_client* est le répertoire racine d'une installation client .NET.

Exemple : `C:\Program Files\IBM\WebSphere\eXtreme Scale .NET Client`

restservice_home

Le répertoire *restservice_home* est le répertoire dans lequel se trouvent les bibliothèques et les exemples du service de données REST d'WebSphere eXtreme Scale. Ce répertoire s'appelle `restservice` et il est le sous-répertoire de *wxs_home*.

- Exemple pour les déploiements autonomes :

  Exemple : `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`

  Exemple : *rép_base_wxs*`\restservice`

- Exemple pour les déploiements intégrés à WebSphere Application Server :

Exemple : `/opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice`

tomcat_root
Le répertoire *home_tomcat* est le répertoire racine de l'installation d'Apache Tomcat.

Exemple : `/opt/tomcat5.5`

wasce_root
wasce_root est le répertoire racine de l'installation WebSphere Application Server Community Edition.

Exemple : `/opt/IBM/WebSphere/AppServerCE`

java_home
Le répertoire *java_home* est le répertoire racine d'une installation de Java Runtime Environment Kit (JRE).

UNIX **Exemple :** `/opt/IBM/WebSphere/eXtremeScale/java`

Windows **Exemple :** *racine_install_wxs*`\java`

samples_home
samples_home est le répertoire dans lequel vous extrayez les exemples de fichiers qui sont utilisés pour les tutoriels.

UNIX **Exemple:** *rép_base_wxs*`/samples`

Windows **Exemple :** *rép_base_wxs*`\samples`

dvd_root
dvd_root est le répertoire racine du DVD qui contient le produit.

Exemple : `dvd_root/docs/`

equinox_root
Le répertoire *equinox_root* est le répertoire racine de l'installation de l'infrastructure OSGi Eclipse Equinox.

Exemple :`/opt/equinox`

user_home
Le répertoire *user_home* est l'emplacement de stockage des fichiers utilisateur, tels que les profils de sécurité.

Windows `c:\Documents and Settings\`*nom_util*

UNIX `/home/`*nom_util*

Fichiers d'exécution pour WebSphere eXtreme Scale intégré à WebSphere Application Server

Des fichiers archives Java (JAR) sont inclus dans l'installation. Vous pouvez voir les fichiers JAR qui sont inclus et l'emplacement dans lequel ils sont installés.

Tableau 5. Fichiers d'exécution pour WebSphere eXtreme Scale. Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
`wxsdynacache.jar`	Client et serveur	`lib`	Le fichier `wxsdynacache.jar` contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique.

Tableau 5. Fichiers d'exécution pour WebSphere eXtreme Scale (suite). Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
wsobjectgrid.jar	Local et client	lib	Le fichier wsobjectgrid.jar contient les exécutions local, client et serveur eXtreme Scale.
ogagent.jar	Local, client et serveur	lib	Le fichier ogagent.jar contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
ogsip.jar	Serveur	lib	Le fichier ogsip.jar contient l'exécution de gestion de sessions SIP (Session Initiation Protocol eXtreme Scale) qui est compatible avec WebSphere Application Server Version 7.0 et les versions suivantes.
sessionobjectgrid.jar	Client et serveur	lib	Le fichier sessionobjectgrid.jar contient l'exécution de la gestion de sessions HTTP eXtreme Scale.
sessionobjectgridsip.jar	Serveur	lib	Le fichier sessionobjectgridsip.jar contient l'exécution de gestion de sessions SIP (Session Initiation Protocol eXtreme Scale) eXtreme Scale qui est compatible avec WebSphere Application Server Version 7.0 et les versions suivantes.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server Version 7.0 ou une versions suivante. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement Java (JRE) Version 1.6 ou version suivante.
oghibernate-cache.jar	Client et serveur	optionalLibraries/ObjectGrid	Le fichier oghibernate-cache.jar contient le plug-in de cache L2eXtreme Scale pour JBoss Hibernate.
ogspring.jar	Local, client et serveur	optionalLibraries/ObjectGrid	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
xsadmin.jar	Utilitaire	optionalLibraries/ObjectGrid	Le fichier xsadmin.jar contient l'exemple d'utilitaire eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	optionalLibraries/ObjectGrid/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
wxshyperic.jar	Utilitaire	optionalLibraries/ObjectGrid/hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
restservice.ear	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
splicerlistener.jar	Utilitaire	optionalLibraries/ObjectGrid/session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.
splicer.jar	Utilitaire	optionalLibraries/ObjectGrid/legacy/session/lib	Le fichier splicer.jar contient l'utilitaire splicer Version 7.0 pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.
wxsra.rar	Utilitaire	optionalLibraries/ObjectGrid/wxsra.rar	Le fichier wxsra.rar contient l'adaptateur de ressources eXtreme Scale permettant de se connecter à la grille à l'aide d'une fabrique de connexions.

Tableau 6. Fichiers d'exécution pour WebSphere eXtreme Scale Client. Le tableau ci-après répertorie les fichiers JAR (archives Java) inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
wxsdynacache.jar	Client et serveur	lib	Le fichier wxsdynacache.jar contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique.
ogagent.jar	Local, client et serveur	lib	Le fichier ogagent.jar contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
ogsip.jar	Serveur	lib	Le fichier ogsip.jar contient l'exécution de gestion de sessions SIP (Session Initiation Protocol eXtreme Scale) qui est compatible avec WebSphere Application Server Version 7.0 et les versions suivantes.
sessionobjectgrid.jar	Client et serveur	lib	Le fichier sessionobjectgrid.jar contient l'exécution de la gestion de sessions HTTP eXtreme Scale.
sessionobjectgridsip.jar	Serveur	lib	Le fichier sessionobjectgridsip.jar contient l'exécution de gestion de sessions SIP (Session Initiation Protocol eXtreme Scale) eXtreme Scale qui est compatible avec WebSphere Application Server Version 7.0 et les versions suivantes.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server Version 7.0 ou une version suivante. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement Java (JRE) Version 1.6 ou version suivante.
oghibernate-cache.jar	Client et serveur	optionalLibraries/ObjectGrid	Le fichier oghibernate-cache.jar contient le plug-in de cache L2eXtreme Scale pour JBoss Hibernate.
ogspring.jar	Local, client et serveur	optionalLibraries/ObjectGrid	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
xsadmin.jar	Utilitaire	optionalLibraries/ObjectGrid	Le fichier xsadmin.jar contient l'exemple d'utilitaire eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	optionalLibraries/ObjectGrid/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
wxshyperic.jar	Utilitaire	optionalLibraries/ObjectGrid/hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
restservice.ear	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	optionalLibraries/ObjectGrid/restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
splicerlistener.jar	Utilitaire	optionalLibraries/ObjectGrid/session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.
splicer.jar	Utilitaire	optionalLibraries/ObjectGrid/legacy/session/lib	Le fichier splicer.jar contient l'utilitaire splicer Version 7.0 pour le filtre de gestionnaire de sessions HTTP eXtreme Scale.

Fichiers d'exécution de l'installation autonome WebSphere eXtreme Scale

Des fichiers JAR (Java) sont inclus dans l'installation. Vous pouvez voir les fichiers JAR qui sont inclus et leur emplacement d'installation.

Tableau 7. Fichiers d'exécution pour une installation complète de WebSphere eXtreme Scale. WebSphere eXtreme Scale s'appuie sur les processus ObjectGrid et les API associées. Le tableau ci-après répertorie les fichiers JAR inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
wxsdynacache.jar	Client et serveur	dynacache/lib	Le fichier wxsdynacache.jar contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique. Le fichier est automatiquement inclus dans l'environnement d'exécution du serveur lorsque vous utilisez les scripts fournis.
wxshyperic.jar	Utilitaire	hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
objectgrid.jar	Local, client et serveur	lib	Le fichier objectgrid.jar est un ensemble OSGi qui est utilisé par l'environnement d'exécution de serveur de Java SE 1.6 et version suivante. Le fichier est automatiquement inclus dans l'environnement d'exécution du serveur lorsque vous utilisez les scripts fournis.
ogagent.jar	Local, client et serveur	lib	Le fichier ogagent.jar contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
ogclient.jar	Local et client	lib	Le fichier ogclient.jar est un ensemble OSGi qui ne contient que les environnements d'exécution local et client. Vous pouvez utiliser ce fichier avec Java SE 1.6 et les versions suivantes.
ogspring.jar	Local, client et serveur	lib	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server ou une version ultérieure. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement d'exécution using Java (JRE) Version 1.6 ou une version ultérieure.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	lib/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
restservice.ear	Client	restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
xsadmin.jar	Utilitaire	samples	Le fichier xsadmin.jar contient l'exemple d'utilitaire d'administration eXtreme Scale.
sessionobjectgrid.jar	Client et serveur	session/lib	Le fichier sessionobjectgrid.jar contient l'exécution de la gestion de sessions HTTP eXtreme Scale.
splicerlistener.jar	Utilitaire	session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer du programme d'écoute de session HTTP eXtreme Scale Version 7.1 et versions suivantes.
xsgbean.jar	Serveur	wasce/lib	Le fichier xsgbean.jar contient le GBean pour l'intégration de serveurs eXtreme Scale dans des serveurs d'applications WebSphere Application Server Community Edition.
splicer.jar	Utilitaire	legacy/session/lib	Utilitaire splicer pour le filtre de gestionnaire de session HTTP WebSphere eXtreme Scale Version 7.0.
wxsra.rar	Client et serveur	session/lib	Le fichier wxsra.rar contient l'adaptateur de ressources eXtreme Scale permettant de se connecter à la grille à l'aide d'une fabrique de connexions.

Tableau 8. Fichiers d'exécution pour WebSphere eXtreme Scale Client. WebSphere eXtreme Scale Client s'appuie sur les processus ObjectGrid et les API associées. Le tableau ci-après répertorie les fichiers JAR inclus dans l'installation. L'emplacement d'installation est relatif au répertoire *rép_base_wxs* que vous choisissez lors de l'installation.

Nom de fichier	Environnement	Emplacement de l'installation	Description
wxsdynacache.jar	Client et serveur	dynacache/lib	Le fichier wxsdynacache.jar contient les classes nécessaires à utiliser avec le fournisseur de cache dynamique. Le fichier est automatiquement inclus dans l'environnement d'exécution du serveur lorsque vous utilisez les scripts fournis.
wxshyperic.jar	Utilitaire	hyperic/lib	Le plug-in de détection des serveurs WebSphere eXtreme Scale pour l'agent de surveillance SpringSource Hyperic.
ogagent.jar	Local, client et serveur	lib	Le fichier ogagent.jar contient les classes d'exécution requises pour exécuter l'agent d'instrumentation Java utilisé avec l'API EntityManager.
ogclient.jar	Local et client	lib	Le fichier ogclient.jar est un ensemble OSGi qui ne contient que les environnements d'exécution local et client. Vous pouvez utiliser ce fichier avec Java SE 1.6 et les versions suivantes.
ogspring.jar	Local, client et serveur	lib	Le fichier ogspring.jar contient les classes de support pour l'intégration de l'infrastructure SpringSource Spring.
wsogclient.jar	Local et client	lib	Fichier wsogclient.jar installé si vous utilisez un environnement qui contient WebSphere Application Server ou une version suivante. Ce fichier ne contient que les environnements d'exécution local et client.
wxssizeagent.jar	Local, client et serveur	lib	Le fichier wxssizeagent.jar est utilisé pour fournir des informations de dimensionnement d'entrée de cache plus précises lors de l'utilisation de l'environnement d'exécution Java (JRE) Version 1.6 ou une version suivante.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client et serveur	lib/endorsed	Cet ensemble de fichiers comprend l'exécution ORB (Object Request Broker) utilisée pour exécuter les applications dans les processus Java SE.
restservice.ear	Client	restservice/lib	Le fichier restservice.ear contient l'archive d'application d'entreprise de service de données eXtreme Scale REST pour les environnements WebSphere Application Server.
restservice.war	Client	restservice/lib	Le fichier restservice.war contient l'archive Web de service de données eXtreme Scale REST pour les serveurs d'applications provenant d'un autre fournisseur.
xsadmin.jar	Utilitaire	samples	Le fichier xsadmin.jar contient l'exemple d'utilitaire d'administration eXtreme Scale.
sessionobjectgrid.jar	Client et serveur	session/lib	Le fichier sessionobjectgrid.jar contient l'exécution de la gestion de sessions HTTP eXtreme Scale.
splicerlistener.jar	Utilitaire	session/lib	Le fichier splicerlistener.jar contient l'utilitaire splicer du programme d'écoute de session HTTP eXtreme Scale Version 7.1 et versions suivantes.
splicer.jar	Utilitaire	legacy/session/lib	Utilitaire splicer pour le filtre de gestionnaire de session HTTP WebSphere eXtreme Scale Version 7.0.
wxsra.rar	Client et serveur	session/lib	Le fichier wxsra.rar contient l'adaptateur de ressources eXtreme Scale permettant de se connecter à la grille à l'aide d'une fabrique de connexions.

A propos de l'installation de WebSphere eXtreme Scale dans un environnement .NET

.NET

Vous pouvez installer WebSphere eXtreme Scale for .NET dans un environnement d'exécution ou un environnement d'exécution et de production. Lors de l'installation dans un environnement d'exécution, il s'agit de l'environnement dans lequel vos applications .NET s'exécuteront. Les assemblages d'exécution sont installés sur le disque et dans le cache GAC (Global Assembly Cache). Pour

générer et tester vos applications .NET, installez WebSphere eXtreme Scale Client for .NET dans votre environnement de développement. Les assemblages d'exécution sont installés sur le disque et dans le cache GAC (Global Assembly Cache). Un exemple de code source .NET eXtreme Scale et de projet Visual Studio est installé dans le répertoire `install_dir\sample`. L'intégration dans Visual Studio IntelliSense fournit des descriptions de classes et de méthodes durant le développement. L'installation dans l'environnement de développement comprend également la documentation des APIs dans le répertoire `install_dir\doc` directory.

Installation de WebSphere eXtreme Scale Client for .NET

.NET

Vous pouvez installer WebSphere eXtreme Scale Client dans un environnement .NET si vous disposez d'applications qui s'exécutent dans cette structure.

Avant de commencer

- Téléchargez WebSphere eXtreme Scale Client depuis le site Support.
- Si vous envisagez d'installer WebSphere eXtreme Scale Client for .NET dans un environnement de développement, vous devez utiliser un système Windows disposant de Microsoft Visual Studio 2010 ou 2008.

Pourquoi et quand exécuter cette tâche

WebSphere eXtreme Scale Client for .NET peut installer un environnement d'exécution dans un environnement d'exécution et un environnement de développement.

Procédure

1. Utilisez l'assistant pour installer le client depuis le DVD ou téléchargez-le depuis le site Support.
2. Exécutez le fichier `setup.exe`.
3. Suivez les invites de l'assistant et cliquez sur **Suivant** pour accéder à la page du type de configuration.
4. Choisissez d'installer WebSphere eXtreme Scale Client dans un environnement d'exécution ou choisissez une installation personnalisée. Une personnalisation personnalisée permet d'installer le produit dans les deux environnements.
5. Si vous avez décidé d'installer WebSphere eXtreme Scale Client dans un environnement d'exécution, cliquez sur **Environnement d'exécution** et procédez comme suit :
   a. Cliquez sur **Installer** pour exécuter le programme d'installation et sur **Terminer**. Le répertoire d'installation par défaut est `C:\Program Files (x86)\IBM\WebSphere\eXtreme Scale .NET Client`
6. Si vous avez décidé d'installer WebSphere eXtreme Scale Client dans un environnement et un environnement d'exécution, choisissez l'installation **personnalisée** et procédez comme suit :
   a. Installez WebSphere eXtreme Scale Client dans le répertoire d'installation par défaut ou choisissez votre propre répertoire d'installation et cliquez sur **Suivant**.
   b. Par défaut, l'environnement d'exécution et l'environnement de développement sont sélectionnés. Vérifiez que vous disposez d'un espace disque suffisant pour installer les deux environnements. Cliquez sur **Suivant**.

c. Choisissez l'emplacement des fichiers journaux et cliquez sur **Suivant**.
d. Cliquez sur **Installer** pour exécuter le programme d'installation et sur **Terminer**.

Que faire ensuite

Vous pouvez mettre à jour ou modifier `Client.Sample` pour tester les API du client eXtreme Scale for .NET. Recherchez `Client.Sample` dans le fichier `install_dir\Sample\ClientSample.sln` dans le répertoire d'installation et chargez ce fichier dans Visual Studio pour afficher l'exemple d'application qui utilise des opérations simples de création, d'extraction, de mise à jour et de suppression. Utilisez `Client.Sample` comme guide pour accéder à la grille de données. Vous pouvez modifier cette application ou écrire une nouvelle application qui utilise le groupe d'API prises en charge du client eXtreme Scale for .NET.

Installation de WebSphere eXtreme Scale Client for .NET en mode silencieux

.NET

Vous pouvez installer WebSphere eXtreme Scale Client dans un environnement .NET en mode silencieux si vous ne souhaitez pas afficher le déroulement de l'installation ou devez installer le produit sur plusieurs machines. L'installation en mode silencieux signifie que vous devez enregistrer d'abord un fichier de réponses et les paramètres transmis à ce fichier.

Avant de commencer

- Téléchargez WebSphere eXtreme Scale Client depuis le site Support.
- Si vous envisagez d'installer WebSphere eXtreme Scale Client for .NET dans un environnement de développement, vous devez utiliser un système Windows disposant de Microsoft Visual Studio 2010 ou 2008.

Pourquoi et quand exécuter cette tâche

WebSphere eXtreme Scale Client for .NET peut être installé dans un environnement d'exécution ou dans un environnement d'exécution et de développement.

Procédure

1. Ouvrez une invite de commande et exécutez le script `setup.exe /r /f1"<install_dir>\Setup.iss"` `install_dir` est l'emplacement dans lequel vous voulez créer le fichier de réponses.
2. Suivez les invites de l'assistant et cliquez sur **Suivant** pour accéder à la page du type de configuration.
3. En fonction des options que vous choisissez, vous pouvez transmettre les valeurs suivantes pour créer le fichier de réponses `Setup.iss` :
   - Choisissez d'installer WebSphere eXtreme Scale Client dans un environnement d'exécution ou une installation personnalisée. Une personnalisation personnalisée permet d'installer le produit dans les deux environnements.
   - Si vous avez décidé d'installer WebSphere eXtreme Scale Client dans un environnement d'exécution, cliquez sur **Environnement d'exécution** et procédez comme suit :

a. Cliquez sur **Installer** pour exécuter le programme d'installation et sur **Terminer**. L'installation par défaut est `C:\Program Files (x86)\IBM\WebSphere\eXtreme Scale .NET Client`

- Si vous avez décidé d'installer WebSphere eXtreme Scale Client dans un environnement et un environnement d'exécution, choisissez l'installation **personnalisée** et procédez comme suit :
  a. Installez WebSphere eXtreme Scale Client dans le répertoire d'installation par défaut ou choisissez votre propre répertoire d'installation. Cliquez sur **Suivant**.
  b. Sélectionnez un environnement d'exécution et un environnement de développement. Si vous souhaitez effectuer l'installation dans les deux environnements, vérifiez que vous disposez d'un espace disque suffisant. Cliquez sur **Suivant**
  c. Choisissez l'emplacement des fichiers journaux et cliquez sur **Suivant**.

4. Cliquez sur **Installer** pour exécuter le programme d'installation et sur **Terminer**.
5. Ouvrez une invite de commande et exécutez le script `setup.exe /s /f1"<install_dir>\Setup.iss"` pour effectuer l'installation en mode silencieux WebSphere eXtreme Scale Client. `install_dir` est l'emplacement dans lequel vous voulez créer le fichier de réponses.

Que faire ensuite

Vous pouvez mettre à jour ou modifier `Client.Sample` pour tester les API du client eXtreme Scale. Recherchez `Client.Sample` dans le fichier `install_dir\Sample\ClientSample.sln` dans le répertoire d'installation et chargez ce fichier dans Visual Studio pour afficher l'exemple d'application qui utilise des opérations simples de création, d'extraction, de mise à jour et de suppression. Utilisez `Client.Sample` comme guide pour accéder à la grille de données. Vous pouvez modifier cette application ou écrire une nouvelle application qui utilise le groupe d'API prises en charge du client eXtreme Scale for .NET.

Désinstallation de WebSphere eXtreme Scale Client for .NET en mode silencieux

.NET

Pour supprimer WebSphere eXtreme Scale Client for .NET dans votre environnement, vous pouvez le désinstaller depuis le panneau de configuration Windows ou enregistrer un fichier de réponses pour le désinstaller en mode silencieux.

Avant de commencer

Si vous souhaitez désinstaller le produit dans un environnement de développement, veillez à arrêter Visual Studio.

Avertissement : Le programme de désinstallation supprime tous les fichiers binaires et la maintenance, telle que les groupes de correctifs et les correctifs temporaires, en même temps.

Procédure

1. Arrêtez les processus .NET eXtreme Scale.

2. Pour enregistrer un fichier de réponses, ouvrez une invite de commande et exécutez le script suivant :

   ```
   setup.exe /uninst /r /f1"<install_dir>\Setup.iss"
   ```

3. L'assistant de désinstallation s'ouvre et une fenêtre de confirmation s'affiche pour vérifier que vous voulez supprimer WebSphere eXtreme Scale Client for .NET et toutes ses fonctions. Cliquez sur **OK**.
4. A la fin de la désinstallation, cliquez sur **Terminer**.
5. Facultatif : pour tester le fichier de réponses et désinstaller WebSphere eXtreme Scale Client en mode silencieux :
   a. Installez WebSphere eXtreme Scale Client for .NET. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale Client for .NET», à la page 202
   b. Ouvrez une invite de commande et exécutez le script suivant pour désinstaller WebSphere eXtreme Scale Client for .NET en mode silencieux :

      ```
      setup.exe /uninst /s /f1"<install_dir>\Setup.iss"
      ```

Que faire ensuite

Vérifiez Windows Explorer pour vous assurer que tous les dossiers ont été supprimés du répertoire d'installation. Vous devez également vérifier le panneau de configuration Windows pour vous assurer que le produit n'est plus répertorié.

Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale

Java

Les offres de produit WebSphere eXtreme Scale sont disponibles dans les référentiels du produit. Avant de pouvoir accéder à ces référentiels, vous devez installer IBM Installation Manager.

Vous pouvez installer Installation Manager en utilisant les fichiers disponibles sur le support du produit, ou en utilisant un fichier obtenu à partir du site Passport Advantage ou en utilisant un fichier disponible sur le site Web de téléchargement d'IBM Installation Manager. Un fichier est un fichier compressé contenant des images d'installation.

Remarque : 8.6+
Installation Manager est disponible pour le téléchargement en version 32 et 64 bits. Vous pouvez utiliser une version d'Installation Manager pour installer WebSphere eXtreme Scale.

Installation Manager vous permet d'accéder aux référentiels de produit nécessaires. Vous devez accéder à ces référentiels afin d'installer les offres de produit WebSphere eXtreme Scale.

Il existe deux possibilités pour accéder aux référentiels de produit.

Option 1 : Accédez aux référentiels de produit sur le support physique et utilisez une installation locale

1. Installez Installation Manager sur votre système.
2. Utilisez Installation Manager pour installer l'offre de produit à partir des référentiels du produit présents sur le support.

Option 2 : Téléchargez les référentiels de produit à partir de Passport Advantage et utilisez une installation locale

1. Téléchargez les référentiels à partir du site Passport Advantage.

   Remarque : Voir Voir Logiciels pris en charge pour obtenir la liste des images d'installation IBM WebSphere eXtreme Scale téléchargeables depuis le site Web IBM Passport Advantage et d'autres informations.
2. Installez Installation Manager sur votre système.
3. Utilisez Installation Manager pour installer le produit à partir des référentiels de produit téléchargés.

Installation d'IBM Installation Manager à l'aide de l'interface graphique

Java

Pour accéder aux référentiels de produit nécessaires à l'installation des offres de produit WebSphere eXtreme Scale , vous devez installer IBM Installation Manager. Vous pouvez installer Installation Manager à l'aide d'une console d'assistant.

Avant de commencer

Vous devez installer IBM Installation Manager et avoir accès aux référentiels nécessaires. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Procédure

1. Accédez à l'emplacement contenant les fichiers d'installation du gestionnaire d'installation et exécutez l'une des commandes suivantes :

   Installation par un administrateur :

   - Windows `install.exe`
   - UNIX Linux `./install`

   Installation par un non administrateur :

   - Windows `userinst.exe`
   - UNIX Linux `./userinst`

   Pour plus d'informations sur les installations administratives et non administratives, voir Installation en tant qu'administrateur, non-administrateur ou groupe

   Installation en mode groupe :

   UNIX Linux `./groupinst`

   Remarques sur le mode groupe :

   - Le mode groupe permet à plusieurs utilisateurs d'utiliser une même instance d'IBM Installation Manager pour gérer des progiciels.

     Le mode Groupe ne signifie pas que deux personnes peuvent utiliser la même instance d'IBM Installation Manager en même temps.
   - Windows Le mode groupe n'est pas disponible sur les systèmes d'exploitation Windows.

- Si vous n'installez pas le gestionnaire d'installation avec le mode Groupe, vous ne pourrez pas utiliser le mode Groupe pour gérer les produits que vous installez ultérieurement à l'aide de cette instance d'Installation Manager.
- Remplacez l'emplacement d'installation par défaut de l'utilisateur en cours par un emplacement accessible à tous les utilisateurs du groupe.
- Configurez vos groupes, droits d'accès et variables d'environnement, comme décrit dans les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5 avant d'effectuer l'installation en mode groupe.
- Pour plus d'informations sur l'utilisation du mode groupe, consultez les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5.

2. Veillez à sélectionner le package du gestionnaire d'installation et cliquez sur **Next**.
3. Acceptez les modalités du contrat de licence et cliquez sur **Next**.
4. Cliquez sur **Next**.
5. Lisez le récapitulatif et cliquez sur **Installer**. Si l'installation a réussi, le programme affichera un message pour confirmer le succès de l'installation. Si l'installation n'aboutit pas, cliquez sur **View Log File** pour corriger l'erreur.
6. Ajoutez le référentiel du produit à vos préférences Installation Manager.
   a. Démarrez Installation Manager.
   b. Dans le menu du haut, cliquez sur **Fichier > Préférences**.
   c. Sélectionnez **Repositories**.
   d. Cliquez sur **Add Repository**.
   e. Entrez le chemin du fichier `repository.config` dans l'emplacement contenant les fichiers du référentiel :
      - Windows `C:\repositories\nom_produit\local-repositories`
      - UNIX Linux `/var/repositories/nom_produit/local-repositories`
   f. Cliquez sur **OK**.
7. Effacez tous les emplacements énumérés dans la fenêtre Repositories, qui ne seront pas utilisés.
8. Cliquez sur **Apply**.
9. Cliquez sur **OK**.
10. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.

Que faire ensuite

Lorsque l'installation du gestionnaire d'installation et la configuration du référentiel aboutissent, vous pouvez continuer d'installer n'importe quel WebSphere eXtreme Scale autonome ou WebSphere eXtreme Scale for WebSphere Application Server pour l'offre de produit. Pour plus d'informations, voir «Installation du produit à l'aide de l'interface graphique»

Installation du produit à l'aide de l'interface graphique

Java

Utilisez Installation Manager à partir de la console de l'assistant pour installer les offres de produit WebSphere eXtreme Scale .

Avant de commencer

Vous devez installer les fichiers de produit nécessaires pour Installation Manager et avoir accès aux référentiels nécessaires. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Procédure

1. Démarrez Installation Manager.

   Conseil : UNIX Linux Vous pouvez démarrer Installation Manager en mode groupe avec la commande **./IBMIM**.
   - Le mode groupe permet à plusieurs utilisateurs d'utiliser une même instance d'IBM Installation Manager pour gérer des packages logiciels.
   - Pour plus d'informations sur l'utilisation du mode groupe, consultez les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5.
2. Cliquez sur **Install**.

   Remarque : Si vous êtes invité à vous authentifier, utilisez l'ID et le mot de passe IBM avec lesquels vous vous êtes enregistré sur le site Web du programme.

   Installation Manager analyse ses référentiels définis à la recherche des packages disponibles.
3. Sélectionnez l'une des offres de produit ci-dessous ainsi que la version appropriée.
   - WebSphere eXtreme Scale dans un environnement autonome
   - WebSphere eXtreme Scale Client dans un environnement autonome
   - WebSphere eXtreme Scale for WebSphere Application Server version 7
   - WebSphere eXtreme Scale for WebSphere Application Server version 8
   - WebSphere eXtreme Scale Client for WebSphere Application Server version 7
   - WebSphere eXtreme Scale Client for WebSphere Application Server version 8

   Si le produit est déjà installé sur votre système, un message s'affichera pour vous en informer. Pour créer une autre installation du produit dans un autre emplacement, cliquez sur **Continuer**.

   Conseil : Si l'option **Search service repositories during installation and updates** est sélectionnée sur la page des préférences de référentiel dans Installation Manager et que vous êtes connecté à Internet, vous pouvez cliquer sur **Check for Other Versions and Extensions**. Ainsi, vous pouvez rechercher les mises à jour dans les référentiels de mise à jour par défaut pour les packages sélectionnés. Dans ce cas, il n'est pas nécessaire d'ajouter l'URL de référentiel de service spécifique à la page des préférences en matière de référentiel d'Installation Manager.
   a. Sélectionnez les correctifs à installer.

      Les correctifs recommandés sont sélectionnés par défaut.

Vous pouvez choisir de n'afficher que les correctifs recommandés et de masquer les autres.

b. Cliquez sur **Next**.

Remarque : Installation Manager peut vous inviter à effectuer une mise à jour vers le dernier niveau d'Installation Manager lors de la connexion au référentiel. Si vous y êtes invité, effectuez la mise à jour vers la nouvelle version avant de poursuivre. Pour plus d'informations sur les mises à jour automatiques, voir le centre de documentation d'IBM Installation Manager Version 1.5.

4. Acceptez les modalités du contrat de licence et cliquez sur **Next**.
5. Indiquez le répertoire principal d'installation du produit.

   Le panneau affiche également le répertoire de ressources partagées et les informations d'espace disque.

   Remarque : La première fois que vous installez un package à l'aide d'Installation Manager, spécifiez le répertoire des ressources partagées. Le répertoire des ressources partagées est l'endroit où les artefacts d'installation se trouvent ; ils peuvent être utilisés par un plusieurs groupe de packages. Utilisez votre unité ayant la taille la plus importante pour cette installation. Vous ne pouvez pas changer l'emplacement du répertoire tant que vous n'avez pas désinstallé tous les packages.

   Restrictions :

   - Si vous supprimez l'emplacement cible par défaut et que vous ne renseignez pas une zone du répertoire d'installation, vous ne pouvez pas continuer.
   - N'utilisez pas de liens symboliques comme répertoire de destination.

     car ils ne sont pas pris en charge.
   - N'utilisez pas de point-virgule dans le nom d'un répertoire.

     Si le répertoire de destination inclut un point-virgule, cela signifie que WebSphere eXtreme Scale n'est pas installé comme prévu.

     Windows Sous Windows, le point-virgule est le caractère utilisé pour construire le chemin d'accès aux classes.
   - Windows La longueur maximale du chemin d'accès sur les systèmes d'exploitation Windows Server 2008, Windows Vista et Windows 7 est de 60 caractères.
6. Cliquez sur **Next**.
7. Sélectionnez les langues pour lesquelles un contenu traduit doit être installé.

   L'anglais est toujours sélectionné.
8. Cliquez sur **Next**.
9. Sélectionnez les fonctions que vous voulez installer.

   En fonction de l'offre de produit sélectionnée, vous pouvez choisir l'une des fonctions suivantes :

   - Client

     Disponible sous forme de fonction obligatoire si vous installez soit WebSphere eXtreme Scale dans un environnement autonome, soit les offres de produit WebSphere eXtreme Scale for WebSphere Application Server. Le client doit être installé pour ces offres de produit.
   - Server

Disponible si vous installez soit WebSphere eXtreme Scale dans un environnement autonome, soit WebSphere eXtreme Scale for WebSphere Application Server. Vous pouvez choisir de ne pas installer le serveur pour ces offres de produit.

- Console

  Disponible pour toutes les offres de produit WebSphere eXtreme Scale. Vous pouvez choisir d'installer la console de surveillance. Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.

- Samples

  Disponible pour toutes les offres de produit WebSphere eXtreme Scale.

10. Cliquez sur **Next**.
11. Lisez le récapitulatif et cliquez sur **Installer**.
    - Si l'installation aboutit, le programme affiche un message indiquant que l'installation a abouti.

      Remarque : Le programme peut également afficher d'importantes instructions post-installation.
    - Si l'installation n'aboutit pas, cliquez sur **View Log File** pour corriger l'erreur.
12. Sélectionnez l'outil que vous souhaitez démarrer à la fin de l'installation.
    - Sélectionnez **Outil de gestion des profils pour créer un profil** si vous souhaitez créer un profil de serveur d'applications avec des paramètres appropriés pour un environnement de production.
    - Sélectionnez **Outil de gestion des profils pour créer un profil de serveur d'applications pour un environnement de développement** si vous souhaitez créer un profil de serveur d'applications avec des paramètres appropriés pour un environnement de développement.

      Remarque : Les paramètres de développement sont adaptés à un environnement de développement au sein desquelles des mises à jour sont effectuées et les ressources systèmes sont au minimum. N'utilisez pas les paramètres de développement pour les serveurs de production.
    - Sélectionnez **Aucun** si vous ne souhaitez pas créer de profil lorsque l'installation est terminée.

    Restriction : L'option de démarrage de l'outil de gestion des profils est disponible uniquement si une version de WebSphere Application Server contenant l'outil de gestion des profils est installée.
13. Cliquez sur **Terminer**.
14. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.

Installation d'IBM Installation Manager à l'aide de la ligne de commande

Java

Pour accéder aux référentiels de produit nécessaires à l'installation des offres de produit WebSphere eXtreme Scale , vous devez installer IBM Installation Manager. Vous pouvez installer Installation Manager à partir de la ligne de commande.

Avant de commencer

Vous devez installer les fichiers de produit nécessaires pour Installation Manager et avoir accès aux référentiels nécessaires. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Procédure

1. Accédez à l'emplacement contenant les fichiers d'installation d'Installation Manager et exécutez une des commandes suivantes :

   Installation par un administrateur :

   - Windows `installc.exe -acceptLicense -log` *`chemin_et_nom_fichier_journal`*
   - UNIX Linux `./installc -acceptLicense -log` *`chemin_et_nom_fichier_journal`*

   Installation par un non administrateur :

   - Windows `userinstc.exe -acceptLicense -log` *`chemin_et_nom_fichier_journal`*
   - UNIX Linux `./userinstc -acceptLicense -log` *`chemin_et_nom_fichier_journal`*

   Installation en mode groupe :

   UNIX Linux `./groupinstc -acceptLicense -dataLocation` *`emplacement_données_application`* `-log` *`chemin_et_nom_fichier_journal`* `-installationDirectory` *`rép_Installation_Manager`*

   Remarques sur le mode groupe :

   - Le mode groupe permet à plusieurs utilisateurs d'utiliser une même instance d'IBM Installation Manager pour gérer des packages logiciels.
   - Windows Le mode groupe n'est pas disponible sur les systèmes d'exploitation Windows.
   - Si vous n'installez pas Installation Manager avec le mode groupe, vous ne pourrez pas utiliser le mode groupe pour gérer les produits que vous installez ultérieurement à l'aide de cette instance d'Installation Manager.
   - Assurez-vous de bien changer l'emplacement d'installation et de le faire passer de l'emplacement par défaut dans le répertoire de base de l'utilisateur actuel à un emplacement accessible par tous les utilisateurs du groupe.
   - Configurez vos groupes, droits d'accès et variables d'environnement, comme décrit dans les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5 avant d'effectuer l'installation en mode groupe.

- Pour plus d'informations sur l'utilisation du mode groupe, consultez les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5.

2. Facultatif : Si le référentiel requiert un nom d'utilisateur et un mot de passe, créez un fichier de clés pour accéder à ce référentiel.

   Pour plus d'informations sur la création d'un fichier de clés pour Installation Manager, reportez-vous au centre de documentation d'IBM Installation Manager version 1.5.

   Conseil : Lors de la création d'un fichier de clés, ajoutez `/repository.config` à la fin de l'emplacement d'URL de référentiel si la commande **imutilsc** ne peut pas trouver l'URL indiquée.

Que faire ensuite

Une fois que l'installation d'Installation Manager et la configuration du référentiel ont abouti, vous pouvez continuer d'installer n'importe quel WebSphere eXtreme Scale autonome ou WebSphere eXtreme Scale pour l'offre de produit WebSphere Application Server. Pour plus d'informations, voir «Installation du produit à l'aide de l'interface graphique», à la page 207.

Installation du produit à l'aide de la ligne de commande

Java

Utilisez Installation Manager à partir de la ligne de commande pour installer les offres de produit WebSphere eXtreme Scale .

Avant de commencer

Vous devez installer les fichiers de produit nécessaires pour Installation Manager et avoir accès aux référentiels nécessaires. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Procédure

1. Ouvrez une session sur votre système.
2. Accédez au sous-répertoire `eclipse/tools` dans le répertoire où vous avez installé Installation Manager.
3. Vérifiez que le référentiel du produit est disponible.

   Windows

   ```
   imcl.exe listAvailablePackages -repositories référentiel_source
   ```

   UNIX Linux

   ```
   ./imcl listAvailablePackages -repositories référentiel_source
   ```

   Vous devez voir un ou plusieurs niveaux de l'offre.
4. Utilisez la commande **imcl** pour installer le produit.

   Windows

   ```
   imcl.exe install com.ibm.websphere.v85_version_offre,ID_fonction_facultative
     -repositories référentiel_source
     -installationDirectory répertoire_installation
     -sharedResourcesDirectory répertoire_partagé
     -accessRights mode_accès
   ```

```
  -preferences clé_préférence=valeur
  -properties clé_propriété=valeur
  -keyring fichier_de_clés -password mot_de_passe
  -acceptLicense
```

UNIX Linux

```
./imcl install com.ibm.websphere.version_offre,ID_fonction_facultative
  -repositories référentiel_source
  -installationDirectory répertoire_installation
  -sharedResourcesDirectory répertoire_partagé
  -accessRights mode_accès
  -preferences clé_préférence=valeur
  -properties clé_propriété=valeur
  -keyring fichier_de_clés -password mot_de_passe
  -acceptLicense
```

Conseils :

- *ID_offre* correspond à l'ID offre répertorié dans la rubrique «ID des offres de produit WebSphere eXtreme Scale», à la page 192.
- La *version_offre*, qui peut, le cas échéant, être associée à l'ID offre à l'aide d'un trait de soulignement, est une version spécifique de l'offre à installer (`8.5.0.20110503_0200`, par exemple).
  - Si la *version_offre* n'est **pas** spécifiée, la version la plus récente de l'offre et **tous** les correctifs temporaires de cette version sont installés.
  - Si la *version_offre* est spécifiée, la version indiquée de l'offre est installée et **aucun** correctif temporaire de cette version n'est installé.

  La version de l'offre peut être rattachée à la fin de l'ID offre à l'aide d'un trait de soulignement lorsque vous exécutez la commande suivante au niveau du référentiel :

  ```
  imcl listAvailablePackages -repositories référentiel_source
  ```
- Vous pouvez également spécifier `none`, `recommended` ou `all` avec l'argument `-installFixes` afin d'indiquer les correctifs temporaires que vous souhaitez installer avec l'offre.
  - Si la version de l'offre n'est **pas** spécifiée, l'option `-installFixes` est paramétrée par défaut sur `all`.
  - Si la version de l'offre est spécifiée, l'option `-installFixes` est paramétrée par défaut sur `none`.
- Vous pouvez ajouter une liste de fonctions séparées par des virgules. Exemple de programme :

```
imcl -acceptLicense install com.ibm.websphere.WXS.v85,xs.console.feature,xs.samples.feature
```

  - `xs.client.standalone.feature` Disponible sous forme de fonction obligatoire si vous installez WebSphere eXtreme Scale dans un environnement autonome ou les offres de produit WebSphere eXtreme Scale for WebSphere Application Server.
  - `xs.server.standalone.feature` En fonction de l'offre de produit que vous souhaitez installer, vous pouvez choisir d'installer le serveur. Le serveur est une fonction sélectionnable dans les offres de produit suivantes :
    - WebSphere eXtreme Scale dans un environnement autonome
    - WebSphere eXtreme Scale for WebSphere Application Server version 7
    - WebSphere eXtreme Scale for WebSphere Application Server version 8
  - `xs.console.feature` Disponible pour toutes les offres de produit. Vous pouvez choisir d'installer la console de surveillance. Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques

préconfigurés pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.

– `xs.samples.feature` Disponible pour toutes les offres de produit. Vous pouvez choisir d'installer des exemples.

Remarques :

- Si vous avez précédemment spécifié le mode d'installation d'Installation Manager, le paramètre -accessRights n'est pas requis.
- Si vous rencontrez des problèmes ultérieurement, Installation Manager peut sauvegarder les versions antérieures d'un package à des fins de rétrogradation. Quand Installation Manager rétrograde un package à une version antérieure, la version actuelle des fichiers est désinstallée et les versions antérieures sont réinstallées. Si vous choisissez de ne pas sauvegarder les fichiers pour rétrogradation, vous pouvez empêcher la sauvegarde des fichiers en indiquant une préférence :

  ```
  -preference com.ibm.cic.common.core.preferences.preserveDownloadedArtifacts=False
  ```

  Pour plus d'informations sur la définition de vos préférences pour Installation Manager, voir le centre de documentation d'IBM Installation Manager Version 1.5.

  Conseil : Même si vous choisissez de ne pas sauvegarder les fichiers pour rétrogradation, vous pouvez accéder aux fichiers de produit pour rétrogradation à partir du référentiel.
- Le programme peut générer des instructions post-installation importantes dans la sortie standard.

Pour plus d'informations sur l'utilisation de la commande **imcl** pour installer le produit, voir le centre de documentation d'IBM Installation Manager Version 1.5.

Installation d'IBM Installation Manager à l'aide de fichiers de réponses

Java

Pour accéder aux référentiels de produit nécessaires à l'installation des offres de produit WebSphere eXtreme Scale , vous devez installer IBM Installation Manager. Vous pouvez installer Installation Manager à l'aide de fichiers de réponses.

Avant de commencer

Vous devez installer les fichiers de produit nécessaires pour Installation Manager et avoir accès aux référentiels nécessaires. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Procédure

Accédez à l'emplacement contenant les fichiers d'installation d'Installation Manager, puis exécutez une des commandes ci-dessous pour installer Installation Manager.

Installation par un administrateur :

- Windows `installc.exe -acceptLicense -log` *chemin_et_nom_fichier_journal*
- UNIX Linux `./installc -acceptLicense -log` *chemin_et_nom_fichier_journal*

Installation par un non administrateur :

- Windows `userinstc.exe -acceptLicense -log` *chemin_et_nom_fichier_journal*
- UNIX Linux `./userinstc -acceptLicense -log` *chemin_et_nom_fichier_journal*

Installation en mode groupe :

UNIX Linux `./groupinstc -acceptLicense -dataLocation` *emplacement_données_application* `-log` *chemin_et_nom_fichier_journal* `-installationDirectory` *rép_Installation_Manager*

Remarques sur le mode groupe :

- Le mode groupe permet à plusieurs utilisateurs d'utiliser une même instance d'IBM Installation Manager pour gérer des packages logiciels.

  Le mode groupe ne signifie pas que deux personnes peuvent utiliser la même instance d'IBM Installation Manager en même temps.
- Windows Le mode groupe n'est pas disponible sur les systèmes d'exploitation Windows.
- Si vous n'installez pas Installation Manager avec le mode groupe, vous ne pourrez pas utiliser le mode groupe pour gérer les produits que vous installez ultérieurement à l'aide de cette instance d'Installation Manager.
- Assurez-vous de bien changer l'emplacement d'installation et de le faire passer de l'emplacement par défaut dans le répertoire de base de l'utilisateur actuel à un emplacement accessible par tous les utilisateurs du groupe.
- Configurez vos groupes, droits d'accès et variables d'environnement, comme décrit dans les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5 avant d'effectuer l'installation en mode groupe.
- Pour plus d'informations sur l'utilisation du mode groupe, consultez les feuilles de route du mode groupe du centre de documentation d'IBM Installation Manager version 1.5.

Que faire ensuite

Une fois que l'installation d'Installation Manager et la configuration du référentiel aboutissent, vous pouvez continuer d'installer n'importe quel WebSphere eXtreme Scale autonome ou WebSphere eXtreme Scale for WebSphere Application Server pour l'offre de produit. Pour plus d'informations, voir «Installation du produit à l'aide de l'interface graphique», à la page 207.

Installation du produit à l'aide d'un fichier de réponses

Java

Utilisez Installation Manager à l'aide d'un fichier de réponses pour installer les offres de produit WebSphere eXtreme Scale .

Avant de commencer

Vous devez installer les fichiers de produit nécessaires pour Installation Manager et avoir accès aux référentiels nécessaires. Pour plus d'informations, voir «Installation d'IBM Installation Manager et des offres de produit WebSphere eXtreme Scale», à la page 205.

Pourquoi et quand exécuter cette tâche

A l'aide d'Installation Manager, vous pouvez utiliser des fichiers de réponses pour installer le produit de plusieurs manières. Vous pouvez enregistrer un fichier de réponses à l'aide de l'interface graphique.

Procédure

1. A partir d'une ligne de commande, accédez au sous-répertoire Eclipse du répertoire dans lequel vous avez installé Installation Manager.
2. Démarrez Installation Manager à partir de la ligne de commande en utilisant l'option d'enregistrement -record.

   Par exemple :

   - Windows **Administrateur ou non administrateur :**

     ```
     IBMIM.exe -skipInstall "C:\temp\imRegistry"
       -record C:\temp\install_response_file.xml
     ```

   - UNIX Linux **Administrateur :**

     ```
     ./IBMIM -skipInstall /var/temp/imRegistry
       -record /var/temp/install_response_file.xml
     ```

   - UNIX Linux **Non administrateur :**

     ```
     ./IBMIM -skipInstall racine_utilisateur/var/temp/imRegistry
       -record rép_base_utilisateur/var/temp/install_response_file.xml
     ```

   Conseil : Lorsque vous enregistrez un nouveau fichier de réponses, vous pouvez sélectionner le paramètre -skipInstall. Celui-ci vous offrira les avantages suivants :

   - Aucun fichier n'est installé, ce qui accélère l'enregistrement.
   - Si vous utilisez un emplacement de données temporaire avec le paramètre -skipInstall, Installation Manager y placera le registre d'installation lors de l'enregistrement. Lorsque vous redémarrerez Installation Manager sans le paramètre -skipInstall, vous pourrez utiliser le fichier de réponses pour l'installation au lieu du vrai registre d'installation.

     L'opération -skipInstall ne doit pas être utilisée sur l'emplacement de données de l'agent en cours utilisé par Installation Manager. Cette opération n'est pas prise en charge. Utilisez un nouvel emplacement inscriptible et utilisez à nouveau l'emplacement pour des sessions d'enregistrement ultérieures.

   Pour plus d'informations, voir le centre de documentation d'IBM Installation Manager Version 1.5.
3. Ajoutez les référentiels appropriés à vos préférences dans Installation Manager.
   a. Dans le menu du haut, cliquez sur **Fichier > Préférences**
   b. Sélectionnez **Référentiels**
   c. Effectuez les actions suivantes pour chaque référentiel :
      1) Cliquez sur **Add Repository**.

2)

3) Entrez le chemin d'accès au fichier `repository.config` du référentiel Web éloigné ou du répertoire local dans lequel vous avez décompressé les fichiers du référentiel.

   Par exemple :

   - Référentiels éloignés :

     `https://downloads.mycorp.com:8080/WXS_85_repository`

   - Référentiels locaux :

     – Windows `C:\repositories\wxs85\local-repositories`

     – UNIX Linux `/var/repositories/wxs85/local-repositories`

4) Cliquez sur **OK**.

5) Cliquez sur **Appliquer**.

6) Cliquez sur **OK**.

d. Cliquez sur **Installer**.

   Remarque : Si vous êtes invité à vous authentifier, utilisez l'ID et le mot de passe IBM avec lesquels vous vous êtes enregistré sur le site Web du programme.

   Installation Manager analyse ses référentiels définis à la recherche des packages disponibles.

4. Sélectionnez l'une des offres de produit suivantes ainsi que la version appropriée :
   - WebSphere eXtreme Scale dans un environnement autonome
   - WebSphere eXtreme Scale Client dans un environnement autonome
   - WebSphere eXtreme Scale for WebSphere Application Server version 7
   - WebSphere eXtreme Scale for WebSphere Application Server version 8
   - WebSphere eXtreme Scale Client for WebSphere Application Server version 7
   - WebSphere eXtreme Scale Client for WebSphere Application Server version 8

   Si le produit est déjà installé sur votre système, un message s'affichera pour vous en informer. Pour créer une autre installation du produit dans un autre emplacement, cliquez sur **Continuer**.

   Conseil : Si l'option **Search service repositories during installation and updates** est sélectionnée sur la page des préférences de référentiel dans Installation Manager et que vous êtes connecté à Internet, vous pouvez cliquer sur **Check for Other Versions and Extensions**. Ainsi, vous pouvez rechercher les mises à jour dans les référentiels de mise à jour par défaut pour les packages sélectionnés. Dans ce cas, il n'est pas nécessaire d'ajouter l'URL de référentiel de service spécifique à la page des préférences en matière de référentiel d'Installation Manager.

5. Sélectionnez les correctifs à installer.

   Les correctifs recommandés sont sélectionnés par défaut.

   Vous pouvez choisir de n'afficher que les correctifs recommandés et de masquer les autres.

6. Cliquez sur **Next**.

7. Acceptez les modalités du contrat de licence et cliquez sur **Next**.

8. Indiquez le répertoire principal d'installation du produit.

   Le panneau affiche également le répertoire de ressources partagées et les informations d'espace disque.

   Remarque : La première fois que vous installez un package à l'aide d'Installation Manager, spécifiez le répertoire des ressources partagées. Le répertoire des ressources partagées est l'endroit où les artefacts d'installation se trouvent ; ils peuvent être utilisés par un plusieurs groupe de packages. Utilisez votre unité ayant la taille la plus importante pour cette installation. Vous ne pouvez pas changer l'emplacement du répertoire tant que vous n'avez pas désinstallé tous les packages.

   Restrictions :

   - Si vous supprimez l'emplacement cible par défaut et que vous ne renseignez pas une zone du répertoire d'installation, vous ne pouvez pas continuer.
   - N'utilisez pas de liens symboliques comme répertoire de destination.

     car ils ne sont pas pris en charge.
   - Windows La longueur maximale du chemin d'accès sur les systèmes d'exploitation Windows Server 2008, Windows Vista et Windows 7 est de 60 caractères.
9. Cliquez sur **Next**.
10. Sélectionnez les langues pour lesquelles un contenu traduit doit être installé.

    L'anglais est toujours sélectionné.
11. Cliquez sur **Next**.
12. Sélectionnez les fonctions que vous voulez installer.
    - Client

      Disponible sous forme de fonction obligatoire si vous installez soit WebSphere eXtreme Scale dans un environnement autonome, soit les offres de produit WebSphere eXtreme Scale for WebSphere Application Server. Le client doit être installé pour ces offres de produit.
    - Server

      Disponible si vous installez soit WebSphere eXtreme Scale dans un environnement autonome, soit WebSphere eXtreme Scale for WebSphere Application Server. Vous pouvez choisir de ne pas installer le serveur pour ces offres de produit.
    - Console

      Disponible pour toutes les offres de produit WebSphere eXtreme Scale. Vous pouvez choisir d'installer la console de surveillance. Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.
    - Samples

      Disponible pour toutes les offres de produit WebSphere eXtreme Scale.
13. Cliquez sur **Next**.
14. Lisez le récapitulatif et cliquez sur **Install**.

- Si l'installation aboutit, le programme affiche un message indiquant que l'installation a abouti.

  Remarque : Le programme peut également afficher d'importantes instructions post-installation.
- Si l'installation n'aboutit pas, cliquez sur **View Log File** pour corriger l'erreur.

15. Cliquez sur **Finish**.
16. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.

Création d'un fichier de clés : Java

Après avoir utilisé Installation Manager pour enregistrer un fichier de réponses afin d'installer les offres du produit WebSphere eXtreme Scale , vous pouvez choisir de créer un fichier de clés. Si vous utilisez un référentiel distant qui requiert une authentification, vous pouvez créer un fichier de clés pour l'installation.

Avant de commencer

Vous devez enregistrer un fichier de réponses. Pour plus d'informations, voir «Installation du produit à l'aide d'un fichier de réponses», à la page 215.

Procédure

1. A partir d'une ligne de commande, passez au sous-répertoire Eclipse du répertoire dans lequel vous avez installé Installation Manager.
2. Démarrez Installation Manager à partir de la ligne de commande en utilisant l'option d'enregistrement -record.

   Par exemple :

   - Windows **Administrateur ou non administrateur :**

     ```
     IBMIM.exe -skipInstall "C:\temp\imRegistry"
       -keyring C:\IM\im.keyring
       -record C:\temp\keyring_response_file.xml
     ```

   - UNIX Linux **Administrateur :**

     ```
     ./IBMIM -skipInstall /var/temp/imRegistry
       -keyring /var/IM/im.keyring
       -record /var/temp/keyring_response_file.xml
     ```

   - UNIX Linux **Non administrateur :**

     ```
     ./IBMIM -skipInstall rép_base_utilisateur/var/temp/imRegistry
       -keyring rép_base_utilisateur/var/IM/im.keyring
       -record rép_base_utilisateur/var/temp/keyring_response_file.xml
     ```

3. Lorsqu'une fenêtre s'affiche, vous demandant vos données d'identification pour le référentiel distant authentifié, entrez-les et **enregistrez**-les.
4. Cliquez sur **Fichier > Quitter** pour fermer le gestionnaire d'installation.

   Pour plus d'informations, voir le centre de documentation IBM Installation Manager version 1.5.

Installation de l'infrastructure OSGi Eclipse Equinox avec Eclipse Gemini pour les clients et les serveurs

Java

Si vous souhaitez déployer WebSphere eXtreme Scale dans la structure OSGi, vous devez configurer l'environnement Eclipse Equinox.

Pourquoi et quand exécuter cette tâche

La tâche nécessite que vous téléchargiez et installiez l'infrastructure Blueprint qui permet de configurer ensuite les JavaBeans et de les exposer en tant que services. L'utilisation de services est importante, car vous pouvez exposer des plug-in en tant que services OSGi pour qu'ils puissent être utilisés par l'environnement d'exécution eXtreme Scale. Le produit prend en charge deux conteneurs Blueprint dans l'infrastructure OSGi principale Eclipse Gemini et Apache Aries. Utilisez cette procédure pour configurer le conteneur Gemini Eclipse.

Procédure

1. Téléchargez Eclipse Equinox SDK Version 3.6.1 ou la version suivante à partir du site Web Eclipse. Créez un répertoire pour l'infrastructure Equinox, par exemple, `/opt/equinox`. Ces instructions font référence à ce répertoire sous la forme `equinox_root`. Extrayez le fichier compressé dans le répertoire `equinox_root`.
2. Téléchargez le fichier compressé gemini-plan d'incubation 1.0.0 depuis le site Web Eclipse. Extrayez le contenu du fichier dans un répertoire temporaire et copiez les fichiers extraits suivants vers le répertoire `equinox_root/plugins` :

   ```
   dist/gemini-blueprint-core-1.0.0.jar
   dist/gemini-blueprint-extender-1.0.0.jar
   dist/gemini-blueprint-io-1.0.0.jar
   ```

   Avertissement : Selon l'emplacement dans lequel vous avez téléchargé le fichier Blueprint compressé, les fichiers extraits peuvent avoir l'extension `RELEASE.jar`, à l'instar des fichiers JAR Spring framework dans l'étape suivante. Vous devez vérifier que les noms de fichier correspondent aux références de fichier dans le fichier `config.ini`.
3. Téléchargez Spring Framework Version 3.0.5 à partir de la page Web SpringSource http://www.springsource.com/download/community. Extrayez le contenu du fichier dans un répertoire temporaire et copiez les fichiers extraits suivants vers le répertoire `equinox_root/plugins` :

   ```
   org.springframework.aop-3.0.5.RELEASE.jar
   org.springframework.asm-3.0.5.RELEASE.jar
   org.springframework.beans-3.0.5.RELEASE.jar
   org.springframework.context-3.0.5.RELEASE.jar
   org.springframework.core-3.0.5.RELEASE.jar
   org.springframework.expression-3.0.5.RELEASE.jar
   ```
4. Téléchargez le fichier AOP Alliance Java archive (JAR) depuis la page Web SpringSource. Copiez com.springsource.org.aopalliance-1.0.0.jar vers le répertoire `equinox_root/plugins` .
5. Téléchargez le fichier JAR Apache commons logging 1.1.1 JAR depuis la page Web SpringSource. Copiez le fichier `com.springsource.org.apache.commons.logging-1.1.1.jar` vers le répertoire `equinox_root/plugins`.
6. Téléchargez le client de ligne de commande Luminis OSGi Configuration Admin. Utilisez cet ensemble de fichiers JAR pour gérer les configurations administratives OSGi. Copiez le fichier `net.luminis.cmc-0.2.5.jar` vers le répertoire `equinox_root/plugins`.

7. Téléchargez l'ensemble Apache Felix file installation Version 3.0.2 depuis la page Web http://felix.apache.org/site/index.html. Copiez le fichier `org.apache.felix.fileinstall-3.0.2.jar` vers le répertoire `equinox_root/plugins`.
8. Créez un répertoire de configuration dans le répertoire `equinox_root/plugins`, par exemple :

   ```
   mkdir equinox_root/plugins/configuration
   ```
9. Créez le fichier `config.ini` suivant dans le répertoire `equinox_root/plugins/configuration` en remplaçant *equinox_root* par le chemin absolu dans le chemin du répertoire `equinox_root` en supprimant tous les espaces après la barre oblique inverse dans chaque ligne. Vous devez placer une ligne blanche à la fin du fichier, par exemple :

   ```
   osgi.noShutdown=true
   osgi.java.profile.bootdelegation=none
   org.osgi.framework.bootdelegation=none
   eclipse.ignoreApp=true
   osgi.bundles=\
   org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
   org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
   org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
   com.springsource.org.apache.commons.logging-1.1.1.jar@1:start, \
   com.springsource.org.aopalliance-1.0.0.jar@1:start, \
   org.springframework.aop-3.0.5.RELEASE.jar@1:start, \
   org.springframework.asm-3.0.5.RELEASE.jar@1:start, \
   org.springframework.beans-3.0.5.RELEASE.jar@1:start, \
   org.springframework.context-3.0.5.RELEASE.jar@1:start, \
   org.springframework.core-3.0.5.RELEASE.jar@1:start, \
   org.springframework.expression-3.0.5.RELEASE.jar@1:start, \
   org.apache.felix.fileinstall-3.0.2.jar@1:start, \
   net.luminis.cmc-0.2.5.jar@1:start, \
   gemini-blueprint-core-1.0.0.jar@1:start, \
   gemini-blueprint-extender-1.0.0.jar@1:start, \
   gemini-blueprint-io-1.0.0.jar@1:start
   ```

   Si vous avez déjà configuré l'environnement, vous pouvez nettoyer le référentiel de plug-in Equinox en supprimant le répertoire `equinox_root\plugins\configuration\org.eclipse.osgi`.
10. Exécutez la commande suivante pour démarrer la console Equinox.

    Si vous exécutez une version différente d'Equinox, le nom du fichier JAR est différent de celui de l'exemple ci-dessous :

    ```
    java -jar plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
    ```

Installation du service de données REST

Java

Nous allons expliquer comment installer sur un serveur Web le service de données REST d'WebSphere eXtreme Scale.

Avant de commencer

Configuration logicielle

Le service de données REST d'WebSphere eXtreme Scale est une application Web Java qui peut être déployée sur tout serveur d'applications prenant en charge la spécification de servlet Java version 2.3 et un environnement d'exécution Java version 5 ou plus récente.

Les logiciels suivants sont requis :

- Java Standard Edition 6 ou version suivante
- le conteneur de servlets Web version 2.3 ou ultérieure, qui inclut l'un des éléments suivants :
  - le serveur d'applications WebSphere version 7.0.0.5 ou ultérieure

– WebSphere Community Edition version 2.1.1.3 ou ultérieure
– Apache Tomcat version 5.5 ou ultérieure

WebSphere eXtreme Scale, Version 7.1 ou ultérieure, y compris la version d'évaluation.

Pourquoi et quand exécuter cette tâche

Le service de données REST WebSphere eXtreme Scale inclut un seul fichier `wxsrestservice.war`. Le fichier `wxsrestservice.war` comporte un seul servlet qui fait office de passerelle entre vos applications client WCF Data Services ou tout autre client REST HTTP et une grille de données.

Le service de données REST contient un exemple qui permet de créer rapidement une grille de données et d'interagir avec elle en utilisant un client eXtreme Scale ou le service de données REST. Voir «Configuration des services de données REST», à la page 435 pour plus d'informations sur l'utilisation de l'exemple.

Lors de l'installation de WebSphere eXtreme Scale 7.1 ou de l'extraction de la version d'évaluation d'eXtreme Scale version 7.1, les répertoires et fichiers suivants sont inclus :

- `base_servicerest/lib`

  Le répertoire lib contient ces fichiers :

  – `wxsrestservice.ear` – L'archive d'application d'entreprise de service de données REST à utiliser avec le serveur d'application WebSphere et le serveur d'application CE WebSphere.
  – `wxsrestservice.war` – Le module Web de service de données REST à utiliser avec Apache Tomcat.

  Le fichier `wxsrestservice.ear` inclut le fichier `wxsrestservice.war` et tous deux sont étroitement couplés à l'environnement d'exécution WebSphere eXtreme Scale. En cas de mise à niveau d'WebSphere eXtreme Scale vers une nouvelle version ou si un groupe de correctifs est appliqué, les fichiers `wxsrestservice.war` file ou `wxsrestservice.ear` devront être mis à niveau manuellement vers la version installée dans ce répertoire.
- `base_servicerest/gettingstarted`

  Le répertoire `gettingstarted` contient un exemple simple montrant comment utiliser le service de données REST WebSphere eXtreme Scale avec une grille de données.

Procédure

Packagez et déployez le service de données REST.
Le service de données REST a été conçu en tant que module WAR autonome. Pour configurer le service de données REST, vous devez commencer par packager dans un fichier JAR ou dans un répertoire la configuration du service de données REST et les éventuels fichiers de configuration d'WebSphere eXtreme Scale. Ce package d'application est ensuite référencé par l'environnement d'exécution du serveur de conteneur Web. La figure suivante illustre les fichiers utilisés par le service de données REST d'eXtreme Scale.

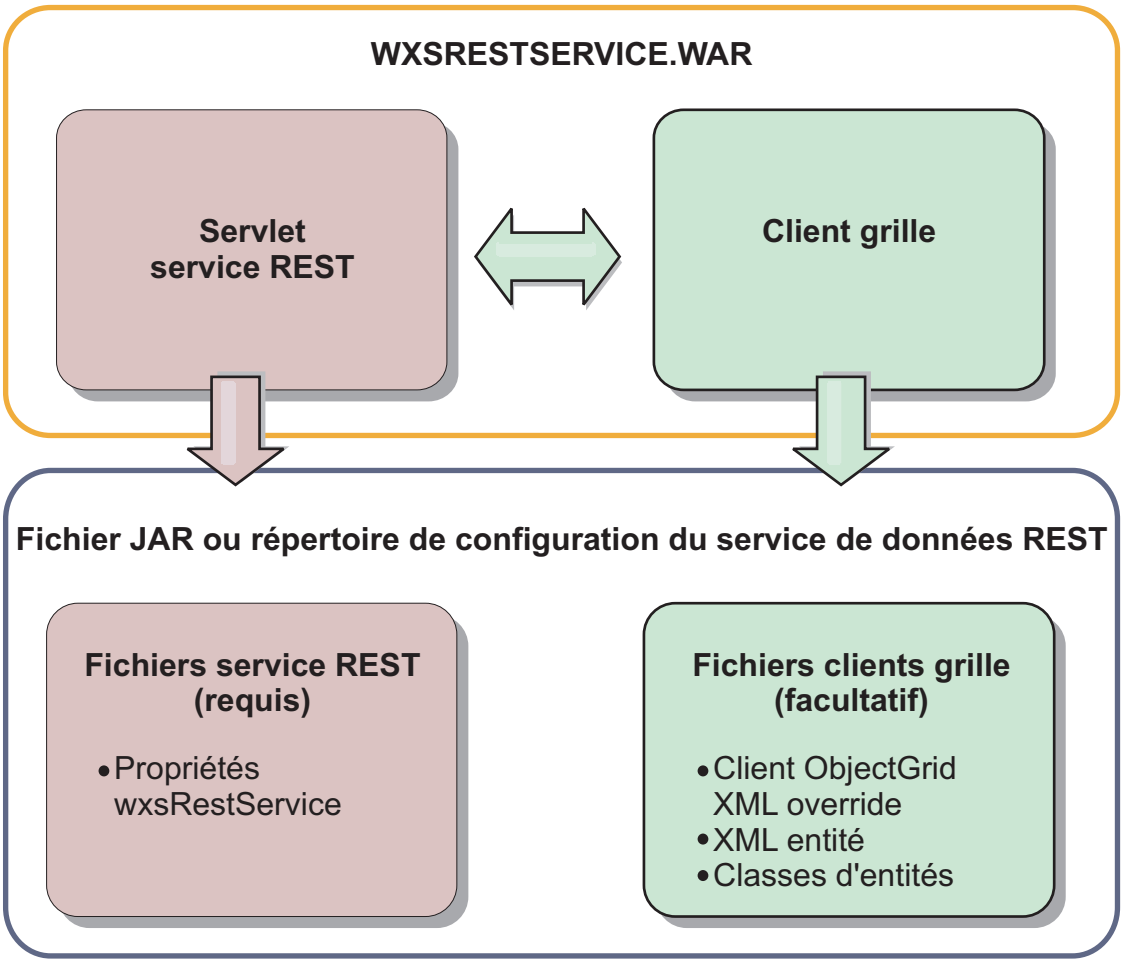


Figure 30. Fichiers du service de données REST d'WebSphere eXtreme Scale

Le fichier JAR de configuration du service REST ou le répertoire doit contenir le fichier suivant :
`wxsRestService.properties` : le fichier `wxsRestService.properties` comprend les options de configuration du service de données REST : points de contact du service de catalogue, noms d'ObjectGrid à exposer, options de suivi, etc. Voir Fichier de propriétés du service de données REST.
Les fichiers suivants du client ObjectGrid sont facultatifs :

- `META-INF/objectGridClient.xml` : le fichier XML de substitution de client ObjectGrid est utilisé pour se connecter à la grille de données distante. Par défaut, ce fichier n'est pas requis. En son absence, le service REST utilise la configuration du serveur en désactivant le cache proche.

  Le nom du fichier peut être remplacé à l'aide de la propriété de configuration du service de données REST objectGridClientXML. S'il est fourni, ce fichier XML doit inclure :

  1. Les ObjectGrids à exposer au service de données REST.
  2. Toute référence au fichier XML descripteur d'entité associé à chaque configuration ObjectGrid.
- `META-INF/`*`fichiers XML de descripteurs d'entités`* : un ou plusieurs fichiers XML de descripteurs d'entités ne sont requis que si le client doit remplacer la définition d'entité du client. Le fichier XML du descripteur d'entité doit être utilisé avec le fichier XML du descripteur d'entité de remplacement ObjectGrid par les clients.

- **Classes entité** Vous pouvez utiliser des classes entité annotées ou un fichier XML descripteur d'entité pour décrire les métadonnées d'entité. Le service REST ne nécessite les classes d'entités dans le chemin d'accès aux classes que si les serveurs eXtreme Scale sont configurés avec les classes de métadonnées d'entités. Aucun fichier XML de descripteur d'entité de remplacement par le client n'est utilisé.

  Voici un exemple avec le fichier de configuration minimum requise, où les entités sont définies dans XML sur les serveurs :

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  Le fichier de propriétés contient :

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  Un exemple avec une entité, des fichiers XML de remplacement et des classes entité :

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  Le fichier de propriétés contient :

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  ```
  com/acme/entities/Customer.class
  META-INF/objectGridClient.xml
  ```

  Le fichier XML descripteur du client ObjectGrid contient :

  ```
  <objectGrid name="CustomerGrid" entityMetadataXMLFile="emd.xml"/>
  META-INF/emd.xml
  ```

  Le fichier XML descripteur des métadonnées d'entité contient :

  ```
  <entity class-name="com.acme.entities.Customer" name="Customer"/>
  ```

Installation des ensembles eXtreme Scale

Java

WebSphere eXtreme Scale inclut des ensembles qui peuvent être installés dans une infrastructure OSGi Eclipse Equinox. Ces ensembles sont nécessaires pour démarrer les serveurs eXtreme Scale ou utiliser les clients eXtreme Scale dans OSGi. Vous pouvez installer les ensembles eXtreme Scale à l'aide de la console Equinox ou du fichier de configuration config.ini.

Avant de commencer

Cette tâche suppose que vous avez déjà installé les produits suivants :

- Infrastructure OSGi Eclipse Equinox
- Client ou serveur autonome eXtreme Scale

Pourquoi et quand exécuter cette tâche

eXtreme Scale inclut deux ensembles. Un seul des ensembles suivants est nécessaire dans une infrastructure OSGi :

objectgrid.jar
: L'ensemble de serveur est le fichier `objectgrid.jar`. Il est installé avec l'installation de serveur autonome eXtreme Scale et il est nécessaire pour exécuter les serveurs eXtreme Scale. Il peut être aussi utilisé pour exécuter les clients eXtreme Scale ou les mémoires caches internes locales. L'ID d'ensemble du fichier `objectgrid.jar` est

com.ibm.websphere.xs.server_<version>, où la version a le format <Version>.<Edition>.<Modification>. Par exemple, l'ensemble serveur pour cette édition est com.ibm.websphere.xs.server_8.5.0.

ogclient.jar
L'ensemble ogclient.jar est installé avec les installations client et autonomes eXtreme Scale et il est utilisé pour exécuter les clients eXtreme Scale ou les mémoires caches internes locales. L'ID d'ensemble du fichier ogclient.jar est com.ibm.websphere.xs.client_<version>, où la version a le format <Version>_<Edition>_<Modification. Par exemple, l'ensemble client pour cette édition est com.ibm.websphere.xs.server_8.5.0.

Pour plus d'informations sur le développement de plug-in eXtreme Scale, voir la rubrique API système et plug-in.

Installez l'ensemble serveur ou client eXtreme Scale dans l'infrastructure Eclipse Equinox OSGi à l'aide de la console Equinox.

Procédure

1. Démarrez l'infrastructure Eclipse Equinox avec la console activée. Par exemple :
   ```
   rép_base_java/bin/java -jar <equinox_root>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
2. Installez l'ensemble serveur ou client eXtreme Scale dans la console Equinox :
   ```
   osgi> install file:///<chemin_ensemble>
   ```
3. Equinox affiche l'ID d'ensemble du nouvel ensemble installé :
   ```
   Bundle id is 25
   ```
4. Démarrez l'ensemble dans la console Equinox, où <id> est l'ID affecté à l'ensemble lors de son installation :
   ```
   osgi>  start <id>
   ```
5. Extrayez l'état du service dans la console Equinox pour vérifier que l'ensemble a démarré. Par exemple :
   ```
   osgi> ss
   ```

   Lorsque l'ensemble a démarré correctement, il affiche l'état ACTIVE, par exemple :
   ```
   25      ACTIVE      com.ibm.websphere.xs.server_8.5.0
   ```

Installez l'ensemble serveur ou client eXtreme Scale dans l'infrastructure Eclipse Equinox OSGi à l'aide du fichier config.ini.

Procédure

1. Copiez l'ensemble client ou serveur eXtreme Scale (objectgrid.jar ou ogclient.jar) de <wxs_install_root>/ObjectGrid/lib vers le répertoire Eclipse Equinox, par exemple : <equinox_root>/plugins
2. Modifiez le fichier de configuration Eclipse Equinox config.ini et ajoutez l'ensemble à la propriété osgi.bundles, par exemple :
   ```
   osgi.bundles=\
   org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
   org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
   org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
   objectgrid.jar@1:start
   ```

   Important : Vérifiez qu'une ligne blanche existe après le dernier nom d'ensemble. Chaque ensemble est séparé par une virgule.

3. Démarrez l'infrastructure Eclipse Equinox avec la console activée. Par exemple :
   ```
   rép_base_java/bin/java -jar <equinox_root>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
4. Extrayez l'état du service dans la console Equinox pour vérifier que l'ensemble a démarré :
   ```
   osgi> ss
   ```
   Lorsque l'ensemble a démarré correctement, il affiche l'état ACTIVE, par exemple :
   ```
   25      ACTIVE      com.ibm.websphere.xs.server_8.5.0
   ```

Résultats

L'ensemble client ou serveur eXtreme Scale est installé et démarré dans l'infrastructure OSGi Eclipse Equinox.

Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server

Java

Vous pouvez installer WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client dans un environnement dans lequel WebSphere Application Server ou WebSphere Application Server Network Deployment est installé. Vous pouvez utiliser les fonctions existantes de WebSphere Application Server ou WebSphere Application Server Network Deployment pour améliorer vos applications eXtreme Scale .

Avant de commencer

- Vérifiez que le répertoire d'installation cible ne contient pas une installation de WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client.
- Arrêtez tous les processus en cours d'exécution dans votre environnement WebSphere Application Server ou WebSphere Application Server Network Deployment. Voir Utilitaires de ligne de commande pour plus d'informations sur les commandes **stopManager**, **stopNode** et **stopServer**.

  ATTENTION :
  Vérifiez que les processus actifs sont arrêtés. Si les processus en cours d'exécution ne sont pas arrêtés, l'installation se poursuit en créant des résultats imprévisibles. Sur certaines plateformes, l'installation peuvent rester dans un état indéterminé.

Important : Lorsque vous installez WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client, il doit se trouver dans le répertoire dans lequel vous avez installé WebSphere Application Server. For example, if you installed WebSphere Application Server in `C:\`*racine_was*, then you should also choose `C:`*racine_was* as the target directory for your WebSphere eXtreme Scale or WebSphere eXtreme Scale Client installation.

Pourquoi et quand exécuter cette tâche

Intégrez eXtreme Scale à WebSphere Application Server ou WebSphere Application Server Network Deployment pour appliquer les fonctions d'eXtreme Scale à vos applications Java Platform, Enterprise Edition. Les applications Java EE hébergent les grilles de données et y accèdent à l'aide d'une connexion client.

Procédure

- Pour installer WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client dans un environnement WebSphere Application Server version 8, procédez comme suit :
  1. Installez IBM Installation Manager. Pour plus d'informations, voir «Installation d'IBM Installation Manager à l'aide de l'interface graphique», à la page 206.
  2. A l'aide d'Installation Manager, installez l'offre de produit eXtreme Scale appropriée :
     - WebSphere eXtreme Scale for WebSphere Application Server version 8
     - WebSphere eXtreme Scale Client for WebSphere Application Server version 8

     Pour plus d'informations, voir «Installation du produit à l'aide de l'interface graphique», à la page 207.
  3. Téléchargez les référentiels WebSphere Application Server version 8 nécessaires à partir du site Passport Advantage. Pour plus d'informations, voir How to download WebSphere Application Server V8.0 from Passport Advantage Online.
  4. Installez WebSphere Application Server version 8 Pour plus d'informations, voir Installation du produit sur des systèmes d'exploitation distribués à l'aide de l'interface graphique.
- Pour installer WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client dans un environnement WebSphere Application Server version 7, procédez comme suit :
  1. Installez IBM Installation Manager. Pour plus d'informations, voir «Installation d'IBM Installation Manager à l'aide de l'interface graphique», à la page 206.
  2. A l'aide d'Installation Manager, installez l'offre de produit eXtreme Scale appropriée :
     - WebSphere eXtreme Scale for WebSphere Application Server version 7
     - WebSphere eXtreme Scale Client for WebSphere Application Server version 7

     Pour plus d'informations, voir «Installation du produit à l'aide de l'interface graphique», à la page 207.
  3. Installez WebSphere Application Server Version 7 en utilisant le programme d'installation InstallShield MultiPlatform (ISMP). Pour plus d'informations, voir Installation de votre environnement d'application.
  4. Ensuite, vous importez WebSphere Application Server Version 7 dans le gestionnaire d'installation pour terminer l'installation. L'importation de WebSphere Application Server version 7 dans Installation Manager vous permet de gérer et d'installer des groupes de correctifs pour le produit à partir d'un seul emplacement. Vous devez veiller à ce que les référentiels nécessaires soient configurés dans Installation Manager pour l'accès aux groupes de correctifs et aux mises à jour. Pour plus d'informations sur l'importation d'une installation existante de WebSphere Application Server 7 dans Installation Manager, voir IBM Installation Manager.

Que faire ensuite

- Commencez à configurer votre installation WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client. Pour plus d'informations, voir «Premières étapes après l'installation», à la page 252.

Installation des groupes de correctifs à l'aide d'IBM Installation Manager

Java

Vous pouvez utiliser IBM Installation Manager pour mettre à jour le produit à l'aide des groupes de correctifs disponibles pour les offres de produit WebSphere eXtreme Scale . Les groupes de correctifs peuvent être installés à partir de l'interface graphique ou de la ligne de commande ou à l'aide de fichiers de réponses.

Installation des groupes de correctifs à l'aide de l'interface graphique

Java

Vous pouvez mettre à niveau ce produit vers une version ultérieure à l'aide de l'assistant d'IBM Installation Manager.

Avant de commencer

Prenez contact avec le centre de support logiciel IBM pour obtenir des informations sur les mises à niveau pour WebSphere eXtreme Scale autonome ou les offres de produit WebSphere eXtreme Scale for WebSphere Application Server. Les informations les plus récentes sont disponibles dans le centre de support logiciel IBM et dans Fix Central.

IBM Installation Manager sert à appliquer un module de maintenance de produit aux offres de produit suivantes :

- WebSphere eXtreme Scale dans un environnement autonome
- WebSphere eXtreme Scale Client dans un environnement autonome
- WebSphere eXtreme Scale for WebSphere Application Server Version 7.0
- WebSphere eXtreme Scale for WebSphere Application Server Version 8.0
- WebSphere eXtreme Scale Client for WebSphere Application Server Version 7.0
- WebSphere eXtreme Scale Client for WebSphere Application Server Version 8.0

Assurez-vous que l'emplacement du référentiel de service Web ou local est répertorié et vérifié ou que l'option **Search service repositories during installation and updates** est sélectionnée dans le panneau Repositories de la page des préférences Installation Manager. Pour plus d'informations sur l'utilisation des référentiels de service avec Installation Manager, accédez au centre de documentation d'IBM Installation Manager Version 1.5.

Pourquoi et quand exécuter cette tâche

Restriction : Vous ne pouvez pas utiliser Installation Manager pour mettre à niveau une installation et ajouter ou supprimer la fonction de profil WebSphere Application Server complète .

Procédure

1. Arrêtez tous les processus en cours d'exécution dans votre environnement.

- Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
- Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.

2. Ouvrez une session sur votre système.
3. Arrêtez tous les serveurs et toutes les applications présents dans l'installation de WebSphere Application Server que vous voulez mettre à jour.
4. Démarrez Installation Manager.
5. Cliquez sur **Mettre à jour**.

   Remarque : Si vous êtes invité à vous authentifier, utilisez l'ID IBM et le mot de passe vous permettant d'accéder aux sites Web protégés d'IBM.
6. Sélectionnez le groupe de packages à mettre à jour.

   Conseil : Si vous sélectionnez **Update all** (Tout mettre à jour), Installation Manager recherche dans tous les référentiels ajoutés et prédéfinis les mises à jour de tous les groupes de packages qu'il a installés. Utilisez cette fonction uniquement si vous contrôlez intégralement quels correctifs sont contenus dans les référentiels cible. Si vous créez et pointez vers un ensemble de référentiels personnalisés incluant uniquement les correctifs spécifiques que vous souhaitez installer, vous devez pouvoir utiliser cette fonction en toute confiance. Si vous activez la recherche dans les référentiels de service ou que vous installez les correctifs directement à partir d'autres référentiels Web en ligne, vous ne souhaitez peut-être pas sélectionner cette option afin de pouvoir sélectionner uniquement les correctifs à installer pour chaque offre dans des panneaux ultérieurs.
7. Cliquez sur **Suivant**.
8. Sélectionnez la version que vous voulez mettre à jour :
   - WebSphere eXtreme Scale dans un environnement autonome
   - WebSphere eXtreme Scale Client dans un environnement autonome
   - WebSphere eXtreme Scale for WebSphere Application Server version 7
   - WebSphere eXtreme Scale for WebSphere Application Server version 8
   - WebSphere eXtreme Scale Client for WebSphere Application Server version 7
   - WebSphere eXtreme Scale Client for WebSphere Application Server version 8
9. Sélectionnez les correctifs que vous voulez installer.

   Les correctifs recommandés sont sélectionnés par défaut.

   Vous pouvez choisir de n'afficher que les correctifs recommandés et de masquer les autres.
10. Cliquez sur **Suivant**.
11. Acceptez les modalités du contrat de licence et cliquez sur **Suivant**.
12. Sélectionnez les fonctions facultatives que vous souhaitez intégrer à votre installation mise à jour.
13. Lisez le récapitulatif et cliquez sur **Mettre à jour**.
    - Si l'installation a réussi, le programme affichera un message pour confirmer le succès de l'installation.

- Si l'installation n'aboutit pas, cliquez sur **View Log File** (Afficher le fichier journal) pour corriger l'erreur.

14. Cliquez sur **Terminer**.
15. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.

Installation des groupes de correctifs à l'aide de la ligne de commande

Java

Vous pouvez utiliser IBM Installation Manager à partir de la ligne de commande pour mettre à jour le produit à l'aide des groupes de correctifs disponibles pour les offres de produit WebSphere eXtreme Scale.

Avant de commencer

. Prenez contact avec le centre de support logiciel IBM pour obtenir des informations sur les mises à niveau pour WebSphere eXtreme Scale autonome ou les offres de produit WebSphere eXtreme Scale for WebSphere Application Server. Les informations les plus récentes sont disponibles dans le centre de support logiciel IBM et dans Fix Central.

IBM Installation Manager sert à appliquer un module de maintenance de produit aux offres de produit suivantes :

- WebSphere eXtreme Scale dans un environnement autonome
- WebSphere eXtreme Scale Client dans un environnement autonome
- WebSphere eXtreme Scale for WebSphere Application Server version 7
- WebSphere eXtreme Scale for WebSphere Application Server version 8
- WebSphere eXtreme Scale Client for WebSphere Application Server version 7
- WebSphere eXtreme Scale Client for WebSphere Application Server version 8

Pourquoi et quand exécuter cette tâche

Restriction : Vous ne pouvez pas utiliser Installation Manager pour mettre à niveau une installation et ajouter ou supprimer la fonction de profil WebSphere Application Server complète .

Procédure

1. Pour obtenir la liste des groupes de correctifs et des correctifs temporaires disponibles pour WebSphere eXtreme Scale 8.5, ainsi que des informations spécifiques sur chaque correctif, procédez comme suit :
   a. Accédez à Fix Central.
   b. Sélectionnez **WebSphere** comme groupe de produits.
   c. Sélectionnez WebSphere eXtreme Scale comme produit.
   d. Sélectionnez **8.5** pour la version installée.
   e. Sélectionnez votre système d'exploitation comme plateforme, puis cliquez sur **Continuer**.
   f. Sélectionnez **Rechercher des correctifs**, puis cliquez sur **Continuer**.
   g. Cliquez sur **Informations complémentaires** sous chaque correctif pour afficher des informations relatives au correctif.
   h. **Recommandation :** Notez le nom du groupe de correctifs que vous souhaitez installer.

2. Mettez à niveau WebSphere eXtreme Scale version 8.5 à l'aide du groupe de correctifs en suivant la procédure ci-après.
   - Téléchargez sur Fix Central le fichier contenant le groupe de correctifs, puis utilisez la mise à jour locale.

     Vous pouvez télécharger à partir de Fix Central un fichier compressé contenant le groupe de correctifs. Chaque fichier de groupe de correctifs compressé contient un référentiel Installation Manager pour le groupe de correctifs et comporte généralement une extension .zip. Après avoir téléchargé et extrait le fichier de groupe de correctifs, utilisez Installation Manager pour mettre à jour WebSphere Application Server version 8.x à l'aide du groupe de correctifs.

     a. Pour télécharger le groupe de correctifs, procédez comme suit :
        1) Accédez à Fix Central.
        2) Sélectionnez **WebSphere** comme groupe de produits.
        3) Sélectionnez **WebSphere eXtreme Scale** comme produit.
        4) Sélectionnez **8.5** pour la version installée.
        5) Sélectionnez votre système d'exploitation comme plateforme, puis cliquez sur **Continuer**.
        6) Sélectionnez **Rechercher des correctifs**, puis cliquez sur **Continuer**.
        7) Sélectionnez le groupe de correctifs à télécharger, puis cliquez sur **Continuer**.
        8) Sélectionnez vos options de téléchargement, puis cliquez sur **Continuer**.
        9) Cliquez sur **J'accepte** pour accepter les dispositions.
        10) Cliquez sur **Télécharger maintenant** pour télécharger le groupe de correctifs.
        11) Transférez le fichier compressé au format binaire vers le système sur lequel il sera installé.
        12) Extrayez les fichiers de référentiel compressés dans un répertoire de votre système.

     b. Pour installer un groupe de correctifs à partir d'un fichier téléchargé, procédez comme suit :
        1) Ouvrez une session sur votre système.
        2) Arrêtez tous les processus en cours d'exécution dans votre environnement. Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
        3) Accédez au répertoire *binaires_Installation_Manager*/eclipse/tools, où *binaires_Installation_Manager* représente le répertoire racine d'installation d'Installation Manager.
        4) Installez le groupe de correctifs.

           UNIX Linux

           ```
           ./imcl install ID_offre_version_offre,ID_fonction_facultative
             -installationDirectory emplacement_installation_produit
             -repositories emplacement_fichiers_étendus
             -acceptLicense
           ```

           Windows

           ```
           imcl.exe install ID_offre_version_offre,ID_fonction_facultative
             -installationDirectory emplacement_installation_produit
             -repositories emplacement_fichiers_étendus
             -acceptLicense
           ```

Conseils :

– L'*ID_offre* représente l'ID d'offre répertorié dans «ID des offres de produit WebSphere eXtreme Scale», à la page 192.

– La *version_offre*, qui peut, le cas échéant, être associée à l'ID offre à l'aide d'un trait de soulignement, est une version spécifique de l'offre à installer (`8.5.0.20110503_0200`, par exemple).

  - Si la *version_offre* n'est **pas** spécifiée, la version la plus récente de l'offre et **tous** les correctifs temporaires de cette version sont installés.
  - Si la *version_offre* est spécifiée, la version indiquée de l'offre est installée et **aucun** correctif temporaire de cette version n'est installé.

  La version de l'offre peut être rattachée à la fin de l'ID offre à l'aide d'un trait de soulignement lorsque vous exécutez la commande suivante au niveau du référentiel :

  ```
  imcl listAvailablePackages -repositories référentiel_source
  ```

– Vous pouvez également spécifier `none`, `recommended` ou `all` avec l'argument `-installFixes` afin d'indiquer les correctifs temporaires que vous souhaitez installer avec l'offre.

  - Si la version de l'offre n'est **pas** spécifiée, l'option `-installFixes` est paramétrée par défaut sur `all`.
  - Si la version de l'offre est spécifiée, l'option `-installFixes` est paramétrée par défaut sur `none`.

– Vous pouvez ajouter une liste de fonctions séparées par des virgules. Si aucune liste de fonctions n'est spécifiée, les fonctions par défaut sont installées.

5) **Facultatif :** Répertoriez tous les packages installés pour vérifier l'installation :

   UNIX Linux

   ```
   ./imcl listInstalledPackages -long
   ```

   Windows

   ```
   imcl.exe listInstalledPackages -long
   ```

Installation des groupes de correctifs à l'aide d'un fichier de réponses

Java

Vous pouvez mettre à jour ce produit vers une version ultérieure en utilisant IBM Installation Manager avec un fichier de réponses.

Avant de commencer

Conseil : Comme alternative à la procédure décrite dans cet article, Installation Manager vous permet d'utiliser la commande **`updateAll`** dans un fichier de réponses ou sur la ligne de commande pour rechercher et mettre à jour tous les packages installés. Utilisez cette commande uniquement si vous contrôlez intégralement quels correctifs sont contenus dans les référentiels cible. Si vous créez et pointez vers un ensemble de référentiels personnalisés incluant uniquement les correctifs spécifiques que vous souhaitez installer, vous devez pouvoir utiliser cette commande en toute confiance. Si vous activez la recherche dans les référentiels de service ou que vous installez les correctifs directement à

partir d'autres référentiels Web en ligne, vous ne souhaitez peut-être pas sélectionner cette option afin de pouvoir sélectionner uniquement les correctifs à installer à l'aide de l'option -installFixes avec la commande **install** sur la ligne de commande ou à l'aide de l'attribut installFixes dans un fichier de réponses.

Procédure

1. Pour obtenir la liste des groupes de correctifs et des correctifs temporaires disponibles pour WebSphere eXtreme Scale et des informations sur chaque correctif, procédez comme suit.
   a. Accédez à Fix Central.
   b. Sélectionnez **WebSphere** comme groupe de produits.
   c. Sélectionnez WebSphere eXtreme Scale comme produit.
   d. Sélectionnez **8.x** pour la version installée.
   e. Sélectionnez votre système d'exploitation comme plateforme, puis cliquez sur **Continuer**.
   f. Sélectionnez **Rechercher des correctifs**, puis cliquez sur **Continuer**.
   g. Cliquez sur **Informations complémentaires** sous chaque correctif pour afficher des informations relatives au correctif.
   h. **Recommandation :** Notez le nom du groupe de correctifs que vous souhaitez installer.
2. Mettez à jour WebSphere eXtreme Scale avec le correctif en procédant comme suit.
   - Téléchargez sur Fix Central le fichier contenant le groupe de correctifs, puis utilisez la mise à jour locale.

     Vous pouvez télécharger à partir de Fix Central un fichier compressé contenant le groupe de correctifs. Chaque fichier de groupe de correctifs compressé contient un référentiel Installation Manager pour le groupe de correctifs et comporte généralement une extension .zip. Après avoir téléchargé et extrait le fichier de groupe de correctifs, utilisez le gestionnaire d'installation pour mettre à jour WebSphere eXtreme Scale avec le groupe de correctifs.

     a. Pour télécharger le groupe de correctifs, procédez comme suit :
        1) Accédez à Fix Central.
        2) Sélectionnez **WebSphere** comme groupe de produits.
        3) Sélectionnez **WebSphere eXtreme Scale** comme produit.
        4) Sélectionnez **8.6** comme version installée.
        5) Sélectionnez votre système d'exploitation comme plateforme, puis cliquez sur **Continuer**.
        6) Sélectionnez **Rechercher des correctifs**, puis cliquez sur **Continuer**.
        7) Sélectionnez le groupe de correctifs à télécharger, puis cliquez sur **Continuer**.
        8) Sélectionnez vos options de téléchargement, puis cliquez sur **Continuer**.
        9) Cliquez sur **J'accepte** pour accepter les dispositions.
        10) Cliquez sur **Télécharger maintenant** pour télécharger le groupe de correctifs.
        11) Transférez le fichier compressé au format binaire vers le système sur lequel il sera installé.
        12) Extrayez les fichiers de référentiel compressés dans un répertoire de votre système.

b. Effectuez les actions suivantes :

1) Ouvrez une session sur votre système.
2) Si le référentiel requiert un nom d'utilisateur et un mot de passe, créez un fichier de clés pour accéder à ce référentiel.

   Pour plus d'informations sur la création d'un fichier de clés pour Installation Manager, reportez-vous au centre de documentation d'IBM Installation Manager version 1.5.

   Conseil : Lors de la création d'un fichier de clés, ajoutez `/repository.config` à la fin de l'emplacement d'URL de référentiel si la commande **imutilsc** ne peut pas trouver l'URL indiquée.
3) Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, consultez le document sur les utilitaires de ligne de commande.
4) Accédez au répertoire *binaires_Installation_Manager*`/eclipse/tools`, où *binaires_Installation_Manager* représente le répertoire racine d'installation d'Installation Manager.
5) Installez le groupe de correctifs à l'aide d'un fichier de réponses.

   Par exemple :

   – Windows **Administrateur ou non administrateur :**

   ```
   imcl.exe -acceptLicense
     input C:\temp\update_response_file.xml
     -log C:\temp\update_log.xml
     -keyring C:\IM\im.keyring
   ```

   – UNIX Linux **Administrateur :**

   ```
   ./imcl -acceptLicense
     input /var/temp/update_response_file.xml
     -log /var/temp/update_log.xml
     -keyring /var/IM/im.keyring
   ```

   – UNIX Linux **Non administrateur :**

   ```
   ./imcl -acceptLicense
     input rép_base_utilisateur/var/temp/update_response_file.xml
     -log rép_base_utilisateur/var/temp/update_log.xml
     -keyring rép_base_utilisateur/var/IM/im.keyring
   ```

Désinstallation des groupes de correctifs à l'aide d'IBM Installation Manager

Java

Vous pouvez utiliser IBM Installation Manager pour rétrograder les offres de produit WebSphere eXtreme Scale vers une version antérieure. Vous pouvez désinstaller les groupes de correctifs à partir de l'interface graphique ou de la ligne de commande ou à l'aide de fichiers de réponses.

Désinstallation des groupes de correctifs à l'aide de l'interface graphique

Java

Vous pouvez rétrograder ce produit vers une version antérieure à l'aide de l'interface graphique d'IBM Installation Manager.

Avant de commencer

Pendant le processus de rétrogradation, Installation Manager doit accéder aux fichiers de la version précédente du module. Par défaut, ces fichiers sont enregistrés sur votre ordinateur lors de l'installation d'un module. Si vous modifiez le paramétrage par défaut ou que supprimez les fichiers enregistrés, Installation Manager a besoin d'accéder au référentiel utilisé pour l'installation de la version précédente.

Pourquoi et quand exécuter cette tâche

Restriction : Vous ne pouvez pas utiliser Installation Manager pour rétrograder une installation et ajouter ou supprimer une fonction.

Procédure

1. Arrêtez tous les processus en cours d'exécution dans votre environnement.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
2. Démarrez Installation Manager.
3. Cliquez sur **Rétrograder**.
4. Sélectionnez le groupe de packages à rétrograder.
5. Cliquez sur **Next**.
6. Sélectionnez la version que vous voulez rétrograder.
7. Cliquez sur **Next**.
8. Lisez le récapitulatif et cliquez sur **Rétrograder**.
   - Si la rétrogradation aboutit, le programme affiche un message pour confirmer le succès de la rétrogradation.
   - Dans le cas contraire, cliquez sur **View Log File** pour corriger l'erreur.
9. Cliquez sur **Terminer**.
10. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.

Désinstallation des groupes de correctifs à l'aide de la ligne de commande

Java

Vous pouvez rétrograder ce produit vers une version antérieure en utilisant IBM Installation Manager à partir de la ligne de commande.

Avant de commencer

Restriction : Pour utiliser cette procédure, Installation Manager version 1.5 ou ultérieure doit être installé sur votre système.

Pendant le processus de rétrogradation, Installation Manager doit accéder aux fichiers de la version précédente du module. Par défaut, ces fichiers sont enregistrés sur votre ordinateur lors de l'installation d'un module. Si vous modifiez le paramétrage par défaut ou que supprimez les fichiers enregistrés, Installation Manager a besoin d'accéder au référentiel utilisé pour l'installation de la version précédente.

Pourquoi et quand exécuter cette tâche

Restriction : Vous ne pouvez pas utiliser Installation Manager pour rétrograder une installation et ajouter ou supprimer la fonction de profil complète WebSphere Application Server .

Procédure

1. Facultatif : Si le référentiel requiert un nom d'utilisateur et un mot de passe, créez un fichier de clés pour accéder à ce référentiel.

   Pour plus d'informations sur la création d'un fichier de clés pour Installation Manager, reportez-vous au centre de documentation d'IBM Installation Manager version 1.5.

   Conseil : Lors de la création d'un fichier de clés, ajoutez `/repository.config` à la fin de l'emplacement d'URL de référentiel si la commande **imutilsc** ne peut pas trouver l'URL indiquée.
2. Ouvrez une session sur votre système.
3. Arrêtez tous les processus en cours d'exécution dans votre environnement.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
4. Accédez au sous-répertoire `eclipse/tools` dans le répertoire où vous avez installé Installation Manager.
5. Utilisez la commande **imcl** pour rétrograder le produit.

   UNIX Linux

   ```
   ./imcl rollback ID_offre_version_offre
     -repositories référentiel_source
     -installationDirectory répertoire_installation
     -preferences clé_préférence=valeur
     -properties clé_propriété=valeur
     -keyring fichier_de_clés -password mot_de_passe
     -acceptLicense
   ```

   Windows

   ```
   imcl.exe rollback ID_offre_version_offre
     -repositories référentiel_source
     -installationDirectory répertoire_installation
     -preferences clé_préférence=valeur
     -properties clé_propriété=valeur
     -keyring fichier_de_clés -password mot_de_passe
     -acceptLicense
   ```

   Conseils :

   - L'*ID_offre* représente l'ID d'offre répertorié dans «ID des offres de produit WebSphere eXtreme Scale», à la page 192.

- La *version_offre*, qui peut, le cas échéant, être associée à l'ID offre à l'aide d'un trait de soulignement, est une version spécifique de l'offre à rétrograder (`8.5.0.20110503_0200`, par exemple).
  - Si la *version_offre* n'est **pas** spécifiée, l'installation est rétrogradée à la version précédemment installée de l'offre et **tous** les correctifs temporaires de cette version sont installés.
  - Si la *version_offre* est spécifiée, l'installation est rétrogradée à la version précédente indiquée de l'offre et **aucun** correctif temporaire de cette version n'est installé.

  La version de l'offre peut être rattachée à la fin de l'ID offre à l'aide d'un trait de soulignement dans la section Package du rapport qui est généré lors de l'exécution de la commande **historyInfo** ou **genHistoryReport** à partir du répertoire *racine_serveur_app*`/bin`.

Pour plus d'informations sur l'utilisation d'Installation Manager, accédez au centre de documentation d'IBM Installation Manager Version 1.5.

6. Facultatif : Répertoriez tous les packages installés pour vérifier la rétrogradation.

   UNIX Linux

   ```
   ./imcl listInstalledPackages -long
   ```

   Windows

   ```
   imcl.exe listInstalledPackages -long
   ```

Désinstallation des groupes de correctifs à l'aide de fichiers de réponses

Java

Vous pouvez rétrograder ce produit vers une version antérieure en utilisant IBM Installation Manager avec un fichier de réponses.

Avant de commencer

Pendant le processus de rétrogradation, Installation Manager doit accéder aux fichiers de la version précédente du module. Par défaut, ces fichiers sont enregistrés sur votre ordinateur lors de l'installation d'un module. Si vous modifiez le paramétrage par défaut ou que supprimez les fichiers enregistrés, Installation Manager a besoin d'accéder au référentiel utilisé pour l'installation de la version précédente.

Pourquoi et quand exécuter cette tâche

Restriction : Vous ne pouvez pas utiliser Installation Manager pour rétrograder une installation et ajouter ou supprimer la fonction de profil complète WebSphere Application Server .

Procédure

1. Facultatif : Si le référentiel requiert un nom d'utilisateur et un mot de passe, créez un fichier de clés pour accéder à ce référentiel.

   Pour plus d'informations sur la création d'un fichier de clés pour Installation Manager, reportez-vous au centre de documentation d'IBM Installation Manager version 1.5.

Conseil : Lors de la création d'un fichier de clés, ajoutez `/repository.config` à la fin de l'emplacement d'URL de référentiel si la commande **imutilsc** ne peut pas trouver l'URL indiquée.

2. Ouvrez une session sur votre système.
3. Arrêtez tous les processus en cours d'exécution dans votre environnement.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
4. Utilisez un fichier de réponses pour rétrograder le produit.

   Accédez au sous-répertoire `eclipse/tools` du répertoire dans lequel vous avez installé Installation Manager, puis rétrogradez le produit.

   Par exemple :

   - Windows **Administrateur ou non administrateur :**

     ```
     imcl.exe
       input C:\temp\rollback_response_file.xml
       -log C:\temp\rollback_log.xml
       -keyring C:\IM\im.keyring
     ```

   - UNIX Linux **Administrateur :**

     ```
     ./imcl
       input /var/temp/rollback_response_file.xml
       -log /var/temp/rollback_log.xml
       -keyring /var/IM/im.keyring
     ```

   - UNIX Linux **Non administrateur :**

     ```
     ./imcl
       input rép_base_utilisateur/var/temp/rollback_response_file.xml
       -log rép_base_utilisateur/var/temp/rollback_log.xml
       -keyring rép_base_utilisateur/var/IM/im.keyring
     ```

   Remarque : Le programme peut générer des instructions post-installation importantes dans la sortie standard.

   Pour plus d'informations sur l'utilisation d'Installation Manager, accédez au centre de documentation d'IBM Installation Manager Version 1.5.
5. Facultatif : Répertoriez tous les packages installés pour vérifier la rétrogradation.

   UNIX Linux

   ```
   ./imcl listInstalledPackages -long
   ```

   Windows

   ```
   imcl.exe listInstalledPackages -long
   ```

Désinstallation du produit à l'aide d'IBM Installation Manager

Java

Utilisez IBM Installation Manager pour désinstaller les offres de produit WebSphere eXtreme Scale .

Désinstallation du produit à l'aide de l'interface graphique

Java

Vous pouvez utiliser la console de l'assistant d'IBM Installation Manager pour désinstaller le produit .

Avant de commencer

Vous devez supprimer l'argument WebSphere eXtreme Scale de tous les profils WebSphere Application Server avant de désinstaller WebSphere eXtreme Scale. Vous ne pourrez pas exécuter l'annulation d'extension après avoir désinstallé WebSphere eXtreme Scale. Utilisez la commande manageprofiles pour réduire les profils existants dans un environnement WebSphere eXtreme Scale. Pour plus d'informations, voir «Commande **manageprofiles**», à la page 245.

Procédure

1. Désinstallez le produit.
   a. Arrêtez tous les processus en cours d'exécution dans votre environnement.
      - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
      - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.

   b. Démarrez Installation Manager.
   c. Cliquez sur **Désinstaller**.
   d. Dans la fenêtre **Désinstaller des packages**, effectuez les actions suivantes.
      1) Sélectionnez l'une des offres de produit suivantes ainsi que la version appropriée :
         - WebSphere eXtreme Scale dans un environnement autonome
         - WebSphere eXtreme Scale Client dans un environnement autonome
         - WebSphere eXtreme Scale for WebSphere Application Server Version 6
         - WebSphere eXtreme Scale for WebSphere Application Server version 7
         - WebSphere eXtreme Scale for WebSphere Application Server version 8
         - WebSphere eXtreme Scale Client for WebSphere Application Server version 6
         - WebSphere eXtreme Scale Client for WebSphere Application Server version 7
         - WebSphere eXtreme Scale Client for WebSphere Application Server version 8

      2) Cliquez sur **Suivant**.

   e. Si l'assistant de désinstallation affiche la liste des profils étendus, WebSphere Application Server, vous devez annuler l'extension de ces profils pour pouvoir poursuivre la désinstallation.
   f. Examinez les informations récapitulatives.
   g. Cliquez sur **Désinstaller**.
      - Si la désinstallation a réussi, le programme affichera un message pour confirmer le succès de l'opération.
      - Dans le cas contraire, cliquez sur **View log** (Afficher le journal) pour corriger l'erreur.

   h. Cliquez sur **Terminer**.
   i. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.
2. Facultatif : Désinstallez IBM Installation Manager.

Important : Avant de pouvoir désinstaller IBM Installation Manager, vous devez désinstaller tous les packages installés par Installation Manager.

Pour plus d'informations sur cette procédure, accédez au centre de documentation d'IBM Installation Manager Version 1.5.

Désinstallation du produit à partir de la ligne de commande

Java

Vous pouvez désinstaller le produit à l'aide d'IBM Installation Manager à partir de la ligne de commande.

Avant de commencer

Vous devez supprimer l'argument WebSphere eXtreme Scale de tous les profils WebSphere Application Server avant de désinstaller WebSphere eXtreme Scale. Vous ne pourrez pas exécuter l'annulation d'extension après avoir désinstallé WebSphere eXtreme Scale. Utilisez la commande manageprofiles pour réduire les profils existants dans un environnement WebSphere eXtreme Scale. Pour plus d'informations, voir «Commande **manageprofiles**», à la page 245.

Procédure

1. Ouvrez une session sur votre système.
2. Arrêtez tous les processus en cours d'exécution dans votre environnement.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
3. Accédez au sous-répertoire `eclipse/tools` dans le répertoire où vous avez installé Installation Manager.
4. Utilisez la commande **imcl** pour désinstaller le produit.

   Windows

   ```
   imcl.exe uninstall com.ibm.websphere.v85,ID_fonction_facultative
     -installationDirectory répertoire_installation
   ```

   UNIX Linux

   ```
   ./imcl uninstall com.ibm.websphere.v85,ID_fonction_facultative
     -installationDirectory répertoire_installation
   ```

   Conseils :

   - L'*ID_offre* représente l'ID d'offre répertorié dans «ID des offres de produit WebSphere eXtreme Scale», à la page 192.
   - Vous pouvez supprimer une liste de fonctions séparées par des virgules (ID fonction). Exemple,

     ```
     imcl uninstall com.ibm.websphere.WXS.v85,xs.console.feature,xs.samples.feature
     ```

     - `client` indique la fonction client autonome
     - `server` indique la fonction serveur autonome
     - `console` indique la console de surveillance Web
     - `samples` indique les exemples
   - Si aucune liste de fonctions n'est spécifiée, la totalité du produit est désinstallée.

Pour plus d'informations, accédez au d'centre de documentation d'IBM Installation Manager Version 1.5.

5. Si le processus de désinstallation affiche la liste des profils WebSphere Application Server étendus, vous devez annuler l'extension de ces profils pour pouvoir poursuivre la désinstallation.
6. Facultatif : Désinstallez IBM Installation Manager.

   Important : Avant de pouvoir désinstaller IBM Installation Manager, vous devez désinstaller tous les packages installés par Installation Manager.

   Pour plus d'informations sur l'utilisation du script de désinstallation pour exécuter cette procédure, consultez le centre de documentation d'IBM Installation Manager version 1.5.

Désinstallation du produit à l'aide de fichiers de réponses

Java

Vous pouvez désinstaller le produit en utilisant IBM Installation Manager avec des fichiers de réponses.

Avant de commencer

Vous devez supprimer l'extension WebSphere eXtreme Scale de tous les profils WebSphere Application Server avant de désinstaller WebSphere eXtreme Scale. Vous ne pourrez pas exécuter l'annulation d'extension après avoir désinstallé WebSphere eXtreme Scale. Utilisez la commande manageprofiles pour réduire les profils existants dans un environnement WebSphere eXtreme Scale. Pour plus d'informations, voir «Commande **manageprofiles**», à la page 245.

Facultatif : Effectuez ou enregistrez l'installation d'Installation Manager et l'installation du produit dans un registre d'installation temporaire sur l'un de vos systèmes de manière à pouvoir enregistrer la désinstallation dans ce registre temporaire et non dans le registre standard ou est installé Installation Manager.

Pourquoi et quand exécuter cette tâche

Avec Installation Manager, vous pouvez utiliser des fichiers de réponses pour désinstaller le produit de plusieurs manières. Vous pouvez enregistrer un fichier de réponses à l'aide de l'interface graphique tel que décrit dans la procédure suivante, ou bien vous pouvez générer un nouveau fichier de réponses à la main ou en modifiant un exemple .

Procédure

1. Arrêtez tous les processus en cours d'exécution dans votre environnement.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement eXtreme Scale autonome, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server, voir le document sur les utilitaires de ligne de commande.
2. Facultatif : **Enregistrez un fichier de réponse pour désinstaller le produit :** Sur l'un de vos systèmes, effectuez les actions suivantes pour enregistrer un fichier de réponse qui vous permettra de désinstaller le module de fonctions:

a. A partir d'une ligne de commande, passez au sous-répertoire eclipse du répertoire dans lequel vous avez installé Installation Manager.
b. Démarrez Installation Manager à partir de la ligne de commande en utilisant l'option d'enregistrement -record.

   Par exemple :

   - Windows **Administrateur ou non administrateur :**

     ```
     IBMIM.exe -skipInstall "C:\temp\imRegistry"
       -record C:\temp\uninstall_response_file.xml
     ```

   - UNIX Linux **Administrateur :**

     ```
     ./IBMIM -skipInstall /var/temp/imRegistry
       -record /var/temp/uninstall_response_file.xml
     ```

   - UNIX Linux **Non administrateur :**

     ```
     ./IBMIM -skipInstall racine_utilisateur/var/temp/imRegistry
       -record racine_utilisateur/var/temp/uninstall_response_file.xml
     ```

   Conseil : Si vous choisissez d'utiliser le paramètre -skipInstall avec un registre d'installation temporaire créé en suivant les instructions *Avant de commencer*, Installation Manager utilise le registre d'installation temporaire lors de l'enregistrement du fichier de réponses. Il est important de savoir que dès lors que le paramètre -skipInstall est spécifié, aucun module de fonction n'est installé ou désinstallé. Toutes les actions effectuées dans Installation Manager mettent tout simplement à jour les données d'installation stockées dans le registre temporaire indiqué. Une fois que le fichier de réponses est généré, il peut être utilisé pour désinstaller le produit : il supprime les fichiers du produit et met à jour le registre d'installation standard.

   L'opération -skipInstall ne doit pas être utilisée sur l'emplacement de données de l'agent en cours utilisé par Installation Manager. Cette opération n'est pas prise en charge. Utilisez un nouvel emplacement inscriptible et utilisez à nouveau l'emplacement pour des sessions d'enregistrement ultérieures.

   Pour plus d'informations, voir le centre de documentation d'IBM Installation Manager Version 1.5.
c. Cliquez sur **Désinstaller**.
d. Dans la fenêtre **Désinstaller des packages**, effectuez les actions suivantes.
   1) Sélectionnez l'une des offres de produit suivantes ainsi que la version appropriée :
      - WebSphere eXtreme Scale dans un environnement autonome
      - WebSphere eXtreme Scale Client dans un environnement autonome
      - WebSphere eXtreme Scale for WebSphere Application Server version 6
      - WebSphere eXtreme Scale for WebSphere Application Server version 7
      - WebSphere eXtreme Scale for WebSphere Application Server version 8
      - WebSphere eXtreme Scale Client for WebSphere Application Server version 6
      - WebSphere eXtreme Scale Client for WebSphere Application Server version 7
      - WebSphere eXtreme Scale Client for WebSphere Application Server version 8
   2) Cliquez sur **Suivant**.

3) Cliquez sur **Suivant**.

e. Examinez les informations récapitulatives.

f. Cliquez sur **Désinstaller**.

- Si la désinstallation a réussi, le programme affichera un message pour confirmer le succès de l'opération.
- Dans le cas contraire, cliquez sur **View log** (Afficher le journal) pour corriger l'erreur.

g. Cliquez sur **Terminer**.

h. Cliquez sur **Fichier > Quitter** pour fermer Installation Manager.

3. **Utilisez le fichier de réponses pour désinstaller le produit :** A partir d'une ligne de commande sur chacun des systèmes desquels vous souhaitez désinstaller le produit, accédez au sous-répertoire `eclipse/tools` du répertoire dans lequel vous avez installé Installation Manager et utilisez le fichier de réponses que vous avez créé pour procéder à la désinstallation du produit.

   Par exemple :

   - Windows **Administrateur ou non administrateur :**

     ```
     imcl.exe
       input C:\temp\uninstall_response_file.xml
       -log C:\temp\uninstall_log.xml
     ```

   - UNIX Linux **Administrateur :**

     ```
     ./imcl
       input /var/temp/uninstall_response_file.xml
       -log /var/temp/uninstall_log.xml
     ```

   - UNIX Linux **Non administrateur :**

     ```
     ./imcl
       input rép_base_utilisateur/var/temp/uninstall_response_file.xml
       -log racine_utilisateur/var/temp/uninstall_log.xml
     ```

   Pour plus d'informations, accédez au d'centre de documentation d'IBM Installation Manager Version 1.5.

4. Facultatif : Répertoriez tous les packages installés pour vérifier la désinstallation.

   UNIX Linux

   ```
   ./imcl listInstalledPackages
   ```

   Windows

   ```
   imcl listInstalledPackages
   ```

5. Si le processus de désinstallation affiche la liste des profils WebSphere Application Server étendus, vous devez annuler l'extension de ces profils pour pouvoir poursuivre la désinstallation.

6. Facultatif : Désinstallez IBM Installation Manager.

   Important : Avant de pouvoir désinstaller IBM Installation Manager, vous devez désinstaller tous les packages installés par Installation Manager.

   Pour plus d'informations sur l'utilisation du script de désinstallation pour exécuter cette procédure, consultez le centre de documentation d'IBM Installation Manager version 1.5.

Création et augmentation de profils pour WebSphere eXtreme Scale

Une fois que vous avez installé le produit, créez des types de profil uniques et étendez les profils existants de WebSphere eXtreme Scale.

Avant de commencer

Installez WebSphere eXtreme Scale. Pour plus d'informations, voir «Présentation de l'installation», à la page 183.

Pourquoi et quand exécuter cette tâche

L'extension de profils en vue de leur utilisation avec WebSphere eXtreme Scale est facultative, mais requise dans les cas suivants :

- Pour démarrer automatiquement un service de catalogue ou un conteneur dans un processus WebSphere Application Server. Si vous n'étendez pas les profils des serveurs, les serveurs peuvent être uniquement démarrés à l'aide d'un programme en utilisant l'API ServerFactory ou comme processus distincts à l'aide d'un script **startOgServer** ou **stopXsServer**.
- Pour utiliser l'infrastructure PMI (Performance Monitoring Infrastructure) afin de surveiller les mesures de WebSphere eXtreme Scale.
- Pour afficher la version de WebSphere eXtreme Scale dans la console d'administration de WebSphere Application Server.

Si vous exécutez WebSphere eXtreme Scale dans WebSphere Application Server Version 7.0, vous pouvez utiliser le plug-in de l'outil de gestion de profil ou la commande **manageprofiles** pour créer et étendre des profils.

Utilisation de l'interface graphique pour créer des profils

Utilisez l'interface graphique, fournie par le plug-in Profile Management Tool, pour créer des profils pour WebSphere eXtreme Scale. n profil est un ensemble de fichiers qui définissent l'environnement d'exécution.

Avant de commencer

Vous ne pouvez pas utiliser l'interface graphique pour étendre des profils dans les scénarios suivants :

- **Installation 64 bits de WebSphere Application Server** :

  L'outil de gestion de profil n'existe pas pour les installations 64 bits de WebSphere Application Server. Utilisez le script **manageprofiles** à partir de la ligne de commande pour ces installations.

Pourquoi et quand exécuter cette tâche

Pour utiliser les fonctions du produit, le plug-in de l'outil de gestion de profil permet à l'interface graphique de vous aider à configurer des profils, tels qu'un profil WebSphere Application Server, un profil de gestionnaire de déploiement, un profil de cellule et un profil personnalisé. Vous pouvez étendre des profils pendant ou après l'installation de WebSphere eXtreme Scale.

Procédure

Utilisez l'interface graphique de l'outil de gestion des profils pour créer des profils. Choisissez l'une des options suivantes pour démarrer l'assistant :

- Sélectionnez **Outil de gestion de profil** dans la console Premiers pas.
- Accédez à l'outil de gestion de profil à partir du menu **Démarrer**.
- Exécutez le script `./pmt.sh|bat` à partir du répertoire *racine_install*`/bin/ProfileManagement`.

Que faire ensuite

Vous pouvez créer d'autres profils ou étendre des profils existants. Pour redémarrer l'outil de gestion de profil, exécutez la commande **./pmt.sh|bat** à partir du répertoire *racine_was*/bin/ProfileManagement, ou sélectionnez **Outil de gestion de profil** dans la console Premiers pas.

Démarrez un service de catalogue, démarrez des conteneurs et configurez les ports TCP dans votre environnement WebSphere Application Server. Pour plus d'informations, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 306.

Utilisation de l'interface graphique pour étendre des profils

Après avoir installé le produit, vous pouvez étendre un profil existant pour le rendre compatible avec WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Lorsque vous étendez un profil existant, vous modifiez le profil en appliquant un modèle d'extension spécifique à un produit. Par exemple, les serveurs WebSphere eXtreme Scale ne démarrent pas automatiquement à moins que le profil du serveur ne soit étendu avec le modèle xs_augment.

- Etendez le profil avec le modèle xs_augment si vous avez installé le client eXtreme Scale ou le client et le serveur.

Procédure

Utilisez l'interface graphique de l'outil de gestion des profils pour étendre les profils pour eXtreme Scale. Choisissez l'une des options suivantes pour démarrer l'assistant :

- Sélectionnez **Outil de gestion de profil** dans la console Premiers pas.
- Accédez à l'outil de gestion de profil à partir du menu **Démarrer**.
- Exécutez le script ./pmt.sh|bat depuis le répertoire *racine_was*/bin/ProfileManagement.

Que faire ensuite

Vous pouvez étendre d'autres profils. Pour redémarrer l'outil de gestion de profil, exécutez la commande **./pmt.sh|bat** à partir du répertoire *racine_was*/bin/ProfileManagement ou sélectionnez **Outil de gestion de profil** dans la console Premiers pas.

Démarrez un service de catalogue, démarrez des conteneurs et configurez les ports TCP dans votre environnement WebSphere Application Server. Pour plus d'informations, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 306.

Commande manageprofiles

Vous pouvez utiliser l'utilitaire **manageprofiles** pour créer des profils à l'aide du modèle WebSphere eXtreme Scale ou étendre et réduire des profils de serveur d'applications existants à l'aide des modèle d'extension de eXtreme Scale. Pour utiliser les fonctions du produit, votre environnement doit contenir au moins un profil étendu pour le produit.

- Pour pouvoir créer et étendre des profils, vous devez installer eXtreme Scale. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226 .

Rôle

La commande **manageprofiles** crée l'environnement d'exécution d'un processus de produit dans un ensemble de fichiers appelé profil. Le profil définit l'environnement d'exécution. Vous pouvez effectuer les actions suivantes à l'aide de la commande **manageprofiles** :

- Création et extension d'un profil de gestionnaire de déploiement
- Création et extension d'un profil personnalisé
- Création et extension d'un profil de serveur d'applications autonome
- Création et extension d'un profil de cellule
- Réduction de tout type de profil

Lorsque vous étendez un profil existant, vous modifiez le profil en appliquant un modèle d'extension spécifique à un produit.

- Etendez le profil avec le modèle xs_augment si vous avez installé le client eXtreme Scale ou le client et le serveur.

Emplacement

Le fichier de commandes se trouve dans le répertoire *racine_install*/bin.

Syntaxe

Pour obtenir une aide détaillée, utilisez le paramètre **-help** :

```
./manageprofiles.sh|bat
-create -templatePath racine_install/profileTemplates/xs_augment/dmgr -help
```

Dans les sections ci-après, chaque tâche que vous pouvez effectuer à l'aide de la commande **manageprofiles** est décrite, avec une liste des paramètres requis. Pour des détails sur les paramètres facultatifs à spécifier pour chaque tâche, reportez-vous à la commande **manageprofiles**, dans le Centre de documentation de WebSphere Application Server.

Création d'un profil de gestionnaire de déploiement

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil de gestionnaire de déploiement. Le gestionnaire de déploiement administre les serveurs d'applications fédérés dans la cellule.

Paramètres

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/dmgr
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -create -templatePath racine_install/profileTemplates/xs_augment/dmgr
```

Création d'un profil personnalisé.

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil personnalisé. Un profil personnalisé est un noeud vide que vous personnalisez via le gestionnaire de déploiement pour inclure des serveurs d'applications, des clusters ou d'autres processus Java.

Paramètres

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/managed
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -create -templatePath racine_install/profileTemplates/xs_augment/managed
```

-

```
./manageprofiles.sh|bat -create -templatePath racine_install/profileTemplates/pf_augment/managed
```

Création d'un profil de serveur d'applications autonome

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil de serveur d'applications autonome.

Paramètres

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/default
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -create -templatePath racine_install/profileTemplates/xs_augment/default
```

Création d'un profil de cellule

Vous pouvez utiliser la commande **manageprofiles** pour créer un profil de serveur, qui comprend un gestionnaire de déploiement et un serveur d'applications.

Paramètres

Indiquez les paramètres suivants dans le modèle de gestionnaire de déploiement :

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/dmgr
```

Indiquez les paramètres suivants avec le modèle de serveur d'applications :

-create
Crée un profil. (Obligatoire)

-templatePath *chemin_modèle*
Indique le chemin du modèle. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/default
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -create -templatePath racine_install/profileTemplates/xs_augment/cell/dmgr
-nodeProfilePath racine_install/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath racine_install/profileTemplates/xs_augment/cell/default
-dmgrProfilePath racine_install/profiles/Dmgr01 -portsFile
racine_install/profiles/Dmgr01/properties/portdef.props -nodePortsFile
racine_install/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

Extension d'un profil de gestionnaire de déploiement

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil de gestionnaire de déploiement.

Paramètres

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/dmgr
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath racine_install/profileTemplates/xs_augment/dmgr
```

Extension d'un profil personnalisé

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil personnalisé.

Paramètres

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/managed
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath racine_install/profileTemplates/xs_augment/managed
```

Extension d'un profil de serveur d'applications autonome

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil de serveur d'applications autonome.

Paramètres

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/default
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath racine_install/profileTemplates/xs_augment/default
```

Extension d'un profil de cellule

Vous pouvez utiliser la commande **manageprofiles** pour étendre un profil de cellule.

Paramètres

Indiquez les paramètres suivants pour le profil de gestionnaire de déploiement :

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/dmgr
```

Indiquez les paramètres suivants pour le profil de serveur d'applications :

-augment
Etend le profil existant. (Obligatoire)

-profileName
Spécifie le nom du profil. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (Obligatoire)

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/cell/default
```

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath racine_install
    /profileTemplates/xs_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath racine_install
    /profileTemplates/xs_augment/cell/default
```

Réduction d'un profil

Pour réduire un profil, spécifiez le paramètre **-ignoreStack** avec le paramètre **-templatePath**, en plus des paramètres **-unaugment** et **-profileName** requis.

Paramètres

-unaugment
Réduit un profil précédemment étendu. (Obligatoire)

-profileName
Spécifie le nom du profil. Le paramètre est généré par défaut si aucune valeur n'est spécifiée. (Obligatoire)

-templatePath *chemin_modèle*
Spécifie le chemin des fichiers de modèle qui se trouvent dans le répertoire racine d'installation. (facultatif).

Utilisez le format suivant :

```
-templatePath racine_install/profileTemplates/type_modèle/type_profil
```

où *type_modèle* est `xs_augment` ou `pf_augment` et *type_profil* correspond à l'un des quatre types de profil suivants :

- `dmgr` : profil du gestionnaire de déploiement
- `managed` : profil personnalisé
- `default` : profil de serveur d'applications autonome
- `cell` : profil de cellule

-ignoreStack
Utilisé avec le paramètre **-templatePath** pour réduire un profil qui a été étendu. (Facultatif).

Exemple

- Utilisation du modèle xs_augment :

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath racine_install/profileTemplates/xs_augment/type_profil
```

Profils non root

Vous pouvez octroyer à l'utilisateur non root des autorisations pour des fichiers et répertoires afin de permettre à cet utilisateur de créer un profil pour le produit. L'utilisateur non root peut également augmenter un profil qui a été créé par un utilisateur root, par un autre utilisateur non root ou par lui-même.

Dans un environnement WebSphere Application Server, les autorisations de création et d'utilisation de profils des utilisateurs non root (non administrateurs) sont limitées. Dans le plug-in de l'outil de gestion de profil, les noms et les valeurs de port uniques sont désactivés pour les utilisateurs non root. Ces derniers doivent modifier dans l'outil de gestion de profils valeurs par défaut des zones de nom du profil, de nom du noeud, de nom de la cellule et d'affectations des ports. Pensez à affecter aux utilisateurs non root une plage de valeurs pour chacune de ces zones. Vous pouvez attribuer la responsabilité à des utilisateurs non root d'adhérer aux plages de valeurs adéquates et de maintenir l'intégrité de leurs propres définitions.

Par *responsable de l'installation*, l'on entend soit l'utilisateur root, soit des utilisateurs non root. En tant que responsable de l'installation, vous pouvez octroyer aux utilisateurs non root des autorisations de création de profils et d'établissement de leurs propres environnements de produit. Par exemple, un utilisateur non root pourra créer un environnement de produit afin de tester un déploiement d'application avec un profil dont il est le propriétaire. Les autorisations de création de profils accordées aux utilisateurs non root comprennent les éléments suivants :

- création d'un profil et attribution de la propriété du répertoire du profil à un utilisateur non root pour lui permettre de démarrer WebSphere Application Server pour un profil spécifique
- octroi de droits d'accès en écriture aux fichiers et répertoires appropriés à un utilisateur non root pour lui permettre de créer le profil. Cette tâche permet de créer un groupe d'utilisateurs autorisés à créer des profils ou d'accorder à des utilisateurs individuels la possibilité de créer des profils
- installation de packages de maintenance pour le produit, ce qui inclut les services requis pour les profils existants appartenant à un utilisateur non root. En tant que responsable de l'installation, c'est vous le propriétaire de tous les fichiers créés par ce package de maintenance

Pour plus d'informations sur la création de profils pour les utilisateurs non root, reportez-vous à la rubrique Création de profils pour les utilisateurs non root.

En tant que responsable de l'installation, vous pouvez également octroyer aux utilisateurs non root des autorisations d'extension de profils. Par exemple, un utilisateur non root peut étendre un profil créé par un responsable d'installation ou par lui-même. Suivez les extensions de profils réalisées par des utilisateurs non root de WebSphere Application Server Network Deployment.

Toutefois, lorsqu'un utilisateur non root étend un profil créé par le responsable de l'installation, il n'a pas besoin de créer auparavant les fichiers indiqué ci-après. Ces fichiers ont été créés en même temps que le profil.

- *racine_was*/logs/manageprofiles.xml
- *racine_was*/properties/fsdb.xml
- *racine_was*/properties/profileRegistry.xml

Lorsqu'un utilisateur non root étend un profil qu'il crée, il doit modifier les autorisations relatives aux documents situés dans les modèles de profils eXtreme Scale.

Avertissement : Vous pouvez également utiliser un profil non root (non administrateur) pour WebSphere eXtreme Scale dans un environnement autonome, extérieur à WebSphere Application Server. Vous devez remplacer le propriétaire du répertoire ObjectGrid par le profil non root. Vous pouvez ensuite ouvrir une session avec ce profil non root et utiliser eXtreme Scale comme vous le feriez avec un profil root (administrateur).

Premières étapes après l'installation

Une fois l'installation terminée et vérifiée, vous pouvez commencer à utiliser WebSphere eXtreme Scale pour créer votre grille de données.

Procédure

1. Mettez à jour votre installation en appliquant la maintenance.

   Plus d'informations : «Mise à jour des serveurs eXtreme Scale», à la page 255.
2. Si vous utilisez WebSphere eXtreme Scale pour la première fois, vous pouvez utiliser les informations du guide de démarrage pour en savoir plus sur l'utilisation du produit.

   Informations complémentaires : Chapitre 1, «Mise en route», à la page 1
3. Configurez le produit. Créez des propriétés et des fichiers XML pour définir la configuration pour les grilles de données, des serveurs et des clients. Vous pouvez également configurer l'intégration du cache ou de la base de données, les services de données REST ou les plug-in OSGi.

   Informations complémentaires : Chapitre 6, «Configuration», à la page 269
4. Développez une application qui accède à la grille de données.

   Informations complémentaires : Développement d'applications
5. Démarrez et administrez les serveurs de conteneur et de catalogue avec vos fichiers de configuration et les données d'application de la grille.

   Informations complémentaires : Chapitre 7, «Administration», à la page 479
6. Surveillez les performances de votre configuration à l'aide de divers outils de surveillance.

   Informations complémentaires : Chapitre 8, «Contrôle», à la page 545

Identification et résolution des incidents liés à l'installation du produit

IBM Installation Manager est un programme d'installation commun à de nombreux logiciels IBM qui vous permet d'installer cette version de WebSphere eXtreme Scale.

Résultats

Remarques sur la journalisation et le traçage :

- Pour consulter aisément les journaux, ouvrez le gestionnaire d'installation et cliquez sur **File > View Log**. Pour ouvrir un fichier journal individuel, il vous suffira alors de le sélectionner dans le tableau et de cliquer sur l'icône **Open log file**.
- Les journaux sont situés dans le répertoire `logs`, à l'emplacement des données applicatives d'Installation Manager. Exemple :

– Windows **Installation par un administrateur :**

```
C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager
```

– Windows **Installation par un non administrateur :**

```
C:\Documents and Settings\nom_utilisateur\Application Data\IBM\Installation Manager
```

– UNIX Linux **Installation par un administrateur :**

```
/var/IBM/InstallationManager
```

– UNIX Linux **Installation par un non administrateur :**

```
rép_utilisateur/var/ibm/InstallationManager
```

- Les principaux fichiers journaux sont horodatés et conservés dans le répertoire `logs` au format XML. Ils peuvent donc être consultés à l'aide d'un navigateur Web standard.
- Le fichier `log.properties` du répertoire `logs` indique le niveau de journalisation ou de traçage appliqué par Installation Manager. Pour activer la fonction de traçage des plug-ins de WebSphere eXtreme Scale, par exemple, créez un fichier `log.properties` contenant les informations suivantes :

```
com.ibm.ws=DEBUG
com.ibm.cic.agent.core.Engine=DEBUG
global=DEBUG
```

Redémarrez Installation Manager au besoin. Installation Manager fournira ainsi les traces des plug-ins de WebSphere eXtreme Scale.

Remarques sur la résolution des problèmes :

- UNIX Linux Par défaut, certains systèmes HP-UX sont configurés pour ne pas utiliser de système DNS pour résoudre les noms d'hôte. Il est par conséquent possible qu'Installation Manager ne puisse pas se connecter à un référentiel externe.

  Vous pouvez vérifier la connexion au référentiel à l'aide de l'utilitaire Ping mais nslookup ne renverra aucun résultat.

  Demandez à votre administrateur système de configurer votre machine afin qu'elle utilise un système DNS ou utilisez l'adresse IP du référentiel.
- Dans certains cas, il peut être nécessaire d'ignorer les mécanismes de vérification existants dans Installation Manager.
  – Sur certains systèmes de fichiers réseau, l'espace disque peut ne pas être signalé correctement et il peut être nécessaire d'ignorer la vérification d'espace disque et de poursuivre l'installation.

    Pour désactiver la vérification d'espace disque, spécifiez la propriété système suivante dans le fichier `config.ini` du répertoire *racine_install_IM*`/eclipse/configuration` et redémarrez Installation Manager :

```
cic.override.disk.space=tailleunité
```

où *taille* est un chiffre entier et *unité* est laissé vide pour octets, a la valeur k pour kilo, m pour mégaoctets ou g pour gigaoctets. Par exemple :

```
cic.override.disk.space=120 (120 octets)
cic.override.disk.space=130k (130 kilooctets)
cic.override.disk.space=140m (140 mégaoctets)
cic.override.disk.space=150g (150 gigaoctets)
cic.override.disk.space=true
```

Installation Manager indique que la taille de l'espace disque est Long.MAX_VALUE. Au lieu d'afficher une grande quantité d'espace disque disponible, N/A s'affiche.

– Pour ignorer la vérification des prérequis pour le système d'exploitation, ajoutez `disableOSPrereqChecking=true` au fichier `config.ini` dans *racine_install_IM*`/eclipse/configuration` et redémarrez Installation Manager.

Si vous devez utiliser l'une de ces méthodes, contactez le support IBM pour obtenir de l'aide et développer une solution n'impliquant pas d'ignorer les mécanismes de vérification d'Installation Manager.

- Pour plus d'informations sur l'utilisation d'Installation Manager, accédez au centre de documentation d'IBM Installation Manager Version 1.5.

  Pour en savoir plus sur la dernière version d'Installation Manager, lisez les notes sur l'édition. Pour accéder aux notes sur l'édition, procédez comme suit :

  – Windows Cliquez sur **Démarrer > Programmes > IBM Installation Manager > Release Notes**.

  – UNIX Linux Accédez au sous-répertoire de la documentation dans le répertoire dans lequel Installation Manager est installé et ouvrez le fichier `readme.html`.

- Si une erreur fatale se produit lors de l'installation du produit, effectuez les étapes suivantes :
  – Faites une copie de sauvegarde de votre répertoire d'installation de produit actuel au cas où le service de support IBM aurait besoin de l'examiner ultérieurement.
  – Utilisez Installation Manager pour désinstaller tout ce que vous avez installé dans l'emplacement d'installation du produit (groupe de packages). Vous risquez de rencontrer des erreurs, mais vous pouvez les ignorer en toute sécurité.
  – Supprimez tout ce qui reste dans le répertoire d'installation du produit.
  – Utilisez Installation Manager pour réinstaller le produit dans le même emplacement ou dans un nouvel emplacement.

Remarque sur les informations de version et d'historique : Les commandes **`versionInfo`** et **`historyInfo`** retournent les informations de version et d'historique basées sur toutes les activités d'installation, de désinstallation, de mise à jour et de rétrogradation effectuées sur le système.

Chapitre 5. Mise à niveau et migration de WebSphere eXtreme Scale

Vous pouvez migrer vers la version 8.6 depuis les versions précédentes ou appliquer des packages de maintenance. Pour éviter les indisponibilités, vous devez prendre en compte l'ordre dans lequel vous appliquez les mises à jour aux serveurs dans votre configuration.

Mise à jour des serveurs eXtreme Scale

Vous pouvez mettre à niveau WebSphere eXtreme Scale vers une nouvelle version, soit en appliquant la maintenance ou en installant une nouvelle version, sans interrompre le service.

Avant de commencer

Vous devez disposer du fichier binaire pour la version majeure ou de maintenance à appliquer. Vous pouvez obtenir les dernières informations sur les versions disponibles et les packages de maintenance depuis le portail du support IBM pour WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Pour effectuer une mise à niveau sans interrompre le service, vous devez d'abord mettre à niveau les serveurs de catalogue, puis les serveurs de conteneur, et enfin les serveurs client.

8.6 Pour prendre en charge les configurations de grille de données d'entreprise, vous devez mettre à niveau le mécanisme de transport depuis ORB (Object Request Broker) vers IBM eXtremeIO (XIO). Si vous n'utilisez pas encore XIO, tous les serveurs et clients doivent être migrés vers la version 8.6 pour pouvoir utiliser le transport XIO. Les serveurs et les clients peuvent utiliser le transport ORB pendant la mise à niveau. A la fin de la mise à niveau, vous pouvez passer à XIO.

Procédure

1. Mettez à jour le niveau de service de catalogue, en répétant les étapes suivantes pour chaque serveur de catalogue dans la grille de données. Mettez à jour le niveau de service de catalogue avant de mettre à niveau les serveurs de conteneur ou les clients. Les serveurs de catalogue individuels peuvent interopérer avec la compatibilité de version de sorte que vous pouvez appliquer des mises à niveau à un seul serveur de catalogue à la fois sans interrompre le service.
   a. Recherchez un état de quorum sain. Exécutez la commande suivante :

      ```
      xscmd -c showQuorumStatus
      ```

      Ce résultat indique que tous les serveurs de catalogue sont connectés.
   b. Si vous utilisez la réplication multimaître entre deux domaines de service de catalogue, supprimez la liaison entre les deux domaines de service de catalogue lorsque vous mettez à niveau les serveurs de catalogue.

      ```
      xscmd –c dismissLink –cep host:2809 -fd domain_name
      ```

Il suffit d'exécuter cette commande depuis l'un des domaines de service de catalogue pour supprimer la liaison entre deux domaines de service de catalogue.

c. Arrêtez l'un des serveurs de catalogue. Vous pouvez utiliser la commande **stopOgServer** ou **stopXsServer** ou bien la commande **xscmd -c teardown** ou arrêter le serveur d'applications qui exécute le service de catalogue dans WebSphere Application Server. Il n'existe pas d'ordre d'arrêt spécifique des serveurs de catalogue, mais l'arrêt du serveur de catalogue principal en dernier réduit la rotation. Pour identifier le serveur de catalogue principal, recherchez le message CWOBJ8106 dans les fichiers journaux. Dans des conditions normales, le quorum est maintenu lorsqu'un serveur de catalogue est arrêté, mais il est recommandé d'identifier l'état du quorum après chaque arrêt avec la commande **xscmd -c showQuorumStatus**.

   Si vous utilisez la commande **xscmd -c teardown**, vous pouvez filtrer les noms de serveur. La commande **stopOgServer** ou **stopXsServer** nécessite un nom de serveur correct ou une liste de noms de serveur pour ne pas activer le mode Parallèle. Vous devez regrouper le processus d'arrêt au lieu d'appeler le processus d'arrêt ou le désassemblage pour de nombreux serveurs en parallèle. En groupant les serveurs à arrêter, la grille de données peut réagir aux serveurs qui sont en cours d'arrêt en déplaçant les fragments sur la grille de données. Vous pouvez utiliser l'une des commandes suivantes pour arrêter vos serveurs :

   Vous pouvez fournir une liste spécifique de serveurs à arrêter à la commande **stopOgServer** ou **xscmd -c teardown** :

   ```
   stopOgServer <server_name>[,<server_name>]
   ```

   8.6+

   ```
   stopXsServer <server_name>[,<server_name>]
   xscmd –c teardown -sl <server_name>[,<server_name>]
   ```

   Avec les exemples précédents, la commande **stopOgServer** , **stopXsServer** ou **xscmd -c teardown** exécute les mêmes tâches d'arrêt. Toutefois, vous pouvez filtrer les serveurs à arrêter avec la commande **xscmd -c teardown**. Voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 508 pour plus d'informations sur le filtrage des serveurs en fonction de la zone ou du nom d'hôte. La commande teardown filtre les serveurs correspondants et demande si les serveurs sélectionnés sont corrects.

d. Installez les mises à jour sur le serveur de catalogue. Vous pouvez soit migrer le serveur de catalogue vers une nouvelle version majeure du produit ou appliquer un package de maintenance. Voir les rubriques suivantes pour plus d'informations :

   - Pour effectuer une migration à partir d'une installation Version 7.1.x : «Migration vers WebSphere eXtreme Scale Version 8.6», à la page 258
   - **8.6+** Pour migrer depuis une installation 8.5 : «Installation des groupes de correctifs à l'aide d'IBM Installation Manager», à la page 228.

e. **8.6+** Mettez à jour la variable d'environnement *JAVA_HOME* pour qu'elle pointe vers une installation Java Development Kit (JDK) prise en charge. Pour les versions prises en charge et les instructions de mise à jour du kit JDK, voir «Java SE : points à prendre en considération», à la page 68.

f. Redémarrez le serveur de catalogue.

   Si vous utilisez un environnement autonome, voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494ou«Démarrage d'un service de catalogue autonome qui utilise le

transport eXtremeIO IBM (XIO)», à la page 480 pour plus d'informations. Si vous utilisez un environnement autonome WebSphere Application Server, voir «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 508 pour plus d'informations.

Le serveur de catalogue s'exécute en mode de compatibilité jusqu'à ce que tous les serveurs de catalogue soient amenés au même niveau. Le mode de compatibilité s'applique principalement aux migrations de versions majeures, car les nouvelles fonctions ne sont pas disponibles sur les serveurs qui ne sont pas migrés. Il n'existe aucune restriction sur la durée d'exécution des serveurs de catalogue en mode de compatibilité, mais il est recommandé de migrer tous les serveurs de catalogue vers le même niveau dès que possible.

g. Appliquez les mises à jour aux serveurs de catalogue restants de la configuration.

2. Mettez à niveau les serveurs de conteneur en répétant les étapes suivantes pour chaque serveur de conteneur dans la grille de données. Vous pouvez mettre à niveau les serveurs de conteneur dans n'importe quel ordre. Toutefois, envisagez de mettre à jour les serveurs d'abord, puis les clients, si vous utilisez des fonctions nouvelles dans la mise à niveau.

   a. Arrêtez les serveurs de conteneur à mettre à niveau. Vous pouvez arrêter les groupes de serveurs de conteneur avec la commande **stopOgserver** , **stopXsServer** ou **teardown**. En créant des lots d'opérations teardown et en exécutant des opérations de démarrage de serveur en parallèle, le mécanisme de placement déplace de plus grands groupes de fragments.

   ```
   xscmd -c teardown -z DefaultZone

   Connecting to Catalog service at localhost:1099

   Processing filter options for Server teardown

   The following servers will be torn down:

     container00
     container01
     container02
     container03
     container04


   Do you want to tear down the listed servers? (Y/N)
   ```

   b. Installez les mises à jour sur le serveur de conteneur. Vous pouvez soit migrer les serveurs de conteneur vers une nouvelle version majeure du produit ou appliquer un module de maintenance. Voir les rubriques suivantes pour plus d'informations :

      - Pour effectuer une migration à partir d'une installation Version 7.1.x : «Migration vers WebSphere eXtreme Scale Version 8.6», à la page 258
      - **8.6+** Pour migrer depuis une installation : «Installation des groupes de correctifs à l'aide d'IBM Installation Manager», à la page 228.

   c. **8.6+** Mettez à jour la variable d'environnement *JAVA_HOME* pour qu'elle pointe vers une installation Java Development Kit (JDK) prise en charge. Pour les versions prises en charge et les instructions de mise à jour du kit JDK, voir «Java SE : points à prendre en considération», à la page 68.

   d. Redémarrez les serveurs de conteneur.

   e. Mettez à niveau les serveurs de conteneur restants de la configuration.

3. Si vous utilisez la réplication multimaître, reconnectez vos domaines de service de catalogue. Utilisez la commande **inattendue -c establishLink** pour reconnecter les domaines de service de catalogue.

```
xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
```

4. Pour vérifier que tous les serveurs utilisent la nouvelle version de WebSphere eXtreme Scale, exécutez la commande **xscmd -c showinfo**.

```
xscmd –c showinfo
```

Que faire ensuite

- Vous pouvez également utiliser ces étapes pour revenir à une version antérieure ou pour désinstaller les modules de maintenance. Toutefois, si vous revenez à la version 7.1.0 lorsque vous utilisez la réplication multimaître, la réplication bidirectionnelle peut échouer lorsque vous rétablissez les liaisons. Dans ce cas, redémarrez les domaines de service de catalogue et liez-les de nouveau avec la commande **establishLink**.
- **8.6+** Après avoir migré tous les serveurs et clients vers la version 8.6, vous pouvez mettre à jour la configuration pour utiliser IBM eXtremeIO (XIO) pour prendre en charge les grilles de données d'entreprise. Pour plus d'informations, voir «Configuration d'IBM eXtremeIO (XIO)», à la page 351.

Migration vers WebSphere eXtreme Scale Version 8.6

Le programme d'installation de WebSphere eXtreme Scale ne vous permet pas de mettre à niveau ou de modifier une précédente installation. Vous devez désinstaller la version précédente avant d'installer la nouvelle version. Vous n'avez pas besoin de faire migrer vos fichiers de configuration car leur compatibilité est ascendante. Mais, si vous avez modifié l'un des scripts qui sont livrés avec le produit, vous devrez réappliquer ces changements aux scripts modifiés.

Avant de commencer

Vérifiez que vos systèmes disposent de la configuration minimale requise pour les versions du produit que vous avez l'intention de migrer et d'installer. Pour plus d'informations, voir «Configurations matérielle et logicielle requises», à la page 65.

Pourquoi et quand exécuter cette tâche

Fusionnez les fichiers script modifiés du produit avec les nouveaux fichiers scripts du produit dans le répertoire `/bin` pour conserver vos modifications.

Conseil : Si vous n'avez pas modifié les fichiers script installés avec le produit, vous n'avez pas besoin d'effectuer les étapes de migration ci-après. A la place, vous pouvez effectuer une mise à niveau vers Version 8.6 en désinstallant la version précédente et en installant la nouvelle version dans le même répertoire.

Procédure

1. Arrêtez tous les processus qui utilisent WebSphere eXtreme Scale.
   - Arrêtez tous les processus en cours d'exécution dans votre environnement WebSphere eXtreme Scale autonome. Pour plus d'informations, voir «Arrêt des serveurs autonomes qui utilisent le transport ORB», à la page 505.
   - Pour arrêter tous les processus en cours d'exécution dans votre environnement WebSphere Application Server ou WebSphere Application Server Network Deployment, reportez-vous aux utilitaires de ligne de commande.
2. Sauvegardez les scripts modifiés de votre répertoire d'installation actuel dans un répertoire temporaire.

3. Désinstallez le produit. Pour plus d'informations, voir «Désinstallation du produit à l'aide d'IBM Installation Manager», à la page 238.
4. Installez WebSphere eXtreme Scale Version 8.6. Pour plus d'informations, voir Chapitre 4, «Installation», à la page 183.
5. Fusionnez vos modifications apportées aux fichiers du répertoire temporaire avec les nouveaux fichiers scripts du produit, dans le répertoire `/bin`.
6. Démarrez tous vos processus WebSphere eXtreme Scale pour commencer à utiliser le produit. Pour plus d'informations, voir Chapitre 7, «Administration», à la page 479.

Mise à jour de WebSphere eXtreme Scale sur WebSphere Application Server

Lorsque vous migrez WebSphere Application Server vers une nouvelle version, vous pouvez également migrer la configuration WebSphere eXtreme Scale vers la nouvelle installation WebSphere Application Server.

Avant de commencer

- On suppose que WebSphere eXtreme Scale version 7 et WebSphere eXtreme Scale version 8 sont installés sur le même serveur.
- Migrez WebSphere Application Server version 7 vers WebSphere Application Server version 8. Pour plus d'informations, voir Migration des configurations de produits.
- Installez WebSphere eXtreme Scale version 8.5 sur votre installation WebSphere Application Server version 8. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226. Tous les scripts de migration WebSphere eXtreme Scale doivent être exécutés à partir de WebSphere eXtreme Scale version 8.5 ou version ultérieure. Par exemple, si vous effectuez une migration de la version 7.x vers la version 8.5, exécutez les scripts de migration à partir du répertoire `<racine_install_WXS_v8>/bin`.

Pourquoi et quand exécuter cette tâche

Lorsque vous installez une nouvelle version de WebSphere Application Server qui dispose d'une version intégrée de WebSphere eXtreme Scale, vous devez tout d'abord mettre à niveau WebSphere Application Server en suivant la procédure normale. Vous installez ensuite la nouvelle version de WebSphere eXtreme Scale sur votre nouvelle installation. Enfin, vous pouvez utiliser le script **xsmigration** pour déplacer les informations de configuration de WebSphere eXtreme Scale vers la nouvelle installation WebSphere Application Server.

Procédure

1. Migrez la configuration liée au gestionnaire de déploiement de la version 7 vers la version 8.
   a. Exécutez le script de sauvegarde WebSphere Application Server. Pour plus d'informations, voir Commande WASPreUpgrade.
   b. Arrêtez le gestionnaire de déploiement.
   c. Accédez au serveur du gestionnaire de déploiement dans votre configuration WebSphere eXtreme Scale et exécutez le script de migration.
      1) Placez-vous dans le répertoire `<racine_install_WXS_v8>/bin`
      2) Exécutez la commande suivante :

```
xsmigration.bat|sh -targetwashome <rép_base_WAS8x>
-sourcewashome <rép_base_WAS7x> -targetprofilepath
<profil_gest_déploiement_WAS8x>
 -sourceprofilepath <profil_gest_déploiement_WAS7x>
```

où

- *<rép_base_WAS8x>* est l'emplacement racine de l'installation de WebSphere Application Server version 8.x. Exemple : `/opt/IBM/WebSphere8`
- *<WAS7x_HOME>* est l'emplacement racine de l'installation de WebSphere Application Server version 7.x Exemple : `/opt/IBM/WebSphere7`
- *<profil_gest_déploiement_WAS8x>* est l'emplacement du profil de gestionnaire de déploiement de WebSphere Application Server version 8.x. Exemple : `/opt/IBM/WebSphere8/profiles/DMgr01`
- *<profil_gest_déploiement_WAS7x>* est l'emplacement du profil de gestionnaire de déploiement de WebSphere Application Server version 7.x. Exemple : `/opt/IBM/WebSphere7/profiles/DMgr01`

2. Migrez la configuration liée au serveur d'applications de la version 7 vers la version 8.
   a. Placez-vous dans le répertoire *<racine_install_WXS_v8>*`/bin`.
   b. Exécutez la commande suivante :

```
xsmigration.bat|sh -targetwashome <rép_base_WAS8x>
-sourcewashome <rép_base_WAS7x> -targetprofilepath <profil_serveur_appl_WAS8x>
 -sourceprofilepath <profil_serveur_appl_WAS7x>
```

   où

   - *<rép_base_WAS8x>* est l'emplacement racine de l'installation de WebSphere Application Server version 8.x. Exemple : `/opt/IBM/WebSphere8`
   - *<WAS7x_HOME>* est l'emplacement racine de l'installation de WebSphere Application Server version 7.x Exemple : `/opt/IBM/WebSphere7`
   - *<profil_serveur_appl_WAS8x>* est l'emplacement du profil de serveur d'applications de WebSphere Application Server version 8.x. Exemple : `/opt/IBM/WebSphere8/profiles/AppServer01`
   - *<profil_serveur_appl_WAS7x>* est l'emplacement du profil de serveur d'applications de WebSphere Application Server version 7.x. Exemple : `/opt/IBM/WebSphere7/profiles/AppServer01`

3. Redémarrez le gestionnaire de déploiement de WebSphere Application Server version 8 et synchronisez tous les noeuds gérés.

Migration de l'outil `xsadmin` vers l'outil `xscmd`

Dans les versions précédentes, l'outil **xsadmin** était un exemple d'utilitaire de ligne de commande pour surveiller l'état de l'environnement. L'outil **inattendue** a été introduit comme outil officiel de ligne de commande d'administration et de surveillance. Si vous utilisiez l'outil **xsadmin**, migrez les commandes vers le nouvel outil **xscmd**.

Equivalents des commandes xsadmin et xscmd

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd**. Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande **xsadmin**	Commande équivalente **xscmd**	Paramètres de commande **xscmd**
-bp	• **-cep** *hostname:listener_port* • **--catalogEndpoint** *hostname:listener_port*	s/o
-ch	• **-cep** *hostname:listener_port* • **--catalogEndpoint** *hostname:listener_port*	s/o
-clear	**-c clearGrid**	**-g, -ms, -v, -m, (-cep)**
-containers	• **-c showPlacement -container***containerName* • **-c showPlacement -server** *serverName*	**-e, -i, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-continuous	s/o	s/o
-coregroups	• • **-c listCoreGroupMembers -cg** *core_group*	s/o
-dismissLink *<catalog_service_domain>*	**-c dismissLink**	• **-fd** *<foreignCatalogServiceDomain>* • **--foreignCatalogServiceDomain** *<foreignCatalogServiceDomain>*
-dmgr	s/o - Cet argument est déterminé automatiquement avec **xscmd**	s/o
-empties	arg spécifique d'une nouvelle commande	s/o
-establishLink *<foreign_domain_name> <host1:port1,host2:port2...>*	**-c establishLink**	• **-fd** *<foreignCatalogServiceDomain>* **-fe** *<host1:port1,host2:port2...>* • **--foreignCatalogServiceDomain** *<foreignCatalogServiceDomain>* **-foreignEndPoints** *<host1:port1,host2:port2...>*
-fc	• **-ct** • **--container**	s/o
-fh	• **-hf** • **--hostFilter**	s/o
-fm	• **-m** • **--map**	s/o
-fnp	• **-snp** • **--serversWithNoPrimaries**	s/o
-fp	• **-p** • **--partitionId**	s/o

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd** *(suite).* Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande **xsadmin**	Commande équivalente **xscmd**	Paramètres de commande **xscmd**
-fs	• **-s** • **--server**	s/o
-fst	• **-st** <*shard_type*> • **--shardType** <*shard_type*> Shard values: P=primary A=asyncReplica S=syncReplica	s/o
-fz	• **-z** • **--zone**	s/o
-force	arg spécifique d'une nouvelle commande	
-g	• **-g** • **--objectGrid**	s/o
-getstatsspec	**-c getStatsSpec**	s/o
-getTraceSpec	**-c getTraceSpec**	s/o
-h	Vous pouvez exécuter l'aide avec ou sans un nom de commande spécifique : • **-h** • **--help** • **-h** <*command_name*> • **--help** <*command_name*>	s/o
-hosts	**-c listHosts**	**-g, -ms, -st, -c, -s, -hf, -z**
-jmxUrl	• **-cep** *hostname*:*listener_port* • **--catalogEndpoint** *hostname*:*listener_port*	s/o
-l	**-c listObjectGridNames**	s/o
-m	• **-ms** • **--mapSet**	s/o
-mapsizes	**-c showMapSizes**	**-g, -ms, -i, [-ct, -z, -s, -hf, sht [P,A,S], -p]**
-mbeanservers	**-c listAllJMXAddresses**	s/o
-overridequorum	**-c overrideQuorum**	s/o
-password	• **-pwd** • **--password**	s/o
-p	• **-cep** *hostname*:*listener_port* • **--catalogEndpoint** *hostname*:*listener_port*	s/o
-placementStatus	**-c placementServiceStatus**	**-g, -ms**
-primaries	**-c showPlacement -sf P**	**-e, -i, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd** *(suite).* Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande **xsadmin**	Commande équivalente **xscmd**	Paramètres de commande **xscmd**
-profile	Pour enregistrer les paramètres de sécurité actuels dans un profil de sécurité : • **-ssp** *profile_name* • **--saveSecProfile** *profile_name* Pour utiliser un profil de sécurité spécifié : • **-sp** *profile_name* • **--securityProfile** *profile_name*	
-quorumstatus	**-c showQuorumStatus**	s/o
-releaseShard <*nom_serveur_conteneur*> <*nom_objectgrid*> <*nom_groupe_mappes*> <*nom_partition*>	**-c releaseShard**	**-c, -g, -ms, -p**
-reserved	• **-sf R** • **--shardFilter R**	s/o
-reserveShard <*nom_serveur_conteneur*> <*nom_objectgrid*> <*nom_groupe_mappes*> <*nom_partition*>	**-c reserveShard**	**-c, -g, -ms, -p**
-resumeBalancing <*objectgrid_name*> <*map_set_name*>	**-c resumeBalancing**	**-g, -ms**
-revisions	**-c revisions**	**-s,-p,-g,-m**
-routetable	**-c routetable**	**-z, -hf,-p,-g,-ms**
-settracespec <*trace_string*>	**-c setTraceSpec**	**-spec** <*trace_string*>
-swapShardWithPrimary <*nom_serveur_conteneur*> <*nom_objectgrid*> <*nom_groupe_mappes*> <*nom_partition*>	**-c swapShardWithPrimary**	**-c -g, -ms, -p**
-setstatsspec <*stats_spec*>	**-c setStatsSpec**	**-spec** <*stats_spec*>
-suspendBalancing <*objectgrid_name*> <*map_set_name*>	**-c suspendBalancing**	**-g, -ms**
-ssl	• **-ssl** • **--enableSSL**	s/o
-teardown	**-c teardown**	**-f, , -st, -snp, -c, -s, -p, -hf, -z, -g, -ms, -m**
-triggerPlacement	**-c triggerPlacement**	**-g, -ms**
-trustPass	• **-tsp** • **--trustStorePassword**	s/o

Tableau 9. Arguments de l'utilitaire **xsadmin** *et commandes équivalentes* **xscmd** *(suite).* Certaines commandes **xscmd** ont une forme courte et une forme longue. La forme courte des commandes a un tiret (-) et la forme longue, deux (--). Vous pouvez utiliser l'une à la place de l'autre et inversement.

Arguments de ligne de commande xsadmin	Commande équivalente xscmd	Paramètres de commande xscmd
-trustPath	• **-ts** • **--trustStore**	s/o
-trustType	• **-tst** • **--trustStoreType**	s/o
-unassigned	**-c showPlacement -sf U**	**-e, -i, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-username	• **-user** • **--username**	s/o
-v	• **-v** • **--verbose**	s/o
-xml	**-c showPlacement**	s/o

Propriétés et API obsolètes

Les propriétés et API suivantes sont obsolètes dans les éditions spécifiées. Utilisez l'action de migration recommandée pour déterminer comment mettre à jour votre configuration.

8.6+ Eléments obsolètes dans la version 8.6

Tableau 10. Propriétés et API obsolètes

Obsolescence	Action de migration recommandée
Attribut numberOfBuckets L'attribut numberOfBuckets dans le fichier XML de descripteur ObjectGrid décrit le nombre de compartiments de l'instance BackingMap à utiliser. Lorsque vous lui affectez la valeur `0`, le cache local client est désactivé.	L'attribut **numberOfBuckets** dans le fichier XML de descripteur ObjectGrid a été remplacé par l'attribut **nearCacheEnabled**. Pour plus d'informations, voir «Configuration du cache local», à la page 362 et Fichier XML du descripteur d'ObjectGrid.
Object Request Broker (ORB) ORB est un transport utilisé pour communiquer sur une pile TCP. ORB dépend de toutes les applications client écrites dans le langage de programmation Java.	Si vous utilisez ORB, envisagez de migrer votre configuration pour utiliser IBM eXtremeIO (XIO). XIO est un nouveau mécanisme de transport qui prend en charge les applications client Java et .NET dans une grille de données d'entreprise. Pour plus d'informations, voir «Configuration d'IBM eXtremeIO (XIO)», à la page 351.
Enumération INSERTUPDATE dans setPutMode La méthode setPutMode(PutMode.UPSERT) est ajoutée pour changer le comportement par défaut des méthodes ObjectMap et JavaMap put() et putAll() pour qu'elles fonctionnent comme des méthodes ObjectMap.upsert() et upsertAll().	La méthode PutMode.UPSERT remplace la méthode setPutMode(PutMode.INSERTUPDATE). Utilisez la méthode PutMode.UPSERT pour indiquer à la mappe de sauvegarde BackingMap et au chargeur qu'une entrée dans la grille de données doit placer la clé et la valeur dans la grille. La mappe de sauvegarde et le chargeur exécutent une insertion ou une mise à jour pour placer la valeur dans la grille et le chargeur. Si vous exécutez l'API upsert dans vos applications, le chargeur reçoit un type LogElement UPSERT qui permet aux chargeurs d'exécuter des appels database merge ou upsert à la place de l'insertion ou de la mise à jour.
Commandes startOgSever et stopOgServer Les commandes **startOgSever** et **stopOgServer** sont utilisées pour démarrer et arrêter les serveurs qui utilisent le transport ORB. Si vous utilisez XIO, vous ne pouvez plus utiliser ces scripts pour démarrer les serveurs.	Si vous utilisez le transport XIO, vous devez utiliser les commandes **startXsSever** et **stopXsServer** pour démarrer et arrêter les serveurs de conteneur et de catalogue. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.

Tableau 10. Propriétés et API obsolètes (suite)

Obsolescence	Action de migration recommandée
wxs_home/ObjectGrid/legacy/session/bin Cet emplacement de chemin de fichier était utilisé pour les scripts de gestion de session avant WebSphere eXtreme Scale Version 7.1 .	Si vous utilisez le script **addObjectFilter** pour étendre l'application Web pour utiliser WebSphere eXtreme Scale pour la gestion de session, utilisez les scripts dans cet emplacement : `wxs_home/ObjectGrid/session/bin`. L'emplacement précédent `wxs_home/ObjectGrid/legacy/session/bin` est obsolète.
Propriétés de port sécurisé et non sécurisé TCP de conteneur XIO Ces ports sont utilisés pour spécifier les numéros de port d'écoute du transport IBM eXtremeIO sur le serveur. Vous définissez ces ports avec les propriétés xioChannel.xioContainerTCPNonSecure.Port et xioChannel.xioContainerTCPSecure.Port dans le fichier de propriétés du serveur.	Vous n'avez plus à spécifier ces propriétés lorsque vous utilisez le transport XIO. La valeur qui est spécifiée par la propriété listenerPort dans le fichier des propriétés du serveur est utilisée. Pour plus d'informations, voir Fichier de propriétés du serveur.

Eléments obsolètes dans la version 8.5

Tableau 11. Propriétés et API obsolètes

Obsolescence	Action de migration recommandée
WebSphereTransactionCallback Ce plug-in était utilisé pour gérer des transactions de grille de données avec des applications d'entreprise exécutées dans un environnement WebSphere Application Server.	L'interface WebSphereTransactionCallback a été remplacée par l'adaptateur de ressources WebSphere eXtreme Scale qui permet la gestion des transactions JTA (Java Transaction API). Vous pouvez installer cet adaptateur de ressources sur WebSphere Application Server ou d'autres serveurs d'applications Java Platform, Enterprise Edition (Java EE). Le plug-in WebSphereTransactionCallback n'est pas une API JTA inscrite et, par conséquent, il n'est pas conçu pour annuler la transaction JTA en cas de validation.

Eléments obsolètes dans la version 7.1.1

Tableau 12. Propriétés et API obsolètes

Obsolescence	Action de migration recommandée
Classe com.ibm.websphere.objectgrid.plugins.builtins.TranPropListener Cette classe était utilisée pour propager les processus de validation de la transaction ObjectGrid ayant abouti aux autres serveurs d'applications WebSphere hébergeant la même instance ObjectGrid en fonction du nom ObjectGrid.	L'interface TranPropListener a été remplacée par l'interface JMSObjectGridEventListener qui est une implémentation JMS de l'interface ObjectGridEventListener. Elle prend en charge l'invalidation de cache local côté client et la réplication entre homologues.
Classe com.ibm.websphere.objectgrid.plugins.OptimisticCallback Cette classe était utilisée pour fournir des opérations de comparaison optimistes pour les valeurs d'une mappe.	Le plug-in OptimisticCallback a été remplacé par l'interface ValueDataSerializer.Versionable que vous pouvez implémenter lorsque vous utilisez le plug-in DataSerializer avec le mode de copie COPY_TO_BYTES ou lorsque vous utilisez l'annotation @Version avec l'API EntityManager. Voir la documentation d'API pour plus d'informations.
Plug-in com.ibm.websphere.objectgrid.plugins.NoVersioning-OptimisticCallback Ce plug-in était utilisé pour le verrouillage optimiste sans vérification de version. Avec ce gestionnaire OptimisticCallback intégré, le chargeur effectuait la vérification de version, mais le verrouillage optimiste était utilisé pour que les données validées soient toujours retournées sur une lecture.	L'interface NoVersioningOptimisticCallback étend l'interface OptimisticCallback. Par conséquent, utilisez la stratégie de verrouillage optimiste avec l'isolement de transaction par défaut READ_COMMITTED ou inférieur. Voir Optimisation des performances de verrouillage pour plus d'informations.
Classe com.ibm.websphere.objectgrid.plugins.ObjectTransformer Ce plug-in était utilisé pour sérialiser, désérialiser et copier des objets dans le cache.	L'interface ObjectTransformer a été remplacée par les plug-in DataSerializer que vous pouvez utiliser pour stocker efficacement les données arbitraires dans WebSphere eXtreme Scale pour que les API de produit existantes puissent interagir efficacement avec vos données.
Méthode com.ibm.websphere.objectgrid.BackingMap.setMapEventListeners Cette méthode était utilisée pour définir la liste des objets MapEventListener.	Utilisez la méthode addMapEventListener(EventListener) ou removeMapEventListener(EventListener) ajouter ou supprimer des programmes d'écoute d'événement à partir dans une mappe de sauvegarde.

Tableau 12. Propriétés et API obsolètes (suite)

Obsolescence	Action de migration recommandée
Méthode com.ibm.websphere.objectgrid.ObjectGrid.setEventListeners Cette méthode était utilisée pour remplacer la liste en cours des objets ObjectGridEventListener par la liste fournie des objets ObjectGridEventListeners.	Utilisez la méthode addEventListener(EventListener) ou removeEventListener(EventListener) pour ajouter ou supprimer des programmes d'écoute d'événement et de cycle de vie dans la grille de données.

Fonctions stabilisées dans la version 7.1.1

Si une fonction est listée comme étant stabilisée, IBM n'envisage pas de la rendre obsolète ou de la supprimer dans une version suivante du produit, mais les efforts porteront sur la fonction alternative. Les utilisateurs n'ont pas besoin de modifier les applications et les scripts existants qui utilisent une fonction stabilisée, mais doivent envisager d'utiliser l'alternative stratégique pour les nouvelles applications.

Tableau 13. Propriétés et API obsolètes

Fonction stabilisée	Action de migration recommandée
xsadmin L'utilitaire xsadmin est fourni comme exemple pour monter comment créer des utilitaires personnalisés pour votre déploiement.	Utilisez l'utilitaire **xscmd** effectuer des tâches d'administration dans l'environnement, telles qu'établir des liens de réplication multimaître, remplacer un quorum et arrêter des groupes de serveurs avec la commande teardown.

Eléments obsolètes dans la version 7.1

Tableau 14. Propriétés et API obsolètes

Obsolescence	Action de migration recommandée
Propriété catalog.services.cluster de cellule et de serveur : cette propriété personnalisée servait à définir un groupe de serveurs de catalogue dans la configuration de WebSphere Application Server.	Cette propriété personnalisée est obsolète depuis la version 7.1. Dans la console d'administration de WebSphere Application Server, créez un domaine de service de catalogue qui crée la même configuration qu'avec la propriété personnalisée. Pour plus d'informations, voir .
Bean géré et interface CoreGroupServicesMBean	Ce bean géré est obsolète depuis la version 7.1. Utilisez plutôt le bean géré CatalogServiceManagementMBean.
Utilisation du bean géré ServerMBean.updateTraceSpec()	Cette opération est obsolète depuis la version 7.1. Utilisez plutôt l'attribut TraceSpec du bean géré DynamicServerMBean.
Bean géré CoreGroupServicesMBean	Ce bean géré est obsolète depuis la version 7.1. Utilisez le bean géré CatalogServiceManagementMbean à la place.
Exception ServiceUnavailableException	Cette exception est obsolète depuis la version 7.1. Utilisez plutôt l'exception TargetNotAvailableException.
	Les fonctionnalités de WPF peuvent également être utilisées dans WebSphere eXtreme Scale.
StreamQuery : Requête continue sur les données en cours stockées dans les mappes ObjectGrid.	Néant
Configuration de grille statique : Topologie statique basée sur les clusters, qui utilise le fichier XML de déploiement des clusters.	Remplacée par la topologie de déploiement dynamique, améliorée, pour la gestion des grilles de données de grande taille.

Tableau 14. Propriétés et API obsolètes (suite)

Obsolescence	Action de migration recommandée
Propriétés système dépréciées : Les propriétés système permettant de spécifier les fichiers de propriétés des serveurs et des clients sont dépréciées.	Vous pouvez toujours utiliser ces arguments, mais vous devrez remplacer vos propriétés par les nouvelles valeurs. **-Dcom.ibm.websphere.objectgrid.CatalogServerProperties** Cette propriété est dépréciée depuis la version 7.0 de WebSphere eXtreme Scale. Utilisez la propriété **-Dobjectgrid.server.props**. **-Dcom.ibm.websphere.objectgrid.ClientProperties** Cette propriété est dépréciée depuis la version 7.0 de WebSphere eXtreme Scale. Utilisez la propriété **-Dobjectgrid.client.props**. **-Dobjectgrid.security.server.prop** Cette propriété est dépréciée depuis la version 6.1.0.3 de WebSphere eXtreme Scale. Utilisez la propriété **-Dobjectgrid.server.prop**. **-serverSecurityFile** Cet argument n'est plus utilisé dans WebSphere eXtreme Scale Version 6.1.0.3. Cette option est transmise dans le script `startOgServer`. Utilisez l'argument **-serverProps**.

Propriétés et API supprimées

Si vous migrez votre configuration depuis une version précédente de WebSphere eXtreme Scale, certaines fonctions risquent d'être supprimées de cette version et des précédentes. Utilisez l'action de migration recommandée pour déterminer comment mettre à jour votre configuration.

Si une fonction est notée comme étant obsolète dans Fonctions obsolètes, IBM peut la supprimer dans une prochaine version du produit. La prochaine action portera sur la fonction stratégique indiquée dans la rubrique Action de migration recommandée de la section Fonctions obsolètes. D'une manière générale, avant de supprimer une fonction, IBM attend trois années complètes ou la publication d'au moins deux éditions majeures après la sortie de l'édition dans laquelle la fonction est réputée obsolète. Dans de rares cas, il peut être nécessaire de supprimer des fonctions plus tôt ; ces cas sont indiqués de façon claire et explicite dans les descriptions de ces fonctions obsolètes dans Fonctions obsolètes.

Les informations suivantes décrivent des éléments supprimés tels que des fonctions, des API, des interfaces de scriptage, des outils et des données de configuration exposées publiquement. Dans la mesure du possible, la procédure de remplacement recommandée est indiquée.

Eléments supprimés dans la version 8.5

Tableau 15. Propriétés et API supprimées

Elément supprimé	Action de migration recommandée
Support de mots clés : Les mots clés sont des balises de chaîne qui peuvent être appliquées aux entrées de cache et faire l'objet d'une requête ultérieure à l'aide des méthodes API ObjectMap.	Utilisez les fonctions d'index ou de requête pour obtenir des objets ayant des attributs spécifiques.
Interface MapAuthorization : Ce plug-in était utilisé pour autoriser ObjectMap et JavaMap à accéder aux principaux qui étaient représentés par un objet Subject.	Utilisez ObjectGridAuthorization pour connecter des implémentations d'autorisation. Un élément ObjectGridAuthorization peut être utilisé pour allouer les permissions d'accès à ObjectGrid, ObjectMap et JavaMap.

Fonction de partitionnement WebSphere (WPF) : La fonction de partitionnement est un ensemble d'API de programmation qui permet aux applications Java EE de prendre en charge le groupement asymétrique.	Vous pouvez configurer le partitionnement avec WebSphere eXtreme Scale.
StreamQuery : Requête continue sur les données en cours stockées dans les mappes ObjectGrid.	Aucune

Chapitre 6. Configuration

Vous pouvez configurer WebSphere eXtreme Scale pour qu'il soit exécuté dans un environnement autonome ou configurer eXtreme Scale pour qu'il soit exécuté dans un environnement avec WebSphere Application Server ou WebSphere Application Server Network Deployment. Pour qu'un déploiement WebSphere eXtreme Scale sélectionne les modifications de configuration sur le serveur de la grille de données, vous devez redémarrer les processus pour que ces modifications entrent en vigueur au lieu d'être appliquées de manière dynamique. Toutefois, côté client, vous ne pouvez pas modifier les paramètres de configuration d'une instance de client existante, mais vous pouvez créer un client avec les paramètres dont vous avez besoin à l'aide d'un fichier XML ou d'un programme. Lorsque vous créez un client, vous pouvez remplacer les paramètres par défaut provenant de la configuration de serveur actuelle.

Méthodes de configuration

Vous pouvez configurer la plupart des éléments du produit avec des fichiers XML et des fichiers de propriétés. Vous pouvez également utiliser des méthodes de programmation, y compris des interfaces de programmation d'applications et système, des plug-in et des beans gérés.

Pourquoi et quand exécuter cette tâche

Utilisez les fichiers suivants pour créer une configuration de base :

Fichier de propriétés du serveur
Utilisez le fichier de propriétés de serveur pour définir les paramètres des serveurs de catalogue et de conteneur (trace, consignation, sécurité, ports, etc.). Vous pouvez envoyer un fichier de propriétés de serveur au script de démarrage du serveur, placer le fichier dans le chemin d'accès aux classes ou définir le fichier avec des propriétés système.

Fichier de propriétés du client
Utilisez le fichier de propriétés de client pour définir les propriétés dans les clients, y compris les ports et les paramètres de sécurité. Vous pouvez spécifier le fichier des propriétés de client à utiliser avec une propriété système en le plaçant dans le chemin d'accès aux classes ou en utilisant la méthode ClientClusterContext.getClientProperties.

Fichier XML descripteur ObjectGrid
Le fichier XML descripteur ObjectGrid décrit la grille de données et la configuration de mappe. Spécifiez le fichier à utiliser avec le script de démarrage du serveur pour les configurations autonomes ou ajoutez le fichier au module d'application pour les configurations WebSphere Application Server.

Fichier XML descripteur de la stratégie de déploiement
Le fichier XML de stratégie de déploiement contrôle la segmentation et le placement des données sur divers serveurs de conteneur dans la configuration. Définissez le fichier à utiliser avec le script de démarrage du serveur pour les configurations autonomes ou ajoutez le fichier au module d'application pour les configurations WebSphere Application Server.

Liste de contrôle opérationnelle

Utilisez la liste de contrôle opérationnelle afin de préparer votre environnement pour le déploiement de WebSphere eXtreme Scale.

Tableau 16. Liste de contrôle opérationnelle

Elément de la liste de contrôle	Pour plus d'informations

Si vous utilisez AIX, optimisez les paramètres suivants du système d'exploitation :
 	
TCP_KEEPINTVL
Le paramètre TCP_KEEPINTVL fait partie d'un protocole de maintien de connexion du socket qui permet la détection des indisponibilités du réseau. La propriété spécifie l'intervalle entre les paquets envoyés pour valider la connexion. Si vous utilisez WebSphere eXtreme Scale, spécifiez la valeur `10`. Pour vérifier le paramètre actuel, exécutez la commande suivante :
`# no –o tcp_keepintvl`
Pour modifier le paramètre actuel, exécutez la commande suivante :
`# no –o tcp_keepintvl=10`
Le paramètre TCP_KEEPINTVL est exprimé en demi secondes.
 	
TCP_KEEPINIT
Le paramètre TCP_KEEPINIT fait partie d'un protocole de maintien de connexion du socket qui permet la détection des indisponibilités du réseau. La propriété spécifie le délai d'attente initial de la connexion TCP. Si vous utilisez WebSphere eXtreme Scale, spécifiez la valeur `40`. Pour vérifier le paramètre actuel, exécutez les commandes suivantes :
`# no –o tcp_keepinit`
Pour modifier le paramètre actuel, exécutez la commande suivante :
`# no –o tcp_keepinit=40`
Le paramètre TCP_KEEPINIT est exprimé en demi secondes.
	• Pour des informations sur l'optimisation d'AIX, voir la rubrique Optimisation des systèmes AIX.

Mettez à jour le fichier `orb.properties` pour modifier le comportement de transport de la grille. Le fichier `orb.properties` se trouve dans le répertoire `java/jre/lib`.	«Propriétés ORB», à la page 604

Tableau 16. Liste de contrôle opérationnelle (suite)

Elément de la liste de contrôle	Pour plus d'informations
Utilisez les paramètres dans le script **startOgServer** ou **startXsServer**. En particulier, utilisez les paramètres suivants : • Définissez les paramètres de segment de mémoire avec l'option **-jvmArgs**. • Définissez les propriétés et le chemin d'accès aux classes avec l'option **-jvmArgs**. • Définissez les paramètres **-jvmArgs** pour configurer la surveillance de l'agent. **Paramètres de port** WebSphere eXtreme Scale doit ouvrir des ports pour les communications pour certains transports. Ces ports sont tous définis de manière dynamique. Toutefois, si un pare-feu est utilisé entre les conteneurs, vous devez spécifier les ports. Utilisez les informations suivantes sur les ports : **Port du programme d'écoute** Vous pouvez utiliser l'argument **-listenerPort** pour spécifier le port utilisé pour les communications entre les processus. **Port du groupe central** Vous pouvez utiliser l'argument **-haManagerPort** pour spécifier le port utilisé pour la détection des incidents. Cet argument correspond à peerPort. Notez que les groupes centraux n'ayant pas besoin de communiquer entre les zones, il n'est pas nécessaire de définir ce port si le pare-feu est ouvert pour tous les membres d'une même zone. **Port du service JMX** Vous pouvez utiliser l'argument **-JMXServicePort** pour spécifier le port à utiliser par le service JMX. **Port SSL** Si vous transmettez `-Dcom.ibm.CSI.SSLPort=1234` comme argument **-jvmArgs**, le port SSL prend la valeur `1234`. Le port SSL est l'homologue de port sécurisé du port du programme d'écoute. **Port du client** Utilisé uniquement dans le service de catalogue. Vous pouvez spécifier cette valeur avec l'argument **-catalogServiceEndPoints**. La valeur de ce paramètre est au format suivant : `nomServeur:nomHôte:portClient:portHomologue`	«Script **startOgServer** (ORB)», à la page 499 «Script **startXsServer** (XIO)», à la page 485
Vérifiez que les paramètres de sécurité sont configurés correctement : • Transport (SSL) • Application (Authentification et autorisation) Pour vérifier vos paramètres de sécurité, vous pouvez essayer d'utiliser un client malveillant pour vous connecter à votre configuration. Par exemple, si le paramètre SSL requis est configuré, un client possédant un paramètre TCP_IP avec ce dernier ou un client possédant un fichier de clés certifiées incorrect ne doit pas pouvoir se connecter au serveur. Si l'authentification est requise, un client sans données d'identification, telles qu'un ID utilisateur et un mot de passe ne doit pas pouvoir se connecter au serveur. Si l'autorisation est appliquée, un client sans autorisation d'accès ne doit pas être autorisé à accéder aux ressources du serveur.	«Intégration de la sécurité à des fournisseurs externes», à la page 638

Tableau 16. Liste de contrôle opérationnelle (suite)

Elément de la liste de contrôle	Pour plus d'informations
Choisissez comment vous allez surveiller votre environnement. • Outil **xscmd** : – Les ports JMX des serveurs de catalogue doivent être visibles de l'outil **xscmd**. Les ports de serveur de conteneur doivent également être accessibles pour certaines commandes qui collectent des informations des conteneurs. • Console de surveillance : Avec la console de surveillance, vous pouvez générer des graphiques des statistiques actuelles et historiques. • Outils de surveillance du fournisseur : – Tivoli Enterprise Monitoring Agent – CA Wily Introscope – Hyperic HQ	• «Surveillance avec l'utilitaire **xscmd**», à la page 571 • «Sécurité JMX (Java Management Extensions)», à la page 635 • «Surveillance à l'aide de la console Web», à la page 547 • «Surveillance à l'aide d'IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale», à la page 588 • «Surveillance d'eXtreme Scale à l'aide de Hyperic HQ», à la page 597 • «Surveillance des applications eXtreme Scale à l'aide de CA Wily Introscope», à la page 594

Configuration des grilles de données

Utilisez un fichier descripteur XML ObjectGrid pour configurer des mappes de données, des mappes de sauvegarde, des plug-in, etc. Pour configurer WebSphere eXtreme Scale, utilisez un fichier XML de descripteur d'ObjectGrid et l'API ObjectGrid. Dans le cas d'une topologie répartie, vous avez besoin d'un fichier descripteur XML d'ObjectGrid et d'un fichier XML de stratégie de déploiement.

Configuration de déploiements locaux

Une configuration eXtreme Scale en mémoire locale peut être créée à l'aide d'un fichier XML descripteur d'ObjectGrid ou des API.

Pourquoi et quand exécuter cette tâche

Pour créer un déploiement local, vous créez un fichier XML descripteur d'ObjectGrid et transmettez le fichier createObjectGrid aux méthodes de l'interface ObjectGridManager.

Vous pouvez également créer le déploiement complet à l'aide d'un programme avec l'interface ObjectGridManager.

Procédure

1. Créez un fichier XML descripteur d'ObjectGrid.

   Le fichier `companyGrid.xml` ci-après est un exemple de XML de descripteur d'ObjectGrid. Les premières lignes de ce fichier incluent l'en-tête requis de chaque fichier XML ObjectGrid. Le fichier définit une instance ObjectGrid nommée "CompanyGrid" et plusieurs mappes de sauvegarde intitulées "Customer," "Item," "OrderLine" et "Order."

   fichier
   companyGrid.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
  <objectGrid name="CompanyGrid">
   <backingMap name="Customer" />
   <backingMap name="Item" />
   <backingMap name="OrderLine" />
   <backingMap name="Order" />
```

```
  </objectGrid>
 </objectGrids>

</objectGridConfig>
```

2. Transmettez le fichier XML à l'une des méthodes createObjectGrid de l'interface ObjectGridManager.

   L'exemple de code ci-après valide le fichier `companyGrid.xml` par rapport au schéma XML et crée l'instance ObjectGrid intitulée "CompanyGrid." L'instance ObjectGrid nouvellement créée n'est pas placée en cache.

```
ObjectGridManager objectGridManager = ObjectGridManagerFactory.getObjectGridManager();
ObjectGrid companyGrid = objectGridManager.createObjectGrid("CompanyGrid",
 new URL("file:etc/test/companyGrid.xml"), true, false);
```

Que faire ensuite

Voir Création d'instances avec l'interface ObjectGrid ObjectGridManager pour plus d'informations sur la définition toutes les mappes à l'aide d'un programme avec les méthodes createObjectGrid sur l'interface ObjectGridManager.

Configuration des grilles de données pour utiliser le format de données eXtreme Scale (XDF)

Si vous utilisez une grille de données d'entreprise, vous devez activer XDF afin que Java et .NET puissent accéder aux mêmes objets dans la grille de données. Utilisez XDF pour sérialiser et stocker les clés et les valeurs dans la grille de données dans un format indépendant du langage.

Avant de commencer

Activez IBM eXtremeIO (XIO) dans l'environnement. Pour plus d'informations, voir «Configuration d'IBM eXtremeIO (XIO)», à la page 351.

Pourquoi et quand exécuter cette tâche

Activez eXtreme Data Format (XDF) pour stocker les objets sérialisés indépendamment du langage. XDF est maintenant la technologie de sérialisation par défaut utilisée lorsque vous exécutez XIO et que le mode de copie de mappe est `COPY_TO_BYTES`. Lorsque vous activez cette fonction les objets Java et C# peuvent partager des données dans une même grille de données. Vous pouvez définir le mode XDF pour les installations de WebSphere eXtreme Scale dans un environnement autonome et pour les installations de WebSphere eXtreme Scale dans un environnement WebSphere Application Server.

L'utilisation de XDF, offre les avantages suivants :

- Sérialisation des données pour le partage entre les applications Java et C#/.NET.
- Indexation des données sur le serveur sans que les classes utilisateur soient présentes, si l'accès aux zones est utilisé.
- Gestion automatique des versions des classes pour pouvoir segmenter les définitions de classe lorsque vous ajoutez des applications qui nécessitent de nouvelles versions de fichiers. Les anciennes versions des données peuvent être utilisées en tirant parti de l'interface Mergable.
- Partitionnement des données avec des annotations dans Java et C# pour un partitionnement cohérente par rapport à l'application.

Procédure

Dans le fichier XML descripteur ObjectGrid, affectez à l'attribut **CopyMode** la valeur XDF dans l'élément backingMap du fichier XLM descripteur ObjectGrid.

```
<backingMap name="Employee" lockStrategy="PESSIMISTIC" copyMode="COPY_TO_BYTES">
```

Que faire ensuite

Développez des applications qui peuvent gérer les données. Pour plus d'informations, voir Développement d'applications de grille de données d'entreprise.

Configuration des expulseurs avec des fichiers XML

Java

Outre la configuration par programmation d'un expulseur TTL (durée de vie) avec l'interface BackingMap, vous pouvez utiliser un fichier XML pour configurer un expulseur sur chaque instance BackingMap.

Avant de commencer

Avant de commencer, choisissez le type d'expulseur que vous allez utiliser :

- **Expulseur TTL basé sur le temps par défaut :** l'expulseur par défaut utilise une règle d'expulsion TTL (time-to-live) pour chaque instance BackingMap.
- **Mécanisme d'expulsion enfichable :** les expulseurs enfichables utilisent généralement une règle d'expulsion basée sur le nombre de tentatives et non pas sur le temps.

Définissez les paramètres de l'expulseur avant de démarrer les serveurs de conteneur.

Procédure

- Pour définir l'expulseur TTL par défaut, ajoutez l'attribut **ttlEvictorType** dans le fichier XML descripteur ObjectGrid.

  L'exemple suivant montre que l'instance BackingMap map1 utilise le type d'expulseur TTL NONE. L'instance BackingMap map2 utilise le ttype d'expulseur TTL LAST_ACCESS_TIME ou LAST_UPDATE_TIME. Indiquez que l'un ou l'autre de ces deux paramètres. L'instance BackingMap map2 a une valeur de durée de vie de 1800 seconds, soit 30 minutes. L'instance BackingMap map3 est définie pour utiliser le type d'expulseur basé sur la durée de vie CREATION_TIME et a une durée de vie de 1200 secondes (20 minutes).

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
    <objectGrid name="grid1">
        <backingMap name="map1" ttlEvictorType="NONE" />
        <backingMap name="map2" ttlEvictorType="LAST_ACCESS_TIME|LAST_UPDATE_TIME"
     timeToLive="1800" />
        <backingMap name="map3" ttlEvictorType="CREATION_TIME"
     timeToLive="1200" />
    </objectgrid>
</objectGrids>
```

Figure 31. Enable TimeToLive evictor with XML

- Pour définir un expulseur enfichable, utilisez l'exemple suivant.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
    <objectGrid name="grid">
        <backingMap name="map1" ttlEvictorType="NONE" pluginCollectionRef="LRU" />
        <backingMap name="map2" ttlEvictorType="NONE" pluginCollectionRef="LFU" />
    </objectGrid>
</objectGrids>
<backingMapPluginCollections>
    <backingMapPlugincollection id="LRU">
        <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
            <property name="maxSize" type="int" value="1000" description="set max size
       for each LRU queue" />
            <property name="sleepTime" type="int" value="15" description="evictor
       thread sleep time" />
            <property name="numberOfLRUQueues" type="int" value="53" description="set number
       of LRU queues" />
        </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="LFU">
        <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LFUEvictor">
            <property name="maxSize" type="int" value="2000" description="set max size for each LFU heap" />
            <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
            <property name="numberOfHeaps" type="int" value="211" description="set number of LFU heaps" />
        </bean>
    </backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>
```

Figure 32. Connexion d'un expulseur en utilisant XML

8.6+ Que faire ensuite

Si vous souhaitez configurer les métadonnées de durée de vie à mettre à jour lorsque le cache local est utilisé, voir «Configuration du cache local pour mettre à jour la valeur LAST_ACCESS_TIME pour l'expulseur TTL dans la grille de données».

Configuration du cache local pour mettre à jour la valeur LAST_ACCESS_TIME pour l'expulseur TTL dans la grille de données

Java

Vous pouvez configurer le cache local configuré sur le client pour propager les opérations de lecture de durée de vie (TTL) vers la grille de données distante. La configuration de cette propagation évite aux entrées d'être évincées prématurément de la grille de données distante.

Avant de commencer

- Vous devez avoir configuré un cache local sur le client. Pour plus d'informations, voir «Configuration du cache local», à la page 362.
- Vous devez disposer d'un expulseur TTL défini dans l'élément backingMap dans le fichier XML de descripteur ObjectGrid avec la valeur `LAST_ACCESS_TIME`. Pour plus d'informations, voir «Configuration des expulseurs avec des fichiers XML», à la page 274.

Pourquoi et quand exécuter cette tâche

Vous pouvez activer le cache local pour mettre à jour les métadonnées TTL sur la grille de données distante en définissant la propriété nearCacheLastAccessTTLSyncEnabled dans le fichier XML de descripteur ObjectGrid.

Procédure

1. Définissez l'attribut **`nearCacheLastAccessTTLSyncEnabled`** dans le fichier XML de descripteur ObjectGrid. Définissez cet attribut dans l'élément backingMap dans lequel vous avez activé l'expulseur TTL. Pour plus d'informations, voir Fichier XML du descripteur d'ObjectGrid.

   8.6+ `nearCacheLastAccessTTLSyncEnabled`
   Définissez la valeur `true` pour synchroniser les informations de durée de vie avec la grille de données distante. Le type d'expulseur TTL (durée de vie) `LAST_ACCESS_TIME` doit être activé lorsque vous activez cette propriété.

   Valeur par défaut : `false` (facultatif)

   Dans l'exemple suivant, les deux mappes de l'élément Grid ObjectGrid ont un expulseur TTL configuré. La synchronisation avec la grille de données distante est également activée dans l'élément Map1 backingMap.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="Grid">
             <backingMap name="Map1" lockStrategy="OPTIMISTIC" nearCacheLastAccessTTLSyncEnabled="true"
      ttlEvictorType="LAST_ACCESS_TIME" timeToLive="60"  copyMode="COPY_TO_BYTES"/>
             <backingMap name="Map2" lockStrategy="OPTIMISTIC" nearCacheLastAccessTTLSyncEnabled="false"
      ttlEvictorType="LAST_ACCESS_TIME" timeToLive="60" copyMode="COPY_TO_BYTES"/>
        </objectGrid>
    </objectGrids>
...
...
</objectGridConfig>
```

2. Redémarrez le serveurs de conteneur et les clients. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479 et «Configuration des serveurs de conteneur dans WebSphere Application Server», à la page 333.

Résultats

Lorsque des entrées de cache sont ajoutées à la grille de données distante, les mêmes clés et valeurs sont insérées dans le cache local sur le client. Lorsque le client reçoit ou extrait des clés à partir de la mémoire cache, les valeurs sont renvoyées à partir du cache local. Les métadonnées TTL sont également envoyées à la grille de données distante afin qu'elle contienne les dernières informations TTL.

Configuration d'une stratégie de verrouillage

Java

Vous pouvez définir une stratégie optimiste, pessimiste ou sans verrouillage sur chaque BackingMap de la configuration de WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Chaque instance de mappe de sauvegarde BackingMap peut être configurée pour utiliser l'une de ces stratégies de verrouillage :

1. mode de verrouillage optimiste
2. mode de verrouillage pessimiste
3. aucun

La stratégie de verrouillage OPTIMISTIC est le mode par défaut. Utilisez le verrouillage optimiste lorsque les données sont modifiées rarement. Les verrous sont uniquement maintenus pendant un laps de temps limité tandis que les données sont lues depuis le cache et copiées dans la transaction. Lorsque le cache de transaction est synchronisé avec le cache principal, tous les objets mis en cache qui ont été mis à jour sont vérifiés avec la version d'origine. Si la vérification échoue, la transaction est annulée et une exception OptimisticCollisionException est provoquée.

La stratégie de verrouillage PESSIMISTIC obtient des verrous pour les entrées de cache et doit être utilisée lorsque les données sont modifiées fréquemment. A chaque lecture d'une entrée de cache, un verrou est obtenu et maintenu de façon conditionnelle jusqu'à la fin de la transaction. La durée de certains verrous peut être paramétrée à l'aide des niveaux d'isolement de transaction pour la session.

Si le verrouillage n'est pas obligatoire car les données ne sont jamais mises à jour ou le sont au cours de période calmes, vous pouvez le désactiver à l'aide de la stratégie de verrouillage NONE. Cette stratégie est très rapide car un gestionnaire de verrou n'est pas requis. La stratégie de verrouillage NONE est idéale pour les tables de recherche et les mappes en lecture seule.

Pour plus d'informations sur les stratégies de verrouillage, voir les Stratégies de verrouillageinformations relatives aux stratégies de verrouillage dans *Présentation du produit*.

Procédure

- **Configurez une stratégie de verrouillage optimiste**
  - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("optimisticMap");
bm.setLockStrategy( LockStrategy.OPTIMISTIC );
```

  - Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="test">
```

```
            <backingMap name="optimisticMap"
                lockStrategy="OPTIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configurez une stratégie de verrouillage pessimiste**
  - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
specify pessimistic strategy programmatically
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("pessimisticMap");
bm.setLockStrategy( LockStrategy.PESSIMISTIC);
```

  - Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid.

```
specify pessimistic strategy using XML
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="pessimisticMap"
                lockStrategy="PESSIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configurez une stratégie sans verrouillage**
  - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("noLockingMap");
bm.setLockStrategy( LockStrategy.NONE);
```

  - Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="noLockingMap"
                lockStrategy="NONE"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

Que faire ensuite

Pour éviter une exception java.lang.IllegalStateException, vous devez appeler la méthode setLockStrategy avant d'appeler les méthodes initialize ou getSession sur l'instance ObjectGrid.

Configuration de la réplication entre homologues avec JMS

Java

Le mécanisme de réplication entre homologues basé sur JMS (Java Message Service) est utilisé dans les environnements WebSphere eXtreme Scale réparti et local. JMS est un processus de réplication de coeur à coeur qui permet aux mises à jour de données de circuler parmi les ObjectGrid locaux et les ObjectGrid répartis. Par exemple, avec ce mécanisme, vous pouvez transférer les mises à jour de données d'une grille de données eXtreme Scale vers une grille eXtreme Scale locale ou d'une grille vers une autre dans un domaine système différent.

Avant de commencer

Le mécanisme JMS de réplication entre homologues repose sur l'ObjectGridEventListener JMS pré-intégré, com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener. Pour des informations plus détaillées sur l'activation du mécanisme de réplication entre homologues, voir «Programme d'écoute d'événement JMS», à la page 283.

Pour plus d'informations, voir «Configuration de la synchronisation du client JMS (Java Message Service)», à la page 366.

Vous trouverez ci-après un exemple de configuration XML permettant d'activer un mécanisme de réplication entre homologues sur une configuration eXtreme Scale :

Configuration de réplication entre homologues - Exemple de XML

```
<bean id="ObjectGridEventListener"
className="com.ibm.websphere.objectgrid.plugins.JMSObjectGridEventListener">
 <property name="replicationRole" type="java.lang.String" value="DUAL_ROLES" description="" />
  <property name="replicationStrategy" type="java.lang.String" value="PUSH" description="" />
  <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
  value="defaultTCF" description="" />
  <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
  <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
  <property name="jms_userid" type="java.lang.String" value="" description="" />
  <property name="jms_password" type="java.lang.String" value="" description="" />
  <property name="jndi_properties" type="java.lang.String"
 value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;
 java.naming.provider.url=tcp://localhost:61616;connectionFactoryNames=defaultTCF;
 topic.defaultTopic=defaultTopic"
  description="jndi properties" />
     </bean>
```

Répartition de modifications entre des machines virtuelles Java homologues

Java

Les objets LogSequence et LogElement répartissent les modifications entre des machines virtuelles Java homologues et, à l'aide d'un plug-in ObjectGridEventListener, ils communiquent les modifications intervenues dans une transaction eXtreme Scale.

Pour plus d'informations sur l'utilisation de JMS (Java Message Service) pour répartir les modifications transactionnelles, voir Répartition des transactions.

Il est impératif au préalable que l'instance ObjectGrid soit mise en cache par ObjectGridManager. Voir les méthodes createObjectGrid pour plus d'informations à ce sujet. La valeur booléenne de cacheInstance doit être définie comme true.

Il n'est pas nécessaire que vous implémentiez vous-même ce mécanisme. Il existe en effet un mécanisme pré-intégré de réplication entre homologues qui se chargera de cette fonction. Voir «Configuration de la réplication entre homologues avec JMS», à la page 279.

Les objets fournissent aux applications le moyen de publier facilement les modifications qui sont intervenues dans un ObjectGrid, à savoir un transport de messages vers des ObjectGrids homologues situés sur des machines virtuelles Java pour appliquer ces modifications sur ces JVM. La classe LogSequenceTransformer est indispensable pour cette prise en charge. Nous allons expliquer ici comment écrire, pour la propagation des messages, un programme d'écoute utilisant un système de messagerie Java Message Service (JMS). En fait un plug-in IBM permet à eXtreme Scale de prendre en charge la transmission de LogSequences qui résultent de la validation d'une transaction eXtreme Scale entre des membres d'un cluster WebSphere Application Server. Cette fonction n'est pas activée par défaut, mais il est possible de la configurer pour la rendre opérationnelle. Mais, lorsque l'utilisateur ou le producteur ne sont pas WebSphere Application Server, le recours à un système externe de messagerie JMS peut s'avérer nécessaire.

Implémenter le mécanisme

La classe LogSequenceTransformer et les API ObjectGridEventListener, LogSequence et LogElement permettent l'utilisation de n'importe quel mécanisme de publication/abonnement pour la répartition des modifications et le filtrage des mappes à répartir. Les fragments de code utilisés ici montrent comment exploiter ces API avec JMS pour construire un ObjectGrid d'égal à égal, partagé par des applications hébergées sur plusieurs sortes de plateformes partageant un transport commun de messages.

Initialisation du plug-in

L'ObjectGrid appelle la méthode initialize du plug-in, qui fait partie du contrat de l'interface ObjectGridEventListener, lorsque l'ObjectGrid démarre. La méthode initialize doit obtenir ses ressources JMS (connexions, sessions et diffuseurs de publications) et elle doit démarrer l'unité d'exécution qu'est le programme d'écoute JMS.

Les exemples suivants illustrent la méthode initialize :

exemple de méthode initialize

```
public void initialize(Session session) {
        mySession = session;
        myGrid = session.getObjectGrid();
        try {
            if (mode == null) {
                throw new ObjectGridRuntimeException("No mode specified");
            }
            if (userid != null) {
                connection = topicConnectionFactory.createTopicConnection(userid,
        password);
            } else
                connection = topicConnectionFactory.createTopicConnection();

            // need to start the connection to receive messages.
            connection.start();
```

```
            // the jms session is not transactional (false).
            jmsSession = connection.createTopicSession(false,
        javax.jms.Session.AUTO_ACKNOWLEDGE);
            if (topic == null)
                if (topicName == null) {
                    throw new ObjectGridRuntimeException("Topic not specified");
                } else {
                    topic = jmsSession.createTopic(topicName);
                }
            publisher = jmsSession.createPublisher(topic);
            // start the listener thread.
            listenerRunning = true;
            listenerThread = new Thread(this);
            listenerThread.start();
         } catch (Throwable e) {
            throw new ObjectGridRuntimeException("Cannot initialize", e);
         }
    }
```

L'unité d'exécution lancée par le code est une unité Java 2 Platform, Standard Edition (Java SE). Pour une exécution sur WebSphere Application Server version 6.x ou sur WebSphere Application Server version 5.x Enterprise, vous devrez utiliser l'API de bean asynchrone pour lancer cette unité d'exécution de démon. Vous pouvez également utiliser les API communes. Voici un exemple de fragment de code montrant la même action effectuée à l'aide d'un gestionnaire de travaux :

```
// start the listener thread.
listenerRunning = true;
workManager.startWork(this, true);
```

Par ailleurs, le plug-in doit implémenter l'interface Work et non l'interface Runnable. Vous devez également ajouter une méthode release pour définir comme false la variable listenerRunning. Le plug-in doit être fourni avec une instance WorkManager dans son constructeur ou par injection si l'on utilise un conteneur IoC (Inversion of Control).

Transmission des modifications

Voici un exemple de méthode transactionEnd pour la publication des modifications locales apportées à un ObjectGrid. Cet exemple utilise JMS, bien qu'il soit possible d'utiliser n'importe quel transport de messages capable de publication/abonnement fiable.

Exemple de méthode transactionEnd

```
// This method is synchronized to make sure the
    // messages are published in the order the transaction
    // were committed. If we started publishing the messages
    // in parallel then the receivers could corrupt the Map
    // as deletes may arrive before inserts etc.
    public synchronized void transactionEnd(String txid, boolean isWriteThroughEnabled,
    boolean committed,
            Collection changes) {
        try {
            // must be write through and commited.
            if (isWriteThroughEnabled && committed) {
                // write the sequences to a byte []
                ByteArrayOutputStream bos = new ByteArrayOutputStream();
                ObjectOutputStream oos = new ObjectOutputStream(bos);
                if (publishMaps.isEmpty()) {
                    // serialize the whole collection
                    LogSequenceTransformer.serialize(changes, oos, this, mode);
                } else {
                    // filter LogSequences based on publishMaps contents
                    Collection publishChanges = new ArrayList();
                    Iterator iter = changes.iterator();
                    while (iter.hasNext()) {
                        LogSequence ls = (LogSequence) iter.next();
```

```
                    if (publishMaps.contains(ls.getMapName())) {
                        publishChanges.add(ls);
                    }
                }
                LogSequenceTransformer.serialize(publishChanges, oos, this, mode);
            }
            // make an object message for the changes
            oos.flush();
            ObjectMessage om = jmsSession.createObjectMessage(bos.toByteArray());
            // set properties
            om.setStringProperty(PROP_TX, txid);
            om.setStringProperty(PROP_GRIDNAME, myGrid.getName());
            // transmit it.
            publisher.publish(om);
        }
    } catch (Throwable e) {
        throw new ObjectGridRuntimeException("Cannot push changes", e);
    }
}
```

Cette méthode utilise plusieurs variables d'instance :

- La variable `jmsSession` : session JMS servant à publier les messages. Elle est créée lors de l'initialisation du plug-in.
- La variable `mode` : le mode de répartition.
- La variable `publishMaps` : ensemble contenant le nom de chacune des mappes dont les modifications sont à publier. La variable vide signifie que la totalité des mappes sont à publier.
- La variable `publisher` : objet TopicPublisher qui est créé durant l'initialisation du plug-in.

Réception et application des messages d'actualisation

Vient à présent la méthode run. Cette méthode s'exécute en boucle jusqu'à ce que l'application arrête la boucle. Chaque itération de la boucle tente de réceptionner un message JMS et de l'appliquer à l'ObjectGrid.

Exemple de méthode run de message JMS

```
private synchronized boolean isListenerRunning() {
        return listenerRunning;
    }

    public void run () {
        try {
            System.out.println("Listener starting");
            // get a jms session for receiving the messages.
            // Non transactional.
            TopicSession myTopicSession;
            myTopicSession = connection.createTopicSession(false, javax.jms.
       Session.AUTO_ACKNOWLEDGE);

            // get a subscriber for the topic, true indicates don't receive
            // messages transmitted using publishers
            // on this connection. Otherwise, we'd receive our own updates.
            TopicSubscriber subscriber = myTopicSession.createSubscriber(topic,
       null, true);
            System.out.println("Listener started");
            while (isListenerRunning()) {
                ObjectMessage om = (ObjectMessage) subscriber.receive(2000);
                if (om != null) {
                    // Use Session that was passed in on the initialize...
                    // very important to use no write through here
                    mySession.beginNoWriteThrough();
                    byte[] raw = (byte[]) om.getObject();
                    ByteArrayInputStream bis = new ByteArrayInputStream(raw);
                    ObjectInputStream ois = new ObjectInputStream(bis);
                    // inflate the LogSequences
```

```
                Collection collection = LogSequenceTransformer.inflate(ois,
        myGrid);
                Iterator iter = collection.iterator();
                while (iter.hasNext()) {
                    // process each Maps changes according to the mode when
                    // the LogSequence was serialized
                    LogSequence seq = (LogSequence) iter.next();
                    mySession.processLogSequence(seq);
                }
                mySession.commit();
            } // if there was a message
          } // while loop
          // stop the connection
          connection.close();
      } catch (IOException e) {
          System.out.println("IO Exception: " + e);
      } catch (JMSException e) {
          System.out.println("JMS Exception: " + e);
      } catch (ObjectGridException e) {
          System.out.println("ObjectGrid exception: " + e);
          System.out.println("Caused by: " + e.getCause());
      } catch (Throwable e) {
          System.out.println("Exception : " + e);
      }
      System.out.println("Listener stopped");
  }
```

Programme d'écoute d'événement JMS

Java

Le programme JMSObjectGridEventListener est conçu pour prendre en charge l'invalidation du cache local côté client et un mécanisme de réplication entre homologues. Il s'agit d'une implémentation JMS (Java Message Service) de l'interface ObjectGridEventListener.

Le mécanisme d'invalidation de client peut être utilisé dans un environnement eXtreme Scale réparti pour garantir la synchronisation des données du cache local avec les serveurs ou les autres clients. Sans cette fonction, le cache local du client pourrait contenir des données obsolètes. Toutefois, même avec ce mécanisme d'invalidation de client JMS, vous devez prendre en compte le délai de mise à jour d'un cache local client en raison du retard de la publication des mises à jour.

Le mécanisme de réplication entre homologues peut être utilisé dans les environnements eXtreme Scale répartis et locaux. Il s'agit d'un processus de réplication de coeur à coeur qui permet aux mises à jour de données de circuler parmi les ObjectGrid locales et les ObjectGrid réparties. Par exemple, avec ce mécanisme, vous pouvez transférer des mises à jour de données d'une grille répartie vers une grille locale ou d'une grille vers une autre grille d'un autre domaine système.

Le programme JMSObjectGridEventListener exige de l'utilisateur la configuration des informations JMS et JNDI (Java Naming and Directory Interface) pour obtenir les ressources JMS requises. En outre, les propriétés de réplication doivent être définies correctement. Dans un environnement JEE, les informations JNDI doivent être disponibles dans les conteneurs Web et EJB (Enterprise JavaBean). Dans ce cas, la propriété JNDI est facultative à moins que vous ne souhaitiez obtenir des ressources JMS externes.

Ce programme d'écoute d'événement comporte des propriétés que vous pouvez configurer avec le langage XML ou à l'aide de programmes et pouvant être

utilisées pour l'invalidation de client, pour la réplication entre homologues ou les deux. La plupart des propriétés sont facultatives afin de personnaliser le comportement permettant d'obtenir les fonctionnalités dont vous avez besoin.

.

Pour plus d'informations, consultez l'API JMSObjectGridEventListener.

Extension du plug-in JMSObjectGridEventListener

Le plug-in JMSObjectGridEventListener permet aux instances ObjectGrid homologues de recevoir des mises à jour lorsque les données de la grille sont modifiées ou expulsées. Il permet également aux clients d'être avertis lors de la mise à jour ou de l'expulsion d'entrées d'une grille eXtreme Scale. Cette rubrique décrit l'extension du plug-in JMSObjectGridEventListener pour permettre aux applications d'obtenir une notification à réception d'un message JMS. Cette fonction est particulièrement utile lors de l'utilisation du paramètre CLIENT_SERVER_MODEL pour l'invalidation de client.

Lors d'une exécution avec le rôle récepteur, la méthode JMSObjectGridEventListener.onMessage substituée est automatiquement appelée par l'exécution eXtreme Scale lorsque l'instance JMSObjectGridEventListener reçoit des mises à jour du message JMS de la grille. Ces messages incluent une collection d'objets LogSequence. Les objets LogSequence sont transmis à la méthode onMessage et l'application utilise l'objet LogSequence pour identifier les entrées de cache qui ont été insérées, supprimées, mises à jour ou invalidées.

Pour utiliser le point d'extension onMessage, les applications suivent les étapes ci-dessous.

1. Crée une classe, en étendant la classe JMSObjectGridEventListener et en substituant la méthode onMessage.
2. Configure la classe étendue JMSObjectGridEventListener de la même manière que la classe ObjectGridEventListener pour l'ObjectGrid.

La classe étendue JMSObjectGridEventListener est un enfant de la classe JMSObjectGridEventListener et peut uniquement se substituer à deux méthodes : les méthodes initialize (facultative) et onMessage. Si une classe enfant de la classe JMSObjectGridEventListener doit utiliser des artefacts ObjectGrid tels que ObjectGrid ou Session dans la méthode onMessage, elle peut obtenir ces artefacts dans la méthode initialize et les mettre en cache en tant que variables d'instance. De même, dans la méthode onMessage, les artefacts ObjectGrid mis en cache doivent être utilisés pour traiter une collection de LogSequences transmise.

Remarque : La méthode initialize remplacée doit appeler la méthode super.initialize pour initialiser la classe parent JMSObjectGridEventListener de manière appropriée.

Voici un exemple de classe étendue JMSObjectGridEventListener.

```
package com.ibm.websphere.samples.objectgrid.jms.price;

import java.util.*;
import com.ibm.websphere.objectgrid.*;
import com.ibm.websphere.objectgrid.plugins.LogElement;
import com.ibm.websphere.objectgrid.plugins.LogSequence;
import com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener;

public class ExtendedJMSObjectGridEventListener extends JMSObjectGridEventListener{
 protected static boolean debug = true;
```

```
   /**
    * Grille associée à ce programme d'écoute d'événement.
    */
   ObjectGrid grid;

   /**
    * Session associée à ce programme d'écoute d'événement.
    */
   Session session;

   String objectGridType;

   public List receivedLogSequenceList = new ArrayList();


/* (non-Javadoc)
 * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
   #initialize(com.ibm.websphere.objectgrid.Session)
 */
public void initialize(Session session) {
 // Remarque : si l'utilisation d'un artefact ObjectGrid est requise,
             cette classe doit obtenir l'ObjectGrid
 //  de l'instance Session transmise et obtenir l'ObjectMap de l'instance Session
 // pour toutes les opérations de mappe ObjectGrid transactionnelles.

 super.initialize(session);  // doit appeler la méthode initialize super.
 this.session = session; // mettez en cache l'instance de session si son utilisation est
 //   requise pour effectuer une opération de mappe.
 this.grid = session.getObjectGrid(); // obtenez ObjectGrid, si vous devez obtenir
 //   des informations d'ObjectGrid.

 if (grid.getObjectGridType() == ObjectGrid.CLIENT)
  objectGridType = "CLIENT";
 else if (grid.getObjectGridType() == ObjectGrid.SERVER)
  objectGridType = "Server";

 if (debug)
  System.out.println("ExtendedJMSObjectGridEventListener[" +
    objectGridType + "].initialize() : grid = " + this.grid);
}

/* (non-Javadoc)
 * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
   #onMessage(java.util.Collection)
 */
protected void onMessage(Collection logSequences) {
 System.out.println("ExtendedJMSObjectGridEventListener[" +
   objectGridType + "].onMessage(): ");

 Iterator iter = logSequences.iterator();

 while (iter.hasNext()) {
           LogSequence seq = (LogSequence) iter.next();

     StringBuffer buffer = new StringBuffer();
     String mapName = seq.getMapName();
     int size = seq.size();
     buffer.append("\nLogSequence[mapName=" + mapName + ", size=" + size + ",
    objectGridType=" + objectGridType
       + "]: ");

     Iterator logElementIter = seq.getAllChanges();
     for (int i = seq.size() - 1; i >= 0; --i) {
      LogElement le = (LogElement) logElementIter.next();
      buffer.append(le.getType() + " -> key=" + le.getCacheEntry().getKey() + ", ");
     }
     buffer.append("\n");

     receivedLogSequenceList.add(buffer.toString());

     if (debug) {
      System.out.println("ExtendedJMSObjectGridEventListener["
     + objectGridType + "].onMessage(): " + buffer.toString());
     }
 }
}

public String dumpReceivedLogSequenceList() {
 String result = "";
```

```
  int size = receivedLogSequenceList.size();
  result = result + "\nExtendedJMSObjectGridEventListener[" + objectGridType
    + "]: receivedLogSequenceList size = " + size + "\n";
  for (int i = 0; i < size; i++) {
   result = result + receivedLogSequenceList.get(i) + "\n";
  }
  return result;
 }

 public String toString() {
  return "ExtendedJMSObjectGridEventListener["
    + objectGridType + " - " + this.grid + "]";
 }
}
```

Configuration

La classe étendue JMSObjectGridEventListener doit être configurée de la même manière pour le mécanisme d'invalidation de client que pour le mécanisme de réplication entre homologues. L'exemple suivant illustre l'approche de la configuration XML.

```
<objectGrid name="PRICEGRID">
   <bean id="ObjectGridEventListener"
    className="com.ibm.websphere.samples.objectgrid.jms.
      price.ExtendedJMSObjectGridEventListener">
    <property name="invalidationModel" type="java.lang.String"
     value="CLIENT_SERVER_MODEL" description="" />
    <property name="invalidationStrategy" type="java.lang.String"
     value="INVALIDATE" description="" />
    <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
     value="jms/TCF" description="" />
    <property name="jms_topicJndiName" type="java.lang.String"
     value="GRID.PRICEGRID" description="" />
    <property name="jms_topicName" type="java.lang.String"
     value="GRID.PRICEGRID" description="" />
    <property name="jms_userid" type="java.lang.String" value=""
     description="" />
    <property name="jms_password" type="java.lang.String" value=""
     description="" />
   </bean>
   <backingMap name="PRICE" pluginCollectionRef="PRICE"></backingMap>
      </objectGrid>
```

Remarque : Le nom de classe du bean ObjectGridEventListener est configuré à l'aide de la classe étendue JMSObjectGridEventListener avec les mêmes propriétés que la classe générique JMSObjectGridEventListener.

Configuration de règles de déploiement

Le fichier XML du descripteur de la règle de déploiement et le fichier XML du descripteur d'ObjectGrid permettent de gérer une topologie répartie. La stratégie de déploiement est codée sous la forme d'un fichier XML qui est fourni au serveur de conteneur. La règle de déploiement fournit des informations concernant les mappes, les groupe de mappes, les partitions, les fragments réplique, etc. Elle contrôle également les comportements pour le positionnement des fragments.

Configuration de déploiements répartis

Utilisez le fichier XML descripteur de la stratégie de déploiement et le fichier XML descripteur d'ObjectGrid pour gérer votre topologie.

La stratégie de déploiement est codée dans un fichier XML qui est fourni au serveur de conteneur eXtreme Scale. Le fichier XML spécifie les informations suivantes :

- Les mappes appartenant à chaque ensemble de mappes
- Le nombre de partitions
- Le nombre de fragments réplique synchrones et asynchrones

La règle de déploiement contrôle également les comportements de positionnement ci-après.

- Le nombre minimal de serveurs de conteneur actifs avant le placement
- Le remplacement automatique des fragments perdus
- Le positionnement de chaque fragment d'une partition sur une autre machine

Les informations sur les points de contact ne sont pas préconfigurées dans l'environnement dynamique. La règle de déploiement ne contient pas de noms de serveur ou d'informations sur la topologie physique. Tous les fragments dans une grille de données sont placés automatiquement dans les serveurs de conteneur par le service de catalogue. Ce dernier utilise les contraintes définies par la règle de déploiement pour gérer automatiquement le positionnement des fragments. Ce placement automatique des fragments facilite la configuration des grilles de données volumineuses. Vous pouvez également ajouter des serveurs à votre environnement, si nécessaire.

Restriction : Dans un environnement WebSphere Application Server, une taille de groupe central de plus de 50 membres n'est pas prise en charge.

Un fichier XML descripteur de stratégie de déploiement est transmis au serveur de conteneur lors du démarrage. Une règle de déploiement doit être utilisée avec un fichier XML d'ObjectGrid. La stratégie de déploiement n'est pas nécessaire pour démarrer un serveur de conteneur, mais elle est recommandée. La règle de déploiement doit être compatible avec le fichier XML d'ObjectGrid qui lui est associé. Pour chaque élément objectgridDeployment de la règle de déploiement, vous devez inclure un élément objectGrid correspondant dans votre fichier XML d'ObjectGrid. Les mappes de l'élément objectgridDeployment doivent être cohérentes avec les éléments backingMap du XML d'ObjectGrid. Chaque mappe de sauvegarde backingMap doit être référencée sans un seul élément mapSet.

Dans l'exemple ci-après, le fichier `companyGridDpReplication.xml` doit être associé au fichier `companyGrid.xml` correspondant.

companyGridDpReplication.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org./2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="CompanyGrid">
  <mapSet name="mapSet1" numberOfPartitions="11"
   minSyncReplicas="1" maxSyncReplicas="1"
   maxAsyncReplicas="0" numInitialContainers="4">
   <map ref="Customer" />
   <map ref="Item" />
   <map ref="OrderLine" />
   <map ref="Order" />
  </mapSet>
 </objectgridDeployment>

</deploymentPolicy>
```

companyGrid.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">

 <objectGrids>
  <objectGrid name="CompanyGrid">
   <backingMap name="Customer" />
   <backingMap name="Item" />
   <backingMap name="OrderLine" />
```

```
    <backingMap name="Order" />
   </objectGrid>
  </objectGrids>

</objectGridConfig>
```

Le fichier `companyGridDpReplication.xml` contient un élément mapSet divisé en 11 partitions. Chaque partition doit posséder exactement une réplique synchrone. Le nombre de fragments réplique synchrones est spécifié par les attributs minSyncReplicas et maxSyncReplicas. L'attribut minSyncReplicas ayant la valeur 1, chaque partition de l'élément mapSet doit disposer d'au moins une réplique synchrone disponible pour traiter les transactions d'écriture. Etant donné que l'attribut maxSyncReplicas a la valeur 1, chaque partition ne peut dépasser une réplique synchrone. Les partitions de cet élément mapSet ne possèdent pas de fragments réplique asynchrones.

L'attribut numInitialContainers indique au service de catalogue de retarder le placement jusqu'à ce que quatre serveurs soient disponibles pour prendre en charge cette instance ObjectGrid. L'attribut numInitialContainers est ignoré lorsque le nombre de serveurs de conteneur est atteint.

Vous pouvez également utiliser la propriété **placementDeferralInterval** et la commande **xscmd -c suspendBalancing** pour retarder le placement des fragments sur les serveurs de conteneur.

Bien que le fichier `companyGridDpReplication.xml` soit un exemple simple, une règle de déploiement peut vous permettre de contrôler intégralement votre environnement.

Topologie répartie

Les mémoires cache cohérentes réparties permettent d'améliorer les performances, la disponibilité et l'évolutivité du système, que vous pouvez configurer.

WebSphere eXtreme Scale équilibre automatiquement les serveurs. Vous pouvez inclure des serveurs supplémentaires sans redémarrer WebSphere eXtreme Scale. L'ajout de serveurs supplémentaires sans avoir à redémarrer eXtreme Scale permet d'avoir des déploiements simples, mais également des déploiements de grande taille se chiffrant en téraoctets et comptant plusieurs milliers de serveurs.

Cette topologie de déploiement est flexible. A l'aide du service de catalogue, vous pouvez ajouter et supprimer des serveurs afin de mieux utiliser les ressources sans supprimer l'intégralité du cache. Vous pouvez utiliser les commandes **startOgServer** et **stopOgServer** ou **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs de conteneur. Ces deux commandes nécessitent de spécifier l'option **-catalogServiceEndPoints**. Tous les clients d'une topologie répartie communiquent avec le service de catalogue via le protocole IIOP (Internet Interoperability Object Protocol). Tous les clients utilisent l'interface ObjectGrid pour communiquer avec les serveurs.

La fonctionnalité de configuration dynamique de WebSphere eXtreme Scale facilite l'ajout de ressources au système. Les conteneurs hébergent les données et le service de catalogue permet aux clients de communiquer avec la grille de serveurs de conteneur. Le service de catalogue transmet les demandes, alloue de l'espace dans les serveurs de conteneur hôtes et gère l'état et la disponibilité de l'ensemble du système. Les clients se connectent à un service de catalogue, extraient la description de la topologie des serveurs de conteneur et communiquent directement avec chaque serveur. Lorsque la topologie des serveurs change suite à

l'ajout de serveurs ou de la défaillance d'autres serveurs, le service de catalogue achemine automatiquement les demandes client au serveur approprié qui héberge les données.

Un service de catalogue existe dans sa propre grille de machines virtuelles Java. Un même serveur de catalogues peut gérer plusieurs serveurs. Vous pouvez démarrer un serveur de conteneur dans une machine virtuelle Java seule ou le charger dans une machine virtuelle Java arbitraire avec d'autres serveurs de conteneur pour différents serveurs. Un client peut exister dans une machine virtuelle Java et communiquer avec un ou plusieurs serveurs. Un client peut également exister dans la même machine virtuelle Java qu'un serveur de conteneur.

Vous pouvez également créer une stratégie de déploiement à l'aide d'un programme lorsque vous intégrez un serveur de conteneur dans un processus ou une application Java. Pour plus d'informations, consultez la documentation de l'API DeploymentPolicy.

Contrôle du placement avec des zones

Utilisez votre stratégie de déploiement pour définir des zones. Les zones vous permettent de contrôler le placement des fragments dans WebSphere eXtreme Scale. Les zones sont un concept logique défini par l'utilisateur. Elles permettent de représenter les regroupements logiques des serveurs physiques.

Configuration de zones pour le positionnement de fragments réplique

La prise en charge des zones permet la configuration avancée du positionnement de réplique pour plusieurs centres de données. Grâce à cette fonctionnalité, des grilles de milliers de partitions peuvent être facilement gérées à l'aide de plusieurs règles de positionnement facultatives. Un centre de données peut correspondre à plusieurs étages d'un bâtiment, plusieurs bâtiments, plusieurs villes ou d'autres distinctions, selon la configuration des règles de zone.

Flexibilité des zones

Il est possible de placer des fragments dans des zones. Cette fonction permet de mieux contrôler la manière dont eXtreme Scale place les fragments dans une grille. Les machines virtuelles Java qui hébergent un serveur eXtreme Scale peuvent être marquées à l'aide d'un identificateur de zone. Le fichier de déploiement peut désormais inclure une ou plusieurs règles de zone, qui sont associées à un type de fragment. La section suivante présente l'utilisation des zones. Pour plus d'informations, voir les informations sur la surveillance du placement des fragments avec des zones dans le *Guide d'administration*.

Les zones de positionnement contrôlent la manière dont eXtreme Scale assigne les fragments primaires et les fragments réplique pour configurer les topologies avancées.

Une machine virtuelle Java peut posséder plusieurs conteneurs mais un seul serveur. Un conteneur peut héberger plusieurs fragments d'un seul objet ObjectGrid.

Cette fonctionnalité permet de s'assurer que les fragments réplique et les fragments primaires sont placés dans différents emplacements ou zones et que leur haute disponibilité est optimale. Généralement, eXtreme Scale ne place pas de fragment principal et de fragment réplique dans les machines virtuelles Java possédant une adresse IP identique. Cette règle simple empêche généralement deux serveurs

eXtreme Scale d'être placés sur le même ordinateur physique. Toutefois, vous pouvez avoir besoin d'un mécanisme plus flexible. Par exemple, vous souhaitez utiliser deux châssis lame sur lesquels *segmentés* les fragments primaires et vous voulez que le fragment réplique de chaque fragment primaire soit positionné sur le châssis de l'autre fragment primaire.

Les fragments primaires *à bandes* désignent les fragments primaires placés dans chaque zone. La réplique de chacun de ces fragments primaires est située dans la zone opposée. Par exemple, le fragment primaire 0 se trouve dans la zone A, et le fragment réplique synchronisé 0 dans la zone B. Le fragment primaire 1 se trouve dans la zone B, et le fragment réplique synchronisé 1 dans la zone A.

Dans ce cas, le nom du châssis correspond à celui de la zone. Vous pouvez également nommer les zones en fonction des étages d'un bâtiment et utiliser les zones pour vous assurer que les fragments primaires et les fragments réplique correspondant aux mêmes données se situent à des étages différents. Vous pouvez également utiliser des bâtiments et des centres de données. Des tests effectués sur les centres de données à l'aide de zones ont permis de s'assurer que les données sont répliquées de manière appropriée entre les centres de données. Si vous utilisez HTTP Session Manager pour eXtreme Scale, vous pouvez également utiliser des zones. Cette fonction vous permet de déployer une seule application Web sur les trois centres de données et de vous assurer que les sessions HTTP des utilisateurs sont répliquées dans les centres de données afin que les sessions puissent être récupérées en cas de défaillance d'un centre de données entier.

WebSphere eXtreme Scale prend en compte la nécessité de gérer une grille volumineuse dans plusieurs centres de données. Il est possible de s'assurer que les sauvegardes et les fragments primaires de la même partition sont situés dans des centres de données différents, le cas échéant. Il permet de placer tous les fragments primaires dans le centre de données 1 et tous les fragments réplique dans le centre de données 2. Il peut également permuter de manière circulaire les fragments primaires et les fragments réplique entre les deux centres de données. Les règles sont flexibles de sorte que de nombreux scénarios sont possibles. eXtreme Scale peut également gérer des milliers de serveurs. Cette fonctionnalité, combinée au positionnement automatique en fonction des centres de données, rend ces grilles volumineuses plus économiques d'un point de vue administratif. Les administrateurs peuvent spécifier ce qu'ils veulent faire de manière simple et efficace.

En tant qu'administrateur, utilisez les zones de positionnement pour définir les emplacements des fragments primaires et des fragments réplique. Vous pouvez ainsi configurer des topologies avancées hautes performances et haute disponibilité. Vous pouvez définir une zone dans tout groupement logique de processus eXtreme Scale, comme indiqué ci-dessus : ces zones peuvent correspondre à des emplacements de stations de travail physiques, tels qu'un centre de données, un étage d'un centre de données ou un châssis lame. Vous pouvez segmenter les données dans les zones, afin de bénéficier d'une disponibilité accrue. Vous pouvez également diviser les fragments primaires et les fragments réplique en zones distinctes si un secours automatique est nécessaire.

Association d'un serveur eXtreme Scale à une zone qui n'utilise pas WebSphere Extended Deployment

Si eXtreme Scale est utilisé avec Java Standard Edition ou un serveur d'application qui n'est pas basé sur WebSphere Extended Deployment version 6.1, il est possible d'associer une JVM utilisée comme conteneur de fragments à une zone, si vous utilisez les méthodes suivantes.

Applications utilisant le script de démarrage.

Le script de démarrage de serveur permet de démarrer une application eXtreme Scale lorsqu'elle n'est pas intégrée dans un serveur existant. Le paramètre **-zone** permet de spécifier la zone à utiliser pour tous les conteneurs du serveur.

Spécification de la zone lors du démarrage d'un conteneur à l'aide d'API

Le nom de zone d'un conteneur peut être spécifié comme décrit dans la documentation de «API de serveurs intégrés», à la page 513.

Association de noeuds WebSphere Extended Deployment à des zones

Si vous utilisez eXtreme Scale avec des applications WebSphere Extended Deployment Java EE, vous pouvez utiliser les groupes de noeuds WebSphere Extended Deployment pour placer les serveurs dans des zones spécifiques.

Dans eXtreme Scale, une JVM ne peut être membre que d'une seule zone. Toutefois, WebSphere autorise un noeud à faire partie de plusieurs groupes. Vous pouvez utiliser cette fonctionnalité des zones eXtreme Scale si vous vous assurez que chacun des noeuds se trouve uniquement dans un groupe de noeuds de zone.

Utilisez la syntaxe suivante pour nommer un groupe de noeuds afin de le déclarer en tant que zone : `ReplicationZone<UniqueSuffix>`. Les serveurs exécutés sur un noeud faisant partie d'un tel groupe sont inclus dans la zone spécifiée par le nom du groupe. Vous trouverez ci-dessous la description d'un exemple de topologie.

Tout d'abord, vous devez configurer quatre noeuds : node1, node2, node3 et node4, chaque noeud possédant deux serveurs. Ensuite, vous créez deux groupes de noeuds que vous nommez ReplicationZoneA et ReplicationZoneB. Vous ajoutez node1 et node2 à ReplicationZoneA , et node3 et node4 à ReplicationZoneB.

Lors du démarrage des serveurs de node1 et node2, ils font partie de ReplicationZoneA. De la même manière, les serveurs de node3 et node4 font partie de ReplicationZoneB.

Une machine virtuelle Java membre de grille vérifie l'appartenance aux zones uniquement lors du démarrage. L'ajout d'un nouveau groupe de noeuds ou la modification de l'appartenance a une incidence uniquement sur les machines virtuelles Java démarrées ou redémarrées.

Règles de zone

Une partition eXtreme Scale possède un fragment primaire et aucune réplique ou plus. Pour cet exemple, considérons les conventions d'attribution de nom suivantes pour les fragments. `P` est le fragment primaire ; `S` est une réplique synchrone et `A` une réplique asynchrone. Une règle de zone comporte trois composants :

- un nom de règle

- une liste de zones
- un indicateur inclusif ou exclusif

Le nom de zone d'un conteneur peut être spécifié comme décrit dans la documentation de «API de serveurs intégrés», à la page 513. Une règle de zone spécifie l'ensemble de zones dans lequel un fragment peut être placé. L'indicateur inclusif indique que le positionnement d'un fragment dans une zone de la liste entraîne le positionnement de tous les autres fragments dans la même liste. Un paramètre exclusif indique que chaque fragment d'une partition est placé dans une zone différente de la liste. Par exemple, si vous utilisez un paramètre exclusif et s'il existe trois fragments (un fragment primaire et deux répliques synchrones), la liste doit alors contenir trois zones.

Chaque fragment peut être associé à une règle de zone. Une règle de zone peut être partagée par deux fragments. Lorsqu'une règle est partagée, l'indicateur inclusif ou exclusif s'applique aux fragments de tout type qui partagent une règle unique.

Exemples

Vous trouverez ci-dessous des exemples illustrant les différents scénarios et la configuration de déploiement permettant d'implémenter ces derniers.

Segmentation des fragments primaires et des fragments réplique dans les zones

Vous disposez de trois châssis lame et souhaitez y répartir les fragments primaires, en plaçant une réplique synchrone dans un châssis différent du fragment primaire. Définissez chaque châssis en tant que zone en les nommant ALPHA, BETA et GAMMA. Exemple de syntaxe XML de déploiement :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:schemaLocation=
 "http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
    xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
   <mapSet name="ms1" numberOfPartitions="37" minSyncReplicas="1"
    maxSyncReplicas="1" maxAsyncReplicas="0">
   <map ref="book" />
   <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="stripeZone"/>
    <shardMapping shard="S" zoneRuleRef="stripeZone"/>
    <zoneRule name ="stripeZone" exclusivePlacement="true" >
     <zone name="ALPHA" />
     <zone name="BETA" />
     <zone name="GAMMA" />
    </zoneRule>
   </zoneMetadata>
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Cette syntaxe de déploiement XML contient une grille appelée "library" (bibliothèque) qui contient une mappe unique appelée "book". Elle utilise quatre partitions avec une seule réplique synchrone. La clause des métadonnées de zone affiche la définition d'une seule règle de zone et l'association des règles de zone à des fragments. Les fragments primaires et synchrones sont associés à la règle de zone "stripeZone". La règle de zone contient les trois zones et utilise le positionnement exclusif. D'après cette règle, si le fragment primaire de la partition 0 est placé dans ALPHA, le fragment réplique de la partition 0 sera placée dans BETA ou dans GAMMA. De la même manière, les fragments primaires des autres partitions sont placés dans d'autres zones que les fragments réplique.

Réplique asynchrone dans une zone différente de celle du fragment primaire et du fragment réplique synchrone

Dans cet exemple, une connexion avec un temps d'attente élevé existe entre deux bâtiments. Vous souhaitez une haute disponibilité sans perte de données pour tous les scénarios. Toutefois, l'incidence de la réplication synchrone sur les performances entre les bâtiments nécessite un compromis. Vous souhaitez un fragment primaire avec une réplique synchrone dans un bâtiment et une réplique asynchrone dans l'autre bâtiment. Généralement, les défaillances qui se produisent sont des arrêts de JVM ou des blocages de l'ordinateur plutôt que des problèmes à grande échelle. Cette topologie permet d'éviter la perte de données en cas de défaillance normale. La perte d'un bâtiment est suffisamment rare pour qu'une perte de données soit acceptable dans ce cas-là. Vous pouvez créer deux zones, une pour chaque bâtiment. Fichier XML de déploiement :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="1"
   maxSyncReplicas="1" maxAsyncReplicas="1">
   <map ref="book" />
   <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primarySync"/>
    <shardMapping shard="S" zoneRuleRef="primarySync"/>
    <shardMapping shard="A" zoneRuleRef="aysnc"/>
    <zoneRule name ="primarySync" exclusivePlacement="false" >
      <zone name="BldA" />
      <zone name="BldB" />
    </zoneRule>
    <zoneRule name="aysnc" exclusivePlacement="true">
      <zone name="BldA" />
      <zone name="BldB" />
    </zoneRule>
   </zoneMetadata>
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Le fragment primaire et le fragment réplique synchrone partagent une règle de zone primaySync avec un paramètre d'indicateur exclusif défini sur "false". Après le positionnement dans une zone du fragment primaire ou de son fragment réplique synchrone, l'autre est placé dans la même zone. La réplique asynchrone utilise une deuxième règle de zone avec les mêmes zones que la règle de zone primarySync mais elle utilise l'attribut **exclusivePlacement** défini sur "true". L'attribut indique qu'un fragment ne peut être placé dans une zone contenant un fragment issu d'une même partition. Par conséquent, le fragment réplique asynchrone n'est pas placé dans la même zone que le fragment primaire ou les fragments réplique synchrones.

Placement de tous les fragments primaires dans une zone et de tous les fragments réplique dans une autre

Dans ce cas, tous les fragments primaires se trouvent dans une zone spécifique et tous les fragments réplique dans une autre zone. Nous obtenons un fragment primaire et une réplique asynchrone unique. Tous les fragments réplique seront placées dans la zone A et les fragments primaires dans la zone B.

```
 <?xml version="1.0" encoding="UTF-8"?>

 <deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
   "http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="library">
   <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
    maxSyncReplicas="0" maxAsyncReplicas="1">
    <map ref="book" />
    <zoneMetadata>
     <shardMapping shard="P" zoneRuleRef="primaryRule"/>
     <shardMapping shard="A" zoneRuleRef="replicaRule"/>
     <zoneRule name ="primaryRule">
      <zone name="A" />
```

```
    </zoneRule>
    <zoneRule name="replicaRule">
     <zone name="B" />
      </zoneRule>
     </zoneMetadata>
    </mapSet>
   </objectgridDeployment>
 </deploymentPolicy>
```

Cet exemple contient deux règles, l'une pour les fragments primaires (P), l'autre pour le fragment réplique (A).

Zones correspondant à des réseaux étendus (WAN)

Vous pouvez souhaiter déployer une instance unique de eXtreme Scale dans plusieurs bâtiments ou centres de données où les interconnexions réseau sont plus lentes. La lenteur accrue des connexions réseau entraîne la réduction de la bande passante et l'augmentation des temps d'attente pour les connexions. Dans ce mode, des partitions réseau sont plus susceptibles de se produire en raison de la congestion du réseau et d'autres facteurs. eXtreme Scale aborde cet environnement difficile de deux manières.

Signal de présence limité entre les zones

Les machines virtuelles Java assemblées en groupes centraux assurent le signal de présence entre elles. Lorsque le service de catalogue organise les machines virtuelles Java en groupes, ces derniers ne s'étendent pas aux zones. Dans ce groupe, un leader transmet les informations d'appartenance au service de catalogue. Ce dernier vérifie les défaillances signalées avant d'entreprendre une action. Pour ce faire, il tente de se connecter aux machines virtuelles Java suspectes. Si le catalogue trouve une fausse détection de défaillance, il n'entreprend pas d'action car la partition de groupe central fonctionnera à nouveau correctement après un court délai.

Le service de catalogue assurera régulièrement le signal de présence des leaders du groupe central à un rythme lent afin de gérer l'isolement du groupe central.

Routage par zone préférée

Avec le routage par zone préférée, vous pouvez définir la manière dont WebSphere eXtreme Scale envoie des transactions vers des zones.

Vous contrôlez l'insertion des fragments dans une grille de données. Voir «Configuration de zones pour le positionnement de fragments réplique», à la page 289 pour plus d'informations sur quelques scénarios de base et la configuration de votre stratégie de déploiement correspondante.

Le routage par zone préférée permet aux clients WebSphere eXtreme Scale de spécifier une préférence pour une zone particulière ou un ensemble de zones. En conséquence, les transactions client sont acheminées vers les zones préférées avant de tenter de les acheminer vers une autre zone.

Exigences concernant le routage par zone préférée

Avant d'utiliser le routage par zone préférée, vérifiez que l'application répond aux exigences de votre scénario.

Le placement de partition par conteneur est nécessaire pour pouvoir utiliser le routage par zone préférée. Cette stratégie de positionnement est adaptée aux applications stockant des données de session dans ObjectGrid. La stratégie de

placement de partition par défaut de WebSphere eXtreme Scale est la `partition fixe`. Les clés sont hachées au moment de la validation de la partition, afin de déterminer quelle partition héberge la paire clé-valeur de la mappe par le biais du positionnement par partition fixe.

Le placement par conteneur affecte les données à une partition aléatoire lorsque la transaction est validée via l'objet SessionHandle. Vous devez pouvoir reconstruire l'objet SessionHandle pour récupérer vos données à partir de la grille de données.

Vous pouvez utiliser des zones pour contrôler plus précisément l'insertion des fragments primaires et de réplique dans votre domaine. L'utilisation de plusieurs zones dans le déploiement offre des avantages lorsque les données se trouvent dans plusieurs emplacements physiques. La séparation géographique des fragments primaires et des fragments de réplique est une façon de s'assurer que la perte irrémédiable d'un centre de données n'affecte pas la disponibilité des données.

Lorsque les données sont réparties dans plusieurs zones, il est probable que les clients soient également répartis dans la topologie. Le routage des clients vers leur zone locale ou leur centre de données local offre à l'évidence un avantage en terme de performance en réduisant la latence réseau. Routez les clients vers des zones locales ou des centres de données locaux lorsque cela est possible.

Configuration de votre topologie pour le routage par zone préférée

Réfléchissez au scénario suivant. Vous disposez de deux centres de données : Chicago et Londres. Pour réduire le temps de réponse des clients, vous souhaitez que les clients lisent et écrivent les données dans leur centre de données local.

Les fragments primaires doivent être placés dans chaque centre de données de sorte que les transactions puissent être écrites localement à partir de chaque emplacement. Les clients doivent avoir connaissance des zones pour pouvoir accéder à la zone locale.

Le placement par conteneur localise les nouveaux fragments primaires dans chaque conteneur qui est démarré. Des répliques sont placées en fonction des règles de zone et de placement définies par la stratégie de déploiement. Par défaut, une réplique est placée dans une zone différente de celle de son fragment primaire. Examinez la règle de déploiement suivante pour ce scénario.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="universe">
  <mapSet name="mapSet1" placementStrategy="PER_CONTAINER"
   numberOfPartitions="3" maxAsyncReplicas="1">
   <map ref="planet" />
  </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Chaque conteneur qui démarre avec la stratégie de déploiement reçoit trois nouveaux fragments. Chaque fragment primaire a un fragment de réplique asynchrone. Démarrez chaque conteneur avec le nom de zone approprié. Utilisez le paramètre **-zone** si vous démarrez les conteneurs avec le script **startOgServer** ou **startXsServer** script.

Pour un serveur de conteneur à Chicago :

- UNIX Linux

```
startOgServer.sh s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

- Windows

```
startOgServer.bat s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

- 8.6+ UNIX Linux

```
startXsServer.sh s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

- 8.6+ Windows

```
startOgServer.bat s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

Si vos conteneurs s'exécutent sur WebSphere Application Server, vous devez créer un groupe de noeuds et le nommer avec le préfixe `ReplicationZone`. Les serveurs qui s'exécutent sur les noeuds de ces groupes de noeuds sont placés dans la zone appropriée. Par exemple, les serveurs s'exécutant sur un noeud de Chicago peuvent appartenir au un groupe de noeuds nommé `ReplicationZoneChicago`.

Pour plus d'informations, voir «Configuration de zones pour le positionnement de fragments réplique», à la page 289.

Les fragments primaires de la zone Chicago ont des répliques dans la zone London. Les fragments primaires de la zone London ont des répliques dans la zone Chicago.

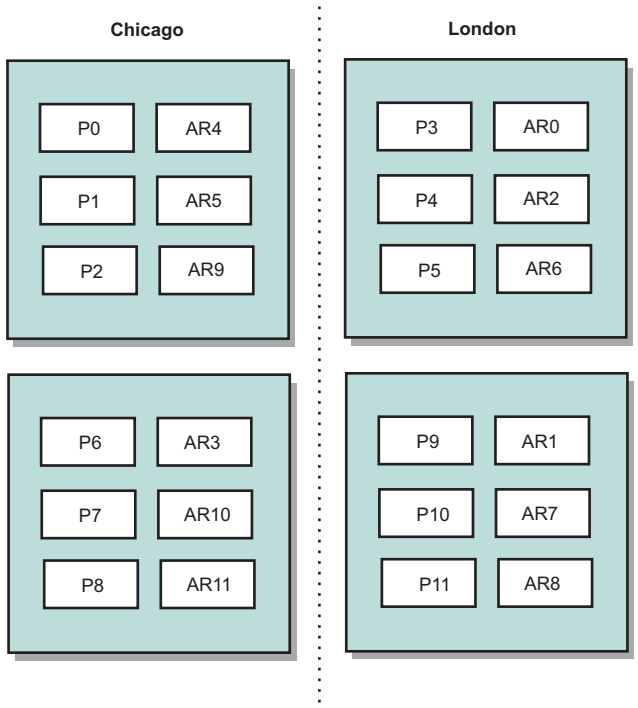


Figure 33. Segments principaux et répliques dans les zones

Définissez les zones préférées pour les clients. Fournissez un fichier de propriétés client à votre machine virtuelle Java (JVM) client. Créez le fichier `objectGridClient.properties` et veillez à le placer dans le chemin d'accès aux classes.

Incluez la propriété **`preferZones`** dans le fichier. Définissez la valeur de propriété sur la zone appropriée. Les clients dans Chicago doivent avoir la valeur suivante dans le fichier `objectGridClient.properties` :

```
preferZones=Chicago
```

Le fichier de propriétés de clients de la zone London doit contenir la valeur suivante :

```
preferZones=London
```

Cette propriété donne l'instruction à chaque client d'acheminer les transactions vers la zone locale dans la mesure du possible. La topologie réplique de manière asynchrone les données insérées dans un fragment primaire dans la zone locale dans la zone externe.

Utilisation de l'interface SessionHandle pour le routage vers la zone locale

La stratégie de placement par conteneur n'utilise pas un algorithme basé sur le hachage pour déterminer l'emplacement de vos paires clé-valeur dans la grille de données. Vous devez utiliser des objet SessionHandle pour que les transactions soient acheminées vers l'emplacement correct lorsque vous utilisez cette stratégie de placement. Lorsqu'une transaction est validée, un objet SessionHandle est lié à la session si aucun objet n'a été défini. L'objet SessionHandle peut également être

lié à la session en appelant la méthode Session.getSessionHandle avant de valider la transaction. Le fragment de code suivant montre un objet SessionHandle lié avant de valider la transaction.

```
Session ogSession = objectGrid.getSession();

// liaison du SessionHandle
SessionHandle sessionHandle = ogSession.getSessionHandle();

ogSession.begin();
ObjectMap map = ogSession.getMap("planet");
map.insert("planet1", "mercury");

// tran est acheminé vers la répartition spécifiée par SessionHandle
ogSession.commit();
```

Supposez que le code précédent a été exécuté sur un client dans le centre de données de Chicago. L'attribut **preferZones** a la valeur `Chicago` pour ce client. Par conséquent, le déploiement routera les transactions vers l'une des partitions principales dans la zone, la partition 0, 1, 2, 6, 7 ou 8.

L'objet SessionHandle fournit un chemin de retour vers la partition qui stocke ces données validées. L'objet SessionHandle doit être réutilisé ou reconstruit et défini dans la session pour revenir à la partition contenant les données validées.

```
ogSession.setSessionHandle(sessionHandle);
ogSession.begin();

// la valeur renvoyée sera "mercury"
String value = map.get("planet1");
ogSession.commit();
```

Le code de cette transaction réutilise l'objet SessionHandle qui a été créé au cours de la transaction d'insertion. La transaction get est routée vers la partition contenant les données insérées. Sans l'objet SessionHandle, la transaction ne peut pas extraire les données insérées.

Conséquences des échecs de conteneur et de zone sur le routage basé sur zones

En règle générale, un client avec la propriété **preferZones** définie route toutes les transactions vers la zone ou les zones spécifiées. Cependant, la perte d'un conteneur entraîne la promotion d'un fragment de réplique comme fragment primaire. Un client qui routait ses transactions vers les partitions de la zone locale doit extraire les données précédemment insérées à partir de la zone distante.

Imaginez le scénario suivant. Un conteneur de la zone de Chicago est perdu. Il contenait précédemment des fragments primaires pour les partitions 0, 1 et 2. Les nouveaux fragments primaires de ces partitions sont ensuite placés dans la zone London, car cette zone contenait les répliques de ces partitions.

Un client de Chicago qui utilise un objet SessionHandle qui pointe vers l'une des partitions basculées reroute maintenant leurs transactions vers London. Les clients de Chicago qui utilisent les nouveaux objets SessionHandle routent leurs transactions vers les fragments primaires de Chicago.

De même, si l'ensemble de la zone Chicago est perdue, toutes les fragments de réplique de la zone London sont promues comme fragments primaires. Dans ce scénario, tous les clients de Chicago routent leurs transactions vers London.

Définition des zones des serveurs de conteneur

Les zones sont des collections de serveurs de conteneur. Un serveur de conteneur peut appartenir uniquement à une seule zone. Un serveur de conteneur est affecté à une zone lorsqu'il démarre.

Pourquoi et quand exécuter cette tâche

Vous devez planifier vos zones avant de démarrer vos serveurs de conteneur, car les serveurs de conteneur définissent leur appartenance à une zone au démarrage. Si vous souhaitez modifier l'appartenance d'un serveur de conteneur à la zone dont il est membre, vous devez redémarrer le serveur avec les informations de la nouvelle zone.

Procédure

- **Définissez les zones des serveurs de conteneur autonomes.**
  1. Utilisez le paramètre **-zone** du script **startOgServer** ou **startXsServer** pour spécifier la zone de tous les conteneurs dans le serveur démarré. Pour plus d'informations sur le démarrage des serveurs, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.
  2. Vous pouvez également affecter des noms à des zones lorsque vous démarrez les serveur de conteneur à l'aide d'un programme avec l'API de serveur intégrée. Pour plus d'informations, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.
- **Définissez des zones pour les serveurs de conteneur exécutés dans WebSphere Application Server.**

  Vous pouvez utiliser des groupes de noeuds pour placer les serveurs de conteneur dans des zones spécifiques. Utilisez la syntaxe suivante pour nommer votre groupe de noeuds pour lui affecter une zone : `ReplicationZone<identifier>`. Lorsque vous définissez des zones dans la stratégie de déploiement, vous devez nommer les zones exactement comme vous avez nommé les groupes de noeuds. Le nom du groupe de noeuds et le nom de zone dans le fichier descripteur XML de stratégie de déploiement doivent être identiques.

  Important : WebSphere Application Server n'interdit pas de placer les noeuds dans plusieurs groupes de noeuds. Etant donné que les serveurs de conteneur ne peuvent se trouver que dans une seule zone, vérifiez que vos noeuds se trouvent exactement dans un groupe de noeuds ReplicationZone.

  Par exemple, divisez quatre noeuds en deux zones, A et B.

  1. Configurez quatre noeuds, noeud 1, noeud 2, noeud 3 et noeud 4, chaque noeud possédant deux serveurs.
  2. Créez le groupe de noeuds ReplicationZoneA et le groupe de noeuds ReplicationZoneB.
  3. Ajoutez le noeud 1 et le noeud 2 à ReplicationZoneA et le noeuds 3 et 4 à ReplicationZoneB.
  4. Définissez ReplicationZoneA et ReplicationZoneB dans le fichier descripteur XML de la stratégie de déploiement. Voir «Exemple : zones dans un environnement WebSphere Application Server», à la page 302 pour un exemple.
  5. Lorsque les serveurs des noeuds 1 et 2 démarrent, ils rejoignent ReplicationZoneA, ou la zone A, dans la configuration WebSphere eXtreme Scale. Les serveurs sur les noeuds 3 et 4 rejoignent ReplicationZoneB, ou la zone B, dans la configuration WebSphere eXtreme Scale.

Exemple : Définitions de zone dans le fichier XML de descripteur de stratégie de déploiement

Vous pouvez définir des zones et des règles de zone avec le fichier XML du descripteur de stratégie de déploiement.

Exemple : fragments primaires et de réplique dans des zones différentes

Cet exemple place les fragments primaires dans une zone donnée et les fragments de réplique dans une zone différente avec une seule réplique asynchrone. Tous les fragments primaires commencent dans la zone DC1. Les fragments de réplique commencent dans la zone DC2.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="library">
 <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
  maxSyncReplicas="0" maxAsyncReplicas="1">
  <map ref="book" />
  <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primaryRule"/>
    <shardMapping shard="A" zoneRuleRef="replicaRule"/>
    <zoneRule name ="primaryRule">
     <zone name="DC1" />
    </zoneRule>
    <zoneRule name="replicaRule">
    </zoneRule>
  </zoneMetadata>
  </mapSet>
  </objectgridDeployment>
 </deploymentPolicy>
```

Une réplique asynchrone est définie dans l'élément ms1 mapSet. Par conséquent, deux fragments existent pour chaque partition : un fragment primaire et un fragment de réplique asynchrone. Dans l'élément zoneMetadata, un élément shardMapping est défini pour chaque fragment: P pour le fragment primaire et DC1 pour le fragment de réplique asynchrone. L'attribut primaryRule spécifie la zone définie des fragments primaires, la zone DC1 simplement, et cette règle doit être utilisée pour le placement des fragments primaires. Les répliques asynchrones sont placées dans la zone DC2.

Toutefois, si la zone DC2 est perdue, les fragments de réplique deviennent indisponibles. La perte ou la défaillance d'un serveur de conteneur dans la zone DC1 peut générer une perte de données, même si une réplique a été spécifiée.

Pour faire face à cette éventualité, vous pouvez ajouter une zone ou une réplique, comme décrit dans les sections suivantes.

Exemple : ajout d'une zone, segmentation des fragments

Le code suivant configure une nouvelle zone :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
   maxSyncReplicas="0" maxAsyncReplicas="1">
     <map ref="book" />
```

```
     <zoneMetadata>
      <shardMapping shard="P" zoneRuleRef="stripeRule"/>
      <shardMapping shard="A" zoneRuleRef="stripeRule"/>
      <zoneRule name="stripeRule" exclusivePlacement="true">
       <zone name="A" />
       <zone name="B" />
      <zone name="C" />
      </zoneRule>
    </zoneMetadata>
   </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Trois zones ont été définies dans ce code : A, B et C. Une règle de zone partagée, stripeRule, est définie à la place de règles de zone de fragments primaires et de fragments de réplique distinctes. Cette règle inclut toutes les zones, avec l'attribut exclusivePlacement affecté de la valeur `true`. La stratégie de placement eXtreme Scale garantit que les fragments primaires et de réplique se trouvent dans des zones distinctes. Avec cette segmentation du placement, les fragments primaires et de réplique sont répartis dans les deux zones conformément à la stratégie. L'ajout d'une troisième zone C permet de ne pas générer une perte de données en cas de perte d'une zone et de conserver les fragments primaire et de réplique de chaque partition. Un problème de zone entraîne la perte du fragment primaire, du fragment de réplique ou d'aucun fragment. Tout fragment perdu est remplacé à partir du fragment survivant dans une zone survivante et il est placé dans l'autres zone survivante.

Exemple : ajout d'une réplique et définition de plusieurs centres de données

Le scénario classique à deux centres de données utilise des réseaux haut débit à faible latence dans chaque centre de données, mais la latence entre les deux centres de données est élevée. Des répliques synchrones sont utilisées dans chaque centre de données où la faible latence réduit l'impact de la réplication sur les temps de réponse. La réplication asynchrone est utilisée entre les centres de données, de sorte que le réseau dont la latence est élevée n'a aucune incidence sur le temps de réponse.

```
<?xml version="1.0" encoding="UTF-8"?>
 <deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="library">
   <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="1"
   maxSyncReplicas="1" maxAsyncReplicas="1">
    <map ref="book" />
    <zoneMetadata>
     <shardMapping shard="P" zoneRuleRef="primarySync"/>
     <shardMapping shard="S" zoneRuleRef="primarySync"/>
     <shardMapping shard="A" zoneRuleRef="async"/>
    <zoneRule name ="primarySync" exclusivePlacement="false" >
      <zone name="DC1" />
      <zone name="DC2" />
     </zoneRule>
     <zoneRule name="async" exclusivePlacement="true">
     <zone name="DC1" />
      <zone name="DC2" />
     </zoneRule>
   </zoneMetadata>
  </mapSet>
  </objectgridDeployment>
 </deploymentPolicy>
```

Les fragments primaires et les fragments de réplique synchrones partagent la règle primarySync avec l'attribut exclusivePlacement affecté de la valeur `false`. L'attribut exclusivePlacement affecté de la valeur false crée une configuration avec les fragments primaires et de réplique synchrone de chaque partition placée dans la même zone. Le fragment de réplique asynchrone utilise une deuxième règle de zone avec principalement les mêmes zones que la règle de zone primarySyn. Toutefois, le fragment de réplique asynchrone utilise l'attribut exclusivePlacement affecté de la valeur `true`. L'attribut exclusivePlacement, lorsqu'il a la valeur true, signifie qu'un fragment ne peut pas être placé dans une zone contenant un fragment issu d'une même partition. En conséquence, le fragment de réplique asynchrone n'est pas placé dans la même zone que le fragment primaire ou de réplique synchrone. Il existe trois fragments par partition dans ce mapSet : un fragment primaire, un fragment de réplique synchrone et un fragment de réplique asynchrone, de sorte qu'il existe trois éléments shardMapping, un pour chaque fragment.

Si une zone est perdue, tous les fragments de réplique asynchrones sont perdus et non régénérés, car ils n'ont pas de zone distinctes. Si les fragments primaire et de réplique sont perdus, le fragment de réplique asynchrone restante devient le fragment primaire et un nouveau fragment de réplique synchrone est créé dans la zone. Les fragments primaires et de réplique sont segmentés dans chaque zone.

Avec le placement exclusif, chaque fragment a sa propre zone ; vous devez avoir suffisamment de zones pour tous les fragments à placer dans leurs propres zones. Si une règle a une zone, un seul fragment peut être placé dans la zone. Avec deux zones, vous pouvez disposer de deux segments dans la zone.

Exemple : zones dans un environnement WebSphere Application Server

Le code suivant configure une nouvelle zone :

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
  <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
   maxSyncReplicas="0" maxAsyncReplicas="1">
     <map ref="book" />
     <zoneMetadata>
      <shardMapping shard="P" zoneRuleRef="stripeRule"/>
      <shardMapping shard="A" zoneRuleRef="stripeRule"/>
      <zoneRule name="stripeRule" exclusivePlacement="true">
      <zone name="ReplicationZoneA" />
       <zone name="ReplicationZoneB" />
      <zone name="ReplicationZoneC" />
      </zoneRule>
    </zoneMetadata>
   </mapSet>
 </objectgridDeployment>
</deploymentPolicy>
```

Pour cet exemple, trois groupes de noeuds sont définis dans l'environnement WebSphere Application Server : ReplicationZoneA, ReplicationZoneB, and ReplicationZoneC. Le nom du groupe de noeuds et le nom de zone dans le fichier XML du descripteur de la stratégie de déploiement doit être identique et contenir le texte `ReplicationZone<identifier>`. Ce fichier définit une configuration similaire à la segmentation des fragments (par exemple, mais indique le nom requis pour une configuration WebSphere Application Server.

Affichage des informations de zone avec l'utilitaire xscmd

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher les informations relatives au déploiement de zone en cours, y compris les données de placement de fragment.

Avant de commencer

- Déployez une grille de données distribuée avec plusieurs centres de données. Pour plus d'informations, voir «Routage par zone préférée», à la page 294.

Pourquoi et quand exécuter cette tâche

Vous pouvez déterminer les informations relatives à votre configuration liées aux paramètres de zone à l'aide de l'utilitaire **xscmd** fourni avec le produit.

Procédure

Utilisez l'utilitaire **xscmd** pour déterminer des informations sur les fragments de données. Exécutez la commande suivante :

```
xscmd -c showPlacement -z zone_name
```

Exemple

Vous pouvez également exécuter un scénario simple à l'aide de l'exemple d'initiation *racine_install_wxs*/ObjectGrid/gettingstarted. Pour plus d'informations, voir «Tutoriel : Démarrer avec WebSphere eXtreme Scale», à la page 1.

1. Démarrerz un serveur de catalogue :

   ```
   runcat.bat
   ```

2. Déterminez le nombre nécessaire de répliques, de règles de zone, de conteneurs, etc., comme dans la commande suivante :

   ```
   startOgServer.bat serverA0 -objectgridFile xml\objectgrid.xml
   -deploymentPolicyFile xml\deployment.xml -zone zoneA
   ```

   8.6+

   ```
   startXsServer.bat serverA0 -objectgridFile xml\objectgrid.xml
   -deploymentPolicyFile xml\deployment.xml -zone zoneA
   ```

3. Vous pouvez arrêter les processus de conteneur pour simuler une défaillance dans la grille de données :

   ```
   stopOgServer.bat serverA0,serverA1,serverB0 -catalogServiceEndPoints
   localhost:2809
   ```

   8.6+

   ```
   stopXsServer.bat serverA0,serverA1,serverB0 -catalogServiceEndPoints
   localhost:2809
   ```

   .

   Si le serveur qui contient le dernier fragment d'une partition est arrêté, eXtreme Scale affecte un nouveau fragment primaire. Vous pouvez rechercher les pertes de données :

   - Le script **runclient** insère et lit l'élément dans votre grille de données.
   - La commande **xscmd -c showMapSizes** indique le nombre d'éléments dans la grille de données.

4. Affichez les serveurs de conteneur actifs avec la commande suivante :

   ```
   xscmd -c showPlacement -z zone_name
   ```

Configuration de serveurs de catalogue et de serveurs de conteneur

WebSphere eXtreme Scale comporte deux types de serveur : les serveurs de catalogue et les serveurs de conteneur. Les serveurs de catalogue contrôlent le positionnement des fragments et détectent et surveillent les serveurs de conteneur. Plusieurs serveurs de catalogue peuvent rejoindre un domaine de service de catalogue pour fournir une haute disponibilité à l'environnement. Un serveur de conteneur est une machine virtuelle Java (JVM) qui stocke les données d'application de la grille de données.

Configuration des domaines de serveur de catalogue et de service de catalogue

Le service de catalogue héberge une logique qui est généralement inactive lorsque l'état est stabilisé. Il a donc très peu d'incidence sur l'évolutivité. Il permet de gérer des centaines de serveurs de conteneur devenant disponibles simultanément. Pour une haute disponibilité, configurez le service de catalogue en un domaine de service de catalogue.

Avant de commencer

Une fois qu'un domaine de service de catalogue a démarré, les membres de la grille de données se lient. La topologie du domaine de service de catalogue doit faire l'objet d'une planification prudente et méticuleuse, car il sera impossible de modifier la configuration du domaine au moment de l'exécution. Etendez la grille de données de manière aussi diversifiée que possible pour éviter d'éventuelles erreurs.

La meilleure pratique pour éviter un point de défaillance unique pour votre domaine de service de catalogue consiste à démarrer un minimum de trois serveurs de catalogue sur trois noeuds différents.

Si vous utilisez seulement deux noeuds, configurez deux serveurs de catalogue sur chacun des deux noeuds pour un total de quatre processus serveur de catalogue. La création de cette configuration garantit que lorsqu'un seul des noeuds est démarré, les deux serveurs de catalogue nécessaires sont actifs. Vous devez démarrer au moins deux serveurs de catalogue en même temps. Lorsque les serveurs de catalogue démarrent, ils recherchent d'autres serveurs de catalogue dans la configuration et ne démarrent pas tant qu'un autre serveur de catalogue au moins n'est pas trouvé.

Procédure

- Configurez les serveurs de catalogue autonome et les domaines de services de catalogue.

  Vous pouvez configurer les serveurs de catalogue autonome et les domaines de services de catalogue avec des paramètres et des fichiers de propriétés que vous envoyez à la commande de démarrage du serveur ou à l'API de serveur embarqué.

  – «Exemple : Configuration des domaines de service de catalogue», à la page 305
  – «Démarrage et arrêt des serveurs sécurisés», à la page 479
  – Propriétés du serveur de catalogue
- Configurez les serveurs de catalogue et les domaines de services de catalogue dans WebSphere Application Server

Configurez les serveurs de catalogue qui s'exécutent dans WebSphere Application Server avec la console d'administration WebSphere Application Server, les tâches d'administration et les fichiers des propriétés de serveur. Le cycle de vie du serveur est contrôlé par le cycle de vie du processus dans WebSphere Application Server. Lorsque les processus démarrent ou s'arrêtent dans WebSphere Application Server, les serveurs de catalogue qui s'exécutent sur ces processus démarrent ou s'arrêtent également.

- «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307
- «Configuration du service de catalogue dans WebSphere Application Server», à la page 306

Exemple : Configuration des domaines de service de catalogue

Lorsque vous utilisez le service de catalogue, un minimum de deux serveurs de catalogue sont requis pour éviter un point de défaillance unique. Selon le nombre de noeuds dans votre environnement, vous pouvez créer des configurations différentes pour qu'au moins deux serveurs de catalogue soient toujours en cours d'exécution.

Exemple : Démarrage de quatre serveurs de catalogue sur deux noeuds dans un environnement autonome

Le script suivant démarre les serveurs de catalogue cs0 et cs1 sur le noeud hôte1 et démarre les serveurs de catalogue cs2 et cs3 sur le noeud host2.

```
./startOgServer.sh|bat cs0 -listenerPort 2809 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startOgServer.sh|bat cs1 -listenerPort 2810 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startOgServer.sh|bat cs2 -listenerPort 2809 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startOgServer.sh|bat cs3 -listenerPort 2810 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m
```

Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

8.6+

```
./startXsServer.sh|bat cs0 -listenerPort 2809 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startXsServer.sh|bat cs1 -listenerPort 2810 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m

./startXsServer.sh|bat cs2 -listenerPort 2809 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m
```

```
./startXsServer.sh|bat cs3 -listenerPort 2810 -catalogServiceEndPoints
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604
-quorum true -jvmArgs -Xmx256m
```

A faire : Vous devez utiliser l'option **-listenerPort** , car les serveurs de catalogue qui s'exécutent sur un noeud requièrent chacun un numéro de port unique.

Exemple : Démarrage de plusieurs serveurs de catalogue dans un environnement WebSphere Application Server

Les serveurs de catalogue démarrent automatiquement dans un environnement WebSphere Application Server. Vous pouvez définir plusieurs serveurs de catalogue afin de commencer en créant un domaine de service de catalogue. Une fois que vous avez indiqué plusieurs noeuds finals dans le domaine de service de catalogue, redémarrez les serveurs d'applications inclus afin que les serveurs de catalogue démarrent en parallèle.

- **WebSphere Application Server Network Deployment** : vous pouvez choisir plusieurs serveurs d'applications existants de la cellule pour les placer dans votre domaine de service de catalogue.
- **Base WebSphere Application Server** : vous pouvez démarrer le service de catalogue sur plusieurs noeuds autonomes. En définissant plusieurs profils dans la même image d'installation à l'aide de l'outil de gestion des profils, vous pouvez créer un ensemble de noeuds autonomes ayant chacun des ports uniques. Dans chaque serveur d'applications, définissez le domaine de service de catalogue. Vous pouvez spécifier n'importe quels autres serveurs d'applications en ajoutant des serveurs distants à la configuration. Après avoir créé cette configuration sur tous les serveurs autonomes, vous pouvez démarrer l'ensemble de serveurs d'applications de base en parallèle en exécutant le script **startServer** ou en utilisant un service Windows pour démarrer les serveurs.

Configuration de WebSphere eXtreme Scale avec WebSphere Application Server

Java

Vous pouvez exécuter les processus des services de catalogue et des serveurs de conteneur dans WebSphere Application Server. La procédure de configuration de ces serveurs est différente d'une configuration autonome. Je service de catalogue peut être automatiquement démarré dans des serveurs ou des gestionnaires de déploiement WebSphere Application Server. Le processus de conteneur démarre lorsqu'une application eXtreme Scale est déployée et démarrée dans l'environnement WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Avertissement : Ne placez pas les serveurs de conteneur avec les serveurs de catalogue dans un environnement de production. Incluez le service de catalogue dans plusieurs processus d'agents de noeud ou sur un serveur d'applications qui n'héberge pas d'application eXtreme Scale.

Configuration du service de catalogue dans WebSphere Application Server :

Java

Les processus de service de catalogue peuvent s'exécuter dans WebSphere Application Server. Le cycle de vie du serveur dans WebSphere Application Server détermine quand le service de catalogue démarre et s'arrête.

Procédure

1. Choisissez un ou plusieurs processus WebSphere Application Server pour l'extension avec le profil WebSphere eXtreme Scale. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 243. Si vous voulez que le service de catalogue démarre automatiquement dans WebSphere Application Server Network Deployment sur le gestionnaire de déploiement, installez WebSphere eXtreme Scale sur le noeud du gestionnaire de déploiement et étendez le profil du gestionnaire de déploiement.
2. Configurez les fichiers de propriétés du serveur pour les processus WebSphere Application Server et ajoutez-les au chemin d'accès aux classes du noeud. Pour plus d'informations, voir Fichier de propriétés du serveur.
3. Configurez un domaine de service de catalogue. Le domaine de service de catalogue est un groupe de serveurs de catalogue dans votre environnement. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server».
4. Démarrez les processus WebSphere Application Server qui hébergent les serveurs de catalogue. Pour plus d'informations, voir «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 508.

Création de domaines de service de catalogue dans WebSphere Application Server :

Java

Un domaine de services de catalogue définit un groupe de serveurs de catalogue qui gèrent le positionnement des fragments et qui surveillent l'état des serveurs conteneurs de la grille de données.

Avant de commencer

- Installer WebSphere eXtreme Scale sur WebSphere Application Server. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226 .

Pourquoi et quand exécuter cette tâche

En créant un domaine de service de catalogue, vous définissez une collection de serveurs de catalogue à haute disponibilité.

Ces serveurs de catalogue peuvent s'exécuter dans WebSphere Application Server au sein d'une simple cellule ou d'un groupe central. Le domaine de service de catalogue peut également définir un groupe distant de serveurs qui s'exécutent dans différents processus Java SE ou dans d'autres cellules WebSphere Application Server.

Pour les serveurs de catalogue exécutés sur les serveurs d'applications existants de la cellule : Lorsque vous définissez un domaine de service de catalogue qui place les serveurs de catalogue sur les serveurs d'applications dans la cellule, les mécanismes des groupes centraux de WebSphere Application Server sont utilisés. Le service de catalogue démarre automatiquement sur les serveurs d'applications de la cellule. Il en résulte que les membres d'un même domaine de service de catalogue ne peuvent donc pas s'étendre au-delà des limites d'un groupe central et qu'un domaine ne peut donc pas s'étendre sur plusieurs cellules. Toutefois, les serveurs de conteneur WebSphere eXtreme Scale et les clients peuvent s'étendre à plusieurs cellules en se connectant à un serveur de catalogue dans les limites de cellule (domaine de service de catalogue autonome ou intégré dans une autre cellule, par exemple).

Pour les serveurs de catalogue distants : vous pouvez connecter les conteneurs et les clients WebSphere eXtreme Scale à un domaine de service de catalogue exécuté dans une autres cellule WebSphere Application Server ou comme processus autonome. Comme les serveurs de catalogue configuré à distance ne démarrent pas automatiquement dans la cellule, vous devez démarrer manuellement les serveurs de catalogue configurés à distance. Lorsque vous configurez un domaine de services de catalogue distant, le nom de domaine doit correspondre au nom de domaine que vous avez défini lorsque vous démarrez les serveurs de catalogue distants. Le nom de domaine de services de catalogue par défaut des serveurs de catalogue autonome est `DefaultDomain`. Définissez un nom de domaine de services de catalogue avec la commande **startOgServer** ou **startXsServer** et le paramètre **-domain**, un fichier de propriétés de serveur ou avec l'API de serveur embarqué. Vous devez démarrer chaque processus de serveur de catalogue distant dans le domaine distant avec le même nom de domaine. Voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494 pour plus d'informations sur le démarrage des serveurs de catalogue.

Avertissement : En environnement de production, ne faites pas cohabiter les services de catalogue avec des serveurs de conteneur WebSphere eXtreme Scale. Incluez le service de catalogue dans plusieurs processus d'agent de noeud ou dans sur un serveur d'applications qui n'héberge pas d'application WebSphere eXtreme Scale.

Procédure

1. Créez le domaine de service de catalogue.
   a. Dans la console d'administration de WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de service de catalogue** > **Nouveau**.
   b. Définissez un nom, une valeur par défaut et des justificatifs d'identification pour l'authentification JMX de votre domaine. Si vous configurez des noeuds finals distants pour le domaine de service de catalogue, le nom de ce dernier doit correspondre au nom du domaine de service de catalogue que vous définissez lorsque vous démarrez les serveurs de catalogue.
   c. Ajoutez des points de contact de serveurs de catalogue. Vous pouvez sélectionner des serveurs d'applications existants ou ajouter des serveurs distants qui exécutent un service de catalogue.
2. Testez la connexion aux serveurs de catalogue dans le domaine de service de catalogue. Pour les serveurs d'applications existants, les serveurs de catalogue démarrent lorsque le serveur d'applications associé est démarré. Pour les serveurs d'applications distants, vous devez démarrer les serveurs manuellement en utilisant la commande **startOgServer** ou **startXsServer** ou l'API de serveur embarqué.
   a. Dans la console d'administration de WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de service de catalogue**.
   b. Sélectionnez le domaine que vous voulez tester et cliquez sur **Tester la connexion**. Lorsque vous cliquez sur ce bouton, tous les points de contact des domaines de service de catalogue définis sont interrogés l'un après l'autre (s'il existe des points de contact) et la procédure retourne un message indiquant que la connexion au domaine a réussi.

Tâches d'administration des domaines de service de catalogue : Java

Les langages de script Jacl ou Jython permettent de gérer les domaines de service de catalogue présents dans votre configuration WebSphere Application Server.

Conditions requises

WebSphere eXtreme Scale Client doit être installé dans votre environnement WebSphere Application Server.

Afficher la liste de toutes les tâches d'administration

Pour obtenir la liste de toutes les tâches d'administration associées aux domaines de services de catalogue, exécutez la commande suivante avec **wsadmin**:

```
wsadmin>$AdminTask help XSDomainManagement
```

Commandes

Les tâches d'administration de domaines de service de catalogue comprennent les commandes suivantes :

- «createXSDomain»
- «deleteXSDomain», à la page 313
- «getDefaultXSDomain», à la page 313
- «listXSDomains», à la page 314
- «modifyXSDomain», à la page 314
- 8.6+ «getTransport», à la page 320
- «testXSDomainConnection», à la page 321
- «testXSServerConnection», à la page 321

createXSDomain

La commande **createXSDomain** enregistre un nouveau domaine de service de catalogue.

Tableau 17. Arguments de la commande createXSDomain

Argument	Description
-name (requis)	Spécifie le nom du domaine de service de catalogue à créer.
-default	Indique si le domaine de services de catalogue est le domaine par défaut de la cellule. La valeur par défaut est `true`. (booléen : a soit la valeur `true`, soit la valeur `false`).
-properties	Spécifie les propriétés personnalisées du domaine de service de catalogue.

Tableau 17. Arguments de la commande createXSDomain (suite)

Argument	Description
8.6+ **-enableXIO**	8.6+ Indique si IBM eXtreme IO (XIO) ou ORB (Object Request Broker) est utilisé pour la communication de transport dans ce domaine de services de catalogue. **true** Indique que XIO est utilisé. **false** Indique qu'ORB est utilisé. Si vous n'indiquez pas de valeur, la valeur par défaut est `true` (XIO activé). Si le domaine de services de catalogue contient des serveurs distants, le paramètre **-enableXIO** ne configure pas XIO ni ORB sur les serveurs distants. Pour configurer le transport sur les serveurs distants, définissez le type de transport lorsque vous les démarrez.

Tableau 18. Arguments de la procédure defineDomainServers

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact du service de catalogue. • **Pour les serveurs d'applications existants** : le nom du noeud final doit avoir le format *cell_name**node_name**server_name* • **Pour les serveurs distantes** : définit le nom d'hôte du serveur distant. Vous pouvez utiliser le même nom pour plusieurs noeuds finals, mais les valeurs de port client doivent être uniques pour chaque noeud final.
custom_properties	Spécifie les propriétés personnalisées du point de contact du domaine de service de catalogue. Si vous ne disposez pas de propriétés personnalisées, utilisez des guillemets doubles ("") pour cet argument.

Tableau 18. Arguments de la procédure defineDomainServers (suite)

Argument	Description
endpoint_ports	Spécifie les numéros de port du point de contact du domaine de service de catalogue. Les ports doivent être définis dans l'ordre suivant : `<client_port>,<listener_port>` **Port client** Spécifie le port utilisé pour la communication entre les serveurs de catalogue dans le domaine de service de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent uniquement dans des processus WebSphere Application Server et elle peut correspondre à n'importe quel port inutilisé autre part. **Port d'écoute** Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finals distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue. **Pour les noeuds finals distants WebSphere eXtreme Scale** : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue via l'ORB. Pour les noeuds finals WebSphere Application Server, la valeur de port d'écoute est facultative, car elle est héritée de la configuration de port BOOTSTRAP_ADDRESS.

Tableau 19. Arguments de la procédure configureClientSecurity

Argument	Description
-securityEnabled	Spécifie que la sécurité du client est activée pour le serveur de catalogue. Le fichier des propriétés du serveur qui est associé au serveur de catalogue sélectionné doit avoir un paramètre **securityEnabled** correspondant dans le fichier des propriétés du serveur. Si ces paramètres ne correspondent pas, une exception est générée. (booléen : a soit la valeur `true`, soit la valeur `false`).

Tableau 19. Arguments de la procédure configureClientSecurity (suite)

Argument	Description
-credentialAuthentication (facultatif)	Indique si l'authentification des données d'identification est imposée ou prise en charge. **Jamais** Aucune authentification de certificat client n'est imposée. **Requis** L'authentification des données d'identification est toujours appliquée. Si le serveur ne prend pas en charge l'authentification des données d'identification, le client ne peut pas se connecter au serveur. **Pris en charge** (Par défaut) L'authentification des données d'identification est imposée seulement si à la fois le client et le serveur prennent en charge l'authentification des données d'identification.
-authenticationRetryCount (facultatif)	Spécifie le nombre de tentatives d'authentification si les données d'identification sont arrivées à expiration. Si vous ne voulez pas réessayer l'authentification, définissez la valeur à `0`. La valeur par défaut est `0`.
-credentialGeneratorClass	Indique la classe d'implémentation de com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator pour que le client récupère les jetons de sécurité de l'unité d'exécution.
-credentialGeneratorProps	Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés sont envoyées à l'objet avec la méthode setProperties(String). La valeur des propriétés du générateur de données d'identification est utilisée seulement si une valeur est spécifiée pour la zone **Classe du générateur de données d'identification**.

Valeur retournée :

Exemples de mode de traitement par lots

Le mode de traitement par lots impose de formater correctement l'entrée de commande. Utilisez le mode interactif pour que les valeurs que vous entrez soient correctement traitées. Lorsque vous utilisez le mode de traitement par lots, vous devez définir les arguments d'étape **-defineDomainServers** en utilisant un tableau de propriétés spécifiques. Ce tableau a le format *name_of_endpoint custom_properties endpoint_ports*. La valeur *endpoint_ports* est la liste des ports qui doivent être définis dans l'ordre suivant : *<client_port>,<listener_port>*.

- Créez un domaine de service de catalogue de noeuds finals distants en utilisant Jacl :

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
{{xhost1.ibm.com "" ,2809}} -configureClientSecurity {-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"}}}
```

- Créez un domaine de service de catalogue de noeuds finals distants en utilisant la chaîne Jython :

```
AdminTask.createXSDomain('[-name TestDomain -default true
-defineDomainServers [[xhost1.ibm.com "" ,2809]
[xhost2.ibm.com "" ,2809]] -configureClientSecurity [-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"] ]')
```

- Créez un domaine de service de catalogue de noeuds finals de serveur d'applications existants en utilisant Jacl :

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
 {{cellName/nodeName/serverName "" 1109}}}
```

Exemples de mode interactif

- Jacl :

```
$AdminTask createXSDomain {-interactive}
```

- Chaîne Jython :

```
AdminTask.createXSDomain ('[-interactive]')
```

deleteXSDomain

La commande **deleteXSDomain** supprime un domaine de service de catalogue.

Paramètres requis :

-name
Spécifie le nom du domaine de service de catalogue à supprimer.

Valeur retournée :

Exemples de mode de traitement par lots

- A l'aide de Jacl :

```
$AdminTask deleteXSDomain {-name TestDomain }
```

- Avec une chaîne Jython :

```
AdminTask.deleteXSDomain('[-name TestDomain ]')
```

Exemples de mode interactif

- A l'aide de Jacl :

```
$AdminTask deleteXSDomain {-interactive}
```

- Avec une chaîne Jython :

```
AdminTask.deleteXSDomain ('[-interactive]')
```

getDefaultXSDomain

La commande **getDefaultXSDomain** renvoie le domaine de service de catalogue par défaut de la cellule.

Paramètres requis : aucun.

Valeur de retour : nom du domaine de service de catalogue.

Exemples de mode de traitement par lots

- A l'aide de Jacl :

```
$AdminTask getDefaultXSDomain
```

- Avec une chaîne Jython :

```
AdminTask.getDefaultXSDomain
```

Exemples de mode interactif

- A l'aide de Jacl :

```
$AdminTask getDefaultXSDomain {-interactive}
```

- Avec une chaîne Jython :

```
AdminTask.getDefaultXSDomain ('[-interactive]')
```

listXSDomains

La commande **listXSDomains** retourne la liste des domaines de services de catalogue existants.

Paramètres requis : aucun.

Valeur de retour : liste de tous les domaines de services de catalogue dans la cellule.

Exemples de mode de traitement par lots

- A l'aide de Jacl :

```
$AdminTask listXSDomains
```

- Avec une chaîne Jython :

```
AdminTask.listXSDomains
```

Exemples de mode interactif

- A l'aide de Jacl :

```
$AdminTask listXSDomains {-interactive}
```

- Avec une chaîne Jython :

```
AdminTask.listXSDomains ('[-interactive]')
```

modifyXSDomain

La commande **modifyXSDomain** modifie un domaine de service de catalogue existant.

Le mode de traitement par lots impose de formater correctement l'entrée de commande. Utilisez le mode interactif pour que les valeurs que vous entrez soient correctement traitées. Lorsque vous utilisez le mode de traitement par lots, vous devez définir les arguments d'étape **-modifyEndpoints**, **-addEndpoints** et **-removeEndpoints** en utilisant un tableau de propriétés spécifiques. Ce tableau a le format *name_of_endpoint host_name custom_properties endpoint_ports*. La valeur *endpoint_ports* est la liste des ports qui doivent être définis dans l'ordre suivant : <*client_port*>,<*listener_port*>.

Tableau 20. Arguments de la commande modifyXSDomain

Argument	Description
-name (requis)	Spécifie le nom du domaine de service de catalogue que vous souhaitez éditer.
-default	Avec la valeur `true`, spécifie que le domaine de service de catalogue est le domaine par défaut de la cellule (booléen).
-properties	Spécifie les propriétés personnalisées du domaine de service de catalogue.

Tableau 20. Arguments de la commande modifyXSDomain (suite)

Argument	Description
8.6+ **-enableXIO**	8.6+ Indique si IBM eXtreme IO (XIO) ou ORB (Object Request Broker) est utilisé pour la communication de transport dans ce domaine de service de catalogue. **true** Indique que XIO est utilisé. **false** Indique qu'ORB est utilisé. Si vous n'indiquez pas de valeur, la valeur par défaut est `true` (XIO activé). Si le domaine de services de catalogue contient des serveurs distants, vous ne pouvez pas configurer XIO sur les serveurs distants.

Tableau 21. Arguments de la procédure modifyEndpoints

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact du service de catalogue. • **Pour les serveurs d'applications existants** : le nom du noeud final doit avoir le format *cell_name**node_name**server_name* • **Pour les serveurs distantes** : définit le nom d'hôte du serveur distant. Vous pouvez utiliser le même nom pour plusieurs noeuds finals, mais les valeurs de port client doivent être uniques pour chaque noeud final.

Tableau 21. Arguments de la procédure modifyEndpoints (suite)

Argument	Description
endpoint_ports	Spécifie les numéros de port du point de contact du domaine de service de catalogue. Les noeuds finals doivent être définis dans l'ordre suivant : `<client_port>,<listener_port>` **Port client** Spécifie le port utilisé pour la communication entre les serveurs de catalogue dans le domaine de service de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent uniquement dans des processus WebSphere Application Server et elle peut correspondre à n'importe quel port inutilisé autre part. **Port d'écoute** Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finals distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue. **Pour les noeuds finaux distants For WebSphere eXtreme Scale** : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue. Pour les noeuds finaux WebSphere Application Server, en spécifiant que la valeur de port d'écoute est facultative. La valeur dépend du type de transport que vous utilisez. Si vous utilisez ORB, la valeur est héritée de la configuration de port BOOTSTRAP_ADDRESS. Si vous utilisez extremeIO, la valeur est héritée de la configuration de port XIO_ADDRESS.

Tableau 22. Arguments de la procédure addEndpoints

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact du service de catalogue. • **Pour les serveurs d'applications existants** : le nom du noeud final doit avoir le format *cell_name**node_name**server_name* • **Pour les serveurs distantes** : définit le nom d'hôte du serveur distant. Vous pouvez utiliser le même nom pour plusieurs noeuds finals, mais les valeurs de port client doivent être uniques pour chaque noeud final.
custom_properties	Spécifie les propriétés personnalisées du point de contact du domaine de service de catalogue. Si vous ne disposez pas de propriétés personnalisées, utilisez des guillemets doubles ("") pour cet argument.

Tableau 22. Arguments de la procédure addEndpoints (suite)

Argument	Description
endpoint_ports	Spécifie les numéros de port du point de contact du domaine de service de catalogue. Les noeuds finals doivent être définis dans l'ordre suivant : `<client_port>,<listener_port>` **Port client** Spécifie le port utilisé pour la communication entre les serveurs de catalogue dans le domaine de service de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent uniquement dans des processus WebSphere Application Server et elle peut correspondre à n'importe quel port inutilisé autre part. **Port d'écoute** Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finals distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue. **Pour les noeuds finals distants WebSphere eXtreme Scale** : définit le port d'écoute ORB (Object Request Broker) qui permet aux conteneurs et aux clients de communiquer avec le service de catalogue via l'ORB. Pour les noeuds finals WebSphere Application Server, la valeur de port d'écoute est facultative, car elle est héritée de la configuration de port BOOTSTRAP_ADDRESS.

Tableau 23. Arguments de la procédure removeEndpoints

Argument	Description
name_of_endpoint	Spécifie le nom du point de contact de domaine de service de catalogue à supprimer.

Tableau 24. Arguments de la procédure configureClientSecurity

Argument	Description
-securityEnabled	Spécifie que la sécurité du client est activée pour le serveur de catalogue. Le fichier des propriétés du serveur qui est associé au serveur de catalogue sélectionné doit avoir un paramètre **securityEnabled** correspondant dans le fichier des propriétés du serveur. Si ces paramètres ne correspondent pas, une exception est générée. (booléen : a soit la valeur `true`, soit la valeur `false`).
-credentialAuthentication (facultatif)	Indique si l'authentification des données d'identification est appliquée ou prise en charge. **Jamais** Aucune authentification de certificat client n'est imposée. **Requis** L'authentification des données d'identification est toujours appliquée. Si le serveur ne prend pas en charge l'authentification des données d'identification, le client ne peut pas se connecter au serveur. **Pris en charge** (Par défaut) L'authentification des données d'identification est imposée seulement si à la fois le client et le serveur prennent en charge l'authentification des données d'identification.
-authenticationRetryCount (facultatif)	Spécifie le nombre de tentatives d'authentification si les données d'identification sont arrivées à expiration. Si vous ne voulez pas réessayer l'authentification, définissez la valeur à `0`. La valeur par défaut est `0`.
-credentialGeneratorClass	Indique la classe d'implémentation de com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator pour que le client récupère les jetons de sécurité de l'unité d'exécution.
-credentialGeneratorProps	Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés sont envoyées à l'objet avec la méthode setProperties(String). La valeur des propriétés du générateur de données d'identification est utilisée seulement si une valeur est spécifiée pour la zone **Classe du générateur de données d'identification**.

Valeur retournée :

Exemples de mode de traitement par lots

- A l'aide de Jacl :

```
$AdminTask modifyXSDomain {-name TestDomain -default true -modifyEndpoints
{{xhost1.ibm.com "" ,2809}} -addEndpoints {{xhost2.ibm.com "" ,2809}}}
-removeEndpoints {{xhost3.ibm.com}}}
```

- Avec une chaîne Jython :

```
AdminTask.modifyXSDomain('[-name TestDomain
-default false -modifyEndpoints [[xhost1.ibm.com "" ,2809]]
-addEndpoints [[xhost3.ibm.com "" ,2809]]
-removeEndpoints [[xhost2.ibm.com]]]')
```

- Utilisation de la sécurité du client avec la commande modify :

```
$AdminTask modifyXSDomain {-name monDomaine -default false
-configureClientSecurity {-securityEnabled true -
Supported -authenticationRetryCount 1 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
 -credentialGeneratorProps "manager manager1"}}
```

- 8.6+ Modifiez un domaine de services de catalogue existant pour activer IBM eXtremeIO :

```
AdminTask.modifyXSDomain('[-name testDomain -enableXIO true]')
```

Exemples de mode interactif

- A l'aide de Jacl :

```
$AdminTask modifyXSDomain {-interactive}
```

- Avec une chaîne Jython :

```
AdminTask.modifyXSDomain ('[-interactive]')
```

8.6+ getTransport

La commande **getTransport** affiche le type de transport pour le domaine de services de catalogue : IBM eXtremeIO (XIO) ou ORB (Object Request Broker). Si vous exécutez cette commande sur un domaine de services de catalogue qui contient des serveurs distants ou que catalogServerName est un serveur distant, une erreur se produit. Vous devez utiliser la commande **xscmd -c showTransport** pour les serveurs distants.

Paramètres requis :

-domainName
Spécifie le nom du domaine de service de catalogue pour lequel vous voulez afficher le type de transport.

-catalogServerName
Spécifie le nom du serveur de catalogue pour lequel vous voulez afficher le type de transport.

Valeur de retour : ORB ou XIO

Affichage du type de transport d'un domaine de service de catalogue

- A l'aide de Jacl :

```
$AdminTask getTransport {-domainName TestDomain }
```

- Avec une chaîne Jython :

```
AdminTask.getTransport('[-domainName testDomain]')
```

Affichage du transport d'un serveur de catalogue

-
- A l'aide de Jacl :

```
$AdminTask getTransport {-catalogServerName myCell01\myNode01\container1 }
```

- Avec une chaîne Jython :

```
AdminTask.getTransport('[-catalogServerName myCell01\myNode01\container1]')
```

Exemples de mode interactif

- A l'aide de Jacl :

```
$AdminTask getTransport {-interactive}
```

- Avec une chaîne Jython :

```
AdminTask.getTransport ('[-interactive]')
```

testXSDomainConnection

La commande **testXSDomainConnection** teste la connexion à un domaine de service de catalogue.

Paramètres requis :

-name
Spécifie le nom du domaine de service de catalogue vers lequel tester la connexion.

Paramètres facultatifs

-timeout
Spécifie le délai d'attente maximal, en secondes, de la connexion.

Valeur de retour : si la connexion peut être établie, retourne `true`. Dans le cas contraire, des informations d'erreur de connexion sont retournées.

Exemples de mode de traitement par lots

- A l'aide de Jacl :
  ```
  $Admintask testXSDomainConnection
  ```
- Avec une chaîne Jython :
  ```
  AdminTask.testXSDomainConnection
  ```

Exemples de mode interactif

- A l'aide de Jacl :
  ```
  $AdminTask testXSDomainConnection {-interactive}
  ```
- Avec une chaîne Jython :
  ```
  AdminTask.testXSDomainConnection ('[-interactive]')
  ```

testXSServerConnection

La commande **testXSServerConnection** teste la connexion à un serveur de catalogue. Cette commande fonctionne pour les serveurs autonomes et les serveurs qui font partie d'un domaine de service de catalogue.

Paramètres requis :

host
Spécifie l'hôte sur lequel réside le serveur de catalogue.

listenerPort
Spécifie le port d'écoute du serveur de catalogue.

Paramètres facultatifs

timeout
Spécifie en secondes pendant combien de temps au maximum attendre la connexion au serveur de catalogue.

domain
Spécifie le nom du domaine de service de catalogue. Si vous définissez une valeur pour ce paramètre, les propriétés de la sécurité du client pour le domaine de service de catalogue spécifié sont utilisées pour tester la

connexion. Sinon, une recherche est effectuée pour trouver le domaine de service de catalogue pour l'hôte et le port d'écoute spécifié. Si un domaine de service de catalogue est trouvé, les propriétés de la sécurité du client qui sont définies pour le domaine de service de catalogue sont utilisées pour tester le serveur. Sinon, aucune propriété de la sécurité du client n'est utilisée lors du test.

Valeur retournée :

Exemples de mode de traitement par lots

- A l'aide de Jacl :

```
$Admintask testXSServerConnection {-host xhost1.ibm.com -listenerPort 2809}
```

- Avec une chaîne Jython :

```
AdminTask.testXSServerConnection('[-host xshost3.ibm.com -listenerPort 2809]')
```

Exemples de mode interactif

- A l'aide de Jacl :

```
$AdminTask testXSServerConnection {-interactive}
```

- Avec une chaîne Jython :

```
AdminTask.testXSServerConnection ('[-interactive]')
```

Collection de domaines de service de catalogue :

Cette page permet de gérer les domaines de service de catalogue. Les domaines de service de catalogue définissent un groupe de serveurs de catalogue qui gèrent le positionnement de fragments et surveille l'état des serveurs de conteneur dans la grille de données.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaine de service de catalogue**. Pour créer un domaine de service de catalogue, cliquez sur **Nouveau**. Pour supprimer un domaine de service de catalogue, sélectionnez le domaine à supprimer et cliquez sur **Supprimer**.

Tester la connexion :

Lorsque vous cliquez sur le bouton de **test de la connexion**, tous les noeuds finals du domaine de service de catalogue défini sont interrogés un par un. Si l'un d'entre eux est disponible, il retourne un message qui indique que la connexion au domaine de service de catalogue a abouti. Vous pouvez utiliser ce bouton pour tester la configuration des informations de connexion et de sécurité.

Définir la valeur par défaut :

Indique le domaine de service de catalogue utilisé comme valeur par défaut. Sélectionnez un domaine de service de catalogue comme valeur par défaut et cliquez sur **Définir la valeur par défaut**. Un seul domaine de serveur de catalogue peut être sélectionné comme valeur par défaut.

Nom :

Indique le nom du domaine de service de catalogue.

Valeur par défaut :

Indique le domaine de service de catalogue de la liste qui est la valeur par défaut.

Le domaine de service de catalogue par défaut est indiqué par l'icône :  .

Paramètres du domaine de service de catalogue :

Cette page permet de gérer les paramètres d'un domaine de service de catalogue spécifique. Les domaines de service de catalogue définissent un groupe de serveurs de catalogue qui gèrent le positionnement de fragments et surveille l'état des serveurs de conteneur dans la grille de données. Vous pouvez définir un domaine de service de catalogue figurant dans la même cellule que le gestionnaire de déploiement. Vous pouvez également définir des domaines de service de catalogue distants si votre configuration WebSphere eXtreme Scale se trouve dans une cellule différente ou que votre grille de données est composée de processus Java SE.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de service de catalogue** > ***catalog_service_domain_name***.

Tester la connexion :

Lorsque vous cliquez sur le bouton de **test de la connexion**, tous les noeuds finals du domaine de service de catalogue défini sont interrogés un par un. Si l'un d'entre eux est disponible, il retourne un message qui indique que la connexion au domaine de service de catalogue a abouti. Vous pouvez utiliser ce bouton pour tester la configuration des informations de connexion et de sécurité.

Nom :

Indique le nom du domaine de service de catalogue.

Activer ce domaine de service de catalogue en tant que valeur par défaut sauf si un autre domaine de service de catalogue est explicitement indiqué :

Si vous cochez cette case, le domaine de service de catalogue sélectionné devient le domaine de service de catalogue par défaut de la cellule. Chaque profil de serveur dans la cellule qui est étendu avec le profil WebSphere eXtreme Scale appartient au domaine du service de catalogue sélectionné.

Pour WebSphere eXtreme Scale, tous les conteneurs eXtreme Scale embarqués dans les modules d'application Java EE se connectent au domaine par défaut. Les clients peuvent se connecter au domaine par défaut en utilisant l'API ServerFactory.getServerProperties().getCatalogServiceBootstrap() pour extraire les noeuds finals de service de catalogue à utiliser lors de l'appel de l'API ObjectGridManager.connect().

Si vous changez le domaine par défaut pour pointer vers un groupe de serveurs de catalogue différents, tous les conteneurs et clients font référence au nouveau domaine après leur redémarrage.

Activer les communications IBM eXtremeIO (XIO) : **8.6+**

Indique si le domaine de service de catalogue utilise les communications XIO. Si vous ne sélectionnez pas cette option, ORB (Object Request Broker) est utilisé.

Remarque : 8.6+ Vous ne pouvez pas activer les communications XIO sur les serveurs distants depuis la console d'administration WebSphere Application Server. Activez XIO sur les serveurs distants lorsque vous démarrez les serveurs avec le script **startXsServer**.

Serveurs de catalogue :

Indique une liste de serveurs de catalogue appartenant à ce domaine de service de catalogue.

Cliquez sur **Nouveau** pour ajouter un serveur de catalogue à la liste. Ce serveur de catalogue doit déjà être défini dans la configuration eXtreme Scale. Vous pouvez également modifier ou supprimer un serveur dans la liste en sélectionnant le noeud final, puis en cliquant sur **Editer** ou **Supprimer**. Définissez les propriétés suivantes pour chaque noeud final du serveur de catalogues :

Noeud final du serveur de catalogues
Indique le nom du serveur d'applications ou du serveur distant existant sur lequel le service de catalogues s'exécute. Un domaine de service de catalogue ne doit pas contenir une combinaison de serveurs d'applications et de serveurs distants existants.

- **Serveur d'applications existant** : définir le chemin d'un serveur d'applications, d'un agent de noeud ou de gestionnaire de déploiement dans la cellule. Un service de catalogue démarre automatiquement dans le serveur sélectionné. Sélectionnez un serveur dans la liste des serveurs d'applications existants. Tous les serveurs d'applications que vous définissez dans le domaine de service de catalogue doivent se trouver dans le même groupe central.
- **Serveur distant** : définit le nom d'hôte du serveur de catalogue distant.

  Pour les noeuds finals distants WebSphere eXtreme Scale : définit le nom d'hôte du processus serveur de catalogue distant. Vous devez démarrer les serveurs distants avec le script **startOgServer** ou **startXsServer** ou l'API de serveur embarqué.

  Remarque : 8.6+ Vous ne pouvez pas activer les communications XIO sur les serveurs distants depuis la console d'administration WebSphere Application Server. Activez XIO sur les serveurs distants lorsque vous démarrez les serveurs avec le script **startXsServer**.

Port client
Spécifie le port utilisé pour la communication entre les serveurs de catalogue dans le domaine de service de catalogue. Cette valeur est nécessaire pour les serveurs de catalogue qui s'exécutent dans des processus WebSphere Application Server. Vous pouvez définir n'importe quel port comme valeur s'il n'est pas utilisé par un autre processus.

Port d'écoute
Indique le port utilisé pour établir des communications avec les clients. Cette valeur est obligatoire pour les noeuds finals distants et elle doit correspondre à la valeur utilisée au démarrage du service de catalogue. Le port d'écoute est utilisé par les clients et les conteneurs pour communiquer avec le service de catalogue.

Pour les noeud finaux distants WebSphere eXtreme Scale : 8.6+

- Si vous utilisez le transport ORB, la valeur **BOOTSTRAP_ADDRESS** de chaque serveur d'applications WebSphere Application Server est utilisée.

- Si vous utilisez le transport IBM eXtremeIO, la valeur **XIO_ADDRESS** est utilisée.

Statut

Tableau 25. Etat de noeud final de serveur de catalogues

Icône	Définition
	Inconnu
	Démarré
	Arrêté

Propriétés de sécurité du client :

Utilisez cette page pour définir la sécurité client d'un domaine de service de catalogue. Ces paramètres s'appliquent à tous les serveurs dans votre domaine de service de catalogue. Ces propriétés peuvent être remplacées en définissant un fichier `splicer.properties` avec la propriété personnalisée com.ibm.websphere.xs.sessionFilterProps ou en raccordant le fichier EAR d'application.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de service de catalogue** > ***catalog_service_domain_name*** > **Propriétés de sécurité du client**.

Activer la sécurité du client :

Indique que la sécurité du client est activée pour le serveur de catalogue. Le fichier des propriétés du serveur qui est associé au serveur de catalogue sélectionné doit avoir un paramètre **securityEnabled** correspondant dans le fichier des propriétés du serveur. Si ces paramètres ne correspondent pas, une exception est générée.

Authentification des données d'identification :

Indique si l'authentification des données d'identification est appliquée ou prise en charge.

Jamais

Aucune authentification n'est appliquée.

Requis

L'authentification des données d'identification est toujours appliquée. Si le serveur ne prend pas en charge l'authentification des données d'identification, le client ne peut pas se connecter au serveur.

Pris en charge

L'authentification des données d'identification est appliquée uniquement si le client et le serveur la prennent en charge.

Nombre de tentatives d'authentification :

Indique le nom de nouvelles tentatives d'authentification si les données d'identification ont expiré.

Si vous ne voulez pas tente d'authentifier de nouveau les données, définissez la valeur `0`.

Classe de générateur de données d'identification :

Indique la classe d'implémentation com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator pour que le client puisse extraire les données d'identification depuis l'objet CredentialGenerator.

Vous pouvez choisir depuis deux classes de générateurs d'identification de données prédéfinis ou vous pouvez définir un générateur personnalisé. Si vous choisissez un générateur personnalisé, vous devez indiquer le nom de la classe du générateur de données d'identification.

- com.ibm.websphere.objectgrid.security.plugins.builtins. UserPasswordCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- Générateur de données d'identification personnalisé

Type de sujet :

Indique si vous utilisez l'appelant J2EE ou le type de sujet J2EE runAs. Vous devez définir cette valeur lorsque vous choisissez le générateur de données d'identification WSTokenCredentialGenerator.

- **runAs** : le sujet contient le principal de l'exécution J2EE comme identité et l'exécution de J2EE en tant que données d'identification.
- **caller** : le sujet contient le principal de l'appelant J2EE et ses données d'identification.

ID utilisateur :

Définissez un ID utilisateur lorsque vous utilisez l'implémentation de générateur de donnée d'identification UserPasswordCredentialGenerator.

Mot de passe :

Définissez un mot de passe lorsque vous utilisez l'implémentation de générateur de donnée d'identification UserPasswordCredentialGenerator.

Propriétés du générateur de données d'identification :

Définissez les propriétés d'une classe d'implémentation CredentialGenerator personnalisée. Les propriétés sont définies dans l'objet avec la méthode setProperties(String). La valeur des propriétés du générateur de données d'identification est utilisée seulement si une valeur est spécifiée pour la zone **Classe du générateur de données d'identification**.

Propriétés personnalisées du domaine de service de catalogue :

Vous pouvez continuer à modifier la configuration du domaine de service de catalogue en définissant les propriétés personnalisées.

Pour afficher cette page de la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de service de catalogue** > **Propriétés personnalisées**. Pour créer une propriété personnalisée, cliquez sur **Nouveau**.

Nom :

Indique le nom de la propriété personnalisée du domaine de service de catalogue.

Valeur :

Indique une valeur pour la propriété personnalisée du domaine de service de catalogue.

Configuration du mécanisme de quorum

Le mécanisme de quorum est configuré pour chaque service de catalogue. Vous devez activer le mécanisme de quorum sur tous les serveurs de catalogue dans le domaine de service de catalogue.

Avant de commencer

Avant d'activer le mécanisme du quorum, vous devez configurer une topologie qui prend en charge ce type de configuration. La configuration doit prendre en charge :

- **Espace d'adresse IP non hiérarchique :** un élément adressable sur le réseau doit être capable de se connecter sans entrave à n'importe quel autre élément adressable du réseau. Vous devez utiliser un espace de nom d'adresse IP non hiérarchique. Tous les pare-feu dans la configuration doivent laisser la totalité du trafic circuler entre les adresses IP et les ports qui sont utilisés pour héberger les serveurs de catalogue et les serveurs de conteneur.
- **Nombre de serveurs de catalogue :** vus devez démarrer au moins un serveur de catalogue pour chaque centre de données dans configuration.
- **Paramètres d'intervalle des pulsations :** si vous ne définissez pas la fréquence, la valeur par défaut est 30 secondes. WebSphere eXtreme Scale vérifie sur les machines virtuelles Java dans une zone unique à la fréquence définie. Par exemple, si une pulsation sur un serveur de conteneur est manquée et qu'un quorum est établi, un événement de basculement se produit pour placer un nouveau serveur de conteneur. Pour plus d'informations, voir «Optimisation de la valeur de l'intervalle des pulsations pour la détection des basculements», à la page 329.
- **Sécurité du transport :** comme les centres de données sont normalement déployés dans des sites géographiquement dispersés, vous pouvez activer la sécurité du transport entre ces centres de données pour des raisons de sécurité. Consultez les informations relatives à la sécurité de la couche de transport dans le document *Guide d'administration*.

Pourquoi et quand exécuter cette tâche

Les quorums sont désactivés par défaut. Activez le mécanisme de quorum dans les cas suivants :

- Lorsque le domaine du service de catalogue couvre un réseau imprévisible et instable. Ce type de réseau peut couvrir plusieurs centres de données.
- Lorsque vous souhaitez empêcher la grille de données d'effectuer les corrections automatiquement au cours d'une microcoupure sur le réseau et suspendre temporairement à la place ses opérations.

Vous pouvez laisser le mécanisme du quorum désactivé si votre domaine de service de catalogue est contenu dans un centre de données unique, ou se trouve sur un réseau local (LAN). Dans ce type de configuration, la pulsation par défaut est utilisée et les microcoupures sont supposées être inférieures à 10 secondes. Comme le délai de détection est d'environ 30 seconds, les microcupures qui se produisent ne modifient pas le placement dans la grille de données.

Si vous activez le quorum, tous les serveurs de catalogue doivent être disponible et communiquer avec la grille de données pour mener des opérations de placement. Si une microcoupure réseau se produit, le placement est suspendu jusqu'à ce que tous les serveurs de catalogue soient disponibles. Si une défaillance de centre de données se produit, des actions manuelles sont requises pour supprimer le serveur de catalogue défaillant du quorum.

Procédure

1. **Activez le quorum sur les serveurs de catalogue.** Dans WebSphere Application Server, vous devez configurer le quorum avec le fichier de propriétés du serveur. Dans un environnement autonome, vous pouvez utiliser la méthode des propriétés ou activez le quorum lorsque vous démarrez le serveur :
   - **Définissez la propriété `enableQuorum=true` dans le fichier de propriétés du serveur.**

     Vous pouvez utiliser cette configuration dans un environnement autonome ou WebSphere Application Server.

```
catalogClusterEndPoints=cat0:cat0.domain.com:6600:6601,
cat1:cat1.domain.com:6600:6601
catalogServiceEndPoints= cat0.domain.com:2809, cat1.domain.com:2809
enableQuorum=true
```

Figure 34. `objectGridServer.properties` file

     Pour plus d'informations sur la configuration du fichier de propriétés, voir Fichier de propriétés du serveur.
   - **Transmettez l'indicateur `-quorum` activé sur la commande `startOgServer` ou `startXsServer`.**

     Vous pouvez utiliser cette méthode de configuration lorsque vous démarrez des serveurs autonomes uniquement.

```
# bin/startOgServer cat0 –serverProps objectGridServer.properties -quorum true
```

     8.6+

```
# bin/startXsServer cat0 –serverProps objectGridServer.properties -quorum true
```

     Pour plus d'informations sur la commande **`startOgServer`** ou **`startXsServer`**, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.
2. **Démarrez les serveurs dans la même zone.**

   Lorsque vous exécutez une grille sur plusieurs centres de données, les serveurs doivent utiliser les informations de zone pour identifier le centre de données dans lequel ils résident. La définition de la zone sur les serveurs de conteneur permet à WebSphere eXtreme Scale de contrôler l'intégrité des serveurs de conteneur qui sont limités au centre de données pour réduire le trafic entre les centres de données. Les machines virtuelles Java de serveur de conteneur dans un groupe central ne doivent jamais couvrir plusieurs réseaux locaux reliés par des liaisons, comme dans un réseau étendu. Voir «Définition des zones des serveurs de conteneur», à la page 299 pour plus d'informations sur la définition de zones pour les serveurs de conteneur.

Les machines virtuelles Java des serveurs de conteneur sont marquées avec un identificateur de zone. La grille de données des machines virtuelles Java de conteneur est automatiquement fractionnée en petits groupes centraux de machines virtuelles. Un groupe central ne comprendra que des machines virtuelles Java de la même zone. Des machines virtuelles Java de zones différentes ne feront jamais partie du même groupe central.

Un groupe central essaiera agressivement de détecter la défaillance de l'une de ses machines virtuelles Java.

Résultats

En configurant le mécanisme de quorum pour qu'il soit activé sur les serveurs de catalogue dans un domaine de service de catalogue, tous les serveurs de catalogue doivent être disponibles pour que les opérations de placement de grille de données puissent être exécutées. En cas de microcoupure réseau brève, les opérations de placement sont temporairement arrêtées jusqu'à ce que tous les serveurs de catalogue dans le quorum soient disponibles.

Vous pouvez ajouter des serveurs de catalogue supplémentaires au quorum en répétant ces étapes.

Que faire ensuite

- Vous pouvez supprimer un serveur de catalogue du quorum en arrêtant le serveur de catalogue à l'aide de la méthode d'administration requise par la configuration. Lorsqu'un serveur de catalogue est arrêté via des actions d'administration, le quorum est automatiquement rétabli entre les serveurs de catalogue restants et le placement se poursuit. Si vous redémarrez le serveur de catalogue avec les étapes décrites dans cette rubrique, le serveur de catalogue peut rejoindre le quorum.
- Si une défaillance permanente ou de longue durée se produit sur un serveur de catalogue qui se trouve dans le quorum actuellement défini, vous devez remplacer le mécanisme du quorum pour que le placement puisse continuer. Voir «Gestion des incidents du centre de données», à la page 522 pour plus d'informations sur le remplacement du mécanisme de quorum.

Optimisation de la valeur de l'intervalle des pulsations pour la détection des basculements

Le paramètre d'intervalle du signal de présence permet de configurer le laps de temps séparant deux vérifications par le système des serveurs en panne. Ce paramètre s'applique uniquement aux serveurs de catalogue.

Pourquoi et quand exécuter cette tâche

La configuration des basculements varie en fonction du type d'environnement que vous utilisez. Si vous utilisez un environnement autonome, vous pouvez configurer les basculements à l'aide de la ligne de commande. Si vous utilisez un environnement WebSphere Application Server Network Deployment, vous devez les configurer à partir de la console d'administration de WebSphere Application Server Network Deployment.

Procédure

- Configurez les basculements pour les environnements autonomes.

  Vous pouvez configurer les intervalles des pulsations du serveur de catalogue à l'aide du paramètre **-heartbeat** dans le fichier script **startOgServer** or **startXsServer**. Affectez à ce paramètre l'une des valeurs suivantes :

Tableau 26. Intervalles de signal de présence

Valeur	Action	Description
0	Standard (par défaut)	Les basculements sont généralement détectés dans les 30 secondes.
-1	Elevé	Les basculements sont généralement détectés dans les 5 secondes.
1	Souple	Les basculements sont généralement détectés dans les 180 secondes.

Un intervalle élevé entre les signaux de présence peut être utile si les processus et le réseau sont stables. Si le réseau ou les processus ne sont pas configurés de manière optimale, il peut manquer des signaux de présence, ce qui peut fausser la détection des incidents.

- Configurez les basculements pour les environnements WebSphere Application Server.

  Vous pouvez configurer WebSphere Application Server Network Deployment Version 7.0 et les versions suivantes pour autoriser WebSphere eXtreme Scale à basculer très rapidement. La durée par défaut de pour les incidents matériels est d'environ 200 secondes. Un incident matériel est un ordinateur physique, une panne du serveur, déconnexion de câble réseau ou une erreur du système d'exploitation. Les incidents dus aux pannes de processus ou à des échecs logiciels sont généralement basculés en moins d'une seconde. La détection des incidents logiciels est effectuée lorsque les sockets réseau du processus inactif sont fermés automatiquement par le système d'exploitation du serveur qui héberge le processus.

  Configuration des signaux de présence du groupe central

  Si WebSphere eXtreme Scale est exécuté dans un processus WebSphere Application Server, il hérite des caractéristiques de reprise en ligne des paramètres du groupe central du serveur d'applications. Les sections suivantes décrivent comment configurer les paramètres des signaux de présence du groupe central pour différentes versions de WebSphere Application Server Network Deployment :

  – **Mise à jour des paramètres des groupes centraux de WebSphere Application Server Network Deployment Version 7.0**

    WebSphere Application Server Network Deployment Version 7.0 fournit deux paramètres de groupe central qui peuvent être ajustés pour augmenter ou réduire le délai de détection des incidents :

    - **Période de transmission du signal de présence.** La valeur par défaut est de 30000 millisecondes.
    - **Période d'expiration du signal de présence.** La valeur par défaut est de 180000 millisecondes.

    Pour plus de détails sur la manière de modifier ces paramètres, voir la rubrique relative à la WebSphere Application Server Network Deployment reconnaissance et de détection des incidents dans le centre de documentation.

    Utilisez les paramètres suivants pour spécifier un délai de détection des incidents de 1500 ms pour les serveurs WebSphere Application Server Network Deployment Version 7 :

    - Spécifiez une période de transmission du signal de présence de 750 millisecondes.
    - Spécifiez une période d'expiration du signal de présence de 1500 millisecondes.

Que faire ensuite

Lorsque vous modifiez ces paramètres pour réduire les délais de basculement, certains points d'optimisation du système sont à prendre en compte. Tout d'abord, Java n'est pas un environnement en temps réel. Des unités d'exécution peuvent être retardées si la JVM connaît des délais de récupération de place importants. Les unités d'exécution risquent également d'être retardées si la charge de la machine qui héberge la JVM est considérable (à cause de la JVM elle-même ou d'autres processus exécutés sur cette machine). Si les unités d'exécution sont retardées, les signaux de présence risquent de ne pas être envoyés à temps. Au pire, ils risquent d'être retardés du délai requis pour la reprise en ligne. Si des unités d'exécution sont retardées, des incidents sont détectés à tort. Le système doit être optimisé et dimensionné de sorte à éviter la détection de faux incidents en production. Il est recommandé pour cela de tester la charge de manière adéquate.

Remarque : La version actuelle d'eXtreme Scale prend en charge WebSphere Real Time.

Configuration de serveurs de conteneur

Le serveur de conteneur stocke les données d'application pour la grille de données. Ces données sont généralement divisées en fractions appelées partitions qui sont hébergés sur plusieurs serveurs de conteneur. Chaque serveur de conteneur à son tour héberge un sous-ensemble de l'ensemble des données.

Pourquoi et quand exécuter cette tâche

- **Serveurs de conteneur autonomes** :

  Configurez les serveurs de conteneur autonomes avec un fichier de propriétés de serveur et un fichier XML de stratégie de déploiement. Contrôlez le cycle de vie d'un serveur de conteneur avec les scripts de démarrage et d'arrêt ou l'API de serveur embarqué.
- **Serveurs de conteneur qui démarrent dans WebSphere Application Server :**

  Configurez les serveurs de conteneur dans WebSphere Application Server avec un fichier de propriétés de serveur et un fichier XML de stratégie de déploiement intégré au module d'application Java EE. Le cycle de vie des serveurs de conteneur est contrôlé par l'application. Les serveurs de conteneur démarrent et s'arrêtent avec l'application.

Propriétés de reconnexion au serveur de conteneur

Utilisez des propriétés Java virtual machine (JVM) pour configurer la façon dont le serveur de conteneur se connecte à la grille de données si le serveur se déconnecte.

Propriétés système JVM

Si un serveur de conteneur se déconnecte de la grille de données, WebSphere eXtreme Scale tente de le reconnecter. En définissant des propriétés système, vous pouvez contrôler le fonctionnement de la fonction de reconnexion du conteneur. Vous pouvez définir ces propriétés lorsque vous démarrez un serveur de conteneur. Certaines de ces propriétés s'appliquent à WebSphere eXtreme Scale dans un environnement autonome, tandis que d'autres sont uniquement applicables lors du démarrage d'un serveur de conteneur pour WebSphere eXtreme Scale for WebSphere Application Server. Par exemple, lors du démarrage d'un serveur de conteneur pour WebSphere eXtreme Scale dans un environnement autonome, vous pouvez définir ces propriétés comme option à partir de la ligne de commande :

```
startOgServer.sh server01 -objectgridFile objectgrid.xml
-deploymentPolicyFile deployment.xml
-Dcom.ibm.websphere.objectgrid.container.reconnect.restart=false
```

8.6+
```
startXsServer.sh server01 -objectgridFile objectgrid.xml
-deploymentPolicyFile deployment.xml
-Dcom.ibm.websphere.objectgrid.container.reconnect.restart=false
```

Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479. Si vous souhaitez définir la propriété appropriée pour WebSphere eXtreme Scale for WebSphere Application Server, vous pouvez utiliser l'outil WebSphere Integrated Solutions Console. Cet outil est une interface graphique qui se trouve dans l'environnement WebSphere Application Server et il est installé en tant qu'extension WebSphere ISC.

com.ibm.websphere.objectgrid.container.reconnect.block.reconnect.time
Définit la quantité de temps (en millisecondes) nécessaire pour bloquer un autre appel de connexion du conteneur. Cette propriété n'est valide que lorsqu'un serveur de conteneur s'exécute sur l'offre de produit WebSphere eXtreme Scale for WebSphere Application Server.

Par défaut : 30000 millisecondes

com.ibm.websphere.objectgrid.container.reconnect.min.successful.heartbeats
Définit le nombre minimum de signaux de présence ayant abouti avant qu'un conteneur puisse être arrêté. Cette propriété n'est valide que lorsqu'un serveur de conteneur est démarré pour l'offre de produit WebSphere eXtreme Scale for WebSphere Application Server.

Par défaut : 10

com.ibm.websphere.objectgrid.container.reconnect.restart
Définit si la reconnexion du conteneur peut redémarrer la machine virtuelle Java. Cette propriété n'est valide que lorsqu'un serveur de conteneur est démarré pour WebSphere eXtreme Scale dans un environnement autonome.

Valeur par défaut : true

com.ibm.websphere.objectgrid.container.reconnect.restart.delay
Définit le délai nécessaire (en millisecondes) après la mort du parent pour pouvoir procéder au démarrage sur le conteneur enfant nouvellement créé lors du redémarrage de la machine virtuelle Java. Cette propriété n'est valide que lorsqu'un serveur de conteneur est démarré pour l'offre de produit WebSphere eXtreme Scale dans un environnement autonome.

Par défaut : 2000 millisecondes

com.ibm.websphere.objectgrid.container.reconnect.restart.parent.timeout
Définit le délai d'attente de la mort du parent (en millisecondes) pouvant être respecté par le conteneur enfant nouvellement créé avant dépassement de délai lors du redémarrage de la machine virtuelle Java. Cette propriété n'est valide que lorsqu'un serveur de conteneur est démarré pour l'offre de produit WebSphere eXtreme Scale dans un environnement autonome.

Par défaut : 180000 millisecondes

com.ibm.websphere.objectgrid.container.reconnect.retry.forever
Indique si le conteneur doit retenter indéfiniment une reconnexion du conteneur. Cette propriété n'est valide que lorsqu'un serveur de conteneur est démarré sur l'offre de produit WebSphere eXtreme Scale for WebSphere Application Server.

Valeur par défaut : false

Configuration des serveurs de conteneur dans WebSphere Application Server

Java

Configurez les serveurs de conteneur dans WebSphere Application Server avec un fichier de propriétés de serveur et un fichier XML de stratégie de déploiement intégré au module d'application Java EE. Les serveurs de conteneur s'arrêtent et démarrent lorsque l'application est arrêtée et démarrée.

Avant de commencer

Configurez un domaine de service de catalogue. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307.

Pourquoi et quand exécuter cette tâche

Pour créer des serveurs de conteneur dans WebSphere Application Server, vous devez imbriquer les fichiers XML de configuration WebSphere eXtreme Scale pour créer les serveurs de conteneur dans le module d'application.

Procédure

1. Identifiez les serveurs d'applications sur lequel vous souhaitez déployer l'application Java EE qui contient les définitions de serveur de conteneur WebSphere eXtreme Scale. Vérifiez que les profils de serveur d'applications cible ont été étendus avec le profil WebSphere eXtreme Scale. Dans un environnement de production, ne faites pas cohabiter les serveurs que vous utilisez pour les serveurs de conteneur et les serveurs de catalogue. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 243.
2. Configurez un fichier de propriétés du serveur et ajoutez-le au chemin d'accès aux classes de chaque noeud de serveur d'applications cible. Pour plus d'informations, voir Fichier de propriétés du serveur.
3. Ajoutez le fichier XML du descripteur d'ObjectGrid et le fichier XML de stratégie de déploiement au module d'application. Pour plus d'informations, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur».

Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur : Java

Les serveurs de conteneur dans un environnement WebSphere Application Server démarrent automatiquement quand un module contenant les fichiers XML eXtreme Scale démarre.

Avant de commencer

WebSphere Application Server et WebSphere eXtreme Scale doivent être installés, et vous devez être capable d'accéder à la console d'administration de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Les applications Java Platform, Enterprise Edition ont des règles de chargeur de classe complexes qui compliquent grandement le chargement des classes lors de l'utilisation d'une grille de données partagée dans un serveur Java EE. Une application Java EE correspond généralement à un seul fichier EAR (Enterprise Archive). Le fichier EAR contient un ou plusieurs modules EJB (Enterprise JavaBeans) ou modules WAR (Web archive) déployés.

WebSphere eXtreme Scale surveille le démarrage de chaque module et recherche des fichiers XML eXtreme Scale. Si le service de catalogue détecte qu'un module démarre avec les fichiers XML, le serveur d'applications est enregistré en tant que conteneur de serveur machine virtuelle Java (JVM). En enregistrant les serveurs de conteneur dans le service de catalogue, une même application peut être déployée dans des grilles de données différents, mais utilisée comme une grille de données unique par le service de catalogue. Le service de catalogue n'est pas concerné par les cellules, les grilles, ou les grilles dynamiques. Une grille de données unique peut couvrir plusieurs cellules, si nécessaire.

Procédure

1. Modularisez le fichier EAR pour disposer de modules incluant les fichiers XML eXtreme Scale dans le dossier `META-INF`. WebSphere eXtreme Scale détecte la présence des fichiers `objectGrid.xml` et `objectGridDeployment.xml` dans le dossier `META-INF` des modules EJB et WEB lorsqu'ils démarrent. Si un seul fichier `objectGrid.xml` est détecté, la machine JVM est supposée être un client. Sinon, la machine virtuelle Java est supposée faire office de grille de données définie dans le fichier `objectGridDeployment.xml`.

   Vous devez utiliser les noms corrects pour ces fichiers XML. Les noms de fichier sont sensibles à la casse. Si les fichiers sont absents, le conteneur ne démarre pas. Vous pouvez vérifier si le fichier `systemout.log` contient des messages indiquant que des fragments sont placés. Un module EJB ou d'archive Web utilisant eXtreme Scale doit avoir des fichiers XML eXtreme Scale dans son répertoire `META-INF`.

   Les fichiers XML eXtreme Scale incluent :

   - Un fichier XML de descripteur d'ObjectGrid nommé `objectGrid.xml`. Pour plus d'informations, voir Fichier XML du descripteur d'ObjectGrid.
   - Un fichier XML de descripteur de déploiement nommé `objectGridDeployment.xml`. Pour plus d'informations, voir Fichier XML du descripteur de la règle de déploiement.
   - (Facultatif) Un fichier XML de descripteur de métadonnées d'entité, si des entités sont utilisées. Le nom du fichier `entity.xml` doit correspondre au nom spécifié dans le fichier `objectGrid.xml`. Pour plus d'informations, voir Fichier XML du descripteur de métadonnées d'entité.

   L'environnement d'exécution détecte ces fichiers, puis contacte le service de catalogue pour l'informer qu'un autre conteneur est disponible pour héberger les fragments pour ce eXtreme Scale.

   Conseil : Si votre application comporte des entités et que vous prévoyez d'utiliser un serveur un conteneur, affectez la valeur **`minSyncReplicas`**) `0` dans le fichier XML du descripteur de déploiement. Sinon, vous risquez de voir l'un des messages suivants dans le fichier `SystemOut.log` car le positionnement ne pourra se produire tant qu'un autre serveur n'a pas démarré pour satisfaire à la règle minSyncReplica :

```
CWPRJ1005E: Erreur lors de la résolution de l'association
d'entités. Entité=nom_entité,
association=nom_association.

CWOBJ3013E: Le référentiel EntityMetadata n'est pas disponible.
Le seuil du délai d'attente a été atteint lors de la tentative
d'inscription de l'entité : nom_entité.
```

2. Déployez et démarrez votre application.

   Le conteneur démarre automatiquement quand le module est démarré. Le service de catalogue commence à placer les serveurs principaux et secondaires de partition (fragments) dès que possible. Ce placement a lieu immédiatement, à moins que vous ne définissiez l'environnement pour le retarder. Pour plus d'informations, voir «Contrôle du placement», à la page 517.

Que faire ensuite

Les applications dans la même cellule que les conteneurs, peuvent se connecter à ces grilles de données à l'aide d'une méthode ObjectGridManager.connect(null, null), puis appeler la méthode getObjectGrid(ccc, "object grid name"). La méthode connect ou getObjectGrid peut être bloquée jusqu'à ce que les conteneurs aient placés les fragments, mais ce blocage représente un problème uniquement quand la grille de données démarre.

Chargeurs de classe

Tout plug-in ou objet stocké dans un eXtreme Scale est chargé sur un certain chargeur de classe. Deux modules EJB dans un même fichier d'archive d'entreprise peuvent inclure ces objets. Ces objets sont les identiques, mais ils sont chargés avec différents chargeurs de classe. Si l'application A stocke un objet Personne dans une mappe qui est locale pour le serveur, l'application B reçoit une exception `ClassCastException` si elle essaie de lire cet objet. Cette exception se produit car l'application B a chargé l'objet Personne sur un chargeur de classe différent.

Une manière de résoudre ce problème consiste à faire en sorte qu'un module racine contienne les plug-in et les objets nécessaires qui sont stockés dans le eXtreme Scale. Chaque module utilisant eXtreme Scale doit référencer ce module pour ses classes. Une autre solution consiste à placer ces objets partagés dans un fichier JAR d'utilitaire qui se trouve dans un chargeur de classe commun partagé par les modules et les applications. Les objets peuvent également être placés dans des classes WebSphere ou le répertoire `lib/ext`, mais cet placement complique le déploiement.

Les modules EJB dans un fichier d'archive d'entreprise partagent généralement le même ClassLoader et ne sont pas affectés par ce problème. Chaque module de fichier d'archive Web possède son propre ClassLoader et est affecté par ce problème.

Connexion à un client de grille de données uniquement

Si la propriété **`catalog.services.cluster`** est définie dans les propriétés personnalisées d'une cellule, d'un noeud ou d'un serveur, un module dans le fichier EAR peut appeler la méthode ObjectGridManager.connect (ServerFactory.getServerProperties().getCatalogServiceBootstrap(), null, null) pour obtenir un ClientClusterContext. Le module peut également appeler la méthode ObjectGridManager.getObjectGrid(ccc, "grid name") pour obtenir une référence à la grille de données. Si des objets d'application sont stockés dans des mappes, vérifiez que ces objets sont présents dans un chargeur de classe commun.

Les clients Java ou les clients en dehors de la cellule peuvent se connecter au port IIOP d'amorçage du service de catalogue. Dans WebSphere Application Server, le gestionnaire de déploiement héberge le service de catalogue par défaut. Le client peut alors obtenir un ClientClusterContext et la grille de données nommée.

Gestionnaire d'entités

Avec le gestionnaire d'entités, les blocs de données sont stockés dans les mappes et non pas les objets d'application, ce qui réduit les problèmes de chargeur de classe. Les plug-in, en revanche, peuvent présenter un problème. Notez également qu'un fichier XML de descripteur ObjectGrid de remplacement client est toujours nécessaire lorsqu'une grille de données a des entités définies : ObjectGridManager.connect("host:port[,host:port], null, objectGridOverride) or ObjectGridManager.connect(null, objectGridOverride).

Configuration d'une grille de données d'entreprise dans un environnement autonome pour la mise en cache dynamique

Copiez et modifiez ces fichiers descripteurs déploiement et objectGrid afin de configurer une grille d'entreprise pour la mise en cache dynamique. Ces fichiers sont utilisés pour démarrer une grille de données d'entreprise.

Pourquoi et quand exécuter cette tâche

Lorsque WebSphere eXtreme Scale est défini comme fournisseur pour une instance de cache dynamique WebSphere Application Server, les serveurs WebSphere eXtreme Scale sont démarrés dans un environnement autonome ou dans un environnement WebSphere Application Server. Voir «Démarrage et arrêt des serveurs sécurisés», à la page 479 pour plus d'informations. Ce processus implique d'utiliser les fichiers descripteurs de déploiement et objectGrid qui sont utilisés pour configurer la grille de données d'entreprise. La mise en cache dynamique nécessite une configuration spéciale. Par conséquent, plusieurs fichiers sont fournis avec WebSphere eXtreme Scale ; ils doivent être copiés, modifiés (en fonction des besoins) et utilisés pour démarrer la grille de données d'entreprise. Ces fichiers peuvent être utilisés tels quels, mais peuvent être modifiés et ils doivent donc être copiés vers un emplacement distinct avant d'être modifiés ou utilisés.

Remarque : Selon la manière dont vous avez installé WebSphere eXtreme Scale, ces fichiers se trouvent dans le répertoire `was_root/optionalLibraries/ObjectGrid/dynacache/etc` pour les installations avec WebSphere Application Server, ou pour une installation dans un environnement autonome, ces fichiers se trouvent dans le `wxs_install_root/ObjectGrid/dynacache/etc`.

Important : Il est vivement recommandé de copier ces fichiers dans un autre emplacement avant de les modifier ou de les utiliser.

Fichier descripteur de cache dynamique (dynacache-remote-deployment.xml)
Ce fichier est le fichier descripteur de déploiement pour démarrer un serveur de conteneur pour la mise en cache dynamique. Voir Fichier XML du descripteur de la règle de déploiement pour plus d'informations. Bien que ce fichier puisse être utilisé tel quel, les éléments ou attributs suivants sont éventuellement modifiés ou sont importants :

- **mapSet name et map ref**

  L'attribut **name** dans mapSet et la valeur définie pour map ref ne correspondent pas directement au nom d'instance de cache dynamique défini pour WebSphere Application Server et sont généralement

modifiés. Toutefois, si ces valeurs sont modifiées, les propriétés personnalisées correspondantes doivent être ajoutées à la configuration de l'instance de cache dynamique. Pour plus d'informations, voir «Personnalisation d'une instance de cache dynamique avec des propriétés personnalisées», à la page 405.

- **numberOfPartitions**

  Cet attribut peut être changé pour représenter le nombre approprié de partitions de votre configuration. Pour plus d'informations, voir «Planification de la capacité de l'environnement», à la page 72.

- **maxAsyncReplicas**

  Cet attribut peut être changé. Un cache dynamique est généralement utilisé dans un modèle de cache secondaire avec une base de données ou une autre ressource comme système d'enregistrement pour les données. Par conséquent, si vous affectez à ce paramètre la valeur OPTIMISTIC ou NONE, vous déclenchez le traitement en cache local, lorsque le type de transport eXtreme I/O (XIO) est utilisé, et les compromis d'espace et de performances nécessaires pour rendre les données hautement disponibles décourage l'utilisation de la réplication. Cependant, dans certains cas, la haute disponibilité est importante.

- **numInitialContainers**

  Cet attribut doit être affecté du nombre de conteneurs à inclure dans le démarrage initial de la grille de données d'entreprise. La définition correcte de ce paramètre facilite le placement et la distribution des partitions dans la grille de données.

Fichier XML descripteur ObjectGrid de cache dynamique (dynacache-remote-objectgrid.xml)

Ce fichier est le fichier descripteur ObjectGrid recommandé pour démarrer un serveur de conteneur pour la mise en cache dynamique. Voir Fichier XML du descripteur d'ObjectGrid pour plus d'informations. Il est configuré pour s'exécuter avec le type de transport eXtreme I/O (XIO) en utilisant le format XDF (eXtreme Data Formatting). En outre, les index Dependency ID et Template ID sont configurés pour utiliser un index global qui améliore les performances d'invalidation. Bien que ce fichier puisse être utilisés tel quel, les éléments ou les attributs suivants sont changés occasionnellement ou ont une importance significative.

- **objectGrid name et backingMap name**

  Les attributs **name** dans les éléments objectGrid et backingMap ne correspondent pas directement au nom d'instance de cache dynamique configuré pour l'instance de cache WebSphere Application Server et ne doivent généralement pas être changés. Si, cependant, ces attributs sont changés, les propriétés personnalisées correspondantes doivent être ajoutées à la configuration de l'instance de cache dynamique. Pour plus d'informations, voir «Personnalisation d'une instance de cache dynamique avec des propriétés personnalisées», à la page 405.

- **copyMode**

  Affectez à cet attribut la valeur COPY_TO_BYTES. Cette valeur active le format XDF (eXtreme Data Format) lorsque le type de transport eXtreme I/O (XIO) est utilisé. Si vous définissez un autre mode CopyMode, vous désactivez XDF et vous devez annuler la mise en commentaire du bean de plug-in ObjectTransformer.

- **lockStrategy**

Affectez la valeur PESSIMISTIC à cet attribut. Si vous définissez la valeur OPTIMISTIC ou NONE vous activez le traitement en cache local et les propriétés du fichier dynamic-nearcache-objectgrid.xml doivent être utilisées.

- **backingMapPluginCollections**

  Cet élément est nécessaire. Les éléments enfant Plug-in Evictor et Plug-in MapIndex sont nécessaires pour la mise en cache dynamique et doivent être supprimés.

- **GlobalIndexEnabled**

  DEPENDENCY_ID_INDEX et TEMPLATE_INDEX contiennent une propriété GlobalIndexEnabled affectée de la valeur true. Si vous définissez la valeur false, vous désactivez la fonction d'index global pour ces index. Il est recommandé de maintenir ces index globaux activés, sauf si vous utilisez un petit nombre de partitions, moins de 40, par exemple.

- **objectTransformer**

  Comme ce fichier descripteur objectGrid doit s'exécuter dans le format XDF (eXtreme Data Format), il est mis en commentaire. Si vous voulez désactiver XDF (en changeant la valeur copyMode value), vous devez annuler la mise en commentaire du plug-in.

Fichier descripteur ObjectGrid de cache local (dynacache-nearCache-ObjectGrid.xml)

Ce fichier est le fichier descripteur ObjectGrid reommandé pour démarrer les serveurs de conteneur de grille pour la mise en cache dynamique lorsqu'un cache local est nécessaire. Il est configuré pour s'exécuter avec le type de transport eXtreme I/O (XIO) en utilisant le format XDF (eXtreme Data Formatting. En outre, les index Dependency ID et Template sont configurés pour utiliser un index global qui améliore les performances d'invalidation. La fonction de cache local de mise en cache dynamique nécessite d'utiliser le type de transport eXtreme I/O (XIO).

Bien que ce fichier puisse être utilisé tel quel, les éléments ou les attributs suivants sont changés occasionnellement ou ont une importance significative :

- **objectGrid name et backingMap name**

  Ces valeurs dans ce fichier ne correspondent pas directement au nom d'instance de cache dynamique configuré pour l'instance de cache de WebSphere Application Server et ne doivent généralement pas être changées. Toutefois, si ces valeurs sont modifiées, les propriétés personnalisées correspondantes doivent être ajoutées à la configuration de l'instance de cache dynamique.

- **lockStrategy**

  Cette propriété doit avoir la valeur OPTIMISTIC ou NONE pour activer un cache local. Aucune autre stratégie lockingStrategy ne prend en charge le cache local.

- **nearCacheInvalidationEnabled**

  Cette propriété doit avoir la valeur true pour activer un cache local de mise en cache dynamique. Cette fonction utilise pub-sub pour envoyer les invalidations du cache distant au cache local en les maintenant synchroniser.

- **nearCacheLastAccessTTLSyncEnabled**

Cette propriété doit avoir la valeur true pour activer un cache local de mise en cache dynamique. Cette fonction utilise pub-sub pour envoyer les expulsions TTL du cache distant au cache local en les maintenant synchroniser.

- **copyMode**

  Cette propriété backingMap a la valeur COPY_TO_BYTES. Cette valeur active le format XDF (eXtreme Data Format) lorsque le type de transport eXtreme I/O (XIO) est utilisé. Si vous définissez un autre mode CopyMode, vous désactivez XDF et vous devez annuler la mise en commentaire du bean de plug-in ObjectTransformer.

- **backingMapPluginCollections**

  MapIndexPlugins et Evictor sont des éléments requis pour la mise en cache dynamique et ils ne doivent pas être supprimés.

- **GlobalIndexEnabled**

  DEPENDENCY_ID_INDEX et TEMPLATE_INDEX contiennent une propriété GlobalIndexEnabled affectée de la valeur true. Si vous définissez la valeur false, vous désactivez la fonction d'index global pour ces index. Il est recommandé de maintenir ces index globaux activés, sauf si vous utilisez un petit nombre de partitions (< 40).

- **ObjectTransformer**

  Comme ce fichier descripteur objectGrid doit s'exécuter dans le format XDF (eXtreme Data Format), il est mis en commentaire. Si XDF doit être désactivé (en changeant copyMode), la mise en commentaire du plug-in doit être annulée.

Fichier descripteur ObjectGrid existant dynamique (dynacache-legacy85-ObjectGrid.xml)

Ce fichier est le fichier descripteur ObjectGrid recommandé pour démarrer un serveur de conteneur de mise en cache dynamique lorsque vous avez choisi un cache local. Bien que ce fichier puisse être utilisé tel quel, les éléments ou les attributs suivants sont changés occasionnellement ou ont une importance significative :

- **objectGrid name et backingMap name**

  Ces valeurs dans ce fichier ne correspondent pas directement au nom d'instance de cache dynamique configuré pour l'instance de cache de WebSphere Application Server et ne doivent généralement pas être changées. Toutefois, si ces valeurs sont modifiées, les propriétés personnalisées correspondantes doivent être ajoutées à la configuration de l'instance de cache dynamique.

- **copyMode**

  Cette propriété backingMap a la valeur COPY_ON_READ_AND_COMMIT. Cette valeur ne doit pas être modifiée.

- **lockStrategy**

  Cette propriété backingMap a la valeur PESSIMISTIC. Ne changez pas cette valeur.

- **backingMapPluginCollections**

  MapIndexPlugins, Evictor, et Object Transformer sont des éléments requis pour la mise en cache dynamique et ils ne doivent pas être supprimés.

Configuration de plusieurs topologies de centres de données

Avec la réplication asynchrone multimaître, vous liez un ensemble de domaines de service de catalogue. Les domaines de service de catalogue connectés sont ensuite synchronisés en utilisant la réplication via les liaisons. Vous pouvez définir les liaisons à l'aide de fichiers de propriétés, lors de l'exécution avec des programmes JMX (Java Management Extensions) ou avec les utilitaires de ligne de commande. Le groupe de liaisons actuel d'un domaine est stocké dans le service de catalogue. Vous pouvez ajouter et supprimer des liens sans redémarrer le domaine de service de catalogue qui héberge la grille de données.

Avant de commencer

- Voir «Planification de plusieurs topologies de centre de données», à la page 45 pour plus d'informations sur les topologies de réplication multimaîtres et les considérations de conception. Vous pouvez configurer des liaisons entre les domaines de service de catalogue avec le fichier de propriétés du serveur pour former la topologie lors du démarrage du serveur. Vous pouvez configurer des liaisons lors de l'exécution.
- Si vous utilisez des chargeurs dans la topologie de réplication multimaître, vous devez planifier la manière dont vous allez gérer des données exactes entre les centres de données. Les méthodes que vous pouvez utiliser varient en fonction de la topologie que vous utilisez. Pour plus d'informations, voir «Remarques sur les chargeurs dans une topologie multimaître», à la page 50.

Procédure

- Définissez des liaisons dans le fichier de propriétés du serveur de catalogue de chaque domaine de service de catalogue de la topologie à des fin d'amorçage.

  Voir Fichier de propriétés du serveur pour plus d'informations sur la définition de ce fichier pour le serveur de catalogue.

  Important : Les noms de propriété sont sensibles à la casse.

  Nom de domaine local :
  Indiquez le nom du domaine de service de catalogue du serveur de catalogue actuel, le domaine A, par exemple.

  ```
  domainName=A
  ```

  Liste facultative des noms de domaines externes :

  Indiquez les noms des domaines de service de catalogue auxquels vous souhaitez vous lier dans la topologie de réplication multimaître ; le domaine B, par exemple :

  ```
  foreignDomains=B
  ```

  Liste facultative des noms de domaines externes :
  Spécifie les informations de connexion des serveurs de catalogue des domaines externes ; le domaine B, par exemple :

  ```
  B.endPoints=hostB1:2809, hostB2:2809
  ```

  Si un domaine externe comporte plusieurs serveurs de catalogue, spécifiez-les tous.
- Utilisez l'utilitaire **xscmd** ou la programmation JMX pour ajouter ou supprimer des liaison lors de l'exécution.

  Les liens d'un domaine sont conservés dans le service de catalogue dans la mémoire répliquée. Cet ensemble de liens peut être modifié à tout moment par

l'administrateur sans nécessiter pour autant un redémarrage de ce domaine ou des autres domaines. L'utilitaire **xscmd** inclut plusieurs options pour l'utilisation des liaisons.

L'utilitaire **xscmd** se connecte à un service de catalogue et donc un domaine de service de catalogue unique. Par conséquent, l'utilitaire **xscmd** peut être utilisé pour créer et supprimer des liaisons entre le domaine auquel il se connecte et n'importe quel autre domaine.

Utilisez la ligne de commande pour créer une liaison, par exemple :

```
xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
```

Cette commande établit une nouvelle liaison entre le domaine local et le domaine externe nommé `dname`. Le service de catalogue `dname` est exécuté à l'adresse `fdHostA:2809` et à l'adresse `fdHostB:2809`. Le domaine de service de catalogue local a un hôte d'écoute de service de catalogue et le port `host:2809`. Indiquez tous les noeuds finals du service de catalogue à partir du domaine externe de sorte que la connectivité à la tolérance aux pannes pour le domaine soit possible. N'utilisez pas une seule paire `host:port` pour le service de catalogue du domaine de service de catalogue externe.

Vous pouvez utiliser n'importe quelle machine virtuelle Java avec **xscmd** et utilisez l'option **-cep**. Si le serveur de catalogue est hébergé sur un gestionnaire de déploiement WebSphere Application Server, le port est habituellement le 9809.

Les ports spécifiés pour le domaine externe ne sont pas des ports JMX. Ce sont les ports que l'on utilise d'ordinaire pour les clients eXtreme Scale.

Une fois que la commande d'ajout de nouveau lien a été émise, le service de catalogue donne instruction à tous les conteneurs qu'il gère de commencer à se répliquer vers le domaine externe. Un lien n'est pas nécessaire des deux côtés. Il suffit d'en créer un sur l'une des deux extrémités.

La ligne de commande permet également de supprimer un lien, par exemple :

```
xscmd –c dismissLink -cep host:2809 -fd dname
```

Cette commande se connecte au service de catalogue d'un domaine et lui donne instruction d'arrêter la réplication vers un domaine spécifique. Une liaison doit être supprimée sur un côté uniquement.

Avertissement : Vous pouvez exécuter les commandes establish ou dismiss link plusieurs fois. Si la liaison n'entre pas dans l'état correct ou est rompue, réexécutez la commande.

Exemples

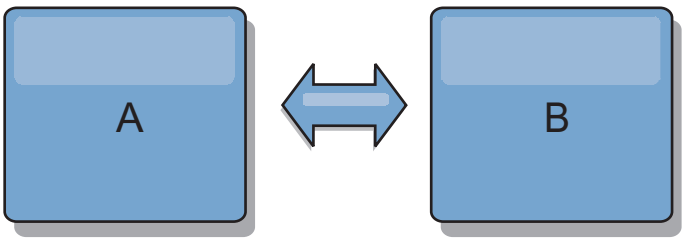


Figure 35. Exemple : liaison entre les domaines de services de catalogue

Supposons que vous souhaitiez définir une configuration à deux domaines comportant les domaines de service de catalogue A et B.

Voici le fichier de propriétés du serveur de catalogue dans le domaine A :

```
domainName=A
foreignDomains=B
B.endPoints=hostB1:2809, hostB2:2809
```

Voici le fichier de propriétés du serveur de catalogue dans le domaine B. Notez les similitudes entres les deux fichiers de propriétés.

```
domainName=B
foreignDomains=A
A.endPoints=hostA1:2809,hostA2:2809
```

Une fois les deux domaines démarrés, toutes les grilles de données ayant les caractéristiques suivantes sont répliquées entre ces domaines.

- Dispose d'un service de catalogue privé avec un nom de domaine unique
- A le même nom de grille de données que les autres grilles du domaine
- A le même nombre de partitions que les autres grilles de données dans le domaine
- Est une grille de données FIXED_PARTITION (les grilles de données PER_CONTAINER ne peuvent pas être répliquées)
- A le même nombre de partitions (sans forcément pour autant avoir le même nombre et le même type de fragments réplique)
- A les mêmes types de données répliqués que les autres grilles du domaine
- A les mêmes nom de groupe de mappes, noms de mappe et modèles de mappes dynamiques que les autres grilles dans le domaine

La règle de réplication d'un domaine de service de catalogue est ignorée.

L'exemple qui précède montre comment configurer chaque domaine pour qu'il ait un lien vers l'autre domaine, mais, en fait, il suffit de définir un lien dans une seule direction. C'est particulièrement utile lorsqu'on a affaire à des topologies en étoile, la configuration s'en trouve considérablement simplifiée. Le fichier de propriétés du concentrateur ne nécessite pas d'être modifié au fur et à mesure que des noeuds sont ajoutés à la topologie et il suffit que le fichier de chacun de ces noeuds comprenne des informations relatives au concentrateur. De la même manière, dans une topologie en anneau, il suffit que chacun des domaines ait un lien avec le domaine qui le précède et avec celui qui le suit dans l'anneau.

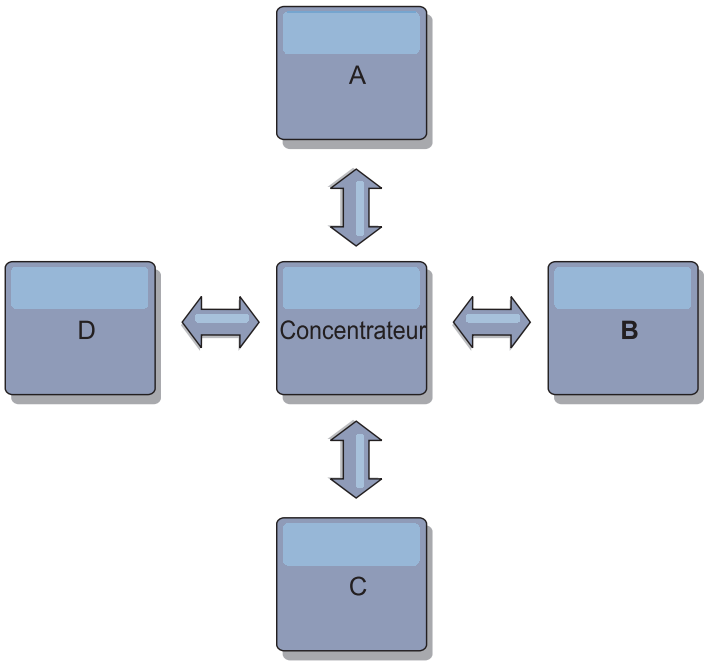


Figure 36. Exemple : topologie en étoile

La topologie avec un concentrateur et quatre domaines A, B, C et D possède des fichiers de propriétés de serveur, comme dans les exemples suivants.

```
domainName=Hub
```

La Branche A a les propriétés de serveur suivantes :

```
domainName=A
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

La branche B a les propriétés de serveur suivantes :

```
domainName=B
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

La branche C a les propriétés de serveur suivantes :

```
domainName=C
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

La branche D a les propriétés de serveur suivantes :

```
domainName=D
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Que faire ensuite

- Pour vérifier ou résoudre les problèmes associés aux liaisons entre les domaines de services de catalogue, vous pouvez utiliser l'utilitaire **xscmd**. Pour plus d'informations sur les commandes permettant de résoudre les problèmes liés à la configuration de centre de données, voir «Traitement des problèmes de plusieurs configurations de centre de données», à la page 686.

- Vous pouvez fournir un arbitre de collisions personnalisé pour résoudre les conflits entre les domaines de service de catalogue. Pour plus d'informations, voir Développement d'arbitres personnalisés pour la réplication multi-maître.

Configuration des ports

Vous devez ouvrir les ports pour communiquer entre les serveurs et avec les serveurs distants.

Configuration de ports en mode autonome

Vous pouvez configurer les ports nécessaires pour les serveurs et les clients dans un déploiement eXtreme Scale à l'aide de paramètres de ligne de commande, de fichiers de propriétés ou à l'aide d'un programme. La plupart des exemples dans les sections suivantes décrivent les paramètres de ligne de commande dans le script **startOgServer** ou **startXsServer**. Des options de configuration équivalentes peuvent être également définies dans des fichiers de propriétés, à l'aide de l'API de serveur embarqué ou l'API client.

Procédure

1. Démarrez les noeuds finals du service de catalogue.

   WebSphere eXtreme Scale utilise IIOP pour la communication entre les machines virtuelles Java. Les machines virtuelles Java de service de catalogue sont les seuls processus qui requièrent la configuration explicite de ports pour les services IIOP et de ports des services de groupe. Les autres processus d'allouer dynamiquement les ports.

   a. Spécifiez les ports client et homologue. Le port du client et le port homologue sont utilisés pour la communication entre les services de catalogue dans un domaine de service de catalogue. Pour indiquer le port client et le port homologue, utilisez l'option de ligne de commande suivante :

      -catalogServiceEndPoints <serverName:hostName:clientPort:peerPort>
      Définit la liste des serveurs de catalogue à lier ensemble dans un domaine de services de catalogue. Chaque attribut est défini comme suit :

      serverName
      Définit le nom du serveur de catalogue.

      hostName
      Spécifie le nom d'hôte de l'ordinateur sur lequel le serveur est lancé.

      clientPort
      Spécifie le port utilisé pour la communication de service de catalogue homologue.

      peerPort
      Cette value est identique à haManagerPort. Spécifie le port utilisé pour la communication de service de catalogue homologue.

      L'exemple suivant démarre le serveur de catalogue, cs1, qui se trouve dans le même domaine de services de catalogue que les serveurs cs2 et cs3 :

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

Si vous démarrez des serveurs de catalogue supplémentaires, ils doivent inclure les mêmes serveurs dans l'argument **-catalogServiceEndPoints**. L'ordre de la liste peut être différent, mais les serveurs contenus dans la liste doivent être les mêmes pour chaque serveur de catalogue. N'insérez aucun espace dans la liste.

Vous pouvez également définir les noeuds finaux de services de catalogue avec la propriété de serveur catalogClusterEndPoints.

b. Spécifiez le port et l'hôte d'écoute. Le port d'écoute ORB (Object Request Broker) est utilisé pour la communication entre les services de catalogue dans un domaine de services de catalogue et pour la communication entre les services de catalogue et les serveurs de conteneur et les clients. Pour indiquer le port d'écoute et l'hôte d'écoute, utilisez les options de ligne de commande suivantes :

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel le transport ORB (Object Request Broker) ou eXtremeIO (XIO) se lie pour les communications. La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que le mécanisme de transport dans la machine JVM connaisse l'adresse IP de liaison. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>

Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

Le port d'écoute et l'hôte d'écoute peuvent être également définis à l'aide des propriétés listenerHost et listenerPort.

c. Facultatif : Définissez le port de service JMX.

Le port de service JMX est utilisé pour la communication entre les clients JMX. Pour indiquer le port de service JMX, utilisez l'option de ligne de commande suivante :

-JMXServicePort <port>

Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)

Valeur par défaut : `1099` pour les serveurs de catalogue

Le port de service JMX peut être également défini à l'aide de la propriété de serveur JMXServicePort.

d. Facultatif : Définissez le port de connecteur JMX.

   Le port de connecteur JMX est utilisé pour la communication entre les clients JMX. Pour indiquer le port de connecteur JMX, utilisez l'option de ligne de commande suivante :

   -JMXConnectorPort <port>
   Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

   Le port de connecteur JMX peut être également défini à l'aide de la propriété de serveur JMXConnectorPort.

e. Définissez le port SSL (Secure Socket Layer).

   Lorsque la sécurité est activée, un port SSL (Secure Socket Layer) est aussi nécessaire. Pour indiquer le port SSL, utilisez l'option de ligne de commande suivante :

   ```
   -jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
   ```

```
./startOgServer.sh cs1 -listenerHost hostA -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

8.6+

```
./startXsServer.sh cs1 -listenerHost hostA -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

Figure 37. Exemple d'utilisation de ligne de commande. Démarrez le premier serveur de catalogues sur hostA. Exemple de commande :

Démarrez le second serveur de catalogues sur hostB. Exemple de commande :

```
./startOgServer.sh cs2 -listenerHost hostB -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

8.6+

```
./startXsServer.sh cs2 -listenerHost hostB -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

2. Démarrez les noeuds finals de serveur de conteneur.

   La commande suivante démarre une machine virtuelle Java de conteneur à utiliser avec l'exemple de service de catalogue :

   ```
   ./startOgServer.sh c0 -catalogServiceEndPoints hostA:2809,hostB:2809
   ```

   8.6+

   ```
   ./startXsServer.sh c0 -catalogServiceEndPoints hostA:2809,hostB:2809
   ```

   Le serveur de conteneur machines virtuelles Java utilise deux ports. Le port du gestionnaire haute disponibilité est utilisé pour les communications internes entre les serveurs de conteneur homologues et serveurs de catalogue. Le port d'écoute est utilisé pour les communications IIOP entre les serveurs de conteneur homologues, les serveurs de catalogue et les clients. L'hôte d'écoute est utilisé pour associer l'ORB à un adaptateur réseau spécifique. Si vous ne spécifiez rien, les deux ports sont dynamiquement sélectionnés. Toutefois, si vous souhaitez configurer les ports de manière explicite, comme dans un environnement de pare-feu, vous pouvez utiliser une option de ligne de commande pour spécifier le port ORB.

   a. Spécifiez le port d'écoute et l'hôte d'écoute. Pour indiquer le port d'écoute et l'hôte d'écoute, utilisez les options de ligne de commande suivantes :

      -listenerHost <nom d'hôte>

Indique le nom d'hôte auquel le transport ORB (Object Request Broker) ou eXtremeIO (XIO) se lie pour les communications. La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que le mécanisme de transport dans la machine JVM connaisse l'adresse IP de liaison. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

`-listenerPort <port>`
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

Le port d'écoute et l'hôte d'écoute peuvent être également définis à l'aide des propriétés listenerHost et listenerPort.

b. Spécifiez le port du gestionnaire haute disponibilité. Pour indiquer le port du gestionnaire haute disponibilité, utilisez l'option de ligne de commande suivante :

`-haManagerPort <port>`
Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, un port libre est choisi. Cette propriété est ignorée dans les environnements WebSphere Application Server.

Le port du gestionnaire haute disponibilité peut être également défini à l'aide de la propriété de serveur HAManagerPort.

c. Facultatif : Spécifiez le port SSL.

Lorsque la sécurité est activée, un port SSL (Secure Socket Layer) est aussi nécessaire. Pour indiquer le port SSL, utilisez l'option de ligne de commande suivante :

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
```

d. Facultatif : Spécifiez le port de service JMX.

`-JMXServicePort <port>`
Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **`-JMXServicePort`** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)

Valeur par défaut : `1099` pour les serveurs de catalogue

Le port de service JMX peut être également défini à l'aide de la propriété de serveur JMXServicePort.

e. Facultatif : Définissez le port de connecteur JMX.

Le port de connecteur JMX est utilisé pour la communication entre les clients JMX. Pour indiquer le port de connecteur JMX, utilisez l'option de ligne de commande suivante :

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

Le port de connecteur JMX peut être également défini à l'aide de la propriété de serveur JMXConnectorPort.

3. Démarrez les noeuds finals client.

   Les clients ont uniquement besoin de connaître les noeuds finals d'écoute de service de catalogue. Les clients extraient les points de contact des machines virtuelles Java du serveur de conteneur, qui sont les machines virtuelles Java qui conservent les données, automatiquement du service de catalogue. Pour vous connecter au service de catalogue dans l'exemple précédente, le client doit envoyer la liste suivante de paires `host:port` à l'API connexion :

   ```
   hostA:2809,hostB:2809
   ```

   Le client peut également recevoir des rappels des serveurs de conteneur lors de l'utilisation de l'API DataGrid. Ces rappels communiquent en utilisant IIOP avec le port d'écoute ORB. Pour indiquer le port et l'adaptateur réseau pour recevoir des rappels, définissez les propriétés **listenerHost** et **listenerPort** dans le fichier de propriétés du client.

   Lorsque la sécurité est activée, un port SSL (Secure Socket Layer) est aussi nécessaire. Pour indiquer le port SSL, utilisez la propriété système suivante lors du démarrage du processus client :

   ```
   -jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
   ```

Configuration de ports dans un environnement WebSphere Application Server

Java

Les services de catalogue WebSphere eXtreme Scale, les serveurs de conteneur et les clients, lorsqu'ils s'exécutent dans des processus WebSphere Application Server, utilisent des ports et des services déjà définis pour le processus.

Pourquoi et quand exécuter cette tâche

Les sections suivantes décrivent les informations relatives à l'utilisation de ports dans le déploiement.

1. Noeuds finals de service de catalogue

   Les services de catalogue WebSphere eXtreme Scale s'exécutent dans n'importe quel processus WebSphere Application Server et sont configurés en utilisant la console d'administration ou des tâches d'administration. Tous les ports sont hérités du processus, sauf pour le port client, qui est explicitement configuré. Pour plus d'informations sur les ports utilisés par le service de catalogue, voir «Planification des ports réseau», à la page 59. Pour plus d'informations sur la configuration d'un domaine de service de catalogue, voir Service de catalogue à haute disponibilité.

2. Noeuds finals du serveur de conteneur

   Les serveurs de conteneur WebSphere eXtreme Scale sont hébergés dans des modules Java EE. Les serveurs de conteneur utilisent les ports définis pour le processus du serveur d'applications. Pour plus d'informations sur les ports utilisés par le service de conteneur, voir «Planification des ports réseau», à la page 59

page 59. Pour plus d'informations sur le démarrage d'un conteneur dans un module Java EE, tel qu'un module Enterprise JavaBeans (EJB) ou un module Web, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur», à la page 333.

3. Noeuds finals du client

   Les clients WebSphere eXtreme Scale sont hébergés dans des modules Web Java EE ou EJB.

   Les clients se connectent à l'aide d'un programme au domaine de service de catalogue en utilisant l'API ObjectGridManager.connect(). Lorsque vous vous connectez à un domaine de service de catalogue hébergé dans la même cellule, la connexion client trouve automatiquement le domaine de service de catalogue par défaut en utilisant l'appel d'API suivant sur ObjectGridManager:

   ```
   connect(securityProps, overRideObjectGridXML)
   ```

   Si le domaine de service de catalogue par défaut est hébergé à distance (en dehors de la cellule), les noeuds finals de service de catalogue doivent être définis en utilisant la méthode suivante dans l'API ObjectGridManager :

   ```
   connect(catalogServerEndpoints, securityProps, overRideObjectGridXml)
   ```

   Si le domaine de service de catalogue par défaut est défini dans la cellule, l'API CatalogServerProperties peut être utilisée pour extraire les adresses de serveur de catalogue. La tâche d'administration XSDomainManagement peut également être utilisée pour extraire un domaine de service de catalogue configuré.

Serveurs avec plusieurs cartes réseau

Vous pouvez exécuter les processus eXtreme Scale sur un serveur doté de plusieurs cartes réseau.

Si un serveur ou un client est en cours d'exécution sur un serveur contenant plusieurs cartes réseau, vous devez spécifier le port réseau et le nom d'hôte dans votre configuration eXtreme Scale à lier à une carte spécifiée. Si cette configuration n'est pas spécifiée, l'environnement d'exécution eXtreme Scale en choisit automatiquement un envoi réseau et un nom d'hôte, ce qui peut entraîner des échecs de connexion ou ralentir les performances.

Lorsque vous définissez le nom d'hôte des processus eXtreme Scale intégrés dans WebSphere Application Server, il peut être nécessaire de tenir compte de WebSphere Application Server ou des autres produits de pile dans la configuration. Par exemple, voir Technote: La configuration de l'agent de noeud dans une carte NIC et de son serveur d'applications dans une autre carte dans un sous-réseau différent peut provoquer des erreurs ORB.

Pour les serveurs de catalogue ou de conteneur, vous devez définir l'hôte d'écoute et le port d'écoute de l'une des manières suivantes :

- Dans le fichier des propriétés du serveur.
- Avec le paramètre de ligne de commande dans le script **startOgServer** or **startXsServer**.

Pour les clients, vous ne pouvez pas utiliser la ligne de commande et vous devez utiliser les propriétés du client.

Configuration des transports

Les transports permettent l'échange d'objets et de données entre différents processus serveur dans votre configuration.

Pourquoi et quand exécuter cette tâche

8.6+ Il existe deux options pour les transports dans WebSphere eXtreme Scale : Object Request Broker (ORB) et IBM eXtremeIO (XIO).

ORB Lorsque vous utilisez l'ORB, les entrées de cache sont stockées dans le segment de mémoire Java. Généralement, le temps de réponse relatif est plus lent que XIO en raison de la récupération de place périodique qui se produit dans cet environnement.

Obsolète : 8.6+ Object Request Broker (ORB) est obsolète. Si vous n'utilisiez pas ORB dans une édition précédente, utilisez IBM eXtremeIO (XIO) pour le mécanisme de transport. Si vous utilisez ORB, migrez la configuration vers XIO.

XIO Lorsque vous utilisez XIO, les entrées de cache sont stockées dans la mémoire native, ou IBM eXtremeMemory. Comme les entrées du cache sont stockées dans la mémoire native, le temps de réponse est plus rapide qu'avec ORB.

Affichage du type de transport du domaine de service de catalogue

Vous pouvez afficher le type de transport qui est actuellement utilisé pour le domaine de service de catalogue.

Avant de commencer

Vous pouvez afficher les types de transports qui sont utilisés dans un domaine de service de catalogue autonome ou un domaine de service de catalogue en cours d'exécution dans WebSphere Application Server.

- Si vous utilisez un domaine de service de catalogue autonome, utilisez l'utilitaire **xscmd** pour afficher les informations de transport sur le domaine de service de catalogue. Pour plus d'informations sur la configuration de l'utilitaire **xscmd**, voir «Administration avec l'utilitaire **xscmd**», à la page 515.
- Si vous disposez d'un domaine de service de catalogue qui s'exécute dans WebSphere Application Server, vous pouvez utiliser l'utilitaire **wsadmin** pour afficher le type de transport. Pour plus d'informations sur l'utilitaire **wsadmin**, voir Démarrage du client de scriptage wsadmin en utilisant le scriptage wsadmin.

Procédure

- Affichez le type de transport d'un domaine de service de catalogue autonome. Dans l'utilitaire **xscmd**, exécutez la commande suivante :
  - UNIX `./xscmd.sh -c showTransport`
  - Windows `xscmd.bat -c showTransport`

  La commande affiche le type de transport. Les valeurs suivantes peuvent s'afficher : `eXtremeIO` ou `Object Request Broker`.
- Affichez le type de transport d'un domaine de service de catalogue qui s'exécute WebSphere Application Server.
  - Dans la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de service de catalogue** > *catalog_service_domain_name*. Vérifiez que **Activer IBM eXtremeIO (XIO) communication** est sélectionné.

– Dans l'utilitaire **wsadmin**, exécutez la commande suivante :
  - En langage Jacl :

    ```
    $AdminTask getTransport {-domainName TestDomain }
    ```

  - A l'aide de la chaîne Jython :

    ```
    AdminTask.getTransport ('[-domainName testDomain]')
    ```

    La commande affiche le type de transport. Les valeurs suivantes peuvent s'afficher : `eXtremeIO` ou `Object Request Broker`. Si vous exécutez cette commande sur un domaine de service de catalogue qui contient des serveurs distants ou que le catalogServerName est un serveur distant, une erreur se produit. Vous devez utiliser la commande **xscmd -c showTransport** pour les serveurs distants. Pour plus d'informations sur la commande **getTransport** dans l'utilitaire **wsadmin**, voir «Tâches d'administration des domaines de service de catalogue», à la page 308.

Configuration d'IBM eXtremeIO (XIO)

IBM eXtremeIO (XIO) est un mécanisme de transport qui remplace ORB (Object Request Broker).

Avant de commencer

- 8.6 Pour configurer XIO, tous les serveurs de conteneur et de catalogue doivent correspondre à l'édition Version 8.6. Pour plus d'informations, voir «Mise à jour des serveurs eXtreme Scale», à la page 255.

8.6+ Vous pouvez configurer XIO pour tous les serveurs de conteneur du domaine de service de catalogue en activant XIO dans les serveurs de catalogue. Les serveurs de conteneur reconnaissent le type de transport du serveur de catalogue et utilisent ce type de transport.

Procédure

8.6+ La façon dont vous activez XIO dépend du type des serveurs que vous utilisez :

- Activez XIO sur les serveurs de catalogue autonomes.

  XIO est activé par défaut lorsque vous démarrez le serveur de catalogue avec la commande **startXsServer**. Pour plus d'informations, voir «Démarrage des serveurs de conteneur qui utilisent le transport IBM eXtremeIO (XIO)», à la page 482.
- Activez XIO sur les serveurs qui s'exécutent dans WebSphere Application Server.

  Vous activez XIO dans le domaine des services de catalogue dans la console d'administration WebSphere Application Server. Cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de services de catalogue** > ***catalog_service_domain***. Sélectionnez **Activer la communication IBM eXtremeIO (XIO)**. Appliquez les modifications. Pour plus d'informations, voir «Configuration du service de catalogue dans WebSphere Application Server», à la page 306.
- Activez XIO sur les serveurs qui s'exécutent dans le Profil Liberty.

  Pour activer XIO dans un serveur Profil Liberty, affectez à l'attribut `transport` la valeur `XIO` dans le fichier `server.xml`. Par exemple, reportez-vous à la propriété sélectionnée dans l'exemple de code suivant :

  ```
  <featureManager>
    ...
    <feature>eXtremeScale.server-1.1</feature>
  ```

```
</featureManager>

<xsServer isCatalog="true" transport="XIO" listenerPort="2809" ... />
```

Avertissement : Le serveur doit être un serveur de catalogue et, par conséquent, `isCatalog` doit avoir la valeur true lorsque vous configurez XIO. Le paramètre `listenerPort` est facultatif, mais XIO peut reconnaître ce port si vous l'activez. Si vous n'activez pas XIO, ORB est utilisé sur ce port à la place.

Ensuite, exécutez la commande **start** pour démarrer les serveurs Profil Liberty. Pour plus d'informations, voir Démarrage et arrêt des serveurs dans le profil Liberty.

8.6+ Vous pouvez utiliser des arguments de ligne de commande et des propriétés de serveur pour configurer le comportement XIO :

- Facultatif : Mettez à jour le fichier de propriétés de chaque serveur de conteneur dans la configuration afin d'activer XIO. Après avoir déterminé les propriétés à définir, vous pouvez spécifier les valeurs dans le fichier de propriétés du serveur ou par programmation à l'aide de l'interface ServerProperties. Pour plus d'informations sur les propriétés que vous pouvez définir, voir «Optimisation d'IBM eXtremeIO (XIO)», à la page 608.

8.6+ Résultats

Les serveurs que vous avez configurés utilisent le transport XIO. Pour vérifier que la configuration est correcte, voir «Affichage du type de transport du domaine de service de catalogue», à la page 350.

Que faire ensuite

Vous pouvez également utiliser IBM eXtremeMemory pour éviter les pauses de récupération d'espace, ce qui permet stabiliser les performances et de bénéficier de temps de réponse plus prévisibles. Pour plus d'informations, voir «Configuration d'IBM eXtremeMemory», à la page 357.

Configuration d'ORB

Java

(Obsolète) ORB (Object Request Broker) est utilisé par WebSphere eXtreme Scale pour communiquer sur une pile TCP. Utilisez le fichier `orb.properties` pour transmettre les propriétés utilisées par l'ORB pour modifier le comportement du transport de la grille de données. Aucune action n'est requise pour utiliser l'ORB fourni par WebSphere eXtreme Scale ou WebSphere Application Server pour vos serveurs WebSphere eXtreme Scale.

8.6+ Pourquoi et quand exécuter cette tâche

Obsolète : 8.6+ Object Request Broker (ORB) est obsolète. Si vous n'utilisiez pas ORB dans une édition précédente, utilisez IBM eXtremeIO (XIO) pour le mécanisme de transport. Si vous utilisez ORB, migrez la configuration vers XIO.

Configuration de la fonction ORB (Object Request Broker) dans un environnement WebSphere Application Server

Java

(Obsolète) Vous pouvez utiliser WebSphere eXtreme Scale avec des applications qui utilisent Object Request Broker (ORB) directement dans l'environnement WebSphere Application Server ou WebSphere Application Server Network Deployment.

8.6+

Pourquoi et quand exécuter cette tâche

Obsolète : 8.6+ Object Request Broker (ORB) est obsolète. Si vous n'utilisiez pas ORB dans une édition précédente, utilisez IBM eXtremeIO (XIO) pour le mécanisme de transport. Si vous utilisez ORB, migrez la configuration vers XIO.

Procédure

1. Nommez vos serveurs d'applications de façon appropriée.

   Dans un environnement WebSphere Application Server, les serveurs ne peuvent pas porter le même nom lorsqu'ils utilisent ORB pour communiquer entre eux. Vous pouvez contourner cette restriction en spécifiant la propriété système **-Dcom.ibm.websphere.orb.uniqueServerName=true** pour les processus de même nom. Par exemple, lorsque des serveurs avec le nom `server1` sur chaque noeud sont utilisés comme domaine de service de catalogue ou plusieurs agents de noeud sont utilisés pour former un domaine de service de catalogue.
2. Optimisez les propriétés ORB dans la configuration WebSphere Application Server.

   Voir «Propriétés ORB», à la page 604 pour plus d'informations sur les propriétés que vous pouvez optimiser. En fonction de la propriété, vous pouvez modifier un paramètre dans la console d'administration ou dans le fichier *racine_was*`properties/orb.properties`.
3. Si vous utilisez plusieurs cartes d'interface réseau, vous devez définir la valeur ORB_LISTENER_ADDRESS dans le panneau des ports dans la console d'administration de WebSphere Application Server. Répétez cette étape pour chaque serveur d'applications dans la configuration.
   a. Pour un serveur d'applications, cliquez sur **Serveurs** > **Serveurs d'applications** > *server_name*. Dans Communications, cliquez sur **Ports**. Le panneau Ports du serveur spécifié s'affiche.
   b. Cliquez sur **Détails** et modifiez la valeur ORB_LISTENER_ADDRESS.
   c. Entrez l'adresse IP dans la zone **Hôte**. Cette valeur doit être une adresse privée pour un environnement à plusieurs interfaces réseau.

      Remarque : Les noms d'hôte DNS ne sont pas pris en charge pour la valeur ORB_LISTENER_ADDRESS.
   d. Entrez le numéro de port dans la zone **Port**. Le numéro de port définit le port pour lequel le service est configuré pour accepter les demandes des client. La valeur de port est utilisée avec le nom d'hôte.

Que faire ensuite

Vous pouvez utiliser l'outil **wxsLogAnalyzer** pour vérifier les paramètres de l'ORB dans votre environnement. Pour plus d'informations, voir «Analyse des journaux et des données de trace», à la page 675.

Configuration d'ORB (Object Request Broker) avec des processus autonomes WebSphere eXtreme Scale

Java

(Obsolète) Vous pouvez utiliser WebSphere eXtreme Scale avec des applications qui utilisent Object Request Broker (ORB) directement dans des environnements ne contenant pas WebSphere Application Server ou WebSphere Application Server Network Deployment.

Avant de commencer

Obsolète :  **8.6+** Object Request Broker (ORB) est obsolète. Si vous n'utilisiez pas ORB dans une édition précédente, utilisez IBM eXtremeIO (XIO) pour le mécanisme de transport. Si vous utilisez ORB, migrez la configuration vers XIO.

Si vous utilisez l'ORB dans le même processus que eXtreme Scale lorsque vous exécutez des applications, ou d'autres composants et infrastructures, non inclus avec eXtreme Scale, il se peut que vous deviez effectuer des tâches supplémentaires pour vous assurer que eXtreme Scale fonctionne correctement dans votre environnement.

Pourquoi et quand exécuter cette tâche

Ajoutez la propriété **`ObjectGridInitializer`** au fichier `orb.properties` pour initialiser l'utilisation de l'ORB dans votre environnement. Utilisez l'ORB pour activer la communication entre les processus eXtreme Scale et les autres processus de votre environnement.

Procédure

1. L'installation autonome n'inclut pas de fichier `orb.properties`. Vous devez placer le fichier `orb.properties` dans le répertoire `java/jre/lib`. Pour les descriptions des propriétés et des paramètres, voir «Propriétés ORB», à la page 604.
2. Dans le fichier `orb.properties`, tapez la ligne suivante et sauvegardez vos modifications :

```
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer
```

Résultats

eXtreme Scale initialise correctement l'ORB et coexiste avec d'autres applications pour lesquelles l'ORB est activé.

Pour utiliser une version personnalisée de l'ORB avec eXtreme Scale, voir «Configuration d'un ORB personnalisé».

Que faire ensuite

Vous pouvez utiliser l'outil **`xsLogAnalyzer`** pour vérifier les paramètres de l'ORB dans votre environnement. Pour plus d'informations, voir «Analyse des journaux et des données de trace», à la page 675.

Configuration d'un ORB personnalisé

(Obsolète)WebSphere eXtreme Scale utilise ORB (Object Request Broker) pour activer les communications entre les processus. Aucune action n'est requise pour utiliser pour vos serveurs WebSphere eXtreme Scale l'ORB fourni par WebSphere eXtreme Scale ou par WebSphere Application Server. L'utilisation des mêmes ORB pour vos clients WebSphere eXtreme Scale ne vous demandera guère plus. Si vous devez utiliser à la place d'un ORB personnalisé, l'ORB fourni avec IBM SDK est un

bon choix, même si vous le configurer. Il est également possible d'utiliser d'autres ORB fournis par d'autres constructeurs, là aussi moyennant quelque configuration.

Avant de commencer

Obsolète : 8.6+ Object Request Broker (ORB) est obsolète. Si vous n'utilisiez pas ORB dans une édition précédente, utilisez IBM eXtremeIO (XIO) pour le mécanisme de transport. Si vous utilisez ORB, migrez la configuration vers XIO.

Déterminez si vous utilisez l'ORB fourni avec WebSphere eXtreme Scale ou WebSphere Application Server, l'ORB fourni avec IBM SDK, ou un ORB de fournisseur externe.

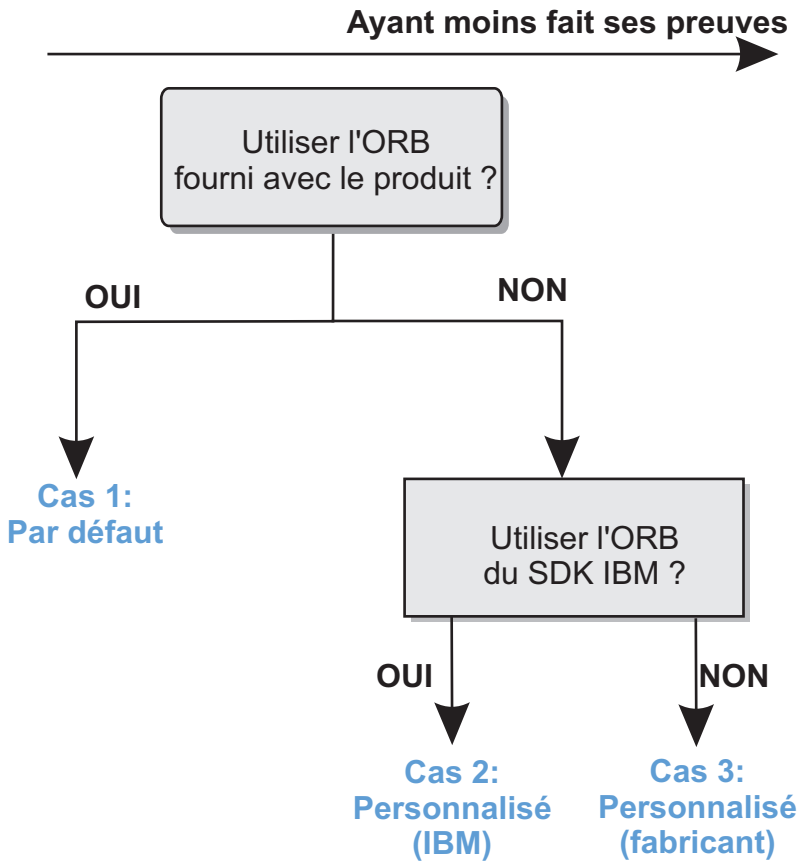


Figure 38. Choix de l'ORB

Vous n'êtes pas obligé de faire les mêmes choix pour les processus serveur WebSphere eXtreme Scale et les processus client WebSphere eXtreme Scale. eXtreme Scale prend en charge les kits de développeur de la plupart des fournisseurs, mais il est recommandé d'utiliser pour vos processus, tant serveur que client, l'ORB fourni avec eXtreme Scale. eXtreme Scale ne prend pas en charge l'ORB fourni avec le Java Development Kit (JDK) Oracle.

Pourquoi et quand exécuter cette tâche

Avant d'utiliser l'ORB que vous avez choisi, familiarisez-vous avec la configuration requise.

Cas n° 1 : ORB par défaut

- Pour vos processus serveur WebSphere eXtreme Scale, aucune configuration n'est requise pour utiliser l'ORB fourni avec WebSphere eXtreme Scale ou avec WebSphere Application Server.
- Pour vos processus client WebSphere eXtreme Scale, un minimum de configuration du chemin d'accès aux classes est requis pour pouvoir utiliser l'ORB fourni avec WebSphere eXtreme Scale ou avec WebSphere Application Server.

Cas n° 2 : ORB personnalisé (IBM)

Pour configurer vos processus client WebSphere eXtreme Scale afin qu'ils utilisent l'ORB fourni avec le IBM SDK, voyez les instructions plus bas. Vous pouvez utiliser l'ORB IBM, que vous utilisiez le IBM SDK ou un autre kit de développement. Vous pouvez utiliser IBM SDK Version 6 ou une version suivante.

Cas 3: ORB personnalisé (fourni par un fournisseur externe)

L'utilisation d'un ORB de fournisseur pour les processus client WebSphere eXtreme Scale est l'option qui a subi le moins de test. Avant de contacter le support technique IBM, vous devez vous assurer que les problèmes rencontrés en utilisant des ORB d'éditeurs de logiciels indépendants sont bien reproductibles avec l'ORB IBM ORB et un JRE compatible.

L'ORB fourni avec le Java Developement Kit (JDK) Oracle n'est pas pris en charge.

Procédure

- Configurez vos processus client pour qu'ils utilisent l'un des ORB par défaut (**Cas 1**). Utilisez l'argument JVM suivant :

```
-jvmArgs –Djava.endorsed.dirs=default_ORB_directory${pathSeparator}JRE_HOME/lib/endorsed
```

Le répertoire ORB par défaut est : *wxs_home*`/lib/endorsed`. Il peut être aussi nécessaire de mettre à jour les propriétés suivantes dans le fichier `orb.properties` :

```
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton
```

- Configurez les processus client ou serveur pour qu'ils utilisent IBM SDK Version 6 **(Cas 2)**.
  1. Copiez les fichiers de l'ORB d'archive Java (JAR) dans un répertoire vide, ou le répertoire *custom_ORB_directory*.
     - `ibmorb.jar`
     - `ibmorbapi.jar`
  2. Dans les scripts qui lancent la commande Java, spécifiez le répertoire *`custom_ORB_directory`* comme répertoire validé.

     Conseil : Si vos commandes Java spécifient déjà un répertoire validé, une autre option consiste à placer le répertoire *`custom_ORB_directory`* sous le répertoire validé existant. En plaçant le répertoire *`custom_ORB_directory`* sous le répertoire validé existant, vous n'avez pas à mettre à jour les scripts. Si vous décidez de mettre à jour les scripts, veillez à ajouter le répertoire *`custom_ORB_directory`* comme préfixe à l'argument existant `–Djava.endorsed.dirs=` au lieu de remplacer complètement l'argument.
     - Modifiez les scripts pour un environnement eXtreme Scale autonome.

Modifiez le chemin pour la variable *OBJECTGRID_ENDORSED_DIRS* dans le fichier `setupCmdLine.bat|sh` pour indiquer le répertoire *custom_ORB_directory*. Sauvegardez vos modifications.

– Modifiez les scripts lorsque'eXtreme Scale est imbriqué dans un environnement WebSphere Application Server.

  Ajoutez la propriété système et les paramètres suivants au script `startOgServer` :

  `-jvmArgs –Djava.endorsed.dirs=`*`répertoire_ORB_personnalisé`*

– Modifiez les scripts personnalisés qui vous servent à démarrer un processus d'application client ou un processus serveur.

  `-Djava.endorsed.dirs=`*`répertoire_ORB_personnalisé`*

Configuration d'IBM eXtremeMemory

En configurant eXtremeMemory, vous pouvez stocker des objets dans la mémoire native plutôt que dans le segment de mémoire Java. La configuration d'eXtremeMemory active le mécanisme de transport IBM eXtremeIO.

Avant de commencer

- Pour en savoir plus sur eXtremeMemory et sur la détermination de la quantité de mémoire maximale à utiliser pour eXtremeMemory, voir «Planification de l'utilisation d'IBM eXtremeMemory», à la page 62.

Pourquoi et quand exécuter cette tâche

Pour plus d'informations sur eXtremeMemory et ses avantages par rapport au segment de mémoire Java pour stocker les objets de grille de données, voir IBM eXtremeMemory. Lorsque vous utilisez eXtremeMemory, eXtremeIO est utilisé pour la communication entre les serveurs de conteneur. Les objets sont sérialisés en octets dans le serveur de conteneur. Pour activer eXtremeMemory, vous définissez les propriétés de serveur nécessaires sur tous les serveurs de conteneur dans la grille de données et redémarrez les serveurs.

eXtremeMemory n'est pas utilisé sur les serveurs de catalogue. Si vous disposez d'un serveur de catalogue et d'un serveur de conteneur regroupés, les serveurs de conteneur utilisent eXtremeMemory, mais le serveur de catalogue ne l'utilise pas.

Procédure

1. Configurez les bibliothèques natives en définissant les variables d'environnement appropriées. Ajoutez le répertoire *racine_install_wxs*`/ObjectGrid/native` au chemin d'accès aux classes.

   Vous pouvez définir la variable d'environnement des manières suivantes :

   - Linux Définissez la variable d'environnement *LD_LIBRARY_PATH* avant d'appeler le programme Java :

     ```
     LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:racine_install_wxs/ObjectGrid/native
     export LD_LIBRARY_PATH
     ```

   - Affectez à la propriété système *java.library.path* Java l'emplacement où se trouvent les bibliothèques natives :

     ```
     java -Djava.library.path=racine_install_wxs/ObjectGrid/native <other args>
     ```

2. Mettez à jour le fichier de propriétés de chaque serveur de conteneur dans la configuration afin d'activer eXtremeMemory. Pour activer eXtremeMemory, vous devez définir la propriété enableXM. Si vous ne souhaitez pas définir la valeur par défaut 25 % du système entier à utiliser pour eXtremeMemory, vous

devez également définir la propriété maxXMSize. Après avoir déterminé les propriétés à définir, vous pouvez définir les valeurs dans le fichier de propriétés du serveur ou par programmation à l'aide de l'interface ServerProperties.

Pour plus d'informations sur la détermination de la valeur maxXMSize à définir, voir «Planification de l'utilisation d'IBM eXtremeMemory», à la page 62.

Propriétés requises

enableXM
Lorsque la valeur est `true`, active IBM eXtremeMemory sur le serveur et configure le serveur pour utiliser IBM eXtremeIO pour la réplication synchrone et asynchrone. Les entrées de cache sont stockées dans la mémoire native et non pas dans le segment de mémoire Java. tous les serveurs de conteneur dans les grilles de données doivent utiliser la même valeur pour la propriété **`enableXM`**.

Valeur par défaut : `false`

Propriétés suggérées

maxXMSize
Définit la quantité maximale de mémoire, en mégaoctets, utilisée par le serveur pour le stockage eXtremeMemory.

Valeur par défaut : 25 % de la mémoire totale du système

3. Activez les serveurs de conteneur pour utiliser eXtremeMemory. Vous pouvez utiliser l'une des méthodes suivantes pour sélectionner les nouvelles valeurs de propriétés :
   - Placez un fichier nommé `objectGridServer.properties` dans le chemin d'accès aux classes. Pour plus d'informations, voir Fichier de propriétés du serveur.
   - Définissez les propriétés de l'application avec l'interface ServerProperties. Pour plus d'informations, voir Interface ServerProperties.
   - Démarrez un ensemble serveur OSGi. Pour plus d'informations, voir «Démarrage des serveurs eXtreme Scale en utilisant l'infrastructure OSGi Eclipse Equinox», à la page 527.
   - Redémarrez les serveurs de conteneur. Pour plus d'informations, voir «Démarrage des serveurs de conteneur qui utilisent le transport IBM eXtremeIO (XIO)», à la page 482 et «Configuration des serveurs de conteneur dans WebSphere Application Server», à la page 333.

Que faire ensuite

Vous pouvez également définir des propriétés pour configurer eXtremeIO. Pour plus d'informations, voir «Configuration d'IBM eXtremeIO (XIO)», à la page 351.

Configuration des clients

Vous pouvez configurer WebSphere eXtreme Scale pour l'exécuter dans un environnement autonome ou dans un environnement avec WebSphere Application Server. Pour qu'un déploiement WebSphere eXtreme Scale sélectionne les modifications de configuration dans la grille de serveurs, vous devez redémarrer les processus pour que ces modifications entrent en vigueur au lieu d'être appliquées de manière dynamique. Toutefois, côté client, vous ne pouvez pas modifier les paramètres de configuration d'une instance de client existante, mais vous pouvez créer une instance de client avec les paramètres nécessaires en

utilisant un fichier XML ou à l'aide d'un programme. Lorsque vous créez un client, vous pouvez remplacer les paramètres par défaut provenant de la configuration de serveur actuelle.

Vous pouvez configurer un client eXtreme Scale des différentes manières suivantes, dont chacune peut être effectuée par programmation ou à l'aide d'un fichier XML de substitution par le client :

- configuration XML
- configuration par programmation
- configuration Spring Framework
- désactivation du cache local

Remplacements client

Java

Vous pouvez configurer un client WebSphere eXtreme Scale en fonction de vos besoins en remplaçant les paramètres serveur. Vous pouvez remplacer plusieurs plug-ins et attributs.

Pour remplacer les paramètres sur un client, vous pouvez utiliser XML ou la configuration par programmation. Pour plus d'informations sur le remplacement des paramètres client, voir «Configuration des clients avec la configuration XML», à la page 360 et «Configuration des clients à l'aide d'un programme», à la page 361.

Vous pouvez remplacer les plug-in suivants sur un client :

- **BackingMap**
  - Evictor
  - MapEventListener
  - Plug-in BackingMapLifecycleListener
  - Plug-in MapSerializerPlugin
- **Attributs BackingMap**
  - attribut numberOfBuckets

    Obsolète : La propriété est obsolète. Utilisez l'attribut nearCacheEnabled pour activer le cache local.
  - attribut timeToLive
  - attribut ttlEvictorType
  - attribut evictionTriggers
  - 8.6+ attribut nearCacheEnabled
  - 8.6+ attribut nearCacheInvalidationEnabled
  - 8.6+ attribut nearCacheLastAccessTTLSyncEnabled
- **ObjectGrid**
  - TransactionCallback
  - ObjectGridEventListener
  - Plug-in ObjectGridLifecycleListener
- **attribut ObjectGrid**
  - attribut entityMetadataXMLFile
  - attribut txTimeout

– attribut txIsolation

Configuration des clients avec la configuration XML

Vous pouvez utiliser le fichier XML de configuration pour modifier les paramètres du client.

Pourquoi et quand exécuter cette tâche

Pour modifier les paramètres d'un client WebSphere eXtreme Scale, créez un fichier XML ObjectGrid dont la structure est similaire à celle du fichier utilisé pour le serveur de conteneur.

Pour la liste des modules d'extension et des attributs que vous pouvez remplacer sur le client, voir «Remplacements client», à la page 359.

Procédure

1. Créez un fichier XML de configuration ObjectGrid pour le client qui ait une structure semblable à celle du fichier pour le serveur de conteneur.

   Supposons que le fichier XML a été associé à un fichier XML de stratégie de déploiement et que ces fichier ont été utilisés pour démarrer un serveur de conteneur.

   companyGridServerSide.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="CompanyGrid">
            <bean id="TransactionCallback"
                className="com.company.MyTxCallback" />
            <bean id="ObjectGridEventListener"
                className="com.company.MyOgEventListener" />
            <backingMap name="Customer"
                pluginCollectionRef="customerPlugins" />
            <backingMap name="Item" />
            <backingMap name="OrderLine" nearCacheEnabled="true"
                timeToLive="1600" ttlEvictorType="LAST_ACCESS_TIME" />
            <backingMap name="Order" lockStrategy="PESSIMISTIC"
                pluginCollectionRef="orderPlugins" />
        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="customerPlugins">
            <bean id="Evictor"
                className="com.ibm.websphere.objectGrid.plugins.builtins.LRUEvictor" />
            <bean id="MapEventListener"
                className="com.company.MyMapEventListener" />
        </backingMapPluginCollection>
        <backingMapPluginCollection id="orderPlugins">
            <bean id="MapIndexPlugin"
                className="com.company.MyMapIndexPlugin" />
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

   Sur un serveur de conteneur, l'instance ObjectGrid nommée CompanyGrid se comporte conformément à ce qui est défini dans le fichier `companyGridServerSide.xml`. Par défaut, les paramètres du client CompanyGrid sont identiques à ceux de l'instance CompanyGrid qui s'exécute sur le serveur.

   Le fichier XML ObjectGrid suivant peut être utilisé pour définir certains attributs et plug-in du client CompanyGrid.

   `companyGridClientSide.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="CompanyGrid">
            <bean id="TransactionCallback"
                className="com.company.MyClientTxCallback" />
            <bean id="ObjectGridEventListener" className="" />
            <backingMap name="Customer" nearCacheEnabled="true"
                pluginCollectionRef="customerPlugins" />
            <backingMap name="Item" />
            <backingMap name="OrderLine" nearCacheEnabled="true"
                timeToLive="800" ttlEvictorType="LAST_ACCESS_TIME" />
            <backingMap name="Order" lockStrategy="PESSIMISTIC"
                pluginCollectionRef="orderPlugins" />
        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="customerPlugins">
            <bean id="Evictor"
                className="com.ibm.websphere.objectGrid.plugins.builtins.LRUEvictor" />
            <bean id="MapEventListener" className="" />
        </backingMapPluginCollection>
        <backingMapPluginCollection id="orderPlugins">
            <bean id="MapIndexPlugin"
                className="com.company.MyMapIndexPlugin" />
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Le fichier XML spécifie les remplacements suivants :

- Le bean TransactionCallback sur le client est com.company.MyClientTxCallback au lieu du paramètre serveur com.company.MyTxCallback.
- Le client n'est associé à aucun plug-in ObjectGridEventListener car la valeur className est la chaîne vide.
- Le client active un cache local pour Customer backingMap, conserve son plug-in Evictor et supprime le plug-in MapEventListener.
- L'attribut timeToLive d'OrderLine backingMap a changé.
- Bien qu'un attribut lockStrategy différent ait été indiqué, les conséquences sont nulles car cet attribut n'est pas pris en charge pour un remplacement par le client.

2. Créez le client avec le fichier XML.

   Pour créer le client CompanyGrid à l'aide du fichier `companyGridClientSide.xml`, transmettez le fichier XML ObjectGrid sous la forme d'une URL à l'une des méthodes de connexion dans l'interface ObjectGridManager :

```
ObjectGridManager ogManager =
 ObjectGridManagerFactory.ObjectGridManager();
ClientClusterContext clientClusterContext =
 ogManager.connect("MyServer1.company.com:2809", null, new URL(
                "file:xml/companyGridClientSide.xml"));
```

Configuration des clients à l'aide d'un programme

Java

Vous pouvez remplacer les paramètres à l'aide d'un programme. Créez un objet ObjectGridConfiguration dont la structure est semblable à celle de l'instance ObjectGrid côté serveur.

Pourquoi et quand exécuter cette tâche

L'exemple de code suivant crée les mêmes substitutions que celles décrites dans «Configuration des clients avec la configuration XML», à la page 360.

Pour la liste des modules d'extension et des attributs que vous pouvez remplacer sur le client, voir «Remplacements client», à la page 359.

Procédure

Le code suivant crée une instance ObjectGrid côté client.

```
ObjectGridConfiguration companyGridConfig = ObjectGridConfigFactory
    .createObjectGridConfiguration("CompanyGrid");
Plugin txCallbackPlugin = ObjectGridConfigFactory.createPlugin(
    PluginType.TRANSACTION_CALLBACK, "com.company.MyClientTxCallback");
companyGridConfig.addPlugin(txCallbackPlugin);

Plugin ogEventListenerPlugin = ObjectGridConfigFactory.createPlugin(
    PluginType.OBJECTGRID_EVENT_LISTENER, "");
companyGridConfig.addPlugin(ogEventListenerPlugin);

BackingMapConfiguration customerMapConfig = ObjectGridConfigFactory
    .createBackingMapConfiguration("Customer");
customerMapConfig.setNumberOfBuckets(1429);
Plugin evictorPlugin = ObjectGridConfigFactory.createPlugin(PluginType.EVICTOR,
    "com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor");
customerMapConfig.addPlugin(evictorPlugin);

companyGridConfig.addBackingMapConfiguration(customerMapConfig);

BackingMapConfiguration orderLineMapConfig = ObjectGridConfigFactory
    .createBackingMapConfiguration("OrderLine");
orderLineMapConfig.setNumberOfBuckets(701);
orderLineMapConfig.setTimeToLive(800);
orderLineMapConfig.setTtlEvictorType(TTLType.LAST_ACCESS_TIME);

companyGridConfig.addBackingMapConfiguration(orderLineMapConfig);

List ogConfigs = new ArrayList();
ogConfigs.add(companyGridConfig);

Map overrideMap = new HashMap();
overrideMap.put(CatalogServerProperties.DEFAULT_DOMAIN, ogConfigs);

ogManager.setOverrideObjectGridConfigurations(overrideMap);
ClientClusterContext client = ogManager.connect(catalogServerEndpoints, null, null);
ObjectGrid companyGrid = ogManager.getObjectGrid(client, objectGridName);
```

L'instance ogManager de l'interface ObjectGridManager recherche uniquement les remplacements dans les objets ObjectGridConfiguration et BackingMapConfiguration que vous incluez dans la mappe overrideMap. Par exemple, le code précédent remplace le nombre de compartiments de la mappe OrderLine. La mappe Order reste cependant inchangée côté client car aucune configuration de cette mappe n'est incluse.

Configuration du cache local

Lorsqu'eXtreme Scale est utilisé dans le cadre d'une topologie répartie, les clients peuvent éventuellement disposer d'un cache local en ligne. L'on appelle cache local ce cache facultatif. Il s'agit d'un ObjectGrid indépendant, présent sur chaque client et faisant office de cache du cache distant côté serveur. Le cache local est activé par défaut lorsque le verrouillage est désactivé ou configuré comme optimiste et ne peut pas être utilisé lorsqu'il est défini comme pessimiste.

Pourquoi et quand exécuter cette tâche

Un cache local est rapide, car il offre un accès en mémoire à un sous-ensemble de l'ensemble des données en cache stockées à distance. Pour plus d'informations sur les caches locaux, voir «Cache réparti», à la page 25.

Vous pouvez modifier les paramètres nécessaires dans le fichier XML ObjectGrid du serveur de conteneur. Les paramètres dans ce fichier s'appliquent à tous les clients si vous ne les modifiez pas. Vous pouvez remplacer le paramètre **nearCacheEnabled** du cache local avec un fichier XML ou par programmation. Pour plus d'informations sur le remplacement des paramètres client, voir «Configuration des clients avec la configuration XML», à la page 360 et «Configuration des clients à l'aide d'un programme», à la page 361.

Procédure

1. Le cache local est activé si vous utilisez les paramètres par défaut. Pour activer le cache local, vous devez définir l'attribut **lockStrategy** en lui affectant la valeur `NONE` or `OPTIMISTIC` dans le fichier XML de descripteur ObjectGrid des serveurs de conteneur. La valeur par défaut est `OPTIMISTIC`. Pour plus d'informations sur l'attribut **lockStrategy**, voir Fichier XML du descripteur d'ObjectGrid. Les clients ne contiennent pas de cache local lorsque le paramètre de verrouillage a la valeur `PESSIMISTIC`.
2. 8.6+ Pour activer ou désactiver le cache local, définissez l'attribut **nearCacheEnabled** dans le fichier XML de descripteur d'ObjectGrid.

   8.6+ **nearCacheEnabled**
   Définissez la valeur `true` pour activer le cache local client. Pour utiliser un cache local, vous devez affecter à l'attribut **lockStrategy** la valeur `NONE` ou `OPTIMISTIC`.

   Valeur par défaut : `true` (facultatif)

   Important : Dans les versions précédentes, vous activiez et désactiviez le cache local avec l'attribut **numberOfBuckets** dans le fichier XML de descripteur d'ObjectGrid. Si cette valeur est `0`, le cache local est désactivé. Ce paramètre remplace l'attribut **nearCacheEnabled**. Si vous ne définissez pas **numberOfBuckets** ou que vous lui affectez une valeur différente de zéro, l'attribut **nearCacheEnabled** détermine si le cache local est activé.
3. Redémarrez les serveurs de conteneur et les clients. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479 et «Configuration des serveurs de conteneur dans WebSphere Application Server», à la page 333.

Résultats

Pour vérifier qu'un cache local est activé, exécutez la méthode BackingMap.isNearCacheEnabled() dans le client. Vous pouvez également rechercher le message CWOBJ1128I dans les fichiers journaux sur le client pour déterminer si le cache local est activé.

Configuration du cache local du cache dynamique

Vous pouvez activer le cache local dans le fournisseur de cache dynamique pour qu'une grille d'objets OjectGrid indépendante existe dans le fournisseur de cache dynamique faisant office de cache rapide pour le cache secondaire distant.

Pourquoi et quand exécuter cette tâche

Modifiez la mappe de sauvegarde backingMap du fichier `objectgrid.xml` pour activer le cache local dans le cache dynamique.

Procédure

1. Ajoutez les propriétés suivantes à la mappe de sauvegarde backingMap du fichier `objectgrid.xml` pour activer le cache local dans le fournisseur de cache dynamique : un cache local de cache dynamique doit contenir les propriétés **`nearCacheEnabled`**, **`nearCacheInvalidationEnabled`**, et **`nearCacheLastAccessTTLSyncEnabled`** affectées de la valeur `true`. Par exemple :

   ```
   <backingMap name="IBM_DC_PARTITIONED_.*" template="true" readOnly="false"
   < pluginCollectionRef="all" preloadMode="false" nearCacheEnabled="true"
   nearCacheInvalidationEnabled="true" nearCacheLastAccessTTLSyncEnabled= "true"
   lockStrategy="OPTIMISTIC"
   copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="NONE"
   nullValuesSupported="false" />
   ```

   Chacun des attributs de l'élément backingMap a les définitions suivantes :

   8.6+ `nearCacheEnabled`
   Définissez la valeur `true` pour activer le cache local client. Pour utiliser un cache local, vous devez affecter à l'attribut **`lockStrategy`** la valeur `NONE` ou `OPTIMISTIC`.

   Valeur par défaut : `true` (facultatif)

   8.6+ `nearCacheInvalidationEnabled`
   Définissez la valeur `true` pour activer la suppression des données obsolètes du cache local aussi vite que possible. Lorsqu'une mise à jour, une suppression ou une invalidation est exécutée sur la grille de données distante, une invalidation asynchrone est déclenchée dans le cache local. Etant donné que l'invalidation est asynchrone, les applications client peuvent accéder à des données périmées pour un court laps de temps après une mise à jour avant que la valeur périmée soit supprimée du cache local. Pour utiliser l'invalidation de cache local, affectez à l'attribut **`lockStrategy`** la valeur `NONE` ou `OPTIMISTIC`.

   Valeur par défaut : `false` (facultatif)

   8.6+ `nearCacheLastAccessTTLSyncEnabled`
   Définissez la valeur `true` pour synchroniser les informations de durée de vie avec la grille de données distante. Le type d'expulseur TTL (durée de vie) `LAST_ACCESS_TIME` doit être activé lorsque vous activez cette propriété.

   Valeur par défaut : `false` (facultatif)

2. Facultatif : Configurez la stratégie de verrouillage optimiste pour les mappes qui prennent en charge les applications qui utilisent le cache dynamique.

   La stratégie de verrouillage OPTIMISTIC est le mode par défaut. Utilisez le verrouillage optimiste lorsque les données sont modifiées rarement. Les verrous sont uniquement maintenus pendant un laps de temps limité tandis que les données sont lues depuis le cache et copiées dans la transaction. Lorsque le cache de transaction est synchronisé avec le cache principal, tous les objets mis en cache qui ont été mis à jour sont vérifiés avec la version d'origine. Si la vérification échoue, la transaction est annulée et une exception OptimisticCollisionException est provoquée.

   - Utilisation de la méthode setLockStrategy à l'aide d'un programme :

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
 ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("optimisticMap");
bm.setLockStrategy( LockStrategy.OPTIMISTIC );
```

- Utilisation de l'attribut lockStrategy dans Fichier XML du descripteur d'ObjectGrid :

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="test">
            <backingMap name="optimisticMap"
                lockStrategy="OPTIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

Configuration de l'invalidation du cache local

Java

Vous pouvez configurer l'invalidation du cache local pour supprimer les données périmées du cache local aussi rapidement que possible. Lorsqu'une mise à jour, une suppression ou une invalidation est exécutée dans la grille de données distante, une invalidation asynchrone est déclenchée dans le cache local. Ce mécanisme est plus rapide que l'utilisation de l'expulsion TTL (durée de vie) dans le cache local.

Avant de commencer

- Vous devez utiliser IBM eXtremeIO. Pour plus d'informations, voir «Configuration d'IBM eXtremeIO (XIO)», à la page 351.
- Vous devez utiliser un cache local. Pour vérifier qu'un cache local est activé, exécutez la méthode BackingMap.isNearCacheEnabled() dans le client. Pour plus d'informations sur la configuration du cache local, voir «Configuration du cache local», à la page 362.
- Si vous utilisez l'authentification, vous devez disposer des droits d'exécution de l'agent. Vous devez également disposer d'une autorisation de lecture sur le cache. Pour plus d'informations, voir Programmation d'autorisations client.

Pourquoi et quand exécuter cette tâche

L'activation de l'invalidation dans le cache local fournit un ensemble plus précis de données de la grille de données distante, car le cache local est mis à jour lorsque les données distantes sont modifiées.

Vous pouvez modifier les paramètres nécessaires dans le fichier XML ObjectGrid du serveur de conteneur. Les paramètres dans ce fichier s'appliquent à tous les clients si vous ne les modifiez pas. Si vous activez l'invalidation de cache local sur le serveur, vous pouvez la désactiver sur le client en remplaçant l'attribut **nearCacheInvalidationEnabled** du cache local par un fichier XML ou par programmation. Toutefois, vous ne pouvez pas remplacer l'attribut pour activer l'invalidation de cache local lorsqu'elle est désactivée sur le serveur. Pour plus

d'informations sur le remplacement des paramètres client, voir «Configuration des clients avec la configuration XML», à la page 360 et «Configuration des clients à l'aide d'un programme», à la page 361.

Procédure

1. Définissez l'attribut **nearCacheInvalidationEnabled** dans le fichier XML de descripteur ObjectGrid. Définissez cet attribut dans l'élément backingMap. Pour plus d'informations, voir Fichier XML du descripteur d'ObjectGrid.

   8.6+ **nearCacheInvalidationEnabled**
   Définissez la valeur `true` pour activer la suppression des données obsolètes du cache local aussi vite que possible. Lorsqu'une mise à jour, une suppression ou une invalidation est exécutée sur la grille de données distante, une invalidation asynchrone est déclenchée dans le cache local. Etant donné que l'invalidation est asynchrone, les applications client peuvent accéder à des données périmées pour un court laps de temps après une mise à jour avant que la valeur périmée soit supprimée du cache local. Pour utiliser l'invalidation de cache local, affectez à l'attribut **lockStrategy** la valeur `NONE` ou `OPTIMISTIC`.

   Valeur par défaut : `false` (facultatif)
2. Redémarrez le serveurs de conteneur et les clients. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479 et «Configuration des serveurs de conteneur dans WebSphere Application Server», à la page 333.

Configuration de la synchronisation du client JMS (Java Message Service)

Vous pouvez utiliser la synchronisation du client JMS pour maintenir le cache local client synchronisé avec les autres serveurs et clients.

Cache local

Vous pouvez utiliser la classe intégrée Java Message Service (JMS) com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener pour activer le mécanisme d'invalidation du client dans un environnement réparti eXtreme Scale.

Le mécanisme d'invalidation de client permet de résoudre les problèmes liés à la présence de données obsolètes dans le cache local du client en environnement eXtreme Scale réparti. Ce mécanisme vérifie que la mémoire est synchronisée avec les serveurs ou les autres clients. Toutefois, même lorsque ce mécanisme existe, la mise à jour de la mémoire n'est pas immédiate. Lorsque l'environnement d'exécution publie des mises à jour, un délai est généré.

Il existe deux modèles pour le mécanisme d'invalidation de client dans un environnement eXtreme Scale réparti :

- Modèle client-serveur : tous les processus serveur ont un rôle de diffuseur de publications qui publie toutes les modifications transactionnelles vers la destination JMS désignée. Tous les processus client ont un rôle de récepteur : ils reçoivent toutes les modifications transactionnelles à partir de la destination JMS désignée.
- Modèle client ayant les deux rôles : aucune interaction n'existe entre les processus serveur et la destination JMS. Tous les processus client ont le rôle de

diffuseur de publications JMS et de récepteur. Les modifications transactionnelles effectuées sur le client sont publiées vers la destination JMS et tous les clients les reçoivent.

Pour plus d'informations, voir «Programme d'écoute d'événement JMS», à la page 283.

8.6+ Si vous ne voulez pas utiliser JMS pour synchroniser le cache local, vous pouvez utiliser l'invalidation de cache local. Pour plus d'informations, voir «Configuration de l'invalidation du cache local», à la page 365.

Modèle client-serveur

Dans un modèle client-serveur, les serveurs ont le rôle de diffuseur de publications JMS et le client a le rôle de récepteur JMS.

Exemple XML de modèle client-serveur

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="AgentObjectGrid">
      <bean id="ObjectGridEventListener"
        className="com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener">
        <property name="invalidationModel" type="java.lang.String" value="CLIENT_SERVER_MODEL" description="" />
        <property name="invalidationStrategy" type="java.lang.String" value="PUSH" description="" />
        <property name="mapsToPublish" type="java.lang.String" value="agent;profile;pessimisticMap" description="" />
        <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String" value="defaultTCF" description="" />
        <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_userid" type="java.lang.String" value="" description="" />
        <property name="jms_password" type="java.lang.String" value="" description="" />
        <property name="jndi_properties" type="java.lang.String"
          value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;
     java.naming.provider.url=
    tcp://localhost:61616;connectionFactoryNames=defaultTCF;topic.defaultTopic=defaultTopic"
          description="jndi properties" />
      </bean>

      <backingMap name="agent" readOnly="false" pluginCollectionRef="agent" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="28800" />
      <backingMap name="profile" readOnly="false" pluginCollectionRef="profile" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="pessimisticMap" readOnly="false" pluginCollectionRef="pessimisticMap" preloadMode="false"
        lockStrategy="PESSIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap1" readOnly="false" pluginCollectionRef="excludedMap1" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap2" readOnly="false" pluginCollectionRef="excludedMap2" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="agent">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.AgentObjectTransformer" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="profile">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.ProfileObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
        <property name="maxSize" type="int" value="2000" description="set max size for LRU evictor" />
        <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
        <property name="numberOfLRUQueues" type="int" value="50" description="set number of LRU queues" />
      </bean>
    </backingMapPluginCollection>

    <backingMapPluginCollection id="pessimisticMap" />
    <backingMapPluginCollection id="excludedMap1" />
    <backingMapPluginCollection id="excludedMap2" />
  </backingMapPluginCollections>

</objectGridConfig>
```

Modèle client ayant les deux rôles

Dans ce modèle, chaque client a le rôle de diffuseur de publications JMS et de récepteur. Le client publie chaque modification transactionnelle validée vers une destination JMS désignée et reçoit toutes les modifications des autres clients. Aucune interaction n'a lieu entre le serveur et JMS.

Exemple XML de modèle où le client a les deux rôles

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="AgentObjectGrid">
      <bean id="ObjectGridEventListener"
        className="com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener">
        <property name="invalidationModel" type="java.lang.String" value="CLIENT_AS_DUAL_ROLES_MODEL" description="" />
        <property name="invalidationStrategy" type="java.lang.String" value="PUSH" description="" />
        <property name="mapsToPublish" type="java.lang.String" value="agent;profile;pessimisticMap" description="" />
        <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String" value="defaultTCF" description="" />
        <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_userid" type="java.lang.String" value="" description="" />
        <property name="jms_password" type="java.lang.String" value="" description="" />
        <property name="jndi_properties" type="java.lang.String"
          value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url=
    tcp://localhost:61616;connectionFactoryNames=defaultTCF;topic.defaultTopic=defaultTopic"
          description="jndi properties" />
      </bean>

      <backingMap name="agent" readOnly="false" pluginCollectionRef="agent" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="28800" />
      <backingMap name="profile" readOnly="false" pluginCollectionRef="profile" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="pessimisticMap" readOnly="false" pluginCollectionRef="pessimisticMap" preloadMode="false"
        lockStrategy="PESSIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap1" readOnly="false" pluginCollectionRef="excludedMap1" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
      <backingMap name="excludedMap2" readOnly="false" pluginCollectionRef="excludedMap2" preloadMode="false"
        lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
        timeToLive="2700" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="agent">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.AgentObjectTransformer" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="profile">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.ProfileObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
        <property name="maxSize" type="int" value="2000" description="set max size for LRU evictor" />
        <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
        <property name="numberOfLRUQueues" type="int" value="50" description="set number of LRU queues" />
      </bean>
    </backingMapPluginCollection>

    <backingMapPluginCollection id="pessimisticMap" />
    <backingMapPluginCollection id="excludedMap1" />
    <backingMapPluginCollection id="excludedMap2" />
  </backingMapPluginCollections>

</objectGridConfig>
```

Définition des valeurs de délai d'attente de nouvelles tentatives de demande

Avec des mappes fiables, vous pouvez fournir une valeur de délai d'attente en millisecondes à WebSphere eXtreme Scale pour les demandes de transaction.

Pourquoi et quand exécuter cette tâche

Vous pouvez définir la valeur du délai d'attente dans le fichier des propriétés du client ou dans une session. La valeur de session remplace la valeur dans les propriétés du client. Si la valeur définie est supérieure à zéro, la demande est

réessayée jusqu'à ce que le dépassement du délai d'attente soit atteint ou qu'une erreur permanente se produise. Une erreur permanente peut être une exception `DuplicateKeyException`. La valeur zéro indique le mode fail-fast et eXtreme Scale ne retente pas la transaction, quelle que soit son type.

Pendant l'exécution, la valeur de délai d'expiration des transactions est utilisée avec la valeur de délai d'attente de nouvelle tentative, ce qui garantit que le délai d'attente de relance ne dépasse pas le délai d'expiration des transactions.

Deux types de transactions existent : les transactions de validation automatique et les transactions qui utilisent des méthodes explicites begin et commit. Les exceptions valides de tentatives diffèrent de ces deux types de transactions :

- Pour les transactions appelées au sein d'une session, les transactions sont retentées pour les exceptions CORBA SystemException et eXtreme Scale TargetNotAvailable.
- Pour les transactions de validation automatique, les transactions sont retentées pour les exceptions de disponibilité CORBA eXtreme Scale. Ces exceptions incluent les exceptions ReplicationVotedToRollbackTransactionException, TargetNotAvailable et AvailabilityException.

Les erreurs d'application et les autres erreurs permanentes s'exécutent immédiatement et le client ne retente pas la transaction. Ces erreurs permanentes incluent les exceptions DuplicateKeyException et KeyNotFoundException. Utilisez le paramètre fail-fast pour exécuter toutes les exceptions sans retenter les transactions.

Exceptions pour lesquels le client retente la transaction :

- ReplicationVotedToRollbackTransactionException (uniquement en validation automatique)
- TargetNotAvailable
- org.omg.CORBA.SystemException
- AvailabilityException (uniquement en validation automatique)
- LockTimeoutException (uniquement en validation automatique)
- UnavailableServiceException (uniquement en validation automatique)

Exceptions permanentes pour lesquelles la transaction n'est pas retentée :

- DuplicateKeyException
- KeyNotFoundException
- LoaderException
- TransactionAffinityException
- LockDeadlockException
- OptimisticCollisionException

Procédure

- Définissez la valeur du délai d'attente dans un fichier de propriétés client.

  Pour définir la valeur de requestRetryTimeout dans un client, ajoutez ou modifiez la propriété dans le Fichier de propriétés du client. Les propriétés du client sont définies par défaut dans le fichier `objectGridClient.properties`. La propriété requestRetryTimeout est définie en millisecondes. Une valeur supérieure à zéro indique que la demande doit être réessayée en cas de survenue d'exceptions pour lesquelles le retry est possible. Une valeur de `0` indique que les échecs ne donnent lieu à aucune nouvelle tentative dans les

exceptions. Pour utiliser le comportement par défaut, supprimez la propriété ou donnez-lui une valeur de -1. Exemple de valeur dans le fichier objectGridClient.properties :

```
requestRetryTimeout = 30000
```

La valeur de requestRetryTimeout est spécifiée en millisecondes. Dans l'exemple, si la valeur est utilisée dans une instance ObjectGrid, la valeur de requestRetryTimeout sera de 30 secondes.

- Définissez la valeur du délai d'attente à l'aide d'un programme.

  Pour définir par programmation les propriétés du client, commencez par créer un fichier de propriétés dans un <emplacement> approprié à votre application. Dans l'exemple suivant, le fichier des propriétés du client correspond au fragment de code objectGridClient.properties de la section précédente. Après vous être connecté à une instance ObjectGridManager, définissez les propriétés du client, comme indiqué. Ensuite, lorsque vous avez une instance ObjectGrid, l'instance a les propriétés client que vous avez définies dans le fichier. A chaque fois que vous serez amené à modifier ce fichier, vous devrez explicitement obtenir une nouvelle instance ObjectGrid.

  ```
  ObjectGridManager manager = ObjectGridManagerFactory.getObjectGridManager();
  String objectGridName = "testObjectGrid";
  URL clientXML = null;
  ClientClusterContext ccc = manager.connect("localhost:2809", null, clientXML);
  File file = new File("<location>/objectGridClient.properties");
  URL url = file.toURI().toURL();
  ccc.setClientProperties(objectGridName, url);
  ObjectGrid objectGrid = ogManager.getObjectGrid(ccc, objectGridName);
  ```

- Définissez le fichier de remplacement pendant une validation de session.

  Pour définir dans l'objet Session pendant combien de temps effectuer de nouvelles tentatives ou pour remplacer la propriété client requestRetryTimeout, appelez la méthode setRequestRetryTimeout(long) dans l'interface Session.

  ```
  Session sessionA = objectGrid.getSession();
  sessionA.setRequestRetryTimeout(30000);
  ObjectMap mapA = sessionA.getMap("payroll");
  String key = "key:" + j;
  mapA.insert(key, "valueA");
  ```

  Cette session utilise à présent la valeur 30 000 millisecondes (30 secondes) pour requestRetryTimeout, quelle que soit la valeur définie dans le fichier des propriétés du client. Pour plus d'informations sur l'interface de session, voir Utilisation des sessions pour accéder aux données de la grille.

Configuration de fabriques de connexions eXtreme Scale

Java

Une fabrique de connexions eXtreme Scale permet aux applications Java EE de se connecter à une grille de données WebSphere eXtreme Scale distante. Utilisez des propriétés personnalisées pour configurer des adaptateurs de ressources.

Avant de commencer

Pour pouvoir créer des fabriques de connexions, vous devez installer l'adaptateur de ressources.

Pourquoi et quand exécuter cette tâche

Une fois l'adaptateur de ressources installé, vous pouvez créer une ou plusieurs fabriques de connexions d'adaptateur de ressources qui représentent des connexions client eXtreme Scale aux grilles de données distantes. Pour configurer

une fabrique de connexions d'adaptateur de ressources et l'utiliser dans une application, effectuez les étapes ci-dessous.

Il est possible de créer une fabrique de connexions eXtreme Scale sur la portée du noeud pour des adaptateurs de ressources autonomes ou dans l'application pour les adaptateurs de ressources intégrés. Pour plus d'informations sur la façon de créer des fabriques de connexions dans WebSphere Application Server, voir les rubriques connexes.

Procédure

1. Utilisez la console d'administration WebSphere Application Server pour créer une fabrique de connexions eXtreme Scale qui représente une connexion client eXtreme Scale. Reportez-vous à la rubrique Configuration des fabriques de connexions Java EE Connector sur la console d'administration. Une fois que vous avez précisé les propriétés de la fabrique de connexions dans le panneau des propriétés générales, vous devez cliquer sur **Appliquer** pour que le lien des propriétés personnalisées devienne actif.
2. Cliquez sur **Propriétés personnalisées** dans la console d'administration. Définissez les propriétés personnalisées suivantes pour configurer la connexion client à la grille de données distante.

Tableau 27. Propriétés personnalisées pour la configuration de fabriques de connexions

Nom de la propriété	Type	Description
ConnectionName	String (chaîne)	Facultatif) Nom de la connexion client eXtreme Scale. ConnectionName permet d'identifier la connexion lorsqu'elle est exposée en tant que bean géré. Cette propriété est facultative. Si elle n'est pas spécifiée, la propriété ConnectionName n'est pas définie.
CatalogServiceEndpoints	String (chaîne)	(Facultatif) Noeuds finaux de domaine de services de catalogue dans le format : `<host>:<port>[,<host><port>]`. Pour plus d'informations, voir «Paramètres du domaine de service de catalogue», à la page 323. Cette propriété est obligatoire si le domaine de service de catalogue n'est pas défini.
CatalogServiceDomain	String (chaîne)	(Facultatif) Nom du domaine de services de catalogue défini dans WebSphere Application Server. Pour plus d'informations, voir «Configuration des domaines de serveur de catalogue et de service de catalogue», à la page 304. Cette propriété est obligatoire si la propriété CatalogServiceEndpoints n'est pas définie.
ObjectGridName	String (chaîne)	(Facultatif) Nom de la grille de données à laquelle est connectée cette fabrique de connexions. S'il n'est pas spécifié, l'application doit fournir le nom lors de l'établissement de la connexion à partir de la fabrique de connexions.
ObjectGridURL	String (chaîne)	(Facultatif) URL du fichier XML de remplacement de la grille de données client. Cette propriété n'est pas valide si la propriété ObjectGridResource est également spécifiée. Pour plus d'informations, voir «Configuration des clients», à la page 358.
ObjectGridResource	String (chaîne)	Chemin d'accès aux ressources du fichier XML de remplacement de la grille de données client. Cette propriété est facultative et non valide si la propriété ObjectGridURL est également spécifiée. Pour plus d'informations, voir «Configuration des clients», à la page 358.
ClientPropertiesURL	String (chaîne)	(Facultatif) URL du fichier de propriétés client. Cette propriété n'est pas valide si la propriété ClientPropertiesResource est également spécifiée. Pour plus d'informations, voir la rubrique Fichier de propriétés du client.
ClientPropertiesResource	String (chaîne)	(Facultatif) Chemin d'accès aux ressources du fichier de propriétés client. Cette propriété n'est pas valide si la propriété ClientPropertiesURL est également spécifiée. Pour plus d'informations, voir la rubrique Fichier de propriétés du client.

WebSphere Application Server permet également d'utiliser d'autres options de configuration pour régler les pools de connexions et gérer la sécurité. Pour plus d'informations sur les liens aux rubriques du centre de documentation WebSphere Application Server, voir les informations connexes.

Que faire ensuite

Créez une référence de fabrique de connexions eXtreme Scale dans l'application. Pour plus d'informations, voir «Configuration d'applications pour une connexion à eXtreme Scale», à la page 373.

Configuration d'environnements Eclipse pour une utilisation de fabriques de connexions eXtreme Scale

Java

L'adaptateur de ressources eXtreme Scale comporte des fabriques de connexions personnalisées. Pour utiliser ces interfaces dans vos applications eXtreme Scale Java Platform, Enterprise Edition (Java EE), vous devez importer le fichier `wxsra.rar` dans votre espace de travail et le lier à votre projet d'application.

Avant de commencer

- Vous devez installer Rational Application Developer version 7 ou ultérieure ou Eclipse Java EE IDE for Web Developers version 1.4 ou ultérieure.
- Un environnement d'exécution de serveur doit être configuré.

Procédure

1. Importez le fichier `wxsra.rar` dans votre projet en sélectionnant **Fichier** > **Importer**. La fenêtre d'importation s'affiche.
2. Sélectionnez **Java EE** > **Fichier RAR**. La fenêtre d'importation de connecteur s'affiche.
3. Pour spécifier le fichier de connecteur, cliquez sur **Parcourir** pour rechercher le fichier `wxsra.rar`. Le fichier `wxsra.rar` est installé lorsque vous installez un adaptateur de ressources. Vous pouvez rechercher le fichier archive d'adaptateur de ressources (RAR) dans l'emplacement suivant :
   - Pour les installations WebSphere Application Server : *racine_install_wxs*`/optionalLibraries/ObjectGrid`
   - Pour les installations autonomes : *racine_install_wxs*`/ObjectGrid/lib directory`
4. Créez un nom pour le nouveau projet de connecteur dans la zone **Projet de connecteur**. Vous pouvez utiliser `wxsra`, qui est le nom par défaut.
5. Choisissez une exécution cible faisant référence à un environnement d'exécution de serveur Java EE.
6. Vous pouvez éventuellement sélectionner l'option d'**ajout du projet à EAR** pour incorporer le fichier RAR à un projet EAR existant.

Résultats

Le fichier RAR est importé dans l'espace de travail Eclipse Eclipse.

Que faire ensuite

Vous pouvez référencer le projet RAR à partir de vos autres projets Java EE en procédant comme suit :

1. Cliquez avec le bouton droit de la souris sur le projet, puis cliquez sur **Propriétés**.
2. D=Sélectionnez **Chemin de génération Java**.
3. Sélectionnez l'onglet Projets.
4. Cliquez sur **Ajouter**.
5. Sélectionnez le projet de connecteur **wxsra** et cliquez sur **OK**.
6. Cliquez à nouveau sur **OK** pour fermer la fenêtre Propriétés.

Les classes de l'adaptateur de ressources eXtreme Scale figurent à présent dans le chemin de classes. Pour installer le produit fichiers JAR d'exécution à partir de la console Eclipse, voir Configuration d'un environnement de développement autonome dans Eclipse.

Configuration d'applications pour une connexion à eXtreme Scale

Les applications utilisent une fabrique de connexions eXtreme Scale pour créer des descripteurs de connexions vers une connexion client eXtreme Scale. Vous pouvez configurer les références de la fabrique de connexions de l'adaptateur de ressources à l'aide de cette tâche.

Avant de commencer

Créez un composant d'application Java Platform, Enterprise Edition (Java EE), tel qu'un conteneur ou un servlet Enterprise JavaBeans (EJB).

Procédure

Créez une référence de ressource javax.resource.cci.ConnectionFactory dans le composant d'application. Les références de ressource sont déclarées dans le descripteur de déploiement par le fournisseur d'application. La fabrique de connexions représente une connexion client eXtreme Scale qui peut être utilisée pour communiquer avec une ou plusieurs grilles de données nommées disponibles dans le domaine de service de catalogue.

Configuration de l'intégration du cache

WebSphere eXtreme Scale peut s'intégrer aux autres produits de mise en cache. Vous pouvez aussi utiliser le fournisseur de cache dynamique WebSphere eXtreme Scale pour connecter WebSphere eXtreme Scale au composant de cache dynamique WebSphere Application Server. Autre extension de WebSphere Application Server : le gestionnaire de sessions HTTP WebSphere eXtreme Scale, qui permet la mise en cache des sessions HTTP.

Configuration de gestionnaires de sessions HTTP

Le gestionnaire de sessions HTTP offre des fonctions de réplication de sessions pour une application associée. Le gestionnaire de sessions fonctionne avec le conteneur Web pour créer et gérer les cycles de vie des sessions HTTP qui sont associés à l'application.

Configuration du gestionnaire de sessions HTTP avec WebSphere Application Server

Java

Alors que WebSphere Application Server offre une fonction de gestion de session, les performances se dégradent alors que le nombre de demandes augmente. WebSphere eXtreme Scale est livré avec une implémentation de la gestion des sessions qui fournit la réplication de sessions, la haute disponibilité, une meilleure évolutivité et des options de configuration plus robustes.

Avant de commencer

- WebSphere eXtreme Scale doit être installé dans votre cellule WebSphere Application Server ou WebSphere Application Server Network Deployment pour pouvoir utiliser le gestionnaire de sessions d'eXtreme Scale. Pour plus

d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226.

- Lorsque vous utilisez WebSphere eXtreme Scale pour la réplication de session HTTP sur WebSphere Application Server, le paramètre Autoriser la gestion des sessions de dépassement doit être vérifié pour chaque application Web applicable et le serveur d'applications hébergeant cette application Web. Pour plus d'informations, voir Paramètres de gestion de session.
- La sécurité globale doit être activée dans la console d'administration de WebSphere Application Server si les serveurs de catalogue de votre domaine de service de catalogue ont SSL activé ou si vous voulez utiliser SSL pour un domaine de service de catalogue avec SSL pris en charge. Vous activer SSL pour un serveur de catalogue en affectant à l'attribut la valeur `SSL-Required` dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.

Pourquoi et quand exécuter cette tâche

Le gestionnaire de sessions HTTP de WebSphere eXtreme Scale prend en charge les serveurs imbriqués et éloignés pour la mise en cache.

- **Scénario de serveurs imbriqués**

  Dans ce scénario, les serveurs WebSphere eXtreme Scale sont regroupés dans les processus où les servlets sont exécutés. Le gestionnaire de sessions peut communiquer directement avec l'instance ObjectGrid locale, pour éviter les retards coûteux du réseau.

  Si vous utilisez WebSphere Application Server, placez dans les répertoires `META-INF` de vos fichiers d'archive Web (WAR) les fichiers *rép_base_wxs*`/session/samples/objectGrid.xml` et *rép_base_wxs*`/session/samples/objectGridDeployment.xml` fournis. eXtreme Scale détecte automatiquement ces fichiers au démarrage de l'application et démarre automatiquement les conteneurs eXtreme Scale dans le même processus que le gestionnaire de sessions.

  Vous pouvez modifier le fichier `objectGridDeployment.xml` suivant que vous souhaitiez utiliser une réplication synchrone ou asynchrone et en fonction du nombre de fragments réplique à configurer.
- **Scénario de serveurs éloignés**

  Dans les scénarios de serveurs distants, les serveurs de conteneur s'exécutent dans des processus différents que les servlets. Le gestionnaire de sessions communique avec un serveur de conteneur distant. Pour pouvoir utiliser un serveur distant connecté à un réseau, le gestionnaire de sessions doit être configuré avec les noms d'hôte et les numéros de port du domaine de service de catalogue. Le gestionnaire de sessions utilise ensuite une connexion client eXtreme Scale pour communiquer avec le serveur de catalogues et avec les serveurs de conteneur.

  Si les serveurs de conteneur sont démarrés dans des processus autonomes indépendants, démarrez les conteneurs eXtreme Scale avec les fichiers `objectGridStandAlone.xml` et `objectGridDeploymentStandAlone.xml` fournis dans le répertoire des exemples du gestionnaire de sessions.

Procédure

1. Raccordez votre application de sorte qu'elle puisse utiliser le gestionnaire de sessions. Pour utiliser le gestionnaire de sessions, vous devez ajouter les déclarations de filtre appropriées aux descripteurs de déploiement Web de l'application. En outre, les paramètres de configuration du gestionnaire de

sessions sont transmis au gestionnaire de sessions sous la forme de paramètres d'initialisation du contexte de servlet dans les descripteurs de déploiement. Vous pouvez introduire ces informations dans votre application de différentes manières :

- **Raccord automatique avec WebSphere Application Server**

  Vous pouvez configurer votre application pour qu'elle utilise un gestionnaire WebSphere eXtreme Scale de sessions HTTP lorsque vous installez votre application. Vous pouvez également modifier la configuration de l'application ou du serveur pour qu'ils utilisent le gestionnaire WebSphere eXtreme Scale de sessions HTTP. Pour plus d'informations, voir «Fractionnement automatique des applications pour la gestion de session HTTP dans WebSphere Application Server», à la page 377.

- **Raccord automatique de l'application avec des propriétés personnalisées**

  Vous n'avez pas besoin de raccorder manuellement vos applications lorsqu'elles s'exécutent dans WebSphere Application Server ou dans WebSphere Application Server Network Deployment.

  Ajoutez une propriété personnalisée à une cellule ou à un serveur pour définir à cette étendue le fichier `splicer.properties` pour toutes les applications Web. Utilisez les étapes suivantes pour configurer la propriété personnalisée :

  a. Dans la console d'administration de WebSphere Application Server, accédez au chemin correct de l'endroit où vous voulez définir la propriété personnalisée pour indiquer l'emplacement du fichier `splicer.properties`.
     - Pour définir la propriété personnalisée pour toutes les applications ou pour une application spécifique, cliquez sur **Administration du système** > **Cellule** > **Propriétés personnalisées**.
     - Pour définir la propriété personnalisée à appliquer à toutes les applications sur un serveur d'applications spécifique, cliquez sur **Serveur d'applications** > ***<nom_serveur>*** > **Administration** > **Propriétés personnalisées**. Le nom de propriété est `com.ibm.websphere.xs.sessionFilterProps` et sa valeur se trouve dans le fichier `splicer.properties` dont a besoin votre application. Exemple de chemin d'emplacement d'un fichier : `/opt/splicer.properties`.
  b. Ajoutez la propriété personnalisée `com.ibm.websphere.xs.sessionFilterProps`. La valeur de cette propriété personnalisée indique l'emplacement du fichier `splicer.properties` à éditer. Le fichier existe dans le gestionnaire_déploiement. Si vous voulez indiquer le fichier `splicer.properties` pour une application spécifique à l'aide d'une propriété personnalisée au niveau de la cellule, entrez le nom de la propriété personnalisée comme `<application_name>,com.ibm.websphere.xs.sessionFilterProps`, où *application_name* est le nom de l'application pour laquelle vous voulez appliquer la propriété personnalisée.

  Important : Vérifiez que le fichier mis à jour `splicer.properties` se trouve dans le même chemin sur tous les noeuds contenant un serveur d'applications hébergeant l'application ou les applications qui sont à raccorder pour la réplication de session.

  Les portées de cellule, de serveur et d'application sont les seules portées disponibles lors d'une exécution dans un gestionnaire de déploiement. Si vous avez besoin d'une autre portée, raccordez manuellement vos applications Web.

A faire : Notez aussi que le raccordement automatique ne fonctionne que si tous les noeuds exécutant l'application contiennent le fichier `splicer.properties` dans le même chemin. Dans le cas d'environnements mixtes contenant des noeuds Windows et UNIX, cette manière de procéder n'est pas possible et vous devez raccorder manuellement l'application.

- **Raccorder l'application avec le script `addObjectGridFilter`**

  Utilisez un script de ligne de commande fourni avec eXtreme Scale pour raccorder une application avec des déclarations de filtre et une configuration sous forme de paramètres d'initialisation de contexte de servlet. Pour un déploiement WebSphere Application Server, ce script se trouve dans `<was_home>/optionalLibraries/ObjectGrid/session/bin/addObjectGridFilter.bat/sh`. Pour un déploiement autonome, le script se trouve dans `WXS_HOME/ObjectGrid/session/bin/addObjectGridFilter.sh/bat`. Le script **`oddObjectGridFilter`** utilise deux paramètres :

  – Application : chemin absolu du fichier archive d'entreprise à raccorder
  – Chemin absolu du fichier de propriétés du raccordeur qui contient des propriétés de configuration.

  Voici le format du script :

  Windows

  ```
  addObjectGridFilter.bat [ear_file] [splicer_properties_file]
  ```

  UNIX

  ```
  addObjectGridFilter.sh [ear_file] [splicer_properties_file]
  ```

  UNIX **Exemple en utilisant eXtreme Scale installé sur WebSphere Application Server sur UNIX :**

  a. `cd rép_base_wxs/optionalLibraries/ObjectGrid/session/bin`
  b. `addObjectGridFilter.sh /tmp/mySessionTest.ear racine_was/optionalLibraries/ObjectGrid/session/samples/splicer.properties`

  UNIX **Exemple en utilisant eXtreme Scale installé dans un répertoire autonome sur UNIX :**

  a. `cd racine_was/session/bin`
  b. `addObjectGridFilter.sh /tmp/mySessionTest.ear racine_was/session/samples/splicer.properties`

  Le filtre de servlet qui est raccordé conserve les valeurs de configuration par défaut. Vous pouvez remplacer ces valeurs par défaut par des options de configuration que vous spécifiez dans le fichier de propriétés, dans le second argument. Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 395.

  Vous pouvez modifier et utiliser l'exemple de fichier `splicer.properties` fourni avec l'installation d'eXtreme Scale. Vous pouvez également utiliser le script **`addObjectGridServlets`**, qui insère le gestionnaire de sessions en étendant chaque servlet. Mais le script recommandé est le script **`addObjectGridFilter`**.

- **Raccorder manuellement l'application avec le script de génération Ant**

  WebSphere eXtreme Scale est fourni avec un fichier `build.xml` qui peut être utilisé par Apache Ant, qui est inclus dans le dossier *racine_was*`/bin` d'une installation WebSphere Application Server. Vous pouvez modifier le fichier `build.xml` pour changer les propriétés de configuration du gestionnaire de sessions. Les propriétés de configuration sont identiques aux noms de propriété dans le fichier `splicer.properties`. Pour modifier le fichier `build.xml`, appelez le processus Ant en exécutant la commande suivante :

– UNIX `ant.sh, ws_ant.sh`

– Windows `ant.bat, ws_ant.bat`

(UNIX) ou (Windows).

- **Mettre à jour manuellement le descripteur Web**

  Editez le fichier `web.xml` intégré à l'application Web pour incorporer la déclaration de filtre, son mappage de servlet et les paramètres d'initialisation du contexte de servlet. N'utilisez pas cette méthode car elle est source d'erreurs possibles.

Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 395.

2. Déployez l'application. Déployez l' application à l'aide de votre procédure normale pour un serveur ou un cluster. Une fois que vous avez déployé l'application, vous pouvez la démarrer.
3. Accédez à l'application. Vous pouvez maintenant accéder à l'application, qui interagit avec le gestionnaire de sessions et WebSphere eXtreme Scale.

Que faire ensuite

Vous pouvez modifier la majorité des attributs de configuration du gestionnaire de sessions lorsque vous instrumentez votre application pour utiliser le gestionnaire de sessions. Ces attributs sont : réplication synchrone ou asynchrone, taille de la table de session en mémoire, etc. En dehors des attributs modifiables lors de l'instrumentation de l'application, les seuls autres attributs de configuration que vous pouvez modifier après le déploiement de l'application sont ceux liés à la topologie des clusters de serveurs WebSphere eXtreme Scale et à la manière dont leurs clients (gestionnaires de sessions) s'y connectent.

Comportement de scénarios distant : si la grille de données complète qui héberge les données de sessions d'application est inaccessible à partir du client du conteneur Web, le client utilise le conteneur Web de base dans WebSphere Application Server pour la gestion des sessions. La grille de données peut être inaccessible dans les scénarios suivants :

- Problème de réseau entre le conteneur Web et les serveurs de conteneur distants.
- Arrêt des processus serveur de conteneur distant

Le nombre de références de session conservées en mémoire, spécifié par le paramètre **`sessionTableSize`** , est toujours maintenu lorsque les sessions sont stockées dans le conteneur Web de base. Les sessions les moins utilisées sont invalidés à partir du cache de session du conteneur Web lorsque la valeur **`sessionTableSize`** est dépassée. Si la grille de données distante devient disponible, les sessions ayant été invalidées à partir du cache du conteneur Web peuvent extraire les données de la grille de données distante et charger les données dans une nouvelle session. Si l'ensemble de la grille de données distante n'est pas disponible et que la session est invalidée dans le cache de session, les données de session utilisateur sont perdues. Compte tenu de ce problème, n'arrêtez pas l'ensemble de la grille de données distante de production lorsque le système est chargé.

Fractionnement automatique des applications pour la gestion de session HTTP dans WebSphere Application Server : Java

Vous pouvez configurer votre application WebSphere Application Server pour qu'elle conserve les sessions vers une grille de données. Cette grille de données

peut être un serveur de conteneur intégré qui s'exécute au sein de WebSphere Application Server. Il peut s'agir également d'une grille de données distantes.

Avant de commencer

Pour pouvoir changer la configuration dans WebSphere Application Server, vous devez avoir :

- le nom de la grille de données de session que vous voulez utiliser. Voir «Configuration du gestionnaire de sessions HTTP avec WebSphere Application Server», à la page 373 pour savoir comment créer une grille de données de session
- si le service de catalogue que vous voulez utiliser pour gérer vos sessions se trouve hors de la cellule dans laquelle vous installez votre application de sessions, vous devez créer un domaine de service de catalogue. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307.
- Si vous configurez un domaine de service de catalogue, il peut être nécessaire d'activer la sécurité du client sur le domaine de service de catalogue si les serveurs de conteneur requièrent une authentification. Ces paramètres indiquent à l'environnement d'exécution l'implémentation à utiliser CredentialGenerator. Cette implémentation génère des données d'identification à envoyer à la grille de données distante. Pour plus d'informations sur la configuration de ces paramètres, voir «Configuration de la sécurité client dans un domaine de service de catalogue», à la page 646.
- activé la sécurité global dans la console d'administration WebSphere Application Server si vous voulez prendre en charge l'un des scénarios suivants :
  - Activer SSL (Secure Socket Layer) pour les serveurs de catalogue dans le domaine de service de catalogue.
  - Utiliser SSL pour le domaine de service de catalogue avec SSL pris en charge.

  Vous spécifiez l'utilisation de SSL pour un serveur de catalogue en affectant à l'attribut **transportType** la valeur `SSL-Required` dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.
- Si vous utilisez la version 7.1.0.3 ou une version suivante, vous pouvez rendre persistantes les sessions qui utilisent la réécriture d'URL ou les cookies comme suivi de session dans la grille de données. Pour les versions antérieures à la version 7.1.0.3, vous ne pouvez pas rendre persistantes les sessions utilisant la réécriture d'URL comme mécanisme de suivi de session. Pour activer la persistance des sessions qui utilisent la réécriture d'URL, affectez à la propriété **useURLEncoding** la valeur `true` dans le fichier `splicer.properties`.
- Lorsque vous raccordez automatiquement des applications pour la gestion de session HTTP dans WebSphere Application Server, tous les serveurs d'applications qui hébergent l'application Web ont la propriété personnalisée de conteneur Web **HttpSessionIdReuse** affectée de la valeur `true`. Cette propriété permet aux sessions qui ont basculé d'un serveur d'applications vers un autre ou qui ont été invalidées depuis le cache de session interne dans un scénario distant, de conserver leur ID de session dans les demandes. Si vous ne voulez pas conserver ce comportement, affectez à la propriété personnalisée de conteneur Web la valeur `false` sur tous les serveurs d'applications applicables avant de configurer la gestion des sessions des applications. Pour plus d'informations sur cette propriété personnalisée, voir «Traitement des problèmes d'intégration du cache», à la page 682.

Procédure

- **Pour configurer la gestion de session lors de l'installation de l'application, effectuez la procédure suivante :**
  1. Dans la console d'administration de WebSphere Application Server, cliquez sur **Applications** > **Nouvelle application** > **Nouvelle application d'entreprise**. Sélectionnez le chemin **Détaillé** pour la création de l'application, puis effectuez les premières étapes de l'assistant.
  2. A l'étape **Paramètres de gestion des sessions eXtreme Scale session** de l'assistant, configurez la grille de données que vous voulez utiliser. Vous avez le choix entre deux types de grille : **Grille de données distante eXtreme Scale** ou **Grille de données imbriquée eXtreme Scale**.
     - Pour l'option **Grille de données distante eXtreme Scale**, choisissez le domaine de service de catalogue qui gère la grille de données des sessions et choisissez une grille de données dans la liste des grilles de données de sessions actives.
     - Pour l'option **Grille de données imbriquée eXtreme Scale**, vous pouvez soit chosir la configuration ObjectGrid par défaut, soit spécifier l'emplacement des fichiers de configuration ObjectGrid.
  3. Terminez l'installation de l'application en effectuant la procédure de l'assistant.

  Vous pouvez également installer l'application à l'aide d'un script wsadmin. Dans l'exemple suivant, le paramètre**-SessionManagement** crée une configuration identique à celle que vous pouvez créer dans la console d'administration :

  Pour la configuration d'une grille de données eXtreme Scale distante :

```
AdminApp.install('C:/A.ear', '[  -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement cs0:!:grid0]]
-MapWebModToVH [[MicroWebApp microwebapp.war,WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdg1app.war,WEB-INF/web.xml
default_host] [MicroDG2App microdg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

  Pour une grille eXtreme Scale imbriquée, avec la configuration par défaut :

```
AdminApp.install('C:/A.ear', '[  -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement  :!: :!:default]]  -MapWebModToVH [[MicroWebApp microwebapp.war,
WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdg1app.war,WEB-INF/web.xml
default_host] [MicroDG2App microdg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

  Pour une grille eXtreme Scale imbriquée, avec une configuration personnalisée :

```
AdminApp.install('C:/A.ear', '[  -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement  :!: :!:custom:!:c:\XS\objectgrid.xml:!:c:\XS\objectgriddeployment.xml]]
```

```
-MapWebModToVH [[MicroWebApp microwebapp.war,WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdg1app.war,WEB-INF/web.xml
default_host] [MicroDG2App microdg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

- **Pour configurer la gestion de session sur une application existante dans la console d'administration de WebSphere Application Server :**
  1. Dans la console d'administration WebSphere Application Server, cliquez sur **Applications** > **Types d'application** > **Applications d'entreprise WebSphere** > *application_name* > **Propriétés du module Web** > **Gestion des sessions** > **Paramètres de gestion des sessions eXtreme Scale**.
  2. Mettez à jour les zones pour activer la persistance de session dans la grille de données.

  Vous pouvez également mettre à jour l'application à l'aide d'un script wsadmin. Dans l'exemple suivant, le paramètre**-SessionManagement** crée une configuration identique à celle que vous pouvez créer dans la console d'administration :

  Pour la configuration de la grille de données eXtreme Scale distante :

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSRemoteSessionManagement cs0:!:grid0]]]')
  ```

  Les caractères **:!:** envoyés sont utilisés comme délimiteurs. Les valeurs envoyées sont les suivantes :

  catalogServiceName:!:*gridName*

  Pour le scénario intégré eXtreme Scale avec la configuration par défaut :

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSEmbeddedSessionManagement :!:!:!:default]]]')
  ```

  Les caractères **:!:** envoyés sont utilisés comme délimiteurs. Les valeurs envoyées sont les suivantes :

  catalogServiceName:!:*gridName*:!:default:!:
  absolutePath_to_objectGridXmlfile:!:*absolutePath_to_DeploymentXmlfile*

  Pour le scénario intégré eXtreme Scale avec la configuration personnalisée :

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSEmbeddedSessionManagement
  :!:!:!:custom:!:c:\XS\objectgrid.xml:!:c:\XS\objectgriddeployment.xml]]]')
  ```

  Les caractères **:!:** envoyés sont utilisés comme délimiteurs. Les valeurs envoyées sont les suivantes :

  catalogServiceName:!:*gridName*:!:custom:!:
  absolutePath_to_objectGridXmlfile:!:*absolutePath_to_DeploymentXmlfile*

  Lorsque vous enregistrez les modifications, l'application utilise la grille de données configurée pour la persistance des sessions sur le dispositif.

- **Pour configurer la gestion de session sur un serveur existant :**
  1. Dans la console d'administration WebSphere Application Server, cliquez sur **Serveurs** > **types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Gestion des sessions** > **Paramètres de gestion des sessions eXtreme Scale**.
  2. Actualisez les zones pour activer la persistance des sessions.

  Les commandes wsadmin suivantes vous permettent de configurer également la gestion des sessions sur un serveur existant :

  Pour la configuration d'une grille de données eXtreme Scale distante :

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSRemoteSessionManagement -XSRemoteSessionManagement
[-catalogService cs0 -csGridName grid0]]')
```

Pour une configuration de grille eXtreme Scale imbriquée :

– La configuration par défaut, si vous utilisez les fichiers XML par défaut :

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement [-embeddedGridType default -objectGridXML -objectGridDeploymentXML ]]')
```

– La configuration personnalisée, si vous utilisez des fichiers XML personnalisés :

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement
[-embeddedGridType custom -objectGridXML c:\XS\objectgrid.xml -objectGridDeploymentXML
c:\XS\objectgriddeployment.xml]]')
```

Lorsque vous enregistrez les modifications, le serveur utilise la grille de données configurée pour la persistance de sessions avec toutes les applications qu'il exécute.

- Si vous souhaitez modifier d'autres aspects de la configuration des sessions HTTP, vous pouvez éditer le fichier `splicer.properties`. Vous pouvez obtenir l'emplacement du chemin du fichier `splicer.properties` en recherchant la propriété personnalisée **`sessionFilterProps`**. Si vous avez configuré la persistance de session au niveau du serveur, le nom de la propriété personnalisée est `com.ibm.websphere.xs.sessionFilterProps`. Si vous l'avez configurée au niveau de l'application, elle s'appelle *<application_name>*`,com.ibm.websphere.xs.sessionFilterProps`. Ces propriétés personnalisées peuvent se trouver dans l'un des emplacements suivants :
  – Dans un environnement WebSphere Application Server Network Deployment : le fichier `splicer.properties` se trouve dans le chemin du profil du gestionnaire de déploiement.
  – Dans un environnement WebSphere Application Server autonome : une propriété personnalisée sur le serveur d'applications

  Vous pouvez ouvrir le fichier indiqué, effectuez les modifications et synchroniser les noeuds pour propager le fichier mis à jour des propriétés vers les autres noeuds de la configuration. Tous les noeuds de serveur d'applications nécessitent que le fichier `splicer.properties` se trouve dans le chemin défini pour que les sessions persistent.

  Avertissement : Si vous souhaitez activer la persistance pour les sessions qui utilisent la réécriture d'URL, affectez à la propriété **`useURLEncoding`** la valeur `true` dans le fichier `splicer.properties`.

  Pour plus d'informations sur les propriétés dans le fichier `splicer.properties`, voir «Fichier `splicer.properties`», à la page 397.

Résultats

Vous avez configuré le gestionnaire de sessions HTTP pour que les sessions soient conservées vers une grille de données. Les entrées sont supprimées de la grille de données lorsque les sessions expirent. Voir Paramètres de gestion des sessions pour plus d'informations sur la mise à jour la valeur de temporisation des sessions dans la console d'administration WebSphere Application Server.

Paramètres de gestion des sessions eXtreme Scale :

Vous pouvez configurer les applications WebSphere Application Server pour utiliser WebSphere eXtreme Scale ou un WebSphere DataPower XC10 Appliance pour la persistance de session.

Vous pouvez éditer ces paramètres dans l'assistant d'installation des applications d'entreprise ou dans les pages d'information de l'application ou du serveur :

- Version 7.0 : **Applications** > **Nouvelle application** > **Nouvelle applications d'enterprise** et choisissez le chemin détaillé pour créer l'application.
- Version 7.0 : **Applications** > **Types d'application** > **Applications d'entreprise WebSphere** > *application_name* > **Propriétés du module Web** > **Gestion de sessionmanagement** > **Paramètres de gestion de session**
- Version 7.0 : **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Paramètres de conteneur** > **Paramètres de gestion de session**

Activer la gestion des sessions :

Permet à la gestion des sessions d'utiliser WebSphere eXtreme Scale imbriquée ou une grille de données distante ou un WebSphere DataPower XC10 Appliance pour la persistance de session.

Gestion de la persistance des sessions par :

Indique comment la persistance des sessions est gérée. Vous pouvez sélectionner l'une des options suivantes :

- WebSphere DataPower XC10 Appliance
- Grille de données distante eXtreme Scale
- Grille de données imbriquée eXtreme Scale

Les autres paramètres que vous configurez varient en fonction du mécanisme de persistance des sessions sélectionné.

Paramètres WebSphere DataPower XC10 Appliance :

Les paramètres suivants sont spécifiques de la configuration de WebSphere DataPower XC10 Appliance pour la persistance de session.

Adresse IP ou nom d'hôte de WebSphere DataPower XC10 Appliance :

Indique l'adresse IP ou le nom d'hôte de l'appliance à utiliser pour stocker les données des sessions.

Information d'identification administrative IBM WebSphere DataPower XC10 Appliance :

Indiquez le **nom d'utilisateur** et le **mot de passe** que vous utilisez pour vous connecter à l'interface utilisateur de DataPower XC10 Appliance. Cliquez sur **Tester la connexion...** pour tester la connexion à l'appliance.

Préférences de persistance des sessions :

Indique la grille de données dans laquelle les sessions sont conservées. Vous pouvez sélectionner l'une des options suivantes :

- **Conserver les sessions dans une nouvelle grille de données sur IBM WebSphere DataPower XC10 Appliance**. Vous pouvez ensuite indiquer un **nom de grille de données**.
- **Conserver les sessions dans une grille de données existante sur IBM WebSphere DataPower XC10 Appliance**. Vous pouvez ensuite entrer ou rechercher un **nom de grille de données existant**.

Configuration d'une grille de données distante eXtreme Scale :

Les paramètres suivants s'appliquent à la configuration de la grille distante eXtreme Scale pour la persistance des sessions :

Domaine de service de catalogue gérant la grille de données de session distante :

Indique le domaine de service de catalogue à utiliser pour gérer les sessions.

Si aucun domaine de service de catalogue n'est affiché ou que vous souhaitez créer un domaine de service de catalogue nouveau, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **domaine de service de catalogue**.

Grille de données distante utilisée pour stocker les informations de session :

Indique le nom de la grille de données du domaine de service de catalogue où les informations de session doivent être stockées. La liste des grilles distantes actives est renseignée lorsque vous sélectionnez un service de catalogue. La grille de données distante doit déjà être définie dans la configuration eXtreme Scale.

Configuration d'une grille de données imbriquée eXtreme Scale :

Les paramètres suivants s'appliquent à une configuration eXtreme Scale imbriquée. Dans le scénario de configuration imbriquée eXtreme Scale, les processus eXtreme Scale sont hébergés par des processus WebSphere Application Server.

Configuration d'une grille de données imbriquée eXtreme Scale :

- **Utiliser la configuration ObjectGrid par défaut**
- **Indiquer des fichiers de configuration ObjectGrid personnalisés**

  Chemin complet du fichier `objectgrid.xml` à copier dans la configuration
  Indique le chemin complet du fichier `objectgrid.xml` correspondant à la configuration à utiliser.

  Chemin complet du fichier `objectgriddeployment.xml` à copier dans la configuration
  Indique le chemin complet du fichier `objectgriddeployment.xml` correspondant à la configuration à utiliser.

Utilisation de WebSphere eXtreme Scale pour la gestion des sessions SIP

Java

Vous pouvez utiliser WebSphere eXtreme Scale comme mécanisme de réplication SIP (Session Initiation Protocol). Ce système se substitue en toute fiabilité au service DRS (Data Replication Service) pour la réplication des sessions SIP.

Configuration de la gestion des sessions SIP

Pour utiliser WebSphere eXtreme Scale comme mécanisme de réplication SIP, définissez la propriété personnalisée com.ibm.sip.ha.replicator.type. Dans la console d'administration, sélectionnez **Application servers** > ***mon_serveur_applications*** > **SIP container** > **Custom properties** pour chaque serveur auquel la propriété personnalisée doit être ajoutée. Entrez `com.ibm.sip.ha.replicator.type` pour le nom et `OBJECTGRID` pour la valeur.

Utilisez les propriétés ci-dessous pour personnaliser le comportement de la valeur ObjectGrid utilisée pour stocker les sessions SIP. Dans la console d'administration, cliquez sur **Application servers** > *mon_serveur_applications* > **SIP container** > **Custom properties** pour chaque serveur auquel la propriété personnalisée doit être ajoutée. Renseignez les zones **Nom** et **Valeur**. Les mêmes propriétés doivent être définies pour chaque serveur pour qu'il fonctionne correctement.

Tableau 28. Propriétés personnalisées pour la gestion des sessions SIP avec ObjectGrid

Propriété	Valeur	Par défaut
com.ibm.sip.ha.replicator.type	`OBJECTGRID` : utilisez ObjectGrid pour stocker les sessions SIP	
min.synchronous.replicas	Nombre minimal de fragments réplique synchrones	0
max.synchronous.replicas	Nombre maximal de fragments réplique synchrones	0
max.asynchronous.replicas	Nombre maximal de fragments réplique asynchrones	1
auto.replace.lost.shards	Pour plus d'informations, voir «Configuration de déploiements répartis», à la page 286.	true
development.mode	• `true` : active les fragments réplique sur le même noeud que les primaires • `false` : les fragments réplique doivent être sur un noeud différent que les primaires	false

Configuration du gestionnaire de sessions HTTP avec WebSphere Portal

Java

Vous pouvez rendre persistantes des sessions HTTP depuis WebSphere Portal dans une grille de données.

Avant de commencer

Votre environnement WebSphere eXtreme Scale et WebSphere Portal doivent satisfaire aux spécifications suivantes :

- La façon dont vous installez WebSphere eXtreme Scale dépend de votre scénario de déploiement. Vous pouvez exécuter les serveurs de conteneurs qui hébergent les grilles de données en dedans ou en dehors de la cellule WebSphere Application Server :
  - Si vous exécutez des serveurs de conteneurs dans la cellule WebSphere Application Server **(scénario imbriqué)** : installez le client et le serveur WebSphere eXtreme Scale sur vos noeuds WebSphere Application Server et WebSphere Portal.
  - Si vous exécutez des serveurs de conteneurs en dehors de la cellule WebSphere Application Server **(scénario à distance)** : installez WebSphere eXtreme Scale Client sur vos noeuds WebSphere Application Server et WebSphere Portal.

  Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226.
- WebSphere Portal Version 7 ou suivante.
- Les portlets personnalisés doivent être configurés dans WebSphere Portal. Les portlets d'administration livrés avec WebSphere Portal ne peuvent actuellement pas être intégrés avec des grilles de données.

Pourquoi et quand exécuter cette tâche

L'introduction de WebSphere eXtreme Scale dans un environnement WebSphere Portal peut être bénéfique dans les scénarios suivants :

Important : Bien que les scénarios suivants apportent des avantages, une utilisation plus importante des processeurs au niveau de WebSphere peut résulter de l'introduction de WebSphere eXtreme Scale dans l'environnement.

- **Lorsque la persistance des sessions est requise.**

  Par exemple, si les données de session de vos portlets personnalisés doivent rester disponibles lors d'une défaillance de WebSphere Portal Server, vous pouvez rendre persistantes les sessions HTTP dans la grille de données WebSphere eXtreme Scale. Les données sont répliquées entre de nombreux serveurs, accroissant la disponibilité des données.
- **Dans une topologie avec plusieurs centres de données.**

  Si votre topologie couvre plusieurs centres de données à travers différents emplacements physiques, vous pouvez rendre persistantes les sessions HTTP de WebSphere Portal dans la grille de données WebSphere eXtreme Scale. Les sessions sont répliquées dans les grilles de données des centres de données. Si un centre de données est défaillant, les sessions sont basculées vers un autre centre de données qui a une copie des données de la grille de données.
- **Pour diminuer la mémoire requise au niveau de WebSphere Portal Server.**

  En déchargeant les données de session sur un groupe de serveurs de conteneurs, un sous-ensemble des sessions se trouve sur les serveurs WebSphere Portal. Ce déchargement de données réduit la mémoire requise au niveau de WebSphere Portal Server.

Procédure

1. Raccordez l'application WebSphere Portal `wps` et les éventuels portlets personnalisés pour permettre aux sessions d'être stockées dans la grille de données.

   Vous pouvez raccorder l'application en configurant la gestion de session HTTP lorsque vous déployez l'application, ou vous pouvez utiliser des propriétés personnalisées pour raccorder automatiquement vos applications. Voir «Configuration du gestionnaire de sessions HTTP avec WebSphere Application Server», à la page 373 pour plus d'informations sur le raccordement de l'application.
2. Si vous utilisez un scénario à distance où vos serveurs de conteneurs se trouvent en dehors de WebSphere Application Server, démarrez explicitement les conteneurs eXtreme Scale distants pour les scénarios de persistance de sessions HTTP à distance. Démarrez les conteneurs avec les fichiers de configuration `XS/ObjectGrid/session/samples/objectGridStandAlone.xml` et `objectGridDeploymentStandAlone.xml`. Par exemple, vous pouvez utiliser la commande suivante : UNIX Linux

   ```
   startOgServer.sh xsContainer1 -catalogServiceEndPoints <hôte>:<port>
   -objectgridFile XS/ObjectGrid/session/samples/objectGridStandAlone.xml -deploymentPolicyFile
   XS/ObjectGrid/session/samples/objectGridDeploymentStandAlone.xml
   ```

   8.6+

   ```
   startXsServer.sh xsContainer1 -catalogServiceEndPoints <host>:<port>
   -objectgridFile XS/ObjectGrid/session/samples/objectGridStandAlone.xml -deploymentPolicyFile
   XS/ObjectGrid/session/samples/objectGridDeploymentStandAlone.xml
   ```

   Pour plus d'informations sur le démarrage des serveurs de conteneur, voir «Démarrage des serveurs de conteneur qui utilisent le transport ORB», à la page 496

page 496 ou «Démarrage des serveurs de conteneur qui utilisent le transport IBM eXtremeIO (XIO)», à la page 482. Si vous utilisez un scénario imbriqué, consultez «Configuration des serveurs de conteneur dans WebSphere Application Server», à la page 333 pour plus d'informations sur la configuration et le démarrage des serveurs de conteneurs.

3. Redémarrez les serveurs WebSphere Portal. Pour plus d'informations, voir WebSphere Portal version 7 : Démarrage et arrêt des serveurs, des gestionnaires de déploiement et des agents de noeud.

Résultats

Vous pouvez accéder à WebSphere Portal Server ; les données de session HTTP pour les portlets personnalisés configurés sont conservées dans la grille de données.

Si l'ensemble de la grille de données qui héberge les données de sessions d'application est inaccessible à partir du client de conteneur Web, le client utilise le conteneur Web de base dans la gestion de sessions WebSphere Application Server. La grille de données peut être inaccessible dans les scénarios suivants :

- Problème de réseau entre le conteneur Web et les serveurs de conteneur distants.
- Arrêt des processus serveur de conteneur distant.

Le nombre de références de session conservées en mémoire, spécifié par le paramètre **sessionTableSize** est toujours maintenu lorsque les sessions sont stockées dans le conteneur Web de base. Les sessions les moins utilisées sont invalidés à partir du cache de session du conteneur Web lorsque la valeur **sessionTableSize** est dépassée. Si la grille de données distante devient disponible, les sessions ayant été invalidées à partir du cache du conteneur Web peuvent extraire les données de la grille de données distante et charger les données dans une nouvelle session. Si l'ensemble de la grille de données distante n'est pas disponible et que la session est invalidée dans le cache de session, les données de session utilisateur sont perdues. Compte tenu de ce problème, n'arrêtez pas l'ensemble de la grille de données distante de production lorsque le système est chargé.

Configuration du gestionnaire de sessions HTTP pour divers serveurs d'applications

WebSphere eXtreme Scale est regroupé avec une implémentation de gestion de session qui remplace le gestionnaire de sessions par défaut pour un conteneur Web. Cette implémentation fournit l réplication de session, la haute disponibilité, améliore l'évolutivité et des options de configuration. Vous pouvez activer le gestionnaire de réplication de session WebSphere eXtreme Scale et le démarrage du conteneur intégré ObjectGrid générique.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser le gestionnaire de sessions HTTP avec d'autres serveurs d'applications qui n'exécutent pas WebSphere Application Server, WebSphere Application Server Community Edition, par exemple. Pour configurer d'autres serveurs d'applications pour qu'ils utilisent la grille de données, vous devez raccorder votre application et incorporer des fichiers WebSphere eXtreme Scale Java archive (JAR) dans votre application.

Procédure

1. Raccordez votre application de sorte qu'elle puisse utiliser le gestionnaire de sessions. Pour utiliser le gestionnaire de sessions, vous devez ajouter les

déclarations de filtre appropriées aux descripteurs de déploiement Web de l'application. En outre, les paramètres de configuration du gestionnaire de sessions sont transmis au gestionnaire de sessions sous la forme de paramètres d'initialisation du contexte de servlet dans les descripteurs de déploiement. Vous disposez de trois manières de présenter ces informations dans votre application :

- Script **addObjectGridFilter** :

  Utilisez un script de ligne de commande fourni avec eXtreme Scale pour raccorder une application avec des déclarations de filtre et une configuration sous forme de paramètres d'initialisation de contexte de servlet. Le script *rép_base_wxs*/session/bin/addObjectGridFilter.sh|bat accepte deux paramètres : le chemin absolu d'accès au fichier EAR (enterprise archive) ou au fichier WAR (web archive) à raccorder et le chemin absolu au fichier des propriétés splicer qui contient diverses propriétés de configuration. La syntaxe de ce script est la suivante :

  Windows

  ```
  addObjectGridFilter.bat <ear_or_war_file> <splicer_properties_file>
  ```

  UNIX

  ```
  addObjectGridFilter.sh <ear_or_war_file> <splicer_properties_file>
  ```

  UNIX Exemple d'utilisation de eXtreme Scale installé dans un répertoire autonome sur UNIX :

  a. `cd rép_base_wxs/session/bin`
  b. `addObjectGridFilter.sh /tmp/mySessionTest.ear rép_base_wxs/session/samples/splicer.properties`

  Le filtre de servlet qui est joint conserve les valeurs de configuration par défaut. Vous pouvez remplacer ces valeurs par défaut par des options de configuration que vous spécifiez dans le fichier de propriétés, dans le second argument. Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 395.

  Vous pouvez modifier et utiliser l'exemple de fichier splicer.properties fourni avec l'installation d'eXtreme Scale. Vous pouvez également utiliser le script **addObjectGridServlets**, qui insère le gestionnaire de sessions en étendant chaque servlet. Mais le script recommandé est le script **addObjectGridFilter**.

- Script de génération Ant :

  WebSphere eXtreme Scale est fourni avec un fichier build.xml qui peut être utilisé par Apache Ant, qui est inclus dans le dossier *racine_was*/bin d'une installation WebSphere Application Server. Vous pouvez modifier le fichier build.xml pour changer les propriétés de configuration du gestionnaire de sessions. Les propriétés de configuration sont identiques aux noms de propriété dans le fichier splicer.properties. Une fois que le fichier build.xml a été modifié, appelez le processus Ant en exécutant ant.sh, ws_ant.sh (UNIX) ou ant.bat, ws_ant.bat (Windows).

- Mise à jour manuelle du descripteur Web :

  Editez le fichier web.xml qui est packagé avec l'application Web pour incorporer la déclaration de filtre, son mappage de servlets et les paramètres d'initialisation du contexte de servlet. N'utilisez pas cette méthode car elle est source d'erreurs possibles.

Pour une liste des paramètres que vous pouvez utiliser, voir «Paramètres d'initialisation du contexte de servlet», à la page 395.

2. Incorporez dans votre application les fichiers JAR du gestionnaire de réplication de sessions d'WebSphere eXtreme Scale. Vous pouvez incorporer les fichiers dans le répertoire `WEB-INF/lib` des modules d'application ou dans le chemin d'accès aux classes du serveur d'applications. Les fichiers JAR requis varient selon le type de conteneurs utilisés :
   - Serveurs de conteneur distants : `ogclient.jar` et `sessionobjectgrid.jar`
   - Serveurs de conteneur intégrés : `objectgrid.jar` et `sessionobjectgrid.jar`
3. Facultatif : Si vous utilisez des serveurs de conteneur distant, démarrez les serveurs de conteneur. Voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494ou «Démarrage d'un service de catalogue autonome qui utilise le transport eXtremeIO IBM (XIO)», à la page 480 pour plus d'informations.
4. Déployez l'application. Déployez l' application à l'aide de votre procédure normale pour un serveur ou un cluster. Une fois que vous avez déployé l'application, vous pouvez la démarrer.
5. Accédez à l'application. Vous pouvez maintenant accéder à l'application, qui interagit avec le gestionnaire de sessions et WebSphere eXtreme Scale.

Que faire ensuite

Vous pouvez modifier la majorité des attributs de configuration du gestionnaire de sessions lorsque vous instrumentez votre application pour utiliser le gestionnaire de sessions. Ces attributs sont des variantes du type de réplication (synchrone ou asynchrone), la taille de la table des sessions en mémoire, etc. En dehors des attributs modifiables lors de l'instrumentation de l'application, les seuls autres attributs de configuration que vous pouvez modifier après le déploiement de l'application sont ceux liés à la topologie des clusters de serveurs WebSphere eXtreme Scale et à la manière dont leurs clients (gestionnaires de sessions) s'y connectent.

Comportement dans le scénario distant : si l'ensemble de la grille de données qui héberge les données de session d'application est inaccessible depuis le client du conteneur Web, le client utilise à la place le conteneur Web de base du serveur d'applications pour gérer les sessions. La grille de données peut être inaccessible dans les scénarios suivants :

- Problème de réseau entre le conteneur Web et les serveurs de conteneur distants
- Arrêt des processus serveur de conteneur distant

Le nombre de références de session conservées en mémoire, spécifié par le paramètre **`sessionTableSize`**, est toujours maintenu lorsque les sessions sont stockées dans le conteneur Web de base. Les sessions les moins utilisées sont invalidées à partir du cache de session du conteneur Web lorsque la valeur **`sessionTableSize`** est dépassée. Si la grille de données distante devient disponible, les sessions ayant été invalidées à partir du cache du conteneur Web peuvent extraire les données de la grille de données distante et charger les données dans une nouvelle session. Si l'ensemble de la grille de données distante n'est pas disponible et que la session est invalidée dans le cache de session, les données de session utilisateur sont perdues. Compte tenu de ce problème, n'arrêtez pas l'ensemble de la grille de données distante de production lorsque le système est chargé.

Fichiers XML de configuration du gestionnaire de sessions HTTP

Lorsque vous démarrez un serveur de conteneur qui stocke les données de session HTTP, vous pouvez utiliser les fichiers XML par défaut ou spécifier des fichiers XML personnalisés. Ces fichiers créent des noms ObjectGrid spécifiques, un nombre de répliques, etc.

Emplacement des fichiers d'exemples

Ces fichiers XML sont regroupés dans *racine_install_wxs*/ObjectGrid/session/samples pour une installation autonome ou *racine_was*/optionalLibraries/ObjectGrid/session/samples pour WebSphere eXtreme Scale installé dans une cellule WebSphere Application Server.

Package XML intégré

Si vous configurez un scénario intégré, le serveur de conteneur commence dans le groupe de serveurs de Web. Utilisez le fichier objectGrid.xml et le fichier objectGridDeployment.xml fournis par défaut. Vous pouvez mettre à jour ces fichiers pour personnaliser le comportement du gestionnaire de session HTTP.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd" xmlns="http://ibm.com/ws/objectgrid/config">
 <objectGrids>
  <objectGrid name="session" txTimeout="30">
   <bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
   <backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata" readOnly="false"
     lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
   <backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
     ttlEvictorType="NONE" copyMode="NO_COPY"/>
       <backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
     ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
  </objectGrid>
 </objectGrids>
    <backingMapPluginCollections>
        <backingMapPluginCollection id="objectgridSessionMetadata">
            <bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Figure 39. Fichier objectGrid.xml

Valeurs que vous pouvez modifier :

Attribut de nom ObjectGrid
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **objectGridName** dans le fichier splicer.properties qui est utilisé pour raccorder l'application Web.
- L'attribut **objectgridName** dans le fichier objectGridDeployment.xml.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

Attribut ObjectGrid txTimeout
Cette valeur détermine le nombre de secondes pendant lequel une transaction peut être ouverte avant que le serveur de conteneur déclenche la fasse expirer. La valeur par défaut est 30 secondes et elle peut être changée en fonction de l'environnement. Si la persistance de session HTTP est configurée avec une valeur de paramètre d'initialisation de contexte de servlet **replicationInterval** supérieure à zéro, les transactions sont traitées par lots dans une unité d'exécution. Si la propriété **replicationInterval** a

la valeur `0`, une transaction démarre généralement lorsqu'une application Web extrait un objet HttpSession valide. La transaction valide la fin de la demande d'application Web. Si l'environnement a des demandes qui durent plus de 30 secondes, définissez cette valeur en conséquence.

Valeurs que vous ne pouvez pas changer :

ObjectGridEventListener
La ligne ObjectGridEventListener ne peut pas être modifiée et elle est utilisée en interne.

objectgridSessionMetadata
La ligne objectgridSessionMetadata fait référence à la mappe où sont stockées les métadonnées de session HTTP. Il existe une entrée pour chaque session HTTP stockée dans la grille de données dans cette mappe.

objectgridSessionTTL.*
Cette valeur ne peut pas être modifiée et elle est réservée à une utilisation future.

objectgridSessionAttribute.*
Le texte `objectgridSessionAttribute.*` définit une mappe dynamique. Cette valeur est utilisée pour créer la mappe où sont stockés les attributs de session HTTP lorsque le paramètre **`fragmentedSession`** a la valeur `true` dans le fichier `splicer.properties`. Cette mappe dynamique s'appelle `objectgridSessionAttribute.*`. Une autre mappe est créée en fonction de ce modèle appelé `objectgridSessionAttributeEvicted` qui stocke les sessions qui ont expiré lorsque le conteneur Web n'a pas invalidé.

Une règle de durée de vie est définie pour la définition de mappe `objectgridSessionMetadata`. L'autre mappe, `objectgridSessionAttribute`, dépend de cette mappe et ne nécessite pas de paramètre de durée de vie. Pour chaque session HTTP active, une entrée est créée dans la mappe `objectgridSessionMetadata` et dans la mappe `objectgridSessionAttribute` pour chaque attribut de session. Si une session en mémoire n'existe pas suite à une erreur du serveur d'applications ou à la suppression d'une session de la mémoire cache interne sur le serveur d'applications, la grille doit lancer l'invalidation de la session à l'aide de la règle d'expulsion TTL (durée de vie). Au moment de l'expulsion, les attributs sont supprimés de la mappe `objectgridSessionAttribute` et insérés dans une mappe dynamique créée appelée `objectgridSessionAttributeEvicted`. Les données sont stockées dans cette mappe jusqu'à ce qu'un serveur d'applications puisse supprimer la session et exécuter l'invalidation de session. Par conséquent, le paramètre TTL est uniquement requis dans la définition de mappe `objectgridSessionMetadata`.

Remarque : `objectgridSessionTTL` n'est pas utilisé par WebSphere eXtreme Scale dans l'édition actuelle.

La ligne **`MapEventListener`** est interne et elle ne peut pas être modifiée.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="session">
  <mapSet name="sessionMapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="0"
    maxAsyncReplicas="1" developmentMode="false" placementStrategy="PER_CONTAINER">
             <map ref="objectgridSessionMetadata"/>
             <map ref="objectgridSessionAttribute.*"/>
             <map ref="objectgridSessionTTL.*"/>
    </mapSet>
   </objectgridDeployment>
</deploymentPolicy>
```

Figure 40. Fichier `objectGridDeployment.xml`

Valeurs que vous pouvez changer :

Attribut de nom ObjectGrid

La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **`objectGridName`** dans le fichier `splicer.properties` qui est utilisé pour raccorder l'application Web.
- L'attribut **`name`** dans le fichier `objectGrid.xml`.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

Attributs d'élément mapSet

Vous pouvez changer toutes les propriétés mapSet, sauf pour l'attribut placementStrategy.

Name Peut être mis à jour avec n'importe quelle valeur.

numberOfPartitions

Spécifie le nombre de partitions primaires qui sont démarrées sur chacun des serveurs hébergeant l'application Web. Au fur et à mesure que l'on ajoute des partitions, les données sont de plus en plus réparties dans l'éventualité d'un basculement. La valeur par défaut est `5` partitions ; elle convient pour la plupart des applications.

minSyncReplicas, maxSyncReplicas et maxAsyncReplicas

Spécifient le nombre et le type des fragments réplique qui stockent les données de session HTTP. La valeur par défaut est 1 réplique asynchrone, ce qui convient pour la plupart des applications. La réplication synchrone intervient pendant le chemin de demande, ce qui peut augmenter les temps de réponse de l'application Web.

developmentMode

Informe le service de placement eXtreme Scale si les fragments réplique d'une partition peuvent être positionnés sur le même noeud que leur fragment primaire. La valeur peut être définie comme true dans un environnement de développement, mais il est conseillé de désactiver cette fonction en environnement de production en raison des risques de pertes de données que pourrait provoquer une défaillance du noeud.

placementStrategy

Ne modifiez pas la valeur de cet attribut.

Le reste du fichier se réfère aux mêmes noms de mappes que dans le fichier `objectGrid.xml`. Ces noms ne peuvent pas être modifiés.

Valeurs non modifiables :

- L'attribut placementStrategy dans l'élément mapSet.

Package XML distant

Lorsque vous utilisez le mode distant dans lequel les conteneurs s'exécutent comme des processus autonomes, vous devez utiliser le fichier `objectGridStandAlone.xml` et le fichier `objectGridDeploymentStandAlone.xml` pour démarrer les processus. Vous pouvez modifier ces fichiers pour adapter la configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
 <objectGrids>
  <objectGrid name="session" txTimeout="30">
   <bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
           <backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata"
       readOnly="false" lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600"
       copyMode="COPY_TO_BYTES"/>
           <backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
       ttlEvictorType="NONE" copyMode="COPY_TO_BYTES"/>
           <backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
       ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="COPY_TO_BYTES"/>
  </objectGrid>
 </objectGrids>
    <backingMapPluginCollections>
        <backingMapPluginCollection id="objectgridSessionMetadata">
            <bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Figure 41. `objectGridStandAlone.xml` file

Valeurs que vous pouvez changer :

Attribut de nom ObjectGrid
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **`objectGridName`** dans le fichier `splicer.properties` qui est utilisé pour raccorder l'application Web.
- L'attribut **`objectgridName`** dans le fichier `objectGridStandAlone.xml`.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

Attribut ObjectGrid txTimeout
Cette valeur détermine le nombre de secondes pendant lequel une transaction peut être ouverte avant que le serveur de conteneur déclenche la fasse expirer. La valeur par défaut est 30 secondes et elle peut être changée en fonction de l'environnement. Si la persistance de session HTTP est configurée avec une valeur de paramètre d'initialisation de contexte de servlet **`replicationInterval`** supérieure à zéro, les transactions sont traitées par lots dans une unité d'exécution. Si la propriété **`replicationInterval`** a la valeur `0`, une transaction démarre généralement lorsqu'une application Web extrait un objet HttpSession valide. La transaction valide la fin de la demande d'application Web. Si l'environnement a des demandes qui durent plus de 30 secondes, définissez cette valeur en conséquence.

Valeurs que vous ne pouvez pas changer :

ObjectGridEventListener
La ligne ObjectGridEventListener ne peut pas être modifiée et elle est utilisée en interne.

objectgridSessionMetadata
La ligne objectgridSessionMetadata fait référence à la mappe où sont stockées les métadonnées de session HTTP. Il existe une entrée pour chaque session HTTP stockée dans la grille de données dans cette mappe.

objectgridSessionTTL.*
Cette valeur ne peut pas être modifiée et elle est réservée à une utilisation future.

objectgridSessionAttribute.*
Le texte `objectgridSessionAttribute.*` définit une mappe dynamique. Cette valeur est utilisée pour créer la mappe où sont stockés les attributs de session HTTP lorsque le paramètre **`fragmentedSession`** a la valeur `true` dans le fichier `splicer.properties`. Cette mappe dynamique s'appelle `objectgridSessionAttribute.*`. Une autre mappe est créée en fonction de ce modèle appelé `objectgridSessionAttributeEvicted` qui stocke les sessions qui ont expiré lorsque le conteneur Web n'a pas invalidé.

Une règle de durée de vie est définie pour la définition de mappe `objectgridSessionMetadata`. L'autre mappe, `objectgridSessionAttribute`, dépend de cette mappe et ne nécessite pas de paramètre de durée de vie. Pour chaque session HTTP active, une entrée est créée dans la mappe `objectgridSessionMetadata` et dans la mappe `objectgridSessionAttribute` pour chaque attribut de session. Si une session en mémoire n'existe pas suite à une erreur du serveur d'applications ou à la suppression d'une session de la mémoire cache interne sur le serveur d'applications, la grille doit lancer l'invalidation de la session à l'aide de la règle d'expulsion TTL (durée de vie). Au moment de l'expulsion, les attributs sont supprimés de la mappe `objectgridSessionAttribute` et insérés dans une mappe dynamique créée appelée `objectgridSessionAttributeEvicted`. Les données sont stockées dans cette mappe jusqu'à ce qu'un serveur d'applications puisse supprimer la session et exécuter l'invalidation de session. Par conséquent, le paramètre TTL est uniquement requis dans la définition de mappe `objectgridSessionMetadata`.

Remarque : `objectgridSessionTTL` n'est pas utilisé par WebSphere eXtreme Scale dans l'édition actuelle.

La ligne **`MetadataMapListener`** est une ligne interne et elle ne peut pas être modifiée.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

 <objectgridDeployment objectgridName="session">
  <mapSet name="sessionMapSet" numberOfPartitions="47" minSyncReplicas="0" maxSyncReplicas="0"
   maxAsyncReplicas="1" developmentMode="false" placementStrategy="FIXED_PARTITIONS">
            <map ref="objectgridSessionMetadata"/>
            <map ref="objectgridSessionAttribute.*"/>
            <map ref="objectgridSessionTTL.*"/>
     </mapSet>
    </objectgridDeployment>
</deploymentPolicy>
```

Figure 42. objectGridDeploymentStandAlone.xml file

Valeurs que vous pouvez changer :

Attribut objectgridName
La valeur doit correspondre aux valeurs suivantes dans les autres fichiers de configuration :

- La propriété **objectGridName** dans le fichier `splicer.properties` qui est utilisé pour raccorder l'application Web.
- L'attribut **name** dans le fichier `objectGrid.xml`.

Si vous disposez de plusieurs applications et voulez stocker les données de session dans des grilles de données différentes, ces applications doivent avoir des valeurs d'attribut de nom ObjectGrid différentes.

Attributs d'élément mapSet
Vous pouvez modifier toutes les propriétés mapSet.

Name Peut être mis à jour avec n'importe quelle valeur.

numberOfPartitions
Lorsque vous utilisez la stratégie de placement FIXED_PARTITIONS par défaut, cet attribut indique le nombre de partitions totales qui seront réparties entre tous les conteneurs de grille en cours d'exécution. La valeur par défaut est 47 partitions ; elle convient pour la plupart des applications. Si une stratégie de placement PER_CONTAINER est utilisée, cette valeur indique le nombre de partitions primaires démarrées dans chaque conteneur de grille. Au fur et à mesure que l'on ajoute des partitions, les données sont de plus en plus réparties dans l'éventualité d'un basculement. La valeur recommandée est 5 pour la stratégie PER_CONTAINER.

minSyncReplicas, maxSyncReplicas et maxAsyncReplicas
Spécifie le nombre de partitions primaires qui sont démarrées sur chacun des serveurs hébergeant l'application Web. Au fur et à mesure que l'on ajoute des partitions, les données sont de plus en plus réparties dans l'éventualité d'un basculement. La valeur par défaut est 5 partitions ; elle convient pour la plupart des applications.

developmentMode
Informe le service de placement eXtreme Scale si les fragments réplique d'une partition peuvent être positionnés sur le même noeud que leur fragment primaire. La valeur peut être définie comme true dans un environnement de développement, mais il est conseillé de désactiver cette fonction en environnement de production en raison des risques de pertes de données que pourrait provoquer une défaillance du noeud.

placementStrategy

Vous pouvez remplacer cet attribut par l'un des éléments suivants :

- FIXED_PARTITIONS Il s'agit de la valeur par défaut et de la méthode recommandée pour l'utilisation d'une topologie de session HTTP distante. Vous devez l'utiliser si vous utilisez la réplication multimaître.
- PER_CONTAINER Cette configuration est toujours prise en charge dans une topologie distante.

Paramètres d'initialisation du contexte de servlet

La liste qui suit de paramètres d'initialisation du contexte de servlet peut être spécifiée dans le fichier splicer.properties en fonction de la méthode de raccord choisie.

Paramètres

objectGridType

Valeur de type chaîne `REMOTE` ou `EMBEDDED`. La valeur par défaut est `REMOTE`.

Si la valeur est `REMOTE`, les données de session sont stockées en dehors du serveur sur lequel l'application Web est exécutée.

Si la valeur est `EMBEDDED`, un conteneur intégré eXtreme Scale démarre dans le processus serveur d'applications sur lequel l'application Web s'exécute.

objectGridName

Valeur de chaîne qui définit le nom de l'instance ObjectGrid utilisée pour une application Web particulière. Le nom par défaut est session.

Cette propriété doit refléter le nom objectGridName dans les fichiers XML ObjectGrid et XLM de déploiement utilisés pour démarrer les serveurs de conteneur eXtreme Scale.

catalogHostPort

Le serveur de catalogues peut être contacté pour obtenir une instance ObjectGrid côté client. La valeur doit avoir le format `host:port<,host:port>`. L'hôte est le programme d'écoute sur lequel le serveur de catalogue s'exécute. Le port est le port d'écoute du processus serveur de catalogue. La longueur de cette liste peut être arbitraire et la liste n'est utilisée que pour l'amorçage. La première adresse viable qui est utilisée. Elle est facultative dans WebSphere Application Server si la propriété **catalog.services.cluster** est défini.

replicationInterval

Entier (en secondes) qui définit le temps séparant deux écritures de sessions actualisées vers la grille. La valeur par défaut est `10` secondes. Les valeurs possibles sont comprises entre 0 et 60. 0 signifie que les sessions actualisées sont écrites dans la grille pour chaque demande dès la fin de l'appel à la méthode de service du servlet. Une valeur **replicationInterval** plus élevée améliore les performances, car un moins grand nombre de mises à jour sont écrites dans la grille de données. Mais, en même temps, une valeur supérieure à 0 rend la configuration moins tolérante aux pannes.

Ce paramètre s'applique uniquement lorsque objectGridType a la valeur `REMOTE`.

sessionIdOverrideClass

Nom de la classe qui implémente l'interface com.ibm.websphere.objectgrid.xs.sessionmanager.SessionIDOverride. Cette classe est utilisée pour remplacer l'identificateur de session unique obtenu avec

la méthode HttpSession.getId() afin que toutes les applications aient le même ID. Par défaut, l'ID provenant de HttpSession.getId() est utilisé.

sessionTableSize

Entier qui définit le nombre de références de session conservées en mémoire. La valeur par défaut est `1000`.

Ce paramètre appartient uniquement à une topologie `REMOTE`, car la topologie `EMBEDDED` a déjà les données de session dans le même groupe que le conteneur Web.

Les sessions sont expulsées de la table interne en fonction de la logique LRU (least recently used). Lorsqu'une session est expulsée de cette table, elle est invalidée dans le conteneur Web. Cependant, les données ne sont pas pour autant supprimées de la grille, ce qui permet aux demandes ultérieures de cette session de continuer à extraire les données. Cette valeur doit être supérieure à la valeur maximale du pool d'unités d'exécution du conteneur Web, ce qui réduit la contention sur le cache de session.

fragmentedSession

Valeur de type chaîne `true` ou `false`. La valeur par défaut est `true`. Ce paramètre permet de contrôler si le produit stocke les données de session en tant qu'entrée entière ou s'il stocke chaque attribut séparément.

Affectez au paramètre fragmentedSession la valeur `true` si la session d'application Web a de nombreux attributs ou des attributs avec des grandes tailles. Affectez à fragmentedSession la valeur `false` si une session a peu d'attributs, car tous les attributs sont stockés dans la même clé dans la grille de données.

Dans la précédente implémentation à base de filtres, il était fait référence à cette propriété en tant que mécanisme de persistance avec, comme valeurs possibles, ObjectGridStore (fragmentation) et ObjectGridAtomicSessionStore (non-fragmentation).

securityEnabled

Valeur de type chaîne `true` ou `false`. La valeur par défaut est `false`. Ce paramètre active la sécurité du client eXtreme Scale. Il doit correspondre au paramètre **securityEnabled** dans le fichier des propriétés sur serveur eXtreme Scale. Si les paramètres ne correspondent pas, une exception est générée.

credentialGeneratorClass

Le nom de la classe qui implémente l'interface com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. Cette classe sert à obtenir les données d'identification des clients.

credentialGeneratorProps

Les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés correspondent à l'objet avec la méthode setProperties(String). La valeur credentialGeneratorProps n'est utilisée que si la valeur de la propriété **credentialGeneratorClass** n'est pas null.

objectGridXML

L'emplacement du fichier `objectgrid.xml`. Le fichier XML intégré regroupé dans la bibliothèque eXtreme Scale est chargé automatiquement si `objectGridType=EMBEDDED` et que la propriété **objectGridXML** n'est pas définie.

objectGridDeploymentXML

Indique l'emplacement du fichier XML de stratégie de déploiement d'objectGrid. Le fichier XML intégré regroupé dans la bibliothèque eXtreme Scale est chargé automatiquement si `objectGridType=EMBEDDED` et que la propriété **objectGridDeploymentXML** n'est pas définie.

traceSpec

Spécifie la spécification de trace d'IBM WebSphere comme une valeur de chaîne. Utilisez ce paramètre pour des serveurs d'applications autres que WebSphere Application Server.

traceFile

Spécifie l'emplacement du fichier de trace sous forme de valeur de chaîne. Utilisez ce paramètre pour des serveurs d'applications autres que WebSphere Application Server.

cookieDomain

Spécifie si vous exigez que les sessions soient accessibles à travers les hôtes. Définissez la valeur avec le nom du domaine commun entre les hôtes.

reuseSessionID

A la valeur `true` si le conteneur Web sous-jacent réutilise les ID de session dans les demandes aux différents hôtes. La valeur par défaut est `false`. La valeur de cette propriété doit être la même que la valeur du conteneur Web. Si vous utilisez WebSphere Application Server et configurez la persistance de session HTTP eXtreme Scale en utilisant la console d'administration ou le scriptage de l'outil **wsadmin**, la propriété personnalisée du conteneur Web `HttpSessionIdReuse=true` est ajoutée par défaut. **reuseSessionID** a également la valeur `true`. Si vous ne voulez pas réutiliser l'ID de session, définissez la propriété `HttpSessionIdReuse=false` dans la propriété personnalisée du conteneur Web avant de configurer la persistance de session eXtreme Scale.

shareSessionsAcrossWebApps

Spécifie si les sessions sont partagées entre des applications Web ; spécifiée comme valeur de chaîne `true` ou `false`. La valeur par défaut est `false`. La spécification de servlet indique que les sessions HTTP ne peuvent pas être partagées entre des applications Web. Une extension à la spécification de servlet est fournie pour permettre ce partage.

useURLEncoding

Affectez-lui la valeur `true` pour activer la réécriture d'URL. La valeur par défaut est `false`, ce qui indique que les cookies sont utilisés pour stocker les données de session. La valeur de ce paramètre doit être identique à celle des paramètres de conteneur Web pour la gestion des sessions.

Fichier `splicer.properties`

Le fichier `splicer.properties` contient toutes les options de configuration pour configurer un gestionnaire de sessions basé sur un filtre de servlet.

Exemple de fichier splicer.properties

Si vous décidez d'utiliser l'une des propriétés supplémentaires décrites dans ce fichier, veillez à mettre en commentaire les lignes des propriétés à activer.

```
# Fichier de propriétés qui contient toutes les options de configuration
# que le gestionnaire de sessions ObjectGrid basé sur un filtre de servlet peut être configuré pour utiliser.
#
# Ce fichier de propriétés peut être généré pour attribuer toutes les
# valeurs par défaut à ces paramètres de configuration, et permettre de
# remplacer les paramètres individuels à l'aide des propriétés de tâche ANT,
# si ce fichier de propriétés est utilisé avec
# la tâche ANT filtersplicer.
```

```
# Valeur de chaîne "REMOTE" ou "EMBEDDED".  La valeur par défaut est REMOTE.
# Si elle est définie sur "REMOTE", les données de session seront stockées en dehors du
# serveur où est exécutée l'application Web. Si sa valeur est
# "EMBEDDED", un conteneur WebSphere eXtreme Scale imbriquée démarre
# dans le processus de serveur d'applications dans lequel l'application Web est exécutée.

objectGridType = REMOTE

# Valeur de chaîne qui définit le nom de l'instance  ObjectGrid
# utilisée pour une applications Web donnée. Le nom par défaut
# est session. Cette propriété doit refléter l'objectGridName dans les deux
# fichiers xml objectgrid et de déploiement utilisés pour démarrer les conteneurs eXtreme
# Scale.

objectGridName = session

# Le serveur de catalogues peut être contacté pour obtenir une instance ObjectGrid
# côté client.  La valeur doit avoir le format
# "host:port<,host:port>", où host est l'hôte d'écoute
# sur lequel le serveur de catalogue est en cours d'exécution, et le port est le
# port d'écoute du processus du serveur de catalogue.
# Cette liste peut être arbitrairement longue et n'est utilisée que pour l'amorçage.
# La première adresse valide est utilisée.  Elle est facultative dans WebSphere
# si la propriété catalog.services.cluster est définie.

# catalogHostPort = host:port<,host:port>

# Entier (secondes) qui définit la durée en secondes entre
# l'écriture de sessions actualisées dans ObjectGrid. La valeur par défaut est 10. Cette propriété
# est utilisée uniquement lorsque objectGridType a la valeur REMOTE. Les valeurs possibles sont
# comprises entre 0 et 60. 0 signifie que les sessions actualisées sont écrites dans l'ObjectGrid
# à la fin de l'appel à la méthode de service de servlet de chaque demande.

replicationInterval = 10

# Entier qui définit le nombre de références de session
# conservées en mémoire. La valeur par défaut est 1 000. Cette propriété est utilisée uniquement lorsque
# objectGridType a la valeur REMOTE. Lorsque le nombre de sessions stockées
# dans la mémoire dans le conteneur Web dépasse cette valeur, la première session ayant fait l'objet d'un accès
# est invalidée depuis le conteneur Web. Si une demande
# arrive pour cette session une fois qu'elle a été invalidée, une nouvelle session
# est créée (avec un nouvel ID de session reuseSessionId=false),
# remplie avec les attributs de la session invalidée. Cette value doit toujours être
# supérieure à la taille maximale du pool d'unités
# d'exécution du conteneur pour éviter les conflits dans ce cache de session.

sessionTableSize = 1000

# Valeur de type chaîne "true" ou "false". La valeur par défaut est "true".
# Permet de contrôler si nous stockons les données de session comme entrée intégrale
# ou de stocker chaque attribut séparément.
# Cette propriété s'appelle persistenceMechanism dans l'implémentation
# basée sur un filtre précédente, avec les valeurs possibles
# ObjectGridStore (fragmenté) et ObjectGridAtomicSessionStore
# (non fragmenté).

fragmentedSession = true

# Valeur de type chaîne "true" ou "false". La valeur par défaut est "false".
# Active la sécurité du client eXtreme Scale. Ce paramètre doit correspondre
# au paramètre securityEnabled dans le fichier des propriétés du serveur eXtreme
# Scale. Si les paramètres ne correspondent pas, une exception
est générée.

securityEnabled = false

# Spécifie la prise en charge de l'authentification des données d'identification du client.
#   Les valeurs possibles sont les suivantes :
#   Jamais : le client ne prend pas en charge l'authentification des données d'identification.
#   Pris en charge* : le client prend en charge l'authentification des données d'identification si et seulement si le serveur
#                la prend en charge également.
#   Obligatoire : le client requiert l'authentification des données d'identification.
#   Elle est prise en charge par défaut.

# credentialAuthentication =

# Indique le nombre de tentatives d'authentification si les données d'identification
# ont expiré. Si la valeur est 0, aucune tentative d'authentification
# n'a lieu.

# authenticationRetryCount =
```

```
# Indique le nom de la classe qui implémente l'interface
# com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator
# . Cette classe utilisée pour obtenir les données d'identification des clients.

# credentialGeneratorClass =

# Spécifie les propriétés de la classe d'implémentation
# CredentialGenerator. Les propriétés sont définies dans l'objet avec la méthode
# setProperties(String). La valeur credentialGeneratorProps est utilisée uniquement si la valeur de la
# propriété credentialGeneratorClass est null.

# credentialGeneratorProps =

# Emplacement du fichier xml objectgrid.
# Le fichier xml pré-intégré qui est regroupé dans la bibliothèque eXtreme Scale
# sera automatiquement chargé si cette propriété
# n'est pas spécifiée et que objectGridType=EMBEDDED

# objectGridXML =

# Emplacement du fichier xml de stratégie de déploiement objectGrid.
# Le fichier xml pré-intégré qui est regroupé dans la bibliothèque eXtreme Scale
# sera automatiquement chargé si cette propriété
# n'est pas spécifiée et que objectGridType=EMBEDDED

# objectGridDeploymentXML =

# Chaîne de spécification de trace IBM WebShere,
# utile pour tous les autres serveurs d'applications, outre WebSphere.

# traceSpec =

# Chaîne d'emplacement de fichier de trace.
# utile pour tous les autres serveurs d'applications, outre WebSphere.

# traceFile=

# Cette propriété doit être définie pour que les sessions soient
# accessibles sur les hôtes. La valeur sera le nom du domaine
# commun aux hôtes.

# cookieDomain=

# Cette propriété doit être affectée du chemin que vous avez configuré
# pour les paramètres de cookie de serveur d'applications. Le chemin par défaut
# est /.

# cookiePath

# A la valeur true si le conteneur Web sous-jacent
# réutilise l'ID dans les demandes à différents hôtes. La valeur par défaut
# est false. La valeur doit correspondre à celle définie dans
# le conteneur Web.

# reuseSessionId=

# Une valeur de type chaîne "true" ou "false". La valeur par défaut est
# "false". Conformément à la spécification de servlet, les sessions HTTP
# ne peuvent pas être partagées dans les applications Web. Une extension à la spécification de servlet
# est fournie pour autoriser le partage.

# shareSessionsAcrossWebApps = false

# Valeur de type chaîne "true" ou "false". La valeur par défaut est "false".
# Affectez-lui la valeur true si vous voulez activer la réécriture d'URL (urlRewriting).  La valeur par défaut est
# false. La valeur doit être identique à celle définie dans
# les paramètres de conteneur Web de la gestion de session.

# useURLEncoding = false

# False si vous voulez désactiver les cookies en tant que
# mécanisme de suivi.  La valeur par défaut est true. La valeur doit être identique à celle définie dans
# les paramètres de conteneur Web de la gestion de session.

# useCookies = true

# Valeur de type chaîne "true" ou "false". La valeur par défaut est "false".
# Active le suivi des statistiques des sessions HTTP client eXtreme Scale.

# enableSessionStats = false
```

```
# Remplace l'ID session extraite d'une application. Par défaut,
l'ID provenant de la méthode HttpSession.getId() est utilisé. Active les sessions HTTP client eXtreme Scale
pour remplacer l'ID de session unique d'une application afin que toutes les applications soient
récupérées avec le même ID.
# Affecté de l'implémentation de l'interface  com.ibm.websphere.xs.sessionmanager.SessionIDOverride. Cette interface
# détermine l'ID HttpSession en fonction de l'objet HttpServletRequest.

# sessionIdOverrideClass =  # Spécification des statistiques HTTP client eXtreme Scale .

# sessionStatsSpec = session.all = enabled

# True si votre environnement contient plusieurs applications qui
# utilisent des noms de cookie. False, qui suppose que toutes les applications
# utilisent le même nom de cookie.

# applicationQualifiedCookies=false
```

Exemple : remplacement de l'ID de session par l'interface sessionIdOverrideClass

Java

Vous pouvez remplacer l'ID session extraite d'une application. L'ID provenant de la méthode HttpSession.getId() est utilisé par défaut.

Exemple de préchargement basé sur le client

Exemple de fragment de code :

```
public class CustomSessionID implements
com.ibm.websphere.xs.sessionmanager.SessionIDOverride {

    public void init(InitializationContext ctx) {
    }

    public void destroy() {
    }

    public String getID(SessionIDContext ctx) {
        HttpServletRequest req = ctx.getRequest();

        String sessionId = (String) req.getAttribute("AppID");
        if (sessionId != null) {
            // sessionId is stored in the request as attribute
             "AppID" for this user
            return sessionId;
        }

        Cookie[] cookies = req.getCookies();
        if (cookies != null) {
            for (int i = 0; i < cookies.length; i ++) {
                if (cookies[i].getName().equals("AppID")) {
     // if the request does not yet contain the AppID attribute,
              then the "AppID" cookie must exist
                      return cookies[i].getValue();
                }
            }
        }
        return null;
    }
}
```

Configuration des instances de cache dynamique

WebSphere Dynamic Cache Service prend en charge la création d'une instance de cache par défaut (baseCache) et des instances supplémentaires de cache de servlet et d'objet.

Pourquoi et quand exécuter cette tâche

L'instance de cache par défaut (baseCache) était initialement la seule instance de cache dynamique prise en charge par WebSphere Application Server et elle est actuellement l'instance de cache dynamique standard utilisée par WebSphere Commerce Suite. Des instances supplémentaires de servlet et d'objet ont été ajoutées dans les dernières versions de WebSphere Application Server et sont configurées dans une section "Instance de cache" de la console d'administration WebSphere.

Configuration de l'instance de cache dynamique par défaut (baseCache)

Java

L'instance de cache dynamique par défaut est créée par le service de cache dynamique WebSphere Application Server. Cette instance de cache dynamique de servlet est utilisée par des produits, tels qu'IBM WebSphere Commerce. Contrairement aux autres instances de cache définies avec WebSphere Application Server, baseCache est spécifique d'un seul serveur ou d'une seule instance de cluster. Utilisez cette procédure pour configurer l'instance baseCache dans WebSphere Application Server pour l'utiliser avec WebSphere eXtreme Scale comme fournisseur de cache dynamique.

Avant de commencer

- Pour pouvoir utiliser le fournisseur de cache dynamique, WebSphere eXtreme Scale doit être installé sur les déploiements de noeud WebSphere Application Server et notamment le noeud du gestionnaire de déploiement. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226 .
- Le domaine de service de catalogue WebSphere eXtreme Scale doit être configuré. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307.
- Un environnement de grille WebSphere eXtreme Scale constitué d'un ou de plusieurs serveurs de catalogue et de conteneur doit être démarré. Pour plus d'informations, voir «Configuration d'une grille de données d'entreprise dans un environnement autonome pour la mise en cache dynamique», à la page 336.
- Si dans les serveurs de catalogue dans le domaine de service de catalogue SSL (Secure Sockets Layer) est activé ou que vous voulez utiliser SSL pour un domaine de service de catalogue avec SSL pris en charge, la sécurité globale doit être activée dans la console d'administration WebSphere Application Server. Vous activez SSL pour un serveur de catalogue en affectant à l'attribut la valeur `SSL-Required` dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.

Pourquoi et quand exécuter cette tâche

Cette procédure s'applique à la version 8.0 de la console d'administration WebSphere Application Server. Ces informations peuvent varier légèrement en fonction de la version de WebSphere Application Server que vous utilisez.

Remarque :

- WebSphere eXtreme Scale Version 8.6 n'est pas pris en charge sur les versions de WebSphere Application Server antérieures à la version 7.0.

- La procédure suivante s'applique à la topologie de cache dynamique WebSphere eXtreme Scale distante. Toutes les autres topologies, notamment intégrées, partitionnées intégrées et locales, sont obsolètes dans WebSphere eXtreme Scale Version 8.6.

Procédure

1. Démarrez la console d'administration WebSphere Application Server.
2. Dans le menu supérieur, cliquez sur **Serveurs > Type de serveur > Serveurs d'applications WebSphere**.
3. Dans la zone **Serveurs d'applications**, sélectionnez le **nom de votre serveur**.
4. Dans le panneau **Configuration**, cliquez sur **Services de conteneur** et sélectionnez **Service de cache dynamique**.
5. Dans la liste déroulante **Fournisseur de cache**, sélectionnez WebSphere eXtreme Scale.

   Remarque : Si WebSphere eXtreme Scale n'apparaît pas comme fournisseur de cache dynamique, cela implique que le profil WebSphere Application Server n'a pas été étendu pour WebSphere eXtreme Scale. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 243.
6. Si vous voulez changer la taille du cache, définissez-la dans la zone de **taille de cache**. La taille de cache définit le nombre maximal d'entrées autorisées dans chaque partition dans une grille WebSphere eXtreme Scale pour l'instance de cache dynamique. La valeur par défaut est 2 000 entrées dans chaque partition.
7. Cochez la case d'**activation de la réplication de cache**. Dans ce cas, les données en cache sont stockées à distance dans la grille et non pas localement. Vous devez cocher la case lorsque vous utilisez WebSphere eXtreme Scale comme fournisseur de cache.
8. Cliquez sur **Appliquer** ou **OK** et enregistrez la configuration.
9. Dans le menu supérieur, cliquez sur **Serveurs > Type de serveur > Serveurs d'applications WebSphere**.
10. Dans la zone **Serveurs d'applications**, sélectionnez le **nom de votre serveur**.
11. Dans le panneau **Configuration**, cliquez sur **Paramètres de conteneur Web** et sélectionnez **Conteneur Web**.
12. Cochez l'option d'**activation de la mise en cache de servlet**.
13. Cliquez sur **Appliquer** ou **OK** et enregistrez la configuration.

Configuration des instances de cache dynamique d'objet ou de servlet

Java

WebSphere Application Server permet de configurer des instances de cache dynamique d'objet ou de servlet en plus de l'instance par défaut. Procédez comme suit pour configurer des instances de cache dynamique d'objet ou de servlet supplémentaires.

Avant de commencer

- Pour pouvoir utiliser le fournisseur de cache dynamique, WebSphere eXtreme Scale doit être installé sur les déploiements de noeud WebSphere Application Server, notamment le noeud du gestionnaire de déploiement. Pour plus

d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226 .

- Le domaine de service de catalogue WebSphere eXtreme Scale doit être configuré. Voir Configuration des serveurs de catalogue et des domaines de service de catalogue.
- Un environnement de grille WebSphere eXtreme Scale constitué d'un ou de plusieurs serveurs de catalogue et de conteneur doit être démarré. Pour plus d'informations, voir Démarrage et arrêt des serveurs autonomes.
- Si dans les serveurs de catalogue dans le domaine de service de catalogue, SSL (Secure Sockets Layer) est activé ou que vous voulez utiliser SSL pour un domaine de service de catalogue avec SSL pris en charge, la sécurité globale doit être activée dans la console d'administration WebSphere Application Server. Vous activez SSL pour un serveur de catalogue en affectant à l'attribut la valeur `SSL-Required` dans Fichier de propriétés du serveur. Pour plus d'informations sur la configuration de la sécurité globale, voir Paramètres de sécurité globale.

Pourquoi et quand exécuter cette tâche

Dans cette procédure, vous pouvez créer deux types d'instances de cache : instances de cache d'objet et instance de cache de servlet. Une instance de cache d'objet est un emplacement en complément du cache dynamique partagé par défaut où les applications Java 2 Platform, Enterprise Edition (J2EE) peuvent stocker, distribuer et partager des objets. Après avoir configuré des instances de cache d'objet, vous pouvez utiliser l'interface DistributedMap ou DistributedObjectCache dans le package com.ibm.websphere.cache pour accéder par programme aux instances de cache. Voir API (Application Programming Interface) supplémentaires pour plus d'informations sur les interfaces DistributedMap et DistributedObjectCache . Les instances de cache de servlet sont des emplacements qui, en plus de la cache dynamique par défaut, permettent à la cache dynamique de stocker, distribuer et partager le résultat et les effets secondaires d'un servlet appelé. En configurant une instance de cache de servlet, vous conférez aux applications une souplesse plus grande et une meilleure optimisation des ressources de la mémoire cache. Le nom JNDI (Java Naming and Directory Interface) défini pour l'instance de cache dans la console d'administration est associé à l'élément d'instance de cache dans le fichier de configuration `cachespec.xml`. Tous les éléments <cache-entry> spécifiés dans un élément <cache-instance> sont créés dans cette instance de cache spécifique. Tous les éléments <cache-entry> spécifiés en dehors d'un élément <cache-instance> sont stockés dans l'instance de cache dynamique par défaut. Voir Instances de cache pour plus d'informations sur les instances de cache de type objet et servlet.

Cette procédure s'applique à la version 8.0 de la console d'administration WebSphere Application Server. Ces informations peuvent varier légèrement en fonction de la version de WebSphere Application Server que vous utilisez.

Remarque :

- WebSphere eXtreme Scale Version 8.6 n'est pas pris en charge sur les versions de WebSphere Application Server antérieures à la version 7.0.
- La procédure suivante s'applique à la topologie de cache dynamique WebSphere eXtreme Scale distante. Toutes les autres topologies, notamment intégrées, partitionnées intégrées et locales, sont obsolètes dans WebSphere eXtreme Scale Version 8.6.

Procédure

- Pour configurer un cache d'objet ou de servlet avec la console d'administration WebSphere Application Server, procédez comme suit :
  1. Démarrez la console d'administration WebSphere Application Server.
  2. Dans le menu supérieur, cliquez sur **Ressources > Instances de cache > Instances de cache d'objet**.
  3. Dans la zone des **instances de cache d'objet**, sélectionnez le type d'instance de cache à créer. Il peut s'agir d'une instance de cache d'objet ou de servlet.
  4. Définissez la portée de l'instance de cache. Spécifiez une portée de cellule pour que l'instance de cache soit disponible pour tous les serveurs de la cellule. Une portée de noeud rend l'instance de cache disponible pour tous les serveurs d'un noeud. Une portée de serveur rend l'instance de cache disponible pour le serveur sélectionné uniquement. Si nécessaire, vous pouvez combiner les portées.
  5. Cliquez sur **Appliquer** et enregistrez la portée.
  6. Cliquez sur **Nouveau** et définissez une instance de cache d'objet.
  7. Dans la liste déroulante **Fournisseur de cache**, sélectionnez WebSphere eXtreme Scale.

     Remarque : Si WebSphere eXtreme Scale n'apparaît pas comme fournisseur de cache dynamique, cela implique que le profil WebSphere Application Server n'a pas été étendu pour WebSphere eXtreme Scale. Pour plus d'informations, voir «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 243.
  8. Spécifiez le nom JNDI de l'instance de cache dynamique. Pour les objets cache, ce nom sera utilisé lors de la consultation du cache. Pour les caches de servlet, il s'agit du nom d'attribut défini dans l'élément <cache-instance> dans le fichier `cachespec.xml`.
  9. Spécifiez le nom JNDI de l'instance de cache dynamique.
  10. Si vous voulez changer la taille du cache, définissez-la dans la zone de **taille de cache**. La taille de cache définit le nombre maximal d'entrées autorisées dans chaque partition dans une grille WebSphere eXtreme Scale pour l'instance de cache dynamique. La valeur par défaut est 2 000 entrées dans chaque partition.
  11. Cochez la case d'**activation de la réplication de cache**. Dans ce cas, les données en cache sont stockées à distance dans la grille et non pas localement. Vous devez cocher la case lorsque vous utilisez WebSphere eXtreme Scale comme fournisseur de cache.
  12. Cliquez sur **Appliquer** ou **OK** et enregistrez la configuration.
  13. Cliquez sur **Nouveau** pour définir les propriétés personnalisées de l'instance de cache.
  14. Dans la zone **Nom**, définissez `com.ibm.websphere.xs.dynacache.topology`
  15. Dans la zone **Valeur**, définissez `remote`.
  16. Sélectionnez `java.lang.String` dans la zone de liste déroulante **Type**.
  17. Cliquez sur **Appliquer** ou **OK** et enregistrez la configuration.
  18. Redémarrez le gestionnaire de déploiement et tous les processus serveur d'applications.
- Pour configurer un cache d'objet ou de servlet en utilisant le fichier `cacheinstances.properties`, procédez comme suit :
  1. Créez un fichier `cacheinstances.properties`. Voir «Fichier de propriétés des instances de cache», à la page 406 pour le contenu nécessaire.

2. Placez le fichier `cacheinstances.properties` dans le serveur d'applications ou le chemin d'accès aux classes d'application. Par exemple, vous pouvez utiliser votre fichier WAR (Web application archive) d'application, le répertoire `WEB-INF\classes` ou créer un répertoire `server_root\classes` et y placer le fichier.

Personnalisation d'une instance de cache dynamique avec des propriétés personnalisées

Java

WebSphere Application Server permet de définir des propriétés personnalisées dans une instance de cache dynamique.

Avant de commencer

Vous devez avoir configuré une instance de cache par défaut ou disposer d'un objet supplémentaire ou d'une instance de cache de type servlet. Voir «Configuration de l'instance de cache dynamique par défaut (baseCache)», à la page 401 or «Configuration des instances de cache dynamique d'objet ou de servlet», à la page 402.

Pourquoi et quand exécuter cette tâche

Vous pouvez définir des propriétés personnalisées spécifiques d'une instance de cache dynamique en procédant comme suit :

- Utilisez la console d'administration WebSphere Application Server avec l'APAR PM71992 si vous voulez définir des propriétés spécifiques d'une instance de cache dynamique donnée. Si vous ne disposez pas de ce correctif, contactez WebSphere Application Server sur la page du support http://www.ibm.com/software/webservers/appserv/was/support.
- Si vous avez créé un fichier `cacheinstances.properties`, vous pouvez définir des propriétés personnalisées dans ce fichier. Cette méthode ne peut pas être utilisée pour définir des propriétés personnalisées pour une instance de cache dynamique (baseCache) par défaut.
- Utilisez la console d'administration WebSphere Application Server pour changer les valeurs des propriétés personnalisées JVM (Java virtual machine).

  Remarque : Les propriétés JVM peuvent affecter toutes les instances de cache dans une machine JVM donnée.

  Remarque : La portée des propriétés JVM peut s'appliquer à toutes les instances de cache dans une machine JVM WebSphere Application Server JVM. Par conséquent, il est préférable d'utiliser des propriétés personnalisées de cache dans la console d'administration WebSphere Application Server (avec l'APAR PM71992 pour une instance de cache par défaut), ou un fichier `cacheinstances.properties` dans la plupart des cas.

Procédure

- Pour définir une propriété personnalisée dans une instance de cache dans la console d'administration WebSphere Application Server, procédez comme suit :
  1. Démarrez la console d'administration WebSphere Application Server.

     Remarque : Ces étapes ne peuvent pas être exécutées pour une instance (baseCache) par défaut tant que vous n'avez pas appliqué l'APAR R PM71992 WebSphere Application Server. Ce correctif est disponible dans les versions

WebSphere Application Server 7.0.0.27, 8.0.0.6, 8.5.0.2 et les versions suivantes. Si vous ne disposez pas de correctif, consultez la page WebSphere Application Server Support, http://www.ibm.com/software/webservers/appserv/was/support.

2. Accédez à l'instance de cache désirée que vous avez configurée.
3. Dans le panneau d'instance de cache, cliquez sur **Propriétés supplémentaires > Propriétés personnalisées**.
4. Sélectionnez **Nouveau** et définissez le nom et la valeur de la propriété personnalisée.
5. Cliquez sur **Appliquer** ou **OK** et enregistrez la configuration.
6. Redémarrez le gestionnaire de déploiement et tous les processus serveur d'applications.

- Pour définir une propriété personnalisée pour une instance de cache en utilisant le fichier `cacheinstances.properties`, procédez comme suit :

  Remarque : Ces étapes ne peuvent pas être exécutées pour une instance (baseCache) par défaut.

  1. Ajoutez la propriété personnalisée à un fichier `cacheinstances.properties`. Si vous devez créer ce fichier, voir «Fichier de propriétés des instances de cache» pour connaître le contenu requis.
  2. Placez le fichier `cacheinstances.properties` dans le serveur d'applications ou le chemin d'accès aux applications. Par exemple, vous pouvez utiliser votre fichier WAR (Web application archive) d'application, le répertoire `WEB-INF\classes` ou créer un répertoire `server_root\classes` pour l'y placer.
- Utilisez la console d'administration WebSphere Application Server pour changer les valeurs des propriétés personnalisées JVM (Java virtual machine). Voir Propriétés personnalisées JVM (java virtual machine) pour plus d'informations.

Fichier de propriétés des instances de cache :

Pour configurer un cache d'objet ou de servlet en utilisant le fichier `cacheinstances.properties`.

Tableau 29. Propriétés des instances de cache

Nom de la propriété : – x est le numéro de l'instance	Requis	Portée	Valeur possible	Description
cache.instance.x	Oui	Par instance de cache	n'importe quelle chaîne (pas de définition par défaut)	Spécifie le nom de l'instance du cache ou le nom JNDI.
cache.instance.x.cacheSize	Non	Par instance de cache	> 0 (par défaut =2000)	Indique le nombre maximal d'entrées autorisées dans une partition dans la grille WebSphere eXtreme Scale. Multipliez cette valeur par le nombre de partitions pour déterminer la capacité du cache dans la grille WebSphere eXtreme Scale.
cache.instance.x.createCacheAtServerStartup	Non	Par instance de cache	True ou false (par défaut=false)	Indique si l'instance de cache configurée est créée lors du démarrage du serveur.
cache.instance.x.enableServletSupport	Non	Par instance de cache	True ou false (par défaut=false)	Spécifie si l'instance de cache correspond au cache des servlets ou au cache d'objets.
cache.instance.x.enableCacheReplication	Oui (uniquement jusqu'à l'APAR)	Par instance de cache	True ou false (par défaut=false)	Indique que le cache est éloigné du serveur d'applications. Cette propriété doit avoir la valeur True dans la topologie distante WebSphere eXtreme Scale.
cache.instance.x.cacheProviderName	Oui	Par instance de cache	com.ibm.ws.objectgrid.dynacache.CacheProviderImpl	Indique le fournisseur de cache dynamique. Le fournisseur WebSphere Application Server est utilisé par défaut si WebSphere eXtreme Scale n'est pas défini.
cache.instance.x.ignoreValueInInvalidationEvent	Non	Par instance de cache	True ou false (par défaut=false)	Spécifie si la valeur de cache de l'événement d'invalidation est ignorée. Si la propriété est définie comme true, la valeur en cache de l'événement est défini comme NULL lorsque le code est retourné à l'appelant.
cache.instance.x.<custom property>	Dépend de la propriété à ajouter.	Par instance de cache	Dépend de la propriété à ajouter.	Vous pouvez ajouter n'importe quelle propriété à ce fichier.

Propriétés personnalisées de cache dynamique :

Reportez-vous à ce tableau pour définir les propriétés personnalisées d'une instance de cache dynamique par défaut ou d'une instance de cache de servlet ou d'objet.

Tableau 30. Propriétés personnalisées de cache dynamique

Nom de la propriété	Requis	Portée	Valeur possible	Description
com.ibm.websphere.xs.dynacache.topology	Non	Par instance de cache	(default=remote)	Indique la topologie de l'instance de cache. Les topologies intégrées, partitionnées intégrées et locales sont obsolètes.
com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent	Non	Par instance de cache	true ou false (false, par défaut)	Spécifie si la valeur en cache de l'événement d'invalidation est ignorée. Si la propriété est définie comme true, la valeur en cache de l'événement est défini comme NULL lorsque le code est retourné à l'appelant.
com.ibm.websphere.xs.dynacache.ignore_value_in_change_event	Non	Par instance de cache	true ou false (false, par défaut)	Spécifie si la valeur en cache de l'événement de changement est ignorée. Si la valeur est true, la valeur de cache de l'événement de changement est NULL lorsque le code est retourné à l'appelant.
com.ibm.websphere.xs.dynacache.cs_override	Non	Par instance de cache	Noeud final de service de catalogue (ex. : 9.5.12.345:2819)	Indique le noeud final de service de catalogue de la grille WXS à associer à l'instance de cache. Cette zone est requise si elle n'est pas définie dans la console d'administration WAS.
com.ibm.websphere.xs.dynacache.grid_name	Non	Par instance de cache	N'importe quelle chaîne (default=DYNACACHE_REMOTE)	Indique le nom de la grille d'objets ObjectGrid distante. Il doit correspondre au nom utilisé lors du démarrage des serveurs ObjectGrid.
com.ibm.websphere.xs.dynacache.map_name	Non	Par instance de cache	N'importe quelle chaîne (non utilisé par défaut)	Indique le mappage ObjectGrid à associer à l'instance de cache. Un modèle de mappe est utilisé par défaut. Il est utilisé uniquement si un modèle de mappe n'est pas désiré.
com.ibm.websphere.xs.dynacache.map_template_name	Non	Par instance de cache	N'importe quelle chaîne (default=IBM_DC_PARTITIONED_.*)	Indique le nom du préfixe du modèle de mappe. Il doit correspondre au nom utilisé lors du démarrage des serveurs ObjectGrid.
com.ibm.websphere.xs.dynacache.cache_name	Non	Par instance de cache	N'importe quelle chaîne (default=valeur dans cache.instance.x)	Indique le nom du suffixe unique utilisé comme nom du modèle de mappe. Par exemple, `IBM_DC_PARTITIONED.<cache_name>`
com.ibm.websphere.xs.dynacache.near_cache_size	Non	Par instance de cache	> 0 (default=valeur définie dans cache.instance.x.cacheSize)	Indique le nombre maximal d'entrées autorisées dans une instance de cache local. Par défaut cette valeur est égale au nombre maximal d'entrées autorisées dans une partition dans la grille d'objets distante de l'instance de cache.

Plug-in de cache niveau 2 (L2) JPA

Java

WebSphere eXtreme Scale inclut des plug-in de mémoire cache de niveau 2 pour les fournisseurs OpenJPA et Hibernate Java Persistence API (JPA). Lorsque vous utilisez l'un de ces plug-in, l'application utilise l'API JPA. Une grille de données est introduite ente l'application et la base de données pour améliorer les temps de réponse.

L'utilisation d'eXtreme Scale en tant que fournisseur de cache de niveau 2 améliore les performances lors de la lecture et de l'interrogation des données et réduit la charge pesant sur la base de données. WebSphere eXtreme Scale présente plusieurs avantages par rapport aux implémentations de cache pré-intégrées car le cache est automatiquement répliqué entre tous les processus. Lorsqu'un client met une valeur en cache, tous les autres clients peuvent utiliser la valeur mise en cache en local.

Vous pouvez configurer la topologie et les propriétés pour le fournisseur de cache L2 dans le fichier `persistence.xml`. Pour plus d'informations sur la configuration de ces propriétés, voir «Propriétés de configuration du cache JPA pour Hibernate Version 4.0», à la page 414.

Conseil : Le plug-in de cache L2 JPA requiert une application qui utilise les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder à une source de données JPA, utilisez le chargeur JPA. Pour plus d'informations, voir Chargeurs JPA.

Remarques relatives à la topologie cache L2 JPA

Les facteurs suivants affectent le type de topologie à configurer :

1. **Quelle quantité de données voulez-vous placer en mémoire cache ?**
   - Si les données peuvent tenir dans un seul segment de mémoire JVM, utilisez la «Topologie imbriquée», à la page 409 ou «Topologie intra-domaine».
   - Dans le cas contraire, utilisez la «Topologie imbriquée et partitionnée», à la page 410 ou «Topologie distante», à la page 412
2. **Quel est le taux de lecture/écriture prévu ?**

   Ce taux affecte les performances du cache L2. Chaque topologie gères différemment les opérations de lecture et d'écriture.
   - «Topologie imbriquée», à la page 409 : lecture locale, écriture distante
   - «Topologie intra-domaine» : lecture locale, écriture locale
   - «Topologie imbriquée et partitionnée», à la page 410 : partitionnée : lecture distante, écriture distante
   - «Topologie distante», à la page 412 : lecture distante, écriture distante.

   Les applications qui fonctionnent principalement en lecture seule doivent utiliser des topologies intra-domaines lorsque cela est possible. Les applications qui exécutent des opération d'écriture principalement doivent utiliser des topologies intra-domaines.
3. **Quel est le pourcentage de données recherchées par rapport au pourcentage de données trouvées par une clé ?**

   Lorsque le cache des requêtes JPA est activé, les opérations d'interrogation l'utilisent. Activez ce cache pour les applications avec des taux de lecture/écriture élevés uniquement, par exemple, lorsque vous approchez de 99 % d'opérations de lecture. Si vous utilisez le cache des requêtes JPA, vous devez utiliser «Topologie imbriquée», à la page 409 ou «Topologie intra-domaine».

   L'opération de recherche par clé recherche une entité cible si l'entité cible n'a pas de relation. Si l'entité cible a des relations avec le type de recherche EAGER, ces relations sont recherchées avec l'entité cible. Dans le cache de données JPA, un petit nombre de réussites en mémoire obtient toutes les données de relation lors de la recherche de ces relations.
4. **Quel est niveau d'obsolescence toléré des données ?**

   Dans un système comportant un petit nombre de machines JVM, il existe une latence de réplication des données pour les opérations d'écriture. Le cache a pour fonction de gérer une vue de données synchronisée dans toutes les machines JVM. Lorsque vous utilisez la topologie intra-domaine, il existe un délai de réplication de données pour les opérations d'écriture. Les applications qui utilisent cette topologie doivent pouvoir tolérer les lectures obsolètes et les écritures simultanées qui peuvent remplacer les données.

Topologie intra-domaine

Avec une topologie intra-domaine, les fragments primaires sont placés sur chaque serveur de conteneur dans la topologie. Ces fragments primaires contiennent l'ensemble des données de la partition. N'importe lequel de ces fragments

primaires peut également exécuter des opérations d'écriture dans la mémoire cache. Cette configuration élimine le goulot d'étranglement dans la topologie intégrée dans lequel toutes les opérations d'écriture de la mémoire cache doivent passer par un fragment primaire unique.

Dans une topologie intra-domaine, aucun fragment de réplique n'est créé, même si vous avez défini des répliques dans vos fichiers de configuration. Chaque fragment primaire redondant contient une copie complète des données, de sorte qu'il peut également être considéré comme un fragment de réplique. Cette configuration utilise une partition unique, similaire à la topologie intégrée.

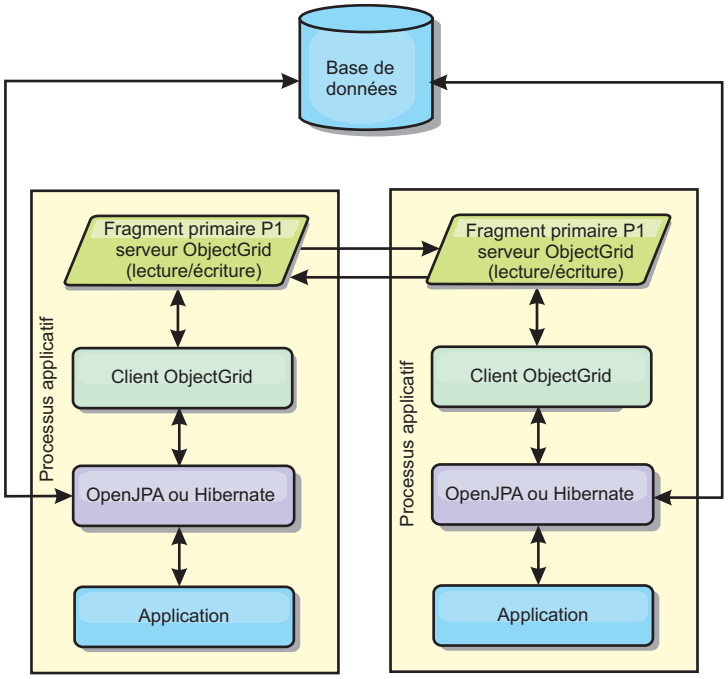


Figure 43. Topologie intra-domaine JPA

Propriétés de configuration du cache JPA associées pour la topologie intra-domaine :

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED,PlacementScope=CONTAINER_SCOPE,PlacementScopeTopology=HUB | RING
```

Avantages :

- Lectures de cache et des mises à jour localement
- Simple à configurer.

Limitations :

- Cette topologie est la mieux adaptée lorsque les serveurs de conteneur peuvent contenir l'ensemble des données de la partition.
- Les fragments de réplique, même s'ils sont configurés, ne sont jamais placés, car chaque serveur de conteneur héberge un fragment primaire. Toutefois, tous les fragments primaires sont répliqués avec les autres fragments primaires, de sorte que ces fragments primaires deviennent des répliques les uns des autres.

Topologie imbriquée

Conseil : Envisagez d'utiliser une topologie intra-domaine pour obtenir de meilleures performances.

Une topologie imbriquée crée un serveur de conteneur dans l'espace de traitement de chaque application. Les plug-in OpenJPA et Hibernate lisent directement la copie en mémoire du cache et écrivent dans toutes les autres copies. Vous pouvez améliorer les performances d'écriture à l'aide de la réplication asynchrone. Cette topologie par défaut produit un résultat optimal lorsque la quantité de données mises en cache est suffisamment réduite pour être traitée par un seul processus. Avec une topologie intégrée, créez une seule partition pour les données.

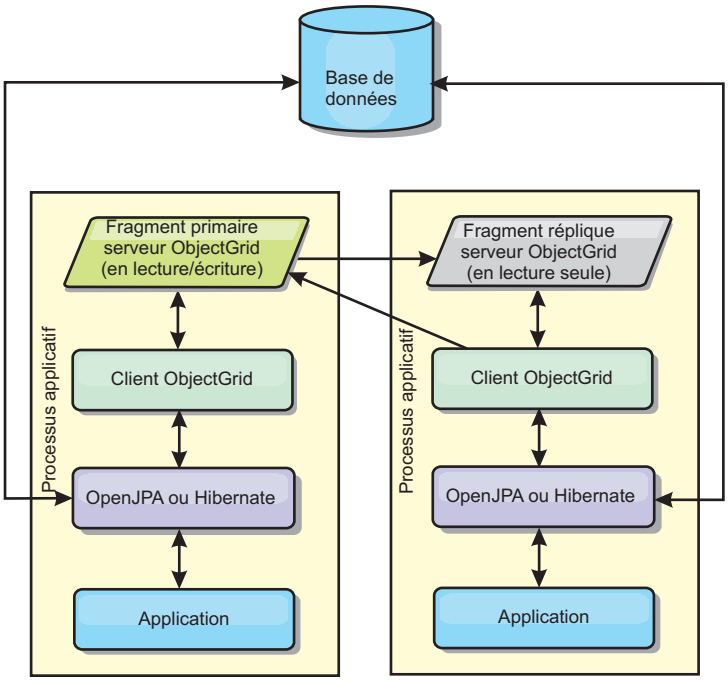


Figure 44. Topologie imbriquée JPA

Propriétés de configuration du cache JPA de la topologie intégrée :

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED,MaxNumberOfReplicas=num_replicas,ReplicaMode=SYNC | ASYNC | NONE
```

Avantages :

- Toutes les lectures de cache sont des accès locaux rapides.
- Simple à configurer.

Limitations :

- La quantité de données est limitée à la taille du processus.
- Toutes les mises à jour de cache sont envoyées via un fragment primaire, ce qui crée un goulot d'étranglement.

Topologie imbriquée et partitionnée

Conseil : Envisagez d'utiliser une topologie intra-domaine pour obtenir de meilleures performances.

ATTENTION :
N'utilisez pas le cache des requêtes JPA avec un topologie partitionnée. Le cache de requêtes stocke les résultats des requêtes qui sont une collection de clés d'entité. Le cache de requêtes recherche toutes les données d'entité dans le cache de données. Comme l'antémémoire données est divisée entre plusieurs processus, ces appels supplémentaires peuvent faire perdre les avantages du cache L2.

Lorsque les données en mémoire cache sont trop volumineuses pour tenir dans un seul processus, vous pouvez utiliser la topologie partitionnée intégrée. Cette topologie divise le données dans plusieurs processus. Les données sont divisées entre les fragments primaires de sortie que chaque fragment primaire contient un sous-ensemble des données. Vous pouvez toujours utiliser cette option lorsque la latence de la base de données est élevée.

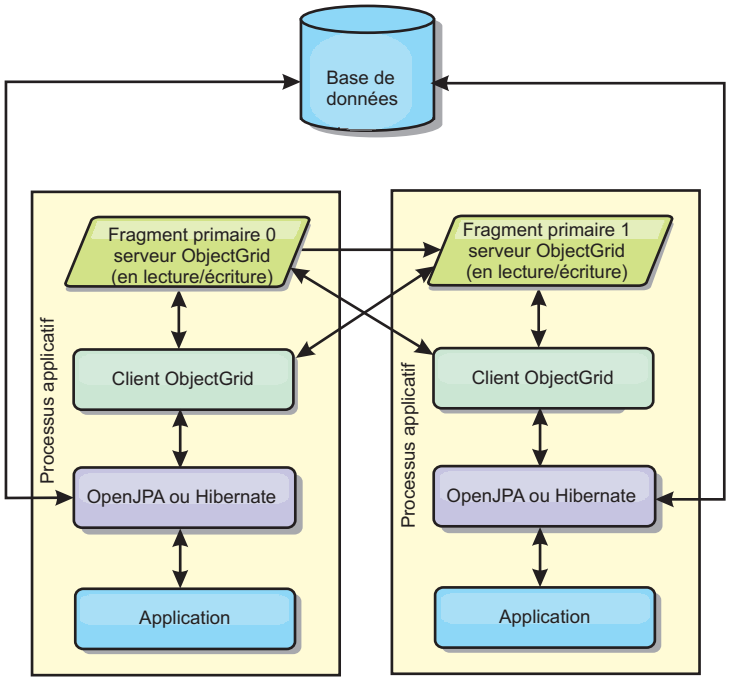


Figure 45. Topologie imbriquée et partitionnée JPA

Propriétés de configuration du cache JPA de la topologie partitionnée intégrée :

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED_PARTITION,ReplicaMode=SYNC | ASYNC | NONE,
NumberOfPartitions=num_partitions,ReplicaReadEnabled=TRUE | FALSE
```

Avantages :

- Stocke de grandes quantités de données.
- Simple à configurer.
- Les mises à jour de cache sont réparties sur plusieurs processus.

Limitation :

- La plupart des lectures et des mises à jour de cache sont distantes.

Par exemple, pour mettre en cache 10 Go de données avec un maximum de 1 Go par machine JVM, 10 machines virtuelles Java sont nécessaires. Le nombre de partitions doit par conséquent être défini sur 10 ou plus. Idéalement, le nombre de partitions doit être un nombre premier où chaque fragment stocke une quantité

raisonnable de mémoire. Le paramètre numberOfPartitions est généralement égal au nombre de machines virtuelles Java. Chaque machine virtuelle Java stocke une partition à l'aide de ce paramètre. Si vous activez la réplication, vous devez augmenter le nombre de machines virtuelles Java dans le système. Dans le cas contraire, chaque machine virtuelle Java stocke également une réplique de partition qui consomme autant de mémoire que la partition principale.

Consultez la rubrique relative à la définition de la taille de la mémoire et au calcul du nombre de partitions dans le *Guide d'administration* pour optimiser les performances de la configuration choisie.

Par exemple, dans un système avec quatre machines virtuelles Java et avec la valeur de paramètre numberOfPartitions 4, chaque machine virtuelle Java héberge une partition principale. Une opération de lecture a 25 pourcents de chances d'extraire des données d'une partition disponible en local, ce qui est sensiblement plus rapide qu'à partir d'une machine virtuelle Java distante. Si une opération de lecture, telle que l'exécution d'une requête, doit extraire une collection de données impliquant une répartition égale de quatre partitions, 75 pourcents des appels sont distants et 25 pourcents sont locaux. Si le paramètre ReplicaMode est défini sur `SYNC` ou `ASYNC` et si le paramètre ReplicaReadEnabled est défini sur `true`, quatre répliques de partitions sont créées et réparties entre quatre machines virtuelles Java. Chaque machine virtuelle Java héberge une partition principale et une réplique. L'opération de lecture a désormais à 50 pourcents de chances de s'exécuter en local. L'opération de lecture qui extrait une collection de données impliquant une répartition égale de quatre partitions comporte 50 pourcents d'appels distants et 50 pourcents d'appels locaux. Les appels locaux sont considérablement plus rapides que les appels distants. Dès que des appels distants sont effectués, les performances chutent.

Topologie distante

ATTENTION :
N'utilisez pas le cache des requêtes JPA avec une topologie distante. Le cache des requêtes stocke les résultats des requêtes qui sont une collection de clés d'entité. Le cache de requêtes recherche toutes les données d'entité dans le cache de données. Comme l'antémémoire données est distante, ces appels supplémentaires peuvent faire perdre les avantages du cache L2.

Conseil : Envisagez d'utiliser une topologie intra-domaine pour obtenir de meilleures performances.

Une topologie distante stocke toutes les donnés mises en cache dans un ou plusieurs processus, ce qui réduit la sollicitation de la mémoire par les processus applicatifs. Vous pouvez tirer parti de la répartition de vos données dans des processus distincts en déployant une grille de données eXtreme Scale partitionnée répliquée. Contrairement aux configurations intégrées et intégrées et partitionnées décrites dans les sections précédentes, si vous souhaitez gérer la grille de données distante, vous devez le faire indépendamment de l'application et du fournisseur JPA.

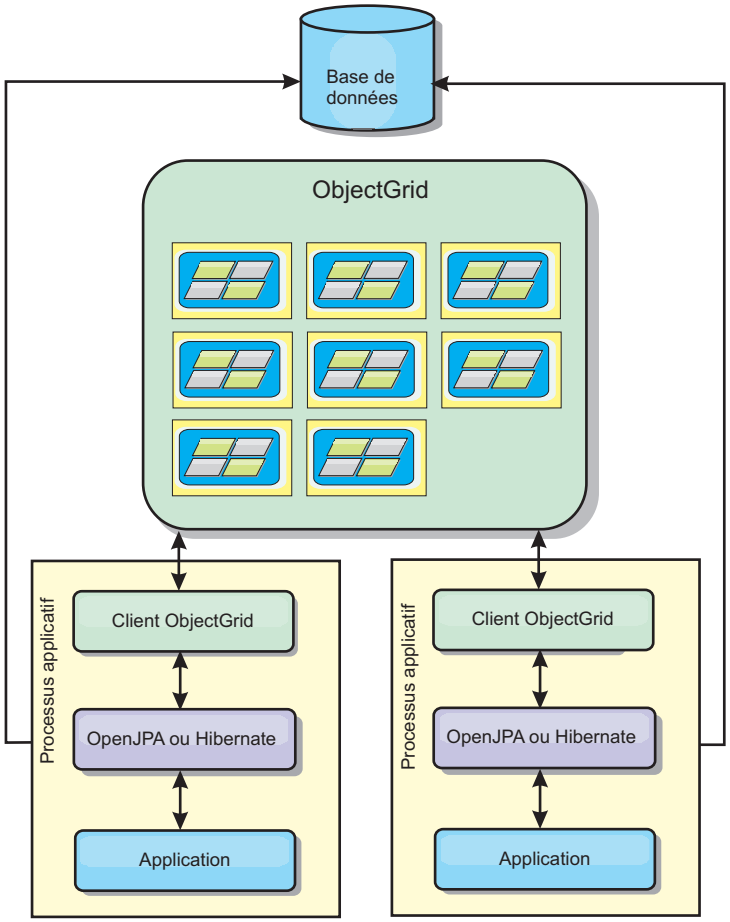


Figure 46. Topologie distante JPA

Propriétés de configuration du cache JPA de la topologie distante :
`ObjectGridName=objectgrid_name,ObjectGridType=REMOTE`

Le type d'ObjectGrid REMOTE ne nécessite pas de paramètres de propriété car l'ObjectGrid et la règle de déploiement sont définis distinctement de l'application JPA. Le plug-in de cache JPA se connecte à distance à un ObjectGrid éloigné existant.

Toute interaction avec la grille d'objets étant éloignée, cette topologie offre les moins bonnes performances parmi tous les types de grille d'objets.

Avantages :

- Stocke de grandes quantités de données.
- Le processus applicatif est exempt de données en cache.
- Les mises à jour de cache sont réparties sur plusieurs processus.
- Options de configuration souples.

Limitation :

- Toutes lectures et mises à jour de cache sont distantes.

Migration vers Hibernate Version 4.0

Java

WebSphere eXtreme Scale Version 8.5 contient deux plug-ins de cache de niveau 2 pour le fournisseur Hibernate Version 3.0 Java Persistence API (JPA).. Si vous voulez migrer vers Hibernate Version 4.0, vous devez changer une propriété de cache L2.

Avant de commencer

Arrêtez les serveurs qui exécutent des applications avec le plug-in Hibernate Version 3.0 ou dans un environnement WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Pour pouvoir migrer vers Hibernate Version 4.0, vous devez remplacer provider_class property par region.factory_class dans le fichier `persistence.xml`.

Procédure

1. Ouvrez le fichier persistence.xml et recherchez la propriété suivante :

```
<property name="hibernate.cache.provider_class"
 value="com.ibm.com.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider"/>
```

2. Mettez à jour la propriété comme suit :

```
<property name="hibernate.cache.region.factory_class"
 value="com.ibm.com.ws.objectgrid.hibernate.cache.WXSRegionFactory"/>
```

Propriétés de configuration du cache JPA pour Hibernate Version 4.0

Java

WebSphere eXtreme Scale inclut des plug-ins de cache L2 pour le fournisseur Hibernate Version 4.0 Java Persistence API (JPA). Pour configurer le plug-in de cache L2, vous devez mettre à jour les propriétés dans le fichier persistence.xml.

Conseil : Le plug-in de cache L2 JPA requiert une application qui utilise les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder à une source de données avec JPA, utilisez le chargeur JPA. Pour plus d'informations, voir «Configuration des chargeurs JPA», à la page 431.

Emplacement des propriétés 4.0

Vous pouvez définir ces propriétés individuellement dans le fichier persistence.xml.

```
<property name="wxs.<name_of_property>"
 value="<value>"/>
```

Vous pouvez définir les propriétés dans DataCache ou QueryCache, par exemple :

```
<property name="wxs.objectgrid_name"
 value="BasicObjectGrid"/>
```

Propriétés

Vous pouvez configurer le plug-in de cache Hibernate Version 4.0 JPA avec les propriétés suivantes. Les valeurs par défaut sont utilisées si vous ne spécifiez pas de valeurs dans la configuration.

wxs.objectgrid_name
: Indique le nom unique d'ObjectGrid. La valeur par défaut est le nom d'unité de persistance défini. Si le nom de l'unité de persistance n'est pas disponible auprès du fournisseur, un nom généré est utilisé.

wxs.objectgrid_type
Indique le type d'ObjectGrid.

Valeurs admises :

EMBEDDED
Type de configuration par défaut et recommandée. es valeurs par défaut sont `wxs.number_of_partitions=1`, `wxs.replica_mode=SYNC`, `wxs.replica_read_enabled=true` et `wxs.max_number_of_replicas=47`. Utilisez le paramètre **wxs.replica_mode** pour définir le mode de réplication et le paramètre **wxs.max_number_of_replicas** pour définir le nombre maximal de répliques. Si un système possède plus de 47 machines virtuelles Java, affectez à **wxs.max_number_of_replicas** une valeur égale au nombre de machines virtuelles Java.

EMBEDDED_PARTITION
Type à utiliser si le système doit mettre en cache une quantité de données importante sur un système réparti. Le nombre de partitions par défaut est 47 avec le mode de réplique `NONE`. Dans un petit environnement comportant quelques machines virtuelles Java, affectez à **wxs.number_of_partitions** une valeur égale ou inférieure au nombre de machines virtuelles Java. Vous pouvez définir les valeurs **wxs.replica_mode**, **wxs.number_of_partitions** et **wxs.replica_read_enabled** pour optimiser le système.

REMOTE Le cache tente de se connecter à un ObjectGrid réparti distant à partir du service de catalogue.

wxs.max_number_of_replicas
Indique le nombre maximal de répliques à utiliser pour le cache. Cette valeur s'applique au type `EMBEDDED` uniquement. Ce nombre doit être égal ou supérieur au nombre de machines virtuelles Java dans un système. La valeur par défaut est 47.

Valeur valides : supérieures ou égales à 1

wxs.max_used_memory

Valeurs valides : `TRUE` ou `FALSE`

Active l'expulsion des entrées du cache si la mémoire est soumise à des contraintes. La valeur par défaut est `TRUE` et les données sont expulsées lorsque le seuil d'utilisation des segments de mémoire de la machine virtuelle Java dépasse 70 pourcent. Vous pouvez modifier le pourcentage de seuil d'utilisation du segment de mémoire JVM par défaut en définissant la propriété memoryThresholdPercentage dans le fichier des propriétés et en plaçant ce fichier dans le chemin d'accès aux classes. Pour plus d'informations sur les expulseurs, voir les Plug-in d'expulsion d'objets du cacheinformations sur les expulseurs dans *Présentation du produit*. Pour plus d'informations sur le fichier de propriétés du serveur, voir Fichier de propriétés du serveur.

wxs.number_of_partitions

Valeurs valides : supérieures ou égale à 1

Indique le nombre de partitions à utiliser pour le cache. Cette propriété s'applique lorsque la valeur **wxs.objectgrid_type** est `EMBEDDED_PARTITION`. La valeur par défaut est 47. Pour le type `EMBEDDED`, la valeur **wxs.number_of_partitions** est toujours 1.

wxs.placement_scope

Indique la granularité d'une instance d'un groupe de mappes.

Valeurs admises :

DOMAIN_SCOPE
(Défaut) Place un fragment primaire pour chaque partition sur un serveur de conteneur dans le domaine de service de catalogue. Les fragments de réplique pour chaque partition sont placés sur les autres serveurs de conteneur dans le domaine de service de catalogue.

CONTAINER_SCOPE
Place un fragment primaire sur chaque serveur de conteneur dans le domaine de service de catalogue

wxs.placement_scope_topology

Définit la topologie de liaison du serveur de conteneur dans le domaine de service de catalogue. Cette valeur est utilisée uniquement lorsque la valeur **wxs.placement_scope** n'est pas DOMAIN_SCOPE.

Valeurs admises :

HUB (Défaut) Si la topologie de concentrateur est sélectionnée, une grille de données unique est sélectionnée pour être le concentrateur. Toutes les autres grilles de données se connectent au concentrateur. Cette topologie est assez évolutive, car branches ont une connexion unique. Le concentrateur peut devenir un goulot d'étranglement et un point unique de défaillance temporaire. Le concentrateur est transféré vers un autre serveur de conteneur lorsque il est défaillant. Cette configuration offre l'avantage de pouvoir écrire plus de code que ne le permet un seul point, le concentrateur, pour gérer toutes les collisions.

RING Si la topologie en anneau est sélectionnée, chaque grille de données est liée à deux autres grilles de données. L'ordre des liaisons n'est pas garantie. Toutefois, chaque conteneur qui démarre est probablement liée au premier conteneur et au dernier conteneur à ajouter à l'anneau. Cette topologie est la plus évolutive, mais il suffit de deux liaisons défaillantes pour qu'une déconnexion temporaire se produise. Si les serveurs de conteneur sont défaillants, des liaisons sont établies parmi les survivants après que l'échec a été détecté.

wxs.replica_mode

Valeurs valides : SYNC/ASYNC/NONE

Indique la méthode utilisée pour copier le cache vers les fragments réplique. Cette propriété s'applique lorsque la valeur de **wxs.objectgrid_type** est EMBEDDED ou EMBEDDED_PARTITION. La valeur par défaut est NONE pour le type EMBEDDED_PARTITION et SYNC pour le type EMBEDDED. Si la valeur de **wxs.replica_mode** est NONE pour EMBEDDED **wxs.objectgrid_type**, le type EMBEDDED utilise toujours **wxs.replica_mode** pour SYNC.

wxs.replica_read_enabled

Valeurs valides : TRUE ou FALSE

Si cette propriété est activée, les clients lisent les valeurs à partir des fragments réplique. Cette propriété s'applique au type EMBEDDED_PARTITION.

La valeur par défaut est FALSE pour le type EMBEDDED_PARTITION. Le type EMBEDDED affecte toujours à **wxs.replica_read_enabled** la valeur TRUE.

wxs.write_behind

Pour les fournisseurs Hibernate uniquement : lorsque **wxs.write_behind** est activé, les mises à jour sont temporairement stockées dans un espace de stockage de données de portée JVM jusqu'à ce que la condition **wxs.write_behind_interval** ou **wxs.write_behind_max_batch_size** soit vraie.

Avertissement : Si la fonction **wxs.write_behind** n'est pas activée, les autres paramètres de configuration de l'écriture différée sont supprimés.

Important : Utilisez la fonction **wxs.write_behind** avec précaution. Les configurations d'écriture différée allongent la latence de synchronisation des données dans toutes les machines JVM et augmentent le risque de perte de données. Dans un système qui utilise la configuration d'écriture différée avec au moins quatre machines JVM, la mise à jour effectuée sur une machine virtuelle Java correspond à un délai d'environ 15 secondes avant que la mise à jour soit disponible pour les autres machines JVM. Si deux des machines virtuelles Java actualisent la même entrée, la première qui vide la mise à jour perd sa mise à jour.

Valeurs valides : TRUE ou FALSE

Valeur par défaut : FALSE

wxs.write_behind_interval

Pour les fournisseurs Hibernate uniquement : spécifie l'intervalle de temps en millisecondes de vidage des mises à jour dans le cache.

Valeurs valides : supérieures ou égales à 1

Valeur par défaut : 5000 (5 secondes).

wxs.write_behind_pool_size

Pour les fournisseurs Hibernate uniquement : spécifie la taille maximale du pool d'unités d'exécution utilisé pour vider les mises à jour dans le cache.

Valeurs valides : supérieures ou égales à 1

Valeur par défaut : 5

wxs.write_behind_max_batch_size

Pour les fournisseurs Hibernate uniquement : spécifie la taille de lot maximale par cache de région pour le vidage des mises à jour dans le cache. Par exemple, si la taille est 1 000 et que les mises à jour stockées dans le stockage d'écriture différée d'un cache de région dépasse 1 000 entrées, les mises à jour sont vidées dans le cache, même si la condition **wxs.write_behind_interval** définie n'est pas satisfaite. Les mises à jour sont vidées dans le cache à peu près dans le délai en secondes spécifié par la valeur **wxs.write_behind_interval** ou lorsque la taille du stockage d'écriture différée dans chacun des caches de région dépasse 1 000 entrées. Notez que si la condition **wxs.write_behind_max_batch_size** est satisfaite, seul le cache de région qui remplit cette condition vide ses mises à jour en écriture différée de dans le cache. Un cache de région correspond généralement à une entité ou à une requête.

Valeurs valides : supérieures ou égales à 1

Valeur par défaut : `1000`

Configuration du plug-in de cache OpenJPA

Java

Vous pouvez configurer les deux implémentations DataCache et QueryCache pour OpenJPA.

Avant de commencer

- Vous devez déterminer la topologie du plug-in de cache JPA à utiliser. Voir «Plug-in de cache niveau 2 (L2) JPA», à la page 407 pour plus d'informations sur les différentes configurations et propriétés à définir pour chaque topologie.
- Vous devez disposer d'une application utilisant les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder aux données avec JPA, utilisez le chargeur JPA. Pour plus d'informations, voir «Configuration des chargeurs JPA», à la page 431.

Procédure

1. Définissez les propriétés dans le fichier `persistence.xml` pour configurer le plug-in de cache OpenJPA : Vous pouvez définir ces propriétés dans l'implémentation de cache DataCache ou Query.

   Les configurations de DataCache et de QueryCache sont indépendantes l'une de l'autre. Vous pouvez activer l'une ou l'autre. Mais, si les deux configurations sont activées, la configuration de QueryCache utilisera celle de DataCache et ses propriétés à elle seront ignorées.

   ```
   <property name="openjpa.DataCache"
             value="<object_grid_datacache_class(<propriété>=<valeur>,...)"/>
   ```

   ou

   ```
   <property name="openjpa.QueryCache"
             value="<object_grid_querycache_class(<propriété>=<valeur>,...)"/>
   ```

   Remarque : Vous pouvez activer le configuration QueryCache pour les topologies intradomaines intégrées et non intégrées uniquement.

   Dans la liste de la classe de cache ObjectGrid, vous pouvez spécifier la propriété ObjectGridName, la propriété ObjectGridType ou toute autre propriétés en rapport avec des règles de déploiement simple pour personnaliser la configuration du cache. Exemple :

   ```
   <property name="openjpa.DataCache"
             value="com.ibm.websphere.objectgrid.openjpa.ObjectGridDataCache(
             ObjectGridName=BasicTestObjectGrid,ObjectGridType=EMBEDDED,
             maxNumberOfReplicas=4)"/>
   <property name="openjpa.QueryCache"
             value="com.ibm.websphere.objectgrid.openjpa.ObjectGridQueryCache()"/>
   <property name="openjpa.RemoteCommitProvider" value="sjvm"/>
   ```

   Voir «Propriétés de configuration du cache JPA pour Hibernate Version 4.0», à la page 414 pour la liste des propriétés que vous pouvez définir.
2. Dans le fichier `persistence.xml`, vous devez affecter à la propriété openjpa.RemoteCommitProvider la valeur `sjvm`.

   ```
   <property name="openjpa.RemoteCommitProvider" value="sjvm"/>
   ```

3. Facultatif : Pour personnaliser davantage la grille de données utilisée par le cache, vous pouvez fournir des paramètres supplémentaires avec des fichiers XML.

   Dans la plupart des cas, la définition des propriétés du cache est amplement suffisante. Pour personnaliser davantage l'ObjectGrid utilisé par le cache, vous pouvez fournir des fichiers XML de configuration OpenJPA ObjectGrid dans le répertoire `META-INF`, similairement au fichier `persistence.xml`. Pendant l'initialisation, le cache tente de localiser ces fichiers XML et les traite s'il les trouve.

   Il existe trois types de fichiers XML de configuration OpenJPA ObjectGrid :

   - `openjpa-objectGrid.xml` (ObjectGrid configuration)

     Chemin du fichier : `META-INF/openjpa-objectGrid.xml`

     Ce fichier sert à personnaliser une configuration d'ObjectGrid de type EMBEDDED ou de type EMBEDDED_PARTITION. Si l'ObjectGrid est de type REMOTE, ce fichier est ignoré. Par défaut, chaque classe d'entité est mappée à sa propre configuration BackingMap désignée au sein de la configuration de l'ObjectGrid sous le nom de la classe. Ainsi, la classe d'entité com.mycompany.Employee sera mappée à la configuration BackingMap com.mycompany.Employee. La configuration BackingMap par défaut est `readOnly="false"`, `copyKey="false"`, `lockStrategy="NONE"` et `copyMode="NO_COPY"`. Vous pouvez tout à fait personnaliser des mappes de sauvegarde avec la configuration que vous choisissez. Le mot clé réservé `ALL_ENTITY_MAPS` représente tous les mappages à l'exclusion des mappages personnalisés répertoriés dans le fichier `openjpa-objectGrid.xml`. Les mappes de sauvegarde qui ne figurent pas dans ce fichier `openjpa-objectGrid.xml` utilisent la configuration par défaut. Si les mappes de sauvegarde personnalisées ne spécifient pas l'attribut ou les propriétés BackingMaps et que ces attributs sont spécifiés dans la configuration par défaut, ce sont les valeurs des attributs dans cette configuration qui s'appliquent. Par exemple, si une classe d'entité est annotée avec `timeToLive=30`, la configuration BackingMap par défaut de cette entité aura un `timeToLive=30`. Si le fichier personnalisé `openjpa-objectGrid.xml` inclut également la mappe de sauvegarde mais sans spécifier de valeur pour timeToLive value, la mappe personnalisée aura la valeur `timeToLive=30` qui est la valeur par défaut. Le fichier `openjpa-objectGrid.xml` a pour finalité de remplacer ou d'étendre la configuration par défaut.

   - `openjpa-objectGridDeployment.xml` (stratégie de déploiement)

     Chemin du fichier : `META-INF/openjpa-objectGridDeployment.xml`

     Ce fichier sert à personnaliser la règle de déploiement. Lorsque celle-ci est personnalisée, si le fichier `openjpa-objectGridDeployment.xml` est fourni, la règle de déploiement par défaut est ignorée. Toutes les valeurs d'attribut de la stratégie de déploiement proviennent du fichier `openjpa-objectGridDeployment.xml` fourni.

   - `openjpa-objectGrid-client-override.xml` (configuration de remplacement ObjectGrid client)

     Chemin du fichier : `META-INF/openjpa-objectGrid-client-override.xml`

     Ce fichier sert à personnaliser un ObjectGrid côté client. Par défaut, le cache de l'ObjectGrid applique une configuration par défaut de substitution des ObjectGrid par les clients, qui désactive les caches locaux (near cache). Vous pouvez activer le cache local en fournissant le fichier `openjpa-objectGrid-client-override.xml` qui remplace cette configuration. Pour plus d'informations sur les paramètres à modifier dans ce fichier pour activer le cache local, voir «Configuration du cache local», à la page 362. Le

fonctionnement du fichier `openjpa-objectGrid-client-override.xml` est semblable à celui du fichier `openjpa-objectGrid.xml`. Le fichier remplace ou étend la configuration d'ObjectGrid.

Pour personnaliser cette topologie, vous pouvez fournir le fichier XML adapté au type de l'eXtreme Scale configuré.

Pour le type EMBEDDED comme pour le type EMBEDDED_PARTITION, vous pouvez fournir n'importe lequel de ces trois fichiers XML pour personnaliser l'ObjectGrid, la règle de déploiement et la configuration de la substitution des ObjectGrid clients.

Dans le cas d'un ObjectGrid REMOTE, le cache ne crée pas d'ObjectGrid dynamique. Le cache ne contient en fait qu'un ObjectGrid côté client provenant du service de catalogue. Dans ce cas, vous pouvez fournir que le fichier `openjpa-objectGrid-client-override.xml` qui personnalisera la configuration de la substitution de l'ObjectGrid client.

4. Facultatif : (Configurations distantes uniquement) Définissez un système eXtreme Scale externe si vous voulez configurer un cache avec un type REMOTE ObjectGrid.

   Afin de pouvoir configurer un cache d'ObjectGrid de type REMOTE, vous devez configurer un système externe eXtreme Scale. Pour configurer ce système externe, vous aurez besoin des deux fichiers XML de configuration ObjectGrid et ObjectGridDeployment basés sur un fichier `persistence.xml`. Pour des exemples de ces fichiers de configuration, voir «Exemple : fichiers XML OpenJPA ObjectGrid», à la page 421.

Résultats

Configuration **EMBEDDED, EMBEDDED_PARTITION, ou intra-domaine** :

Lors du démarrage d'une application, le plug-in détecte automatiquement un service de catalogue ou en démarre un, démarre un serveur de conteneur et connecte les serveurs de conteneur au service de catalogue. Le plug-in communique alors avec le conteneur ObjectGrid et ses homologues exécutés dans d'autres processus de serveur d'applications à l'aide de la connexion client.

Configuration REMOTE :

La stratégie de déploiement est spécifiée séparément de l'application JPA. Un système ObjectGrid externe comporte le service de catalogue et les processus de serveurs de conteneur. Vous devez démarrer le service de catalogue avant les serveurs de conteneur. Pour plus d'informations, reportez-vous aux rubriques «Démarrage des serveurs autonomes qui utilisent le transport ORB», à la page 493 et «Démarrage des serveurs de conteneur qui utilisent le transport ORB», à la page 496.

Que faire ensuite

- Développez une application OpenJPA qui utilise la configuration. Pour plus d'informations, voir Exemple: Utilisation du plug-in Hibernate pour précharger les données dans le cache ObjectGrid.
- Dans un environnement de production, créez des domaines de service de catalogue pour les processus automatiquement créés pour votre configuration EMBEDDED ou EMBEDDED_PARTITION.
  - Environnement autonome :

Si vous n'exécutez pas vos serveurs dans un processus WebSphere Application Server, les hôtes et les ports du domaine de service de catalogue sont spécifiés à l'aide du fichier de propriétés `objectGridServer.properties`. Ce fichier doit être stocké dans le chemin d'accès aux classes de l'application et la propriété **catalogServiceEndPoints** doit être définie. Le domaine de service de catalogue est démarré indépendamment des processus d'application et doit être démarré avant les processus d'application.

Le format du fichier `objectGridServer.properties` est le suivant :

```
catalogServiceEndPoints=<hostname1>:<port1>,<hostname2>:<port2>
```

– Environnement WebSphere Application Server :

Lors d'une exécution à l'intérieur d'un processus WebSphere Application Server, le plug-in de cache JPA se connecte automatiquement au service de catalogue (ou au domaine de service de catalogue) qui est défini pour la cellule WebSphere Application Server.

- Si vous utilisez le type de grille d'objets `EMBEDDED` ou `EMBEDDED_PARTITION` dans un environnement Java SE, utilisez la méthode System.exit(0) à la fin du programme pour arrêter le serveur eXtreme Scale imbriqué. Sinon, le programme peut ne pas répondre.

Exemple : fichiers XML OpenJPA ObjectGrid : Java

Les fichiers XML OpenJPA ObjectGrid XML doivent être créés à partir de la configuration de l'unité de persistance.

Fichier `persistence.xml`

Voici à titre d'exemple un fichier `persistence.xml` représentant la configuration d'une unité de persistance :

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0">
  <persistence-unit name="AnnuityGrid">
    <provider>org.apache.openjpa.persistence.PersistenceProviderImpl</provider>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.FixedAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.EquityAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Person</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityHolder</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Address</class>
    <exclude-unlisted-classes>true</exclude-unlisted-classes>

    <properties>
    <!-- Database setting -->


    <!--  enable cache -->
      <property name="openjpa.DataCache"
        value="com.ibm.websphere.objectgrid.openjpa.ObjectGridDataCache(objectGridName=Annuity,
               objectGridType=EMBEDDED, maxNumberOfReplicas=4)" />
      <property name="openjpa.RemoteCommitProvider" value="sjvm" />
      <property name="openjpa.QueryCache"
                value="com.ibm.websphere.objectgrid.openjpa.ObjectGridQueryCache()"/>
    </properties>
  </persistence-unit>
</persistence>
```

Fichier `openjpa-objectGrid.xml`

Le fichier `openjpa-objectGrid.xml`sert à personnaliser une configuration d'ObjectGrid de type EMBEDDED et de type EMBEDDED_PARTITION. Et voici le fichier `openjpa-objectGrid.xml` correspondant à ce fichier `persistence.xml` :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Annuity">
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject"
                  readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <backingMap name="ObjectGridQueryCache" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" pluginCollectionRef="ObjectGridQueryCache"
                  evictionTriggers="MEMORY_USAGE_THRESHOLD"  />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection
            id="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout">
      <bean id="ObjectTransformer"
            className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="ObjectGridQueryCache">
      <bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.HashIndex" >
          <property name="Name" type="java.lang.String"
                    value="QueryCacheKeyIndex" description="name of index"/>
```

```
        <property name="POJOKeyIndex" type="boolean" value="true" description="POJO Key Index "/>
      </bean>
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Important :

1. Chaque entité est mappée à une mappe de sauvegarde qui porte le nom qualifié complet de la classe de cette entité.

   Par défaut, les entités font partie de la mémoire cache de second niveau. Dans les classes Entity qui doivent être exclues de la mise en cache, vous pouvez inclure l'annotation `@DataCache(enabled=false)` dans la classe Entity à exclure du cache L2:

   ```
   import org.apache.openjpa.persistence.DataCache;
   @Entity
   @DataCache(enabled=false)
   public class OpenJPACacheTest { ... }
   ```

2. Si les classes d'entités sont dans une hiérarchie d'héritage, les classes enfants se mappent à la mappe de sauvegarde parent. La hiérarchie d'héritage partage une même mappe de sauvegarde.
3. Le mappage ObjectGridQueryCache map est indispensable pour la prise en charge de QueryCache.
4. L'ObjectTransformer de la backingMapPluginCollection de chaque mappage d'entrée doit utiliser la classe com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer.
5. L'index de clé de la backingMapPluginCollection d'un mappage d'ObjectGridQueryCache être nommé QueryCacheKeyIndex (voir l'exemple).
6. L'expulseur (evictor) est facultatif pou chaque mappage.

Fichier `openjpa-objectGridDeployment.xml`

Le fichier `openjpa-objectGridDeployment.xml` permet de personnaliser la stratégie de déploiement. Voici le fichier `openjpa-objectGridDeployment.xml` qui correspond au fichier `persistence.xml` :

openjpa-objectGridDeployment.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Annuity">
    <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1"
            minSyncReplicas="0" maxSyncReplicas="4" maxAsyncReplicas="0"
            replicaReadEnabled="true">
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <map ref="ObjectGridQueryCache" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Remarque : Le mappage ObjectGridQueryCache map est indispensable pour la prise en charge de QueryCache.

Configuration du plug-in de mémoire cache Hibernate

Java

Vous pouvez activer le cache pour utiliser le plug-in de cache Hibernate en définissant des fichiers de propriétés.

Avant de commencer

- Vous devez déterminer la topologie du plug-in de cache JPA à utiliser. Voir «Plug-in de cache niveau 2 (L2) JPA», à la page 407 pour plus d'informations sur les différentes configurations.
- Vous devez disposer d'une application utilisant les API JPA. Si vous souhaitez utiliser les API WebSphere eXtreme Scale pour accéder aux données avec JPA, utilisez le chargeur JPA. Pour plus d'informations, voir «Configuration des chargeurs JPA», à la page 431.

Procédure

1. Si vous utilisez WebSphere Application Server, placez les fichiers JAR (Java Archive) dans les emplacements appropriés.

   Le plug-in de cache Hibernate est regroupé dans le fichier `wxshibernate.jar` et installé dans le répertoire *racine_install_wxs*`/opt/IBM/eXtremeScale/ObjectGrid`. Dans un environnement intégré WebSphere Application Server, le plug-in est installé dans le répertoire *racine_was*`/optionalLibraries/ObjectGrid`. Pour utiliser le plug-in de cache Hibernate, vous devez inclure le fichier `wxshibernate.jar` dans la bibliothèque Hibernate. Par exemple, si vous incluez la bibliothèque Hibernate dans l'application, vous devez également inclure le fichier `wxshibernate.jar`. Si vous définissez une bibliothèque partagée pour inclure la bibliothèque Hibernate, vous devez ajouter le fichier `wxshibernate.jar` dans le répertoire de cette bibliothèque partagée.

   eXtreme Scale n'installe pas le fichier `cglib.jar` dans l'environnement WebSphere Application Server. Si vous avez des applications existantes ou des bibliothèques partagées comme Hibernate, qui dépendent de `cglib.jar`, localisez le fichier `cglib.jar` et incluez-le dans le chemin d'accès aux classes. Par exemple, si votre application inclut tous les fichiers JAR de la bibliothèque Hibernate, mais exclut le fichier `cglib.jar` fourni avec Hibernate, vous devez inclure le fichier `cglib.jar` fourni par Hibernate dans votre application.
2. Définissez les propriétés dans le fichier `persistence.xml` pour configurer le plug-in de cache Hibernate.

   Syntaxe de la définition des propriétés dans le fichier `persistence.xml` :

   ```
   <property name="hibernate.cache.region.factory_class"
        value="com.ibm.ws.objectgrid.hibernate.cache.WXSRegionFactory"/>
   <property name="hibernate.cache.use_second_level_cache" value="true"/>
   <property name="hibernate.cache.use_query_cache" value="true" />
   ```

   - **`hibernate.cache.region.factory_class`**: La valeur de la propriété **`region.factory_class`** est la classe com.ibm.ws.objectgrid.hibernate.cache.WXSRegionFactory.
   - **`hibernate.cache.use_query_cache`** : pour activer le cache des requêtes, affectez la valeur `true` à la propriété **`use_query_cache`**.

     Remarque : Vous pouvez activer le cache des requêtes pour les topologies intradomaines intégrées et non intégrées uniquement.
   - Pour activer la mise en cache d'écriture différée, utilisez les attributs d'écriture différée suivants dans la propriété PROPERTY_WRITE_BEHIND. Lorsque la mise en cache d'écriture différée est activée, les mises à jour sont provisoirement stockées dans une mémoire de données de portée JVM jusqu'à ce que la condition wxs.write_behind_interval ou wxs.write_behind_max_batch_size soit vérifiée lorsque les données sont vidées dans le cache.

```
wxs.write_behind=true, wxs.write_behind_interval=5000, wxs.write_behind_Pool_Size=10,
wxs.write_behind_max_batch_size=1000
```

Avertissement : Si wxs.write_n'est pas activée, les autres paramètres de configuration de l'écriture différée sont ignorés.

3. Facultatif : Pour personnaliser davantage la grille de données utilisée par le cache, vous pouvez fournir des paramètres supplémentaires avec des fichiers XML.

   Dans la plupart des cas, la définition des propriétés du cache est amplement suffisante. Toutefois, si vous souhaitez peaufiner la personnalisation de la grille d'objets utilisée par le cache, vous pouvez mettre à disposition dans votre répertoire `META-INF` des fichiers XML de configuration de Hibernate ObjectGrid, à la manière du fichier `persistence.xml`. Pendant l'initialisation, le cache tente de localiser ces fichiers XML et les traite s'il les trouve.

   Il existe trois types de fichiers XML de configuration Hibernate ObjectGrid :

   - `hibernate-objectGrid.xml` (configuration ObjectGrid)

     Chemin du fichier : `META-INF/hibernate-objectGrid.xml`

     Par défaut, chaque classe d'entité est associée à une valeur regionName (par défaut au nom de la classe d'entité) mappée sur une configuration BackingMap désignée sous regionName au sein de la configuration de l'ObjectGrid. Par exemple, la classe d'entité com.mycompany.Employee est associée à une valeur regionName qui a la valeur par défaut com.mycompany.Employee BackingMap. La configuration BackingMap par défaut est readOnly="false", copyKey="false", lockStrategy="NONE" et `copyMode="NO_COPY"`. Vous pouvez tout à fait personnaliser des mappes de sauvegarde avec la configuration que vous choisissez. Le mot clé réservé ALL_ENTITY_MAPS représente tous les mappages à l'exclusion des mappages personnalisés répertoriés dans le fichier `hibernate-objectGrid.xml`. Les mappes de sauvegarde qui ne figurent pas dans ce fichier `hibernate-objectGrid.xml` utilisent la configuration par défaut.

   - `hibernate-objectGridDeployment.xml` (stratégie de déploiement)

     Chemin du fichier : `META-INF/hibernate-objectGridDeployment.xml`

     Ce fichier sert à personnaliser la règle de déploiement. Lorsque celle-ci est personnalisée, si le fichier `hibernate-objectGridDeployment.xml` est fourni, la règle de déploiement par défaut est ignorée. Toutes les valeurs d'attribut de la règle de déploiement proviennent du fichier `hibernate-objectGridDeployment.xml` fourni.

   - `hibernate-objectGrid-client-override.xml` (configuration de remplacement ObjectGrid client)

     Chemin de fichier : `META-INF/hibernate-objectGrid-client-override.xml`

     Ce fichier sert à personnaliser un ObjectGrid côté client. Par défaut, le cache de l'ObjectGrid applique une configuration par défaut de remplacement par les clients, qui désactive le cache local. Vous pouvez activer le cache local en fournissant le fichier `hibernate-objectGrid-client-override.xml` qui remplace cette configuration. Pour plus d'informations sur les paramètres à modifier dans ce fichier pour activer le cache local, voir «Configuration du cache local», à la page 362. Le fonctionnement du fichier `hibernate-objectGrid-client-override.xml` est semblable à celui du fichier `hibernate-objectGrid.xml`. Le fichier remplace ou étend la configuration d'ObjectGrid.

   Pour personnaliser cette topologie, vous pouvez fournir le fichier XML adapté au type de l'eXtreme Scale configuré.

Pour le type EMBEDDED comme pour le type EMBEDDED_PARTITION, vous pouvez fournir n'importe lequel de ces trois fichiers XML pour personnaliser la grille d'objets, la règle de déploiement et la configuration des remplacements par les clients ObjectGrid.

Dans le cas d'un ObjectGrid REMOTE, le cache ne crée pas d'ObjectGrid dynamique. Le cache ne contient en fait qu'un ObjectGrid côté client provenant du service de catalogue. Dans ce cas, vous ne pouvez fournir qu'un fichier `hibernate-objectGrid-client-override.xml` qui personnalisera la configuration de la substitution de l'ObjectGrid client.

4. Facultatif : (Configurations distantes uniquement) Définissez un système eXtreme Scale externe si vous voulez configurer un cache avec un type REMOTE ObjectGrid.

   Afin de pouvoir configurer un cache d'ObjectGrid de type REMOTE, vous devez configurer un système externe eXtreme Scale. Pour configurer ce système externe, vous aurez besoin des deux fichiers XML de configuration ObjectGrid et ObjectGridDeployment basés sur un fichier `persistence.xml`. Pour des exemples de ces fichiers de configuration, voir «Exemple : fichiers XML ObjectGrid Hibernate», à la page 427.

Résultats

Configuration EMBEDDED ou EMBEDDED_PARTITION :

Lors du démarrage d'une application, le plug-in détecte automatiquement un service de catalogue ou en démarre un, démarre un serveur de conteneur et connecte les serveurs de conteneur au service de catalogue. Le plug-in communique alors avec le conteneur ObjectGrid et ses homologues exécutés dans d'autres processus de serveur d'applications à l'aide de la connexion client.

Chaque entité JPA possède une mappe de sauvegarde indépendante affectée à l'aide du nom de classe de l'entité. Chaque mappe de sauvegarde possède les attributs ci-après.

- `readOnly="false"`
- `copyKey="false"`
- `lockStrategy="NONE"`
- `copyMode="NO_COPY"`

Configuration REMOTE :

La règle de déploiement est spécifiée séparément de l'application JPA. Un système ObjectGrid externe comporte le service de catalogue et les processus serveur de conteneur. Vous devez démarrer le service de catalogue avant les serveurs de conteneur. Pour plus d'informations, reportez-vous aux rubriques «Démarrage des serveurs autonomes qui utilisent le transport ORB», à la page 493 et «Démarrage des serveurs de conteneur qui utilisent le transport ORB», à la page 496.

Que faire ensuite

- Développez une application Hibernate qui utilise la configuration. Pour plus d'informations, voir Exemple: Utilisation du plug-in Hibernate pour précharger les données dans le cache ObjectGrid.
- Dans un environnement de production, créez des domaines de service de catalogue pour vos processus automatiquement créés pour votre configuration.
  - Environnement autonome :

Si vous n'exécutez pas vos serveurs dans un processus WebSphere Application Server, les hôtes et les ports du domaine de service de catalogue sont spécifiés à l'aide du fichier de propriétés `objectGridServer.properties`. Ce fichier doit être stocké dans le chemin d'accès aux classes de l'application et la propriété **catalogServiceEndPoints** doit être définie. Le domaine de service de catalogue est démarré indépendamment des processus d'application et doit être démarré avant les processus d'application.

Le format du fichier `objectGridServer.properties` est le suivant :

```
catalogServiceEndPoints=<hostname1>:<port1>,<hostname2>:<port2>
```

– Environnement WebSphere Application Server :

Lors d'une exécution à l'intérieur d'un processus WebSphere Application Server, le plug-in de cache JPA se connecte automatiquement au service de catalogue (ou au domaine de service de catalogue) qui est défini pour la cellule WebSphere Application Server.

- Si vous utilisez le type de grille d'objets `EMBEDDED` ou `EMBEDDED_PARTITION` dans un environnement Java SE, utilisez la méthode System.exit(0) à la fin du programme pour arrêter le serveur eXtreme Scale imbriqué. Sinon, le programme peut ne pas répondre.

Exemple : fichiers XML ObjectGrid Hibernate : Java

Créer des fichiers XML Hibernate ObjectGrid en fonction de la configuration dans une unité de persistance.

Fichier `persistence.xml`

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0">
<persistence-unit name="AnnuityGrid">
<provider>org.hibernate.ejb.HibernatePersistence</provider>

<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.FixedAnnuity</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.EquityAnnuity</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Person</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityHolder</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact</class>
<class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Address</class>

<exclude-unlisted-classes>true</exclude-unlisted-classes>

<properties>
<property name="hibernate.show_sql" value="false" />
<property name="hibernate.connection.url" value="jdbc:db2:Annuity" />
<property name="hibernate.connection.driver_class" value="com.ibm.db2.jcc.DB2Driver" />
<property name="hibernate.default_schema" value="EJB30" />

<!-- Cache -->

<property name="hibernate.cache.region.factory_class" value="com.ibm.websphere.objectgrid.hibernate.cache.WXSRegionFactory"/>
<property name="hibernate.cache.use_query_cache" value="true" />
<property name="wxs.objectgrid_name" value="Annuity" />
<property name="wxs.objectgrid_type" value="EMBEDDED" />
<property name="wxs.max_number_of_replicas" value="4" />
</properties>

</persistence-unit>

</persistence>
```

Fichier `hibernate-objectGridDeployment.xml`

```
?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Annuity">
    <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1" minSyncReplicas="0"
           maxSyncReplicas="4" maxAsyncReplicas="0" replicaReadEnabled="true">
```

```
      <map ref="IBM_HIBERNATE_GENERAL_.*" />
      <map ref="IBM_HIBERNATE_TIMESTAMPS_.*" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Fichier hibernate-objectGrid.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Annuity">
      <backingMap name="IBM_HIBERNATE_TIMESTAMPS_.*" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY"
                  pluginCollectionRef="IBM_HIBERNATE_TIMESTAMPS_.*"
                  template="true" />
      <backingMap name="IBM_HIBERNATE_GENERAL_.*" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="IBM_HIBERNATE_GENERAL_.*"
                  template="true" />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="IBM_HIBERNATE_TIMESTAMPS_.*">
    </backingMapPluginCollection>
    <backingMapPluginCollection id="IBM_HIBERNATE_GENERAL_.*">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
      </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Remarque : Les mappes IBM_HIBERNATE_GENERAL_.* et IBM_HIBERNATE_TIMESTAMPS_.* sont nécessaires.

Configuration d'un fournisseur de cache Spring

Java

Spring Framework Version 3.1 a introduit une nouvelle abstraction de cache. Celle-ci vous permet d'ajouter de manière transparente la mise en cache à une application Spring existante. Vous pouvez utiliser WebSphere eXtreme Scale comme fournisseur de cache pour l'abstraction de cache.

Avant de commencer

- Vous devez disposer d'une application utilisant Spring Framework version 3.1 ou ultérieure.
- Votre application doit déclarer les méthodes de mise en cache à l'aide d'annotations. Pour plus d'informations sur la mise à jour de votre application pour l'abstraction de cache, voir Spring Framework Reference Documentation : Cache abstraction.
- Vérifiez que le fichier `ogclient.jar` se trouve dans le chemin de classes pour l'application Spring.
- Si la machine virtuelle Java sur laquelle votre application s'exécute n'est pas la machine JVM qui est installée par WebSphere eXtreme Scale Client, vous devez ajouter l'argument JVM suivant afin que l'ORB (Object Request Broker) IBM soit utilisé :

  ```
  -Djava.endorsed.dirs=wxs_root/lib/endorsed
  ```

- Vous devez démarrer un serveur de catalogue. Pour plus d'informations, voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494.

Pourquoi et quand exécuter cette tâche

A l'aide de l'abstraction de cache dans Spring Framework, vous pouvez réduire le nombre d'exécutions de la méthode d'application Spring. Lorsqu'elle est configurée, les résultats d'une méthode particulière sont placés dans la mémoire cache. Quand la méthode est ré-exécutée, l'abstraction vérifie la mémoire cache pour déterminer si les résultats de la méthode y figurent déjà. Si les résultats sont dans le cache, les résultats sont renvoyés à partir de la mémoire cache et la méthode ne s'exécute pas. La mise en oeuvre de l'abstraction peut donc réduire le nombre d'exécutions des méthodes coûteuses, ce qui réduit également le temps de réponse moyen de l'application.

Procédure

1. Configurez vos serveurs de conteneur afin qu'ils utilisent les fichier de configuration de Spring.

   Vous devez démarrer les serveurs de conteneur avant le démarrage de l'application Spring qui accède à la mémoire cache. Pour démarrer les serveurs de conteneur, voir «Démarrage des serveurs autonomes qui utilisent le transport ORB», à la page 493.

   Les fichiers de configuration XML par défaut permettant de démarrer un serveur de conteneur pour le fournisseur de cache eXtreme Scale Spring se trouvent dans l'un des emplacements suivants :

   - Installations autonomes : *racine_install_wxs*`/ObjectGrid/spring/etc`
   - Installations WebSphere Application Server : *racine_was*`/optionalLibraries/ObjectGrid/spring/etc`

   Les fichiers sont intitulés `spring-remote-objectgrid.xml` et `spring-remote-deployment.xml`. Vous pouvez utiliser ces fichiers en l'état, les personnaliser ou créer vos propres fichiers de configuration.

   Exécutez la commande suivante pour démarrer un serveur de conteneur autonome pour le fournisseur de cache eXtreme Scale Spring. Exécutez la commande suivante à partir du répertoire *rép_base_wxs*`/ObjectGrid/bin` :

   Windows

   ```
   startOgServer.bat container1 -objectGridFile ../spring/etc/spring-remote-objectgrid.xml
   -deploymentPolicyFile ../spring/etc/spring-remote-deployment.xml
   ```

   UNIX

   ```
   startOgServer.sh container1 -objectGridFile ../spring/etc/spring-remote-objectgrid.xml
   -deploymentPolicyFile ../spring/etc/spring-remote-deployment.xml
   ```

   Windows 8.6+

   ```
   startXsServer.bat container1 -objectGridFile ../spring/etc/spring-remote-objectgrid.xml
   -deploymentPolicyFile ../spring/etc/spring-remote-deployment.xml
   ```

   UNIX 8.6+

   ```
   startXsServer.sh container1 -objectGridFile ../spring/etc/spring-remote-objectgrid.xml
   -deploymentPolicyFile ../spring/etc/spring-remote-deployment.xml
   ```

2. Configurez le conteneur Spring IoC (Inversion of Control) de sorte qu'il utilise WebSphere eXtreme Scale comme fournisseur de cache. L'implémentation du cache WebSphere eXtreme Scale réside sous le package com.ibm.websphere.objectgrid.spring. Définissez les beans ci-dessous dans votre configuration de conteneur Spring IoC.

```
<bean id="wxsCSDomain" class="com.ibm.websphere.objectgrid.spring.ObjectGridCatalogServiceDomainBean"
  p:catalog-service-endpoints="CATALOG_SERVICE_ENDPOINTS"
  p:client-override-xml="CLIENT_OVERRIDE_XML (facultatif)"
  p:client-security-config="CLIENT_SECURITY_CONFIG (facultatif)" />
```

```
<bean id="wxsGridClient" class="com.ibm.websphere.objectgrid.spring.ObjectGridClientBean"
  p:object-grid-name="OBJECT_GRID_NAME(facultatif)"
  p:catalog-service-domain-ref="wxsCSDomain" />

<bean id="cacheManager" class="org.springframework.cache.support.SimpleCacheManager">
  <property name="caches">
    <set>
      <bean class="com.ibm.websphere.objectgrid.spring.ObjectGridCache"
         p:name="CACHE_NAME"
         p:map-name="MAP_NAME (facultatif)"
         p:object-grid-client-ref="wxsGridClient" />
     </set>
   </property>
</bean>
```

CATALOG_SERVICE_ENDPOINTS
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port.

CLIENT_OVERRIDE_XML **(facultatif)**
Indique le chemin d'accès relatif ou absolu à un fichier XML ObjectGrid dans lequel modifier les paramètres côté client sous la forme d'une ressource Spring. Pour plus d'informations sur la spécification des ressources dans Spring, voir Spring Framework Reference Documentation: Resources.

Exemple : `p:client-override-xml="file:/path/to/objectgrid.xml"`

Exemple : `p:client-override-xml="classpath:com/example/app/override-objectgrid.xml"`

Exemple : `p:client-override-xml="http://myserver/override-objectgrid.xml"`

Exemple : `p:client-override-xml="ftp://myserver/override-objectgrid.xml"`

CLIENT_SECURITY_CONFIG **(facultatif)**
Indique le chemin d'accès relatif ou absolu à un fichier `client.properties` sous la forme d'une ressource Spring. Pour plus d'informations sur la spécification des ressources dans Spring, voir Spring Framework Reference Documentation: Resources.

Exemple : `p:client-security-config="file:/path/to/client.properties"`

OBJECT_GRID_NAME **(facultatif)**
Spécifie le nom d'ObjectGrid. Ce paramètre n'est pas requis si les serveurs de conteneur sont démarrés à l'aide des fichiers de configuration XML fournis. Ce paramètre doit être cohérent avec les fichiers de configuration XML qui sont utilisés pour le démarrage des serveurs de conteneur.

CACHE_NAME
Indique le nom de la mémoire cache spécifiée dans l'application de mise en cache Spring.

MAP_NAME **(facultatif)**
Indique le nom de la mappe de sauvegarde pour une mémoire cache. Ce paramètre n'est pas requis si les serveurs de conteneur sont démarrés avec les fichiers de configuration XML fournis. Ce paramètre doit être compatible avec les fichiers de configuration XML utilisés pour le démarrage des serveurs de conteneur. Si vous utilisez les fichiers de configuration XML fournis, la valeur *MAP_NAME* n'est pas requise. Les mappes de la grille de données sont créées automatiquement lors de l'exécution de l'application Spring. Le nom de la mappe dynamique commence par `IBM_SPRING_PARTITIONED_`. Par exemple : `IBM_SPRING_PARTITIONED_1`, `IBM_SPRING_PARTITIONED_2`, etc.

Exemple

Le fragment suivant permet de créer deux caches, appelés `default` et `books` hébergés par le domaine de service de catalogue à l'adresse `localhost:2809`.

```
<bean id="wxsCSDomain" class="com.ibm.websphere.objectgrid.spring.ObjectGridCatalogServiceDomainBean"
 p:catalog-service-endpoints ="localhost:2809" />
<bean id="wxsGridClient" class="com.ibm.websphere.objectgrid.spring.ObjectGridClientBean"
 p:catalog-service-domain-ref="wxsCSDomain" />
<bean id="cacheManager" class="org.springframework.cache.support.SimpleCacheManager">
 <property name="caches">
  <set>
   <bean class="com.ibm.websphere.objectgrid.spring.ObjectGridCache"
    p:name="default"
    p:object-grid-client-ref="wxsGridClient" />
   <bean class="com.ibm.websphere.objectgrid.spring.ObjectGridCache"
    p:name="books"
    p:object-grid-client-ref="wxsGridClient" />
  </set>
 </property>
</bean>
```

Configuration de l'intégration de base de données

Vous pouvez utiliser WebSphere eXtreme Scale pour réduire la charge sur les bases de données. Vous pouvez utiliser une API de persistance Java (JPA) entre WebSphere eXtreme Scale et la base de données pour intégrer les modifications comme chargeur.

Avant de commencer

Pour le récapitulatif des différentes topologies que vous pouvez créer avec une base de données, voir «Intégration de la base de données : caches avec écriture différée, caches en ligne et caches secondaires», à la page 27.

Configuration des chargeurs JPA

Java

Un chargeur Java Persistence API (JPA) est une implémentation de plug-in qui utilise JPA pour interagir avec la base de données.

Avant de commencer

- Vous devez disposer d'une implémentation JPA, comme Hibernate ou OpenJPA.
- Votre base de données peut correspondre à tout programme d'arrière plan prise en charge par le fournisseur JPA choisi.
- Déterminez si vous allez utiliser le plug-in JPALoader ou JPAEntityLoader. Utilisez le plug-in JPALoader lorsque vous stockez des données à l'aide de l'API ObjectMap. Utilisez le plug-in JPAEntityLoader lorsque vous stockez des données à l'aide de l'API EntityManager.

  Remarque : Si vous utilisez les API JPA pour accéder à la source de données JPA, utilisez le plug-in de mémoire cache L2 JPA. Ce plug-in place la grille de données entre votre application et les données source JPA, tout en continuant à utiliser une application JPA. Pour plus d'informations, voir «Plug-in de cache niveau 2 (L2) JPA», à la page 407.

Pourquoi et quand exécuter cette tâche

Pour plus d'informations sur le fonctionnement de Java Persistence API (JPA) Loader, voir Chargeurs JPA.

Procédure

1. Configurez les paramètres requis par JPA pour interagir avec une base de données.

   Les paramètres ci-après sont requis. Ces paramètres sont configurés dans le bean JPALoader ou JPAEntityLoader et le bean JPATxCallback.

   - **persistenceUnitName** : Indique le nom de l'unité de persistance. Ce paramètre est requis à deux titres : pour créer une fabrique de gestionnaire d'entités JPA et pour rechercher les métadonnées d'entité JPA dans le fichier `persistence.xml`. Cet attribut est défini dans le bean JPATxCallback.
   - **JPAPropertyFactory** : Indique la fabrique permettant de créer une mappe de propriétés de persistance pour remplacer les propriétés de persistance par défaut. Cet attribut est défini dans le bean JPATxCallback. Pour définir cet attribut, une configuration de style Spring est requise.
   - **entityClassName** : Indique le nom de classe d'entité requis pour utiliser les méthodes JPA (par exemple, EntityManager.persist, EntityManager.find, etc.). Le plug-in JPALoader requiert ce paramètre, mais ce paramètre est facultatif pour JPAEntityLoader. Pour le plug-in JPAEntityLoader, si aucun paramètre **entityClassName** n'est défini, la classe d'entités configurée dans la mappe d'entités d'ObjectGrid est utilisée. Vous devez utiliser le même nom de classe pour le gestionnaire d'entités eXtreme Scale et le fournisseur JPA. Cet attribut est défini dans le bean JPALoader ou JPAEntityLoader.
   - **preloadPartition** : Indique la partition à partir de laquelle le préchargement de la mappe démarre. Si la partition de préchargement est inférieure à zéro ou supérieure au nombre total de partitions moins 1, le préchargement de la mappe n'est pas démarré. La valeur par défaut est -1, ce qui signifie que le préchargement ne démarre pas par défaut. Cet attribut est défini dans le bean JPALoader ou JPAEntityLoader.

   Outre les quatre paramètres JPA devant être définis dans eXtreme Scale, des métadonnées JPA sont utilisées pour extraire la clé des entités JPA. Les métadonnées JPA peuvent être configurées comme annotation ou comme fichier `orm.xml` spécifié dans le fichier `persistence.xml`. Elles ne font pas partie de la configuration d'eXtreme Scale.

2. Configurez les fichiers XML de la configuration JPA.

   Pour configurer un JPALoader ou JPAEntityLoader, voir Plug-in pour communiquer avec les bases de données.

   Configuration d'un rappel de transaction JPATxCallback avec la configuration du chargeur. L'exemple suivant est un fichier de descripteur XML ObjectGrid (`objectgrid.xml`), qui dispose d'un bean JPAEntityLoader et d'un bean JPATxCallback configurés :

   configuration d'un chargeur avec rappel - Exemple XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
      <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
        <bean id="TransactionCallback"
           className="com.ibm.websphere.objectgrid.jpa.JPATxCallback">
           <property
              name="persistenceUnitName"
              type="java.lang.String"
              value="employeeEMPU" />
       </bean>
       <backingMap name="Employee" pluginCollectionRef="Employee" />
    </objectGrid>
  </objectGrids>
```

```
  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
        <bean id="Loader"
          className="com.ibm.websphere.objectgrid.jpa.JPAEntityLoader">
       <property
              name="entityClassName"
              type="java.lang.String"
              value="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
        </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Si vous souhaitez configurer un JPAPropertyFactory, vous devez utiliser une configuration de type Spring. Voici un exemple de fichier de configuration XML,JPAEM_spring.xml qui configure un bean Spring à utiliser pour les configurations eXtreme Scale.

configuration d'un chargeur avec une fabrique de propriétés JPA - Exemple XML

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">

  <objectgrid:JPAEntityLoader id="jpaLoader"
  entityClassName="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
  <objectgrid:JPATxCallback id="jpaTxCallback" persistenceUnitName="employeeEMPU" />
</beans>
```

Le fichier de configuration XML Objectgrid.xml est présenté ci-après. Notez que le nom ObjectGrid est JPAEM, qui correspond au nom ObjectGrid dans le fichier de configuration Spring JPAEM_spring.xml.

Configuration du chargeur JPAEM - Exemple de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
      <bean id="TransactionCallback"
            className="{spring}jpaTxCallback"/>
        <backingMap name="Employee" pluginCollectionRef="Employee"
                    writeBehind="T4"/>
      </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
       <bean id="Loader" className="{spring}jpaLoader" />
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Une entité peut être annotée avec les annotations JPA et les annotations du gestionnaire d'entités d'eXtreme Scale. Chaque annotation possède un équivalent XML qui peut être utilisé. eXtreme Scale a donc ajouté l'espace de noms Spring. Vous pouvez également les configurer à l'aide de la prise en charge de l'espace de noms Spring. Pour plus d'informations, voir Présentation de l'infrastructure Spring.

Configuration d'un programme de mise à jour de données JPA en fonction de la date/heure

Java

Vous pouvez configurer une mise à jour de base de données en fonction de la date/heure à l'aide d'une configuration XML eXtreme Scale locale ou répartie. Vous pouvez également configurer une configuration locale à l'aide d'un programme.

Pourquoi et quand exécuter cette tâche

Pour plus d'information sur le fonctionnement du programme de mise à jour en fonction de la date/heure Java Persistence API (JPA), voir Programme de mise à jour de données JPA en fonction de la date/heure.

Procédure

Créez une configuration timeBasedDBUpdate.

- **Avec un fichier XML :**

  L'exemple suivant illustre un fichier `objectgrid.xml` qui contient une configuration timeBasedDBUpdate :

```
Programme de mise à jour
JPA en fonction de la date/heure - Exemple de XML
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="changeOG"
            entityMetadataXMLFile="userEMD.xml">
            <backingMap name="user" >
                <timeBasedDBUpdate timestampField="rowChgTs"
                      persistenceUnitName="userderby"
                      entityClass="com.test.UserClass"
                      mode="INVALIDATE_ONLY"
               />
            </backingMap>
        </objectGrid>
    </objectGrids>
    <backingMapPluginCollections>
</objectGridConfig>
```

  Dans cet exemple, la mappe "user" est configurée avec une mise à jour de base de données en fonction de la date/heure. Le mode de mise à jour de base de données est INVALIDATE_ONLY et la zone d'horodatage possède la valeur rowChgTs.

  Si l'ObjectGrid réparti "changeOG" est démarré sur le serveur de conteneur, une unité d'exécution de mise à jour de base de données en fonction de la date/heure est automatiquement démarrée dans la partition 0.

- **A l'aide d'un programme :**

  Si vous créez un ObjectGrid local, vous pouvez également créer un objet TimeBasedDBUpdateConfig et le définir sur l'instance BackingMap :

```
public void setTimeBasedDBUpdateConfig(TimeBasedDBUpdateConfig dbUpdateConfig);
```

  Pour plus d'informations sur la définition d'un objet sur l'instance BackingMap, voir les informations sur l'interface BackingMap dans la documentation de l'API.

  Vous pouvez également annoter la zone d'horodatage dans la classe d'entité à l'aide de l'annotation com.ibm.websphere.objectgrid.jpa.dbupdate.annotation.Timestamp. En configurant la valeur dans la classe, vous n'avez pas besoin de configurer la zone d'horodatage dans la configuration XML.

Que faire ensuite

Démarrez le programme de mise à jour de base de données JPA en fonction de la date/heure. Pour plus d'informations, voir Démarrage du programme de mise à jour en fonction de la date/heure.

Configuration des services de données REST

Java

Vous pouvez utiliser le service de données RESTWebSphere eXtreme Scale WebSphere Application Server, WebSphere Application Server Community Edition et Apache Tomcat.

Pourquoi et quand exécuter cette tâche

L'exemple fourni comprend le code source et les fichiers binaires compilés permettant d'exécuter une grille de données partitionnées. Cet exemple montre comment créer une grille de données simple, modéliser les données en utilisant des entités et fournit deux applications client de ligne de commande qui permettent d'ajouter et d'interroger des entités en utilisant Java ou C#.

L'exemple de client Java utilise L'API Java EntityManager pour conserver et interroger dans la grille de données. Ce client peut être exécuté dans Eclipse ou à l'aide d'un script de ligne de commande. Notez que l'exemple de client Java n'illustre pas le service de données REST, mais permet de mettre à jour les données dans la grille de façon à ce qu'un navigateur Web ou un autre client puisse lire les données.

L'exemple de client Microsoft WCF Data Services C# communique avec la grille de données eXtreme Scale via le service de données REST à l'aide de l'infrastructure .NET. Le client WCF Data Services peut être utilisé à la fois pour mettre à jour et interroger la grille de données.

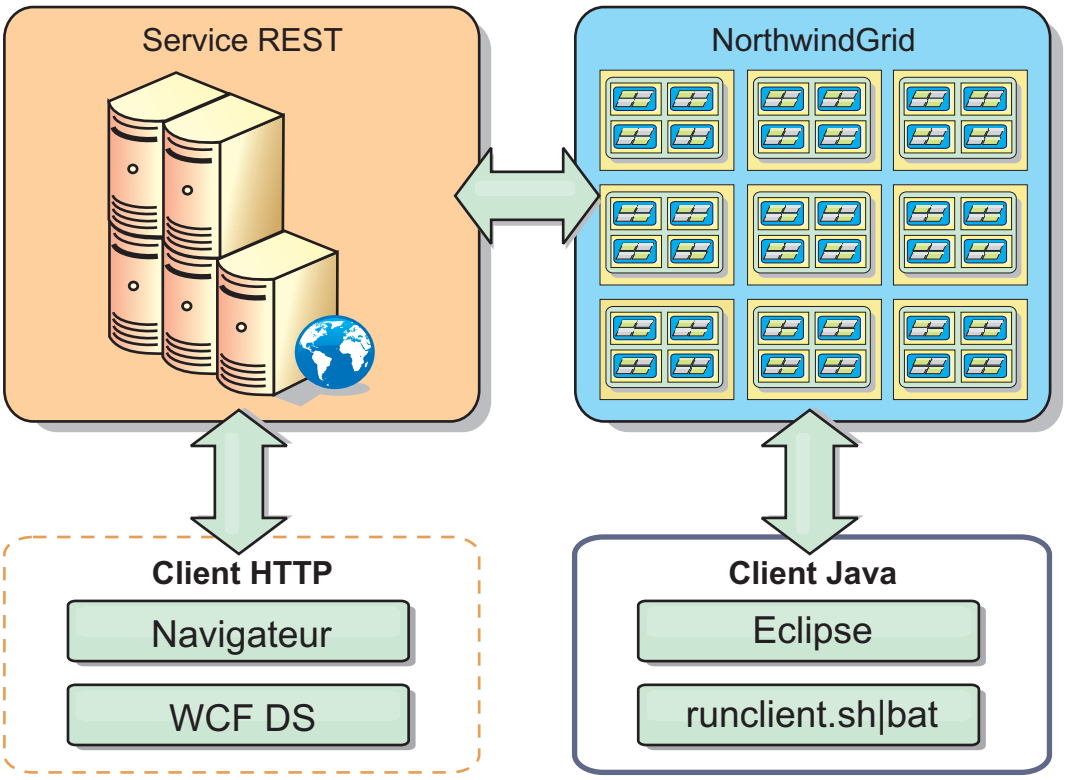


Figure 47. Exemple Mise en route de topologie. Les clients HTTP en utilisant le service de données REST et les clients Java peuvent accéder à la même grille de données.

Procédure

1. Configurez et démarrez une grille de données eXtreme Scale. Voir «Activation du service de données REST».
2. Configurez et démarrez le service de données REST dans un navigateur Web. Voir «Configuration de serveurs d'applications pour le service de données REST», à la page 445.
3. Exécutez un client pour interagir avec le service de données REST. Deux options sont disponibles :
   a. Exécutez l'exemple de client Java pour remplir la grille avec les données en utilisant l'API EntityManager et interrogez les données de la grille par le biais d'un navigateur Web et du service de données REST d'eXtreme Scale. Voir «Utilisation d'un client Java avec les services de données REST», à la page 462.
   b. Exécutez l'exemple de client WCF Data Services C#. Voir «Client WCF de Visual Studio 2008 avec le service de données REST», à la page 464.

Activation du service de données REST

Le service de données REST peut représenter les métadonnées d'entités WebSphere eXtreme Scale pour représenter chaque entité sous la forme d'un EntitySet.

Démarrage d'une exemple de grille de données eXtreme Scale

En général, avant de lancer le service de données REST, démarrez la grille de données eXtreme Scale. La procédure qui suit va démarrer un processus de service de catalogue eXtreme Scale et deux processus de serveurs de conteneur.

WebSphere eXtreme Scale peut être installé selon trois méthodes différentes :

- installation d'essai
- déploiement autonome
- déploiement intégré à WebSphere Application Server

Evolutivité du modèle de données dans eXtreme Scale

L'exemple Microsoft Northwind utilise la table OrderDetail pour établir une association plusieurs-à-plusieurs entre les commandes et les produits.

Les spécifications ORM (Object to relational mapping) comme l'ADO.NET Entity Framework et JPA (Java Persistence API) peuvent mapper les tables et les relations à l'aide d'entités. Mais cette architecture n'est pas évolutive. Pour bien fonctionner, tout doit se trouver sur la même machine ou sur un cluster coûteux de machines.

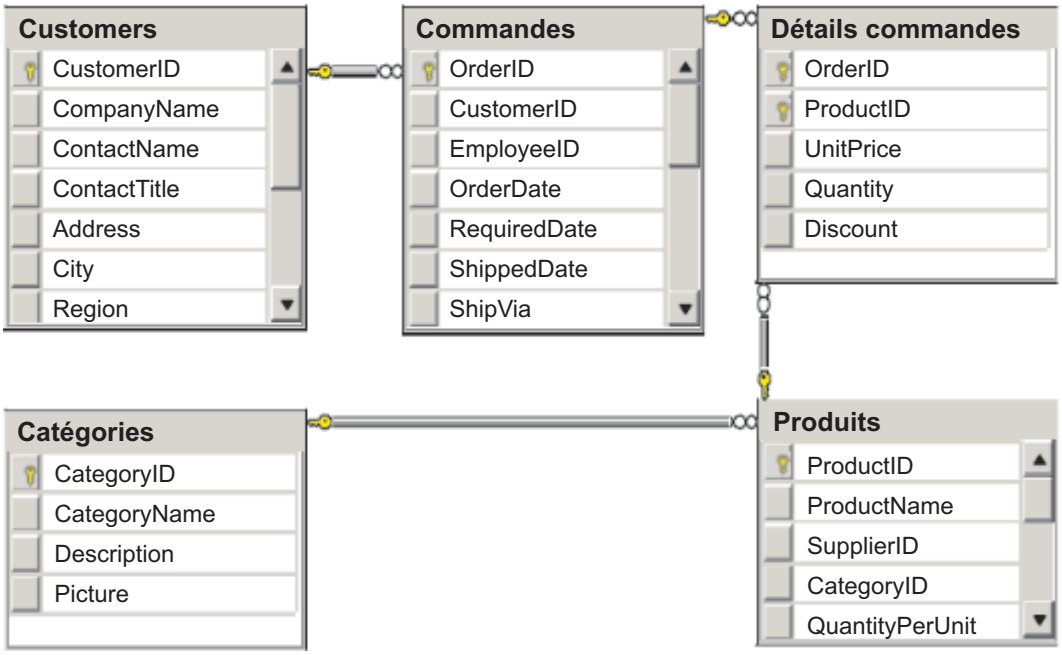


Figure 48. Schéma de l'exemple Microsoft SQL Server Northwind

Pour que puisse être créée une version évolutive de l'exemple, les entités doivent être modélisées de manière à ce que chaque entité ou chaque groupe d'entités en rapport puissent être partitionnées à partir d'une seule clé. De ce fait, les demandes peuvent être réparties entre plusieurs serveurs indépendants. Pour y arriver, les entités ont été divisées en deux arborescences : l'arborescence Customer et l'arborescence Product. Dans ce modèle, chaque arborescence peut être partitionnée de manière indépendante et peut donc croître à des rythmes différents, d'où une plus grande évolutivité.

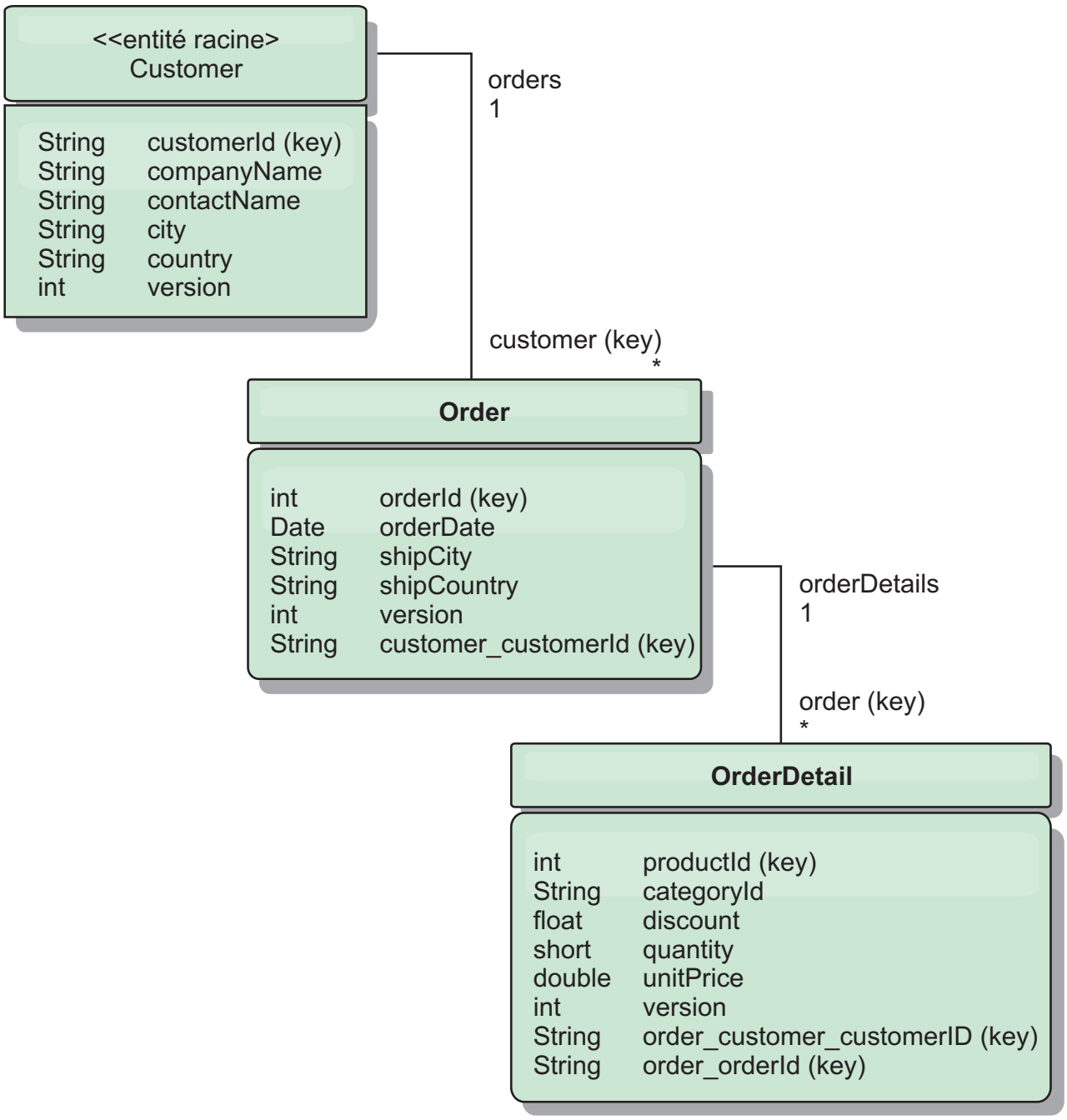


Figure 49. Schéma des entités Customer et Order

Par exemple, Order et Product ont tous les deux comme clés des entiers distincts et uniques. Et de fait, la table Order et la table Product sont deux tables réellement indépendantes l'une de l'autre. Par exemple, considérez l'effet de la taille d'un catalogue, le nombre de produits que vous vendez, avec le nombre total de commandes. A première vue, il peut sembler qu'avoir un grand nombre de produits implique d'avoir également un grand nombre de commandes, mais ce n'est pas nécessairement le cas. Si c'était vrai, il suffirait d'ajouter des produits au catalogue pour augmenter les ventes. Les commandes et les produits ont leurs propres tables indépendantes. L'on peut étendre ce concept en imaginant que les commandes et les produits aient chacun leurs propres grilles de données. Avec des grilles de données indépendantes, vous pouvez contrôler le nombre de partitions et

de serveurs, en plus de la taille de chaque grille de données séparément afin que votre application puisse évoluer. Si vous doublez la taille du catalogue, vous devez doubler la grille de données de produits, mais la grille des commandes peut rester telle quelle. L'inverse est vrai pour un afflux de commandes.

Dans le schéma, un client a zéro ou plusieurs commandes et une commande a des articles (OrderDetail), chacun avec un produit spécifique. Un produit est identifié par un ID (la clé Product) dans chaque OrderDetail. Une grille de données unique stocke les clients, les commandes et les détails des commande, Customer étant l'entité racine de la grille de données. L'on peut extraire les clients à partir de leur ID, mais l'on doit faire partir les commandes des ID clients. L'ID client est donc ajouté à la commande comme partie de sa clé. De la même manière, l'ID client et l'ID commande font partie de l'ID du détail de commande.

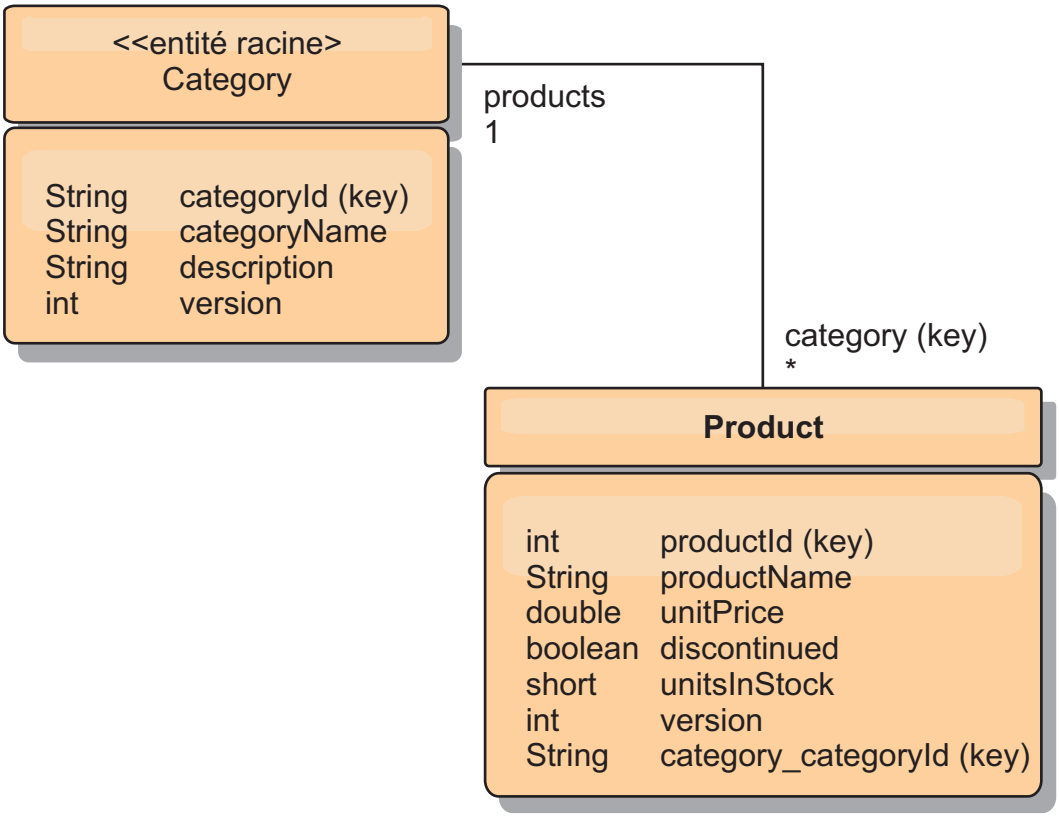


Figure 50. Schéma des entités Category et Product

Dans le schéma Category et Product, Category est la racine du schéma. Avec ce schéma, les clients peuvent rechercher des produits à partir de leur catégorie. Voir «Extraction et actualisation des données avec REST» pour d'autres détails sur les associations de clés et leur importance.

Extraction et actualisation des données avec REST

Le protocole OData requiert que toutes les entités soient adressables à partir de leur forme canonique. Cela signifie que chaque entité doit inclure la clé de l'entité racine partitionnée, la racine de schéma.

Voici un exemple de la manière d'utiliser l'association à partir d'une entité racine pour définir l'adresse d'un enfant dans :

```
/Customer('ACME')/order(100)
```

Dans WCF Data Services, l'entité enfant doit être directement adressable, c'est-à-dire que la clé dans la racine du schéma doit faire partie de la clé de l'enfant : `/Order(customer_customerId='ACME', orderId=100)`. L'on y parvient en créant une association à l'entité racine dans laquelle l'association un-à-un ou plusieurs-à-un à l'entité racine est également identifiée comme une clé. Lorsque des entités sont incluses comme faisant partie de la clé, les attributs de l'entité parent sont exposés comme propriétés de la clé.

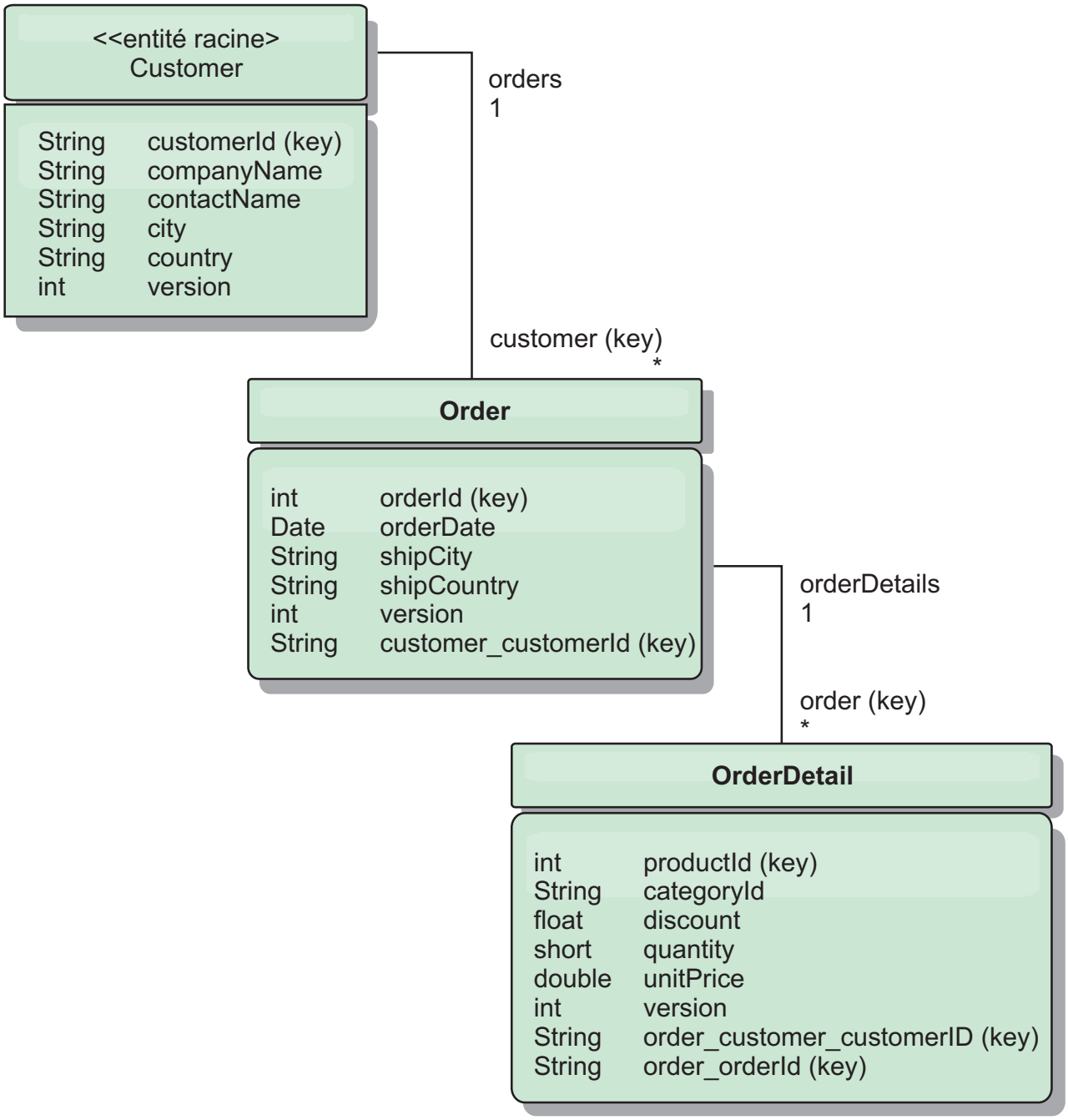


Figure 51. Schéma des entités Customer et Order

Le schéma des entités Customer/Order illustre la manière dont chaque entité est partitionnée à l'aide de Customer. L'entité Order inclut Customer comme partie de sa clé et elle est donc directement accessible. Le service de données REST expose

toutes les associations de clés comme des propriétés individuelles : Order a customer_customerId et OrderDetail a order_customer_customerId and order_orderId.

L'API EntityManager permet de trouver la commande avec l'ID du client et celui de la commande :

```
transaction.begin();
// L'on recherche l'Order à l'aide du Customer.  Nous n'incluons que l'Id
// dans la classe Customer lorsqu'on génère l'instance de clé OrderId.
Order order = (Order) em.find(Order.class,
    new OrderId(100, new Customer('ACME')));
...
transaction.commit();
```

Lorsqu'on utilise le service de données REST, la commande peut être extraite avec l'une des URL :

- `/Order(orderId=100, customer_customerId='ACME')`
- `/Customer('ACME')/orders?$filter=orderId eq 100`

L'adresse de la clé Customer est créée avec l'attribut name de l'entité Customer, un caractère de soulignement et l'attribut name de l'ID Customer : customer_customerId.

Une entité peut également inclure une entité non racine comme faisant partie de sa clé si tous les ancêtres de cette entité non racine ont des associations de clés à la racine. Dans notre exemple, OrderDetail a une association de clés à Order et Order a une association de clés à l'entité Customer racine. Avec l'API EntityManager :

```
transaction.begin();
// L'on construit une instance de clé OrderDetailId.  Elle inclut
// Order et Customer avec uniquement le jeu de clés.
Customer customerACME = new Customer("ACME");
Order order100 = new Order(100, customerACME);
OrderDetailId orderDetailKey =
    new OrderDetailId(order100, "COMP");
OrderDetail orderDetail = (OrderDetail)
    em.find(OrderDetail.class, orderDetailKey);
...
```

Le service de données REST permet l'adressage direct d'OrderDetail :

```
/OrderDetail(productId=500, order_customer_customerId='ACME', order_orderId =100)
```

L'association partant de l'entité OrderDetail vers l'entité Product a été rompue pour permettre le partitionnement indépendant des commandes et du stock de produits. L'entité OrderDetail stocke la catégorie et l'ID de produit au lieu d'une relation en dur. En découpant les deux schémas d'entités, il n'est accédé qu'à une seule partition à la fois.

Le schéma Category et Product montre que l'entité racine est Category et que chaque Product a une association à une entité Category. L'entité Category est incluse dans l'identité Product. Le service de données REST expose une propriété de clé : category_categoryId qui permet l'adressage direct de Product.

Category étant l'entité racine, dans un environnement partitionné, Category doit être connu pour que Product puisse être trouvé. Avec l'API EntityManager, la transaction doit être fixée à l'entité Category avant toute recherche de Product.

Avec l'API EntityManager :

```
transaction.begin();
// L'on crée l'entité racine Category avec uniquement la clé.  Cela
// nous permet de construire un ProductId sans avoir besoin de trouver
// d'abord la Category.  La transaction est à présent fixée
// à la partition où est stockée la Category "COMP".
Category cat = new Category("COMP");
Product product = (Product) em.find(Product.class,
    new ProductId(500, cat));
...
```

Le service de données REST permet l'adressage direct de Product :

```
/Product(productId=500, category_categoryId='COMP')
```

Démarrage d'une grille de données autonome pour les services de données REST

Suivez ces étapes pour démarrer l'exemple de grille de données du service REST WebSphere eXtreme Scale pour un déploiement eXtreme Scale autonome.

Avant de commencer

Installez la version d'évaluation de WebSphere eXtreme Scale ou le produit complet :

- Installez la version autonome du produit et appliquez tous les correctifs ultérieurs.
- Téléchargez et extrayez la version d'évaluation WebSphere eXtreme Scale 7.1 qui inclut le service de données REST WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Démarrez l'exemple de grille de données WebSphere eXtreme Scale.

Procédure

1. Démarrez le processus de service de catalogue. Ouvrez une ligne de commande ou une fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`
2. `cd base_servicerest/gettingstarted`
3. Démarrez le processus de service de catalogue. Pour démarrer le service *sans* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Pour démarrer le service avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
4. Démarrez deux processus de serveur de conteneur. Ouvrez une autre ligne de commande ou fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`

5. `cd base_servicerest/gettingstarted`
6. Démarrez un processus de serveur de conteneur :

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

   - Linux UNIX `./runcontainer.sh container0`
   - Windows `runcontainer.bat container0`

   Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

   - Linux UNIX `./runcontainer_secure.sh container0`
   - Windows `runcontainer_secure.bat container0`
7. Ouvrez une autre ligne de commande ou fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=`*`base_java`*
   - Windows `set JAVA_HOME=`*`base_java`*
8. `cd base_servicerest/gettingstarted`
9. Démarrez un second processus de serveur de conteneur.

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes.

   - Linux UNIX `./runcontainer.sh container1`
   - Windows `runcontainer.bat container1`

   Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes.

   - Linux UNIX `./runcontainer_secure.sh container1`
   - Windows `runcontainer_secure.bat container1`

Résultats

Attendez que les conteneurs eXtreme Scale soient prêts avant de passer aux étapes suivantes. Les serveurs de conteneur sont prêts lorsque le message suivant s'affiche dans la fenêtre de terminal :

`CWOBJ1001I: Le serveur ObjectGrid `*`nom_conteneur`*` est prêt à traiter les requêtes.`

où *nom_conteneur* correspond au nom du conteneur qui a été démarré.

Démarrage d'une grille de données pour les services de données REST dans WebSphere Application Server

Appliquez la procédure exposée ici pour démarrer une grille de données exemple de service REST WebSphere eXtreme Scale autonome pour un déploiement de WebSphere eXtreme Scale intégré à WebSphere Application Server. Bien que WebSphere eXtreme Scale soit intégré à WebSphere Application Server, cette procédure démarre un processus de service de catalogue WebSphere eXtreme Scale autonome et un conteneur.

Avant de commencer

Installez le produit dans un répertoire d'installation WebSphere Application Server Version 7.0.0.5 ou suivante avec la sécurité activée. Etendez au moins un profil de serveur d'applications.

Pourquoi et quand exécuter cette tâche

Démarrage de l'exemple de grille de données WebSphere eXtreme Scale.

Procédure

1. Démarrez le processus de service de catalogue. Ouvrez une fenêtre de ligne de commande ou de terminal et définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`

   `cd base_servicerest/gettingstarted`
2. Démarrez le processus de service de catalogue.

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes.
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
3. Démarrez deux processus de serveur de conteneur. Ouvrez une autre fenêtre de ligne de commande ou de terminal et définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`
4. Démarrez un processus de serveur de conteneur.

   Pour démarrer le serveur sans la sécurité d'eXtreme Scale, utilisez les commandes suivantes.
   a. Ouvrez une fenêtre de ligne de commande.
   b. `cd base_servicerest/gettingstarted`
   c. Pour démarrer le serveur *sans* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :
      - Linux UNIX `./runcontainer.sh container0`
      - Windows `runcontainer.bat container0`
   d. Pour démarrer le serveur avec la sécurité d'eXtreme Scale, utilisez les commandes suivantes.
      - Linux UNIX `./runcontainer_secure.sh container0`
      - Windows `runcontainer_secure.bat container0`
5. Démarrez un second processus de serveur de conteneur.
   a. Ouvrez une fenêtre de ligne de commande.

b. `cd base_servicerest/gettingstarted`

c. Pour démarrer le serveur *sans* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

- Linux UNIX `./runcontainer.sh container1`
- Windows `runcontainer.bat container1`

d. Pour démarrer le serveur *avec* la sécurité d'eXtreme Scale, utilisez les commandes suivantes :

- Linux UNIX `./runcontainer_secure.sh container1`
- Windows `runcontainer_secure.bat container1`

Résultats

Attendez que les serveurs de conteneur soient prêts avant de passer aux étapes suivantes. Les serveurs de conteneur sont prêts lorsque le message suivant s'affiche :

`CWOBJ1001I: Le serveur ObjectGrid nom_conteneur est prêt à traiter les requêtes.`

où *nom_conteneur* correspond au nom du conteneur qui a été démarré à l'étape précédente.

Configuration de serveurs d'applications pour le service de données REST

Vous pouvez configurer des serveurs d'applications différents pour utiliser le service de données REST.

Déploiement du service de données REST sur WebSphere Application Server

Java

Cette rubrique explique comment configurer le service de données REST WebSphere eXtreme Scale sur WebSphere Application Server ou WebSphere Application Server Network Deployment Version ou une version suivante. Ces instructions s'appliquent également aux déploiements où WebSphere eXtreme Scale est intégré avec le déploiement WebSphere Application Server.

Avant de commencer

Vous devez disposer de l'un des environnements suivants sur votre système pour configurer et déployer le service de données REST pour WebSphere eXtreme Scale.

- WebSphere Application Server avec le client autonome WebSphere eXtreme Scale :
  - WebSphere eXtreme Scale Trial Version 7.1 avec le service de données REST est téléchargé et extrait ou WebSphere eXtreme Scale Version 7.1.0.0 avec le groupe de correctif 2 est installé dans un répertoire autonome.
  - WebSphere Application Server Version 7.0.0.5 ou version suivante, est installé et actif.
- WebSphere Application Server intégré à WebSphere eXtreme Scale:

WebSphere eXtreme Scale Version 7.1.0.0 avec le groupe de correctifs 2 ou suivant installé sur WebSphere Application Server Version 7.0 ou version suivante.

Conseil : Le service de données REST d'WebSphere eXtreme Scale requiert uniquement que le client WebSphere eXtreme Scale soit installé. Il n'est pas nécessaire d'étendre le profil.

Découvrez comment activer la sécurité Java 2 dans le centre de documentation WebSphere Application Server.

Procédure

1. Configurez et démarrez une grille de données.
   a. Pour plus de détails sur la configuration d'une grille de données pour l'utiliser avec le service de données REST, voir «Démarrage d'une grille de données pour les services de données REST dans WebSphere Application Server», à la page 443.
   b. Vérifiez qu'un client peut se connecter et accéder aux entités dans la grille de données. Pour un exemple, voir «Tutoriel : Démarrer avec WebSphere eXtreme Scale», à la page 1.
2. Générez le répertoire ou le fichier JAR de configuration du service REST d'eXtreme Scale. Reportez-vous dans «Installation du service de données REST» , à la page 221 aux explications sur le packaging et le déploiement du service REST.
3. Ajoutez le fichier JAR ou le répertoire de la configuration du service de données REST au chemin d'accès aux classes du serveur d'applications :
   a. Ouvrez la console d'administration de WebSphere Application Server.
   b. Accédez à **Environnement** > **Bibliothèques partagées**
   c. Cliquez sur **Nouveau**
   d. Ajoutez les entrées suivantes dans les zones appropriées :
      - Nom : `extremescale_rest _configuration`
      - Chemin de classes : <répertoire ou fichier JAR de configuration du service REST>
   e. Cliquez sur **OK**
   f. Sauvegardez les modifications apportées à la configuration principale
4. Ajoutez le fichier JAR d'exécution client WebSphere eXtreme Scale, `wsogclient.jar` et le fichier JAR fichier JAR du service de données REST ou le répertoire d'accès au chemin d'accès aux classes du serveur. Cette étape n'est pas nécessaire si WebSphere eXtreme Scale est intégré à l'installation WebSphere Application Server.
   a. Ouvrez la console d'administration de WebSphere Application Server.
   b. Accédez à **Environnement** > **Bibliothèques partagées**.
   c. Cliquez sur **Nouveau**.
   d. Ajoutez les entrées suivantes dans les zones :
      - Nom : `extremescale_client_v71`
      - Chemin d'accès aux classes : `base_wxs/lib/wsogclient.jar`

      A faire : Ajoutez chaque chemin sur une ligne séparée.
   e. Cliquez sur **OK**.
   f. Sauvegardez les modifications apportées à la configuration principale.

5. Installez le fichier EAR du service de données REST (wxsrestservice.ear) sur WebSphere Application Server à partir de la console d'administration :
   a. Ouvrez la console d'administration de WebSphere Application Server.
   b. Cliquez sur **Applications** > **Nouvelle application**.
   c. Accédez au fichier /lib/wxsrestservice.ear dans le système de fichiers, sélectionnez-le et cliquez sur **Suivant**.
      - Si vous utilisez WebSphere Application Server Version 7.0, cliquez sur suivant.
   d. Choisissez l'option d'installation détaillée et cliquez sur **Suivant**.
   e. Dans l'écran d'avertissements de sécurité de l'application, cliquez sur **Continuer**.
   f. Sélectionnez les options d'installation par défaut, puis cliquez sur **Suivant**.
   g. Choisissez un serveur pour y associer l'application et cliquez sur **Suivant**.
   h. Sur la page de rechargement JSP, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.
   i. Dans la page des bibliothèques partagées, associez le module wxsrestservice.war aux bibliothèques partagées que vous avez définies :
      - extremescale_rest_configuration
      - extremescale_client_v71

      Conseil : Cette bibliothèque partagée est nécessaire uniquement si WebSphere eXtreme Scale n'est pas intégré à WebSphere Application Server.
   j. Sur la page de mappe des relations de bibliothèques partagées, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.
   k. Sur la page de mappe des hôtes virtuels, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.
   l. Dans la page des racines de contexte des mappes, spécifiez wxsrestservice comme racine de contexte.
   m. Sur l'écran récapitulatif, cliquez sur **Fin** pour terminer la installation.
   n. Sauvegardez les modifications apportées à la configuration principale.
6. Démarrez l'application de service de données wxsrestservice REST :
   a. Accédez à l'application dans la console d'administration.
      - WebSphere Application Server Version 7.0 : Dans la console d'administration, cliquez sur **Applications** > **Types d'application** > **Applications WebSphere**.
   b. Cochez la case de l'application wxsrestservice et cliquez sur **Démarrer**.
   c. Vérifiez le fichier SystemOut.log du profil de serveur d'applications. Lorsque le service de données REST a démarré, le message suivant apparaît dans le journal SystemOut.log du profil du serveur :

      ```
      CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
      ```
7. Vérifiez que le service de données REST fonctionne : Le numéro de port se trouve dans le fichier SystemOut.log dans le répertoire des journaux du profil du serveur d'applications et vous pouvez le trouver en recherchant le premier port affiché pour l'identificateur de message : SRVE0250I. Le port par défaut est 9080.

   Par exemple : http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/ Résultat : le document du service AtomPub s'affiche.

Par exemple : `http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata`. Le document Entity Model Data Extensions (EDMX) s'affiche.

8. Pour arrêter les processus de grille de données, utilisez `CTRL+C` dans la fenêtre de commande correspondante.

Démarrage de services de données REST avec WebSphere eXtreme Scale intégré à WebSphere Application Server 7.0 : Java

Cette rubrique explique comment configurer et démarrer le service de données REST d'eXtreme Scale à l'aide de WebSphere Application Server version 7.0 qui a été intégré et étendu avec WebSphere eXtreme Scale.

Avant de commencer

Vérifiez que l'exemple de grille de données autonome eXtreme Scale est démarré. Voir «Activation du service de données REST», à la page 436 pour plus d'informations sur le démarrage de la grille.

Pourquoi et quand exécuter cette tâche

Pour commencer avec le service de données REST de WebSphere eXtreme Scale REST à l'aide de WebSphere Application Server, procédez comme suit :

Procédure

1. Ajoutez au chemin d'accès aux classes le fichier JAR d'exemple de configuration du service de données REST de WebSphere eXtreme Scale :
   a. Ouvrez la console d'administration WebSphere
   b. Accédez à Environnement -> Bibliothèques partagées
   c. Cliquez sur Nouveau
   d. Ajoutez les entrées suivantes dans les zones appropriées :
      1) Nom : extremescale_gettingstarted_config
      2) Chemin d'accès aux classes
         - `base_servicerest/gettingstarted/restclient/bin`
         - `base_servicerest/gettingstarted/common/bin`

         A faire : Chaque chemin doit se trouver sur une ligne distincte.
   e. Cliquez sur **OK**
   f. Enregistrez les modifications apportées à la configuration principale.
2. Installez le fichier d'archive d'entreprise du service de données REST sur le serveur à l'aide de la console d'administration WebSphere :
   a. Ouvrez la console d'administration WebSphere
   b. Accédez à Applications -> Nouvelle application
   c. Allez au fichier `base_servicerest/lib/wxsrestservice.ear`. Sélectionnez le fichier, puis cliquez sur **Suivant**.
   d. Sélectionnez les options d'installation détaillées, puis cliquez sur **Suivant**.
   e. Dans l'écran d'avertissements de sécurité de l'application, cliquez sur **Continuer**.
   f. Sélectionnez les options d'installation par défaut, puis cliquez sur **Suivant**.
   g. Choisissez un serveur sur lequel mapper le module wxsrestservice.war, puis cliquez sur **Suivant**.

h. Sur la page de rechargement JSP, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

i. Sur la page des bibliothèques partagées, mappez le module wxsrestservice.war" sur les bibliothèques partagées suivantes, à savoir celles définies au cours de la première étape : `extremescale_ gettingstarted _config`

j. Sur la page de mappe des relations de bibliothèques partagées, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

k. Sur la page de mappe des hôtes virtuels, utilisez les valeurs par défaut, puis cliquez sur **Suivant**.

l. Sur la page de mappe de racine de contexte, définissez la racine de contexte sur : wxsrestservice.

m. Sur l'écran récapitulatif, cliquez sur **Fin** pour terminer la installation.

n. Enregistrez les modifications apportées à la configuration principale.

3. Si la grille de données eXtreme Scale a été démarrée avec la sécurité eXtreme Scale activée, définissez la propriété suivante dans le fichier `restservice_home/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=base_servicerest/gettingstarted/security/security.ogclient.properties
```

4. Démarrez le serveur d'applications et l'application wxsrestservice du service de données REST d'eXtreme Scale.

   Une fois que l'application a démarré, ouvrez le journal SystemOut.log du serveur d'applications et vérifiez que le message suivant est présent : `CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.`

5. Vérifiez que le service de données REST fonctionne :

   a. Ouvrez un navigateur et rendez-vous à l'adresse suivante :

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid

      Le document de service de la NorthwindGrid s'affiche.

   b. Rendez-vous à l'adresse suivante :

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Le document Entity Model Data Extensions (EDMX) s'affiche

6. Pour arrêter les processus de grille de données, utilisez CTRL+C dans la fenêtre de commande correspondante.

Déploiement du service de données REST sur WebSphere Application Server Community Edition

Java

Vous pouvez configurer le service de données REST eXtreme Scale sur WebSphere Application Server Community Edition Version 2.1.1.3 et les versions suivantes.

Avant de commencer

- Un JRE ou JDK Oracle IBM (recommandé) Version 6 ou version suivante est installé et la variable d'environnement JAVA_HOME est définie.
- Téléchargez et installez WebSphere Application Server Community Edition version 2.1.1.3 ou plus récente dans le répertoire `racine_wasce`, par exemple le répertoire `/opt/IBM/wasce`. Pour plus d'informations sur version 2.1.1 ou (autres versions), lisez les instructions d'installation.

Procédure

1. Configurez et démarrez une grille de données.
   a. Pour plus d'informations sur la configuration d'une grille de données eXtreme Scale pour l'utiliser avec le service de données REST, voir «Démarrage d'une grille de données autonome pour les services de données REST», à la page 442.
   b. Vérifiez qu'un client eXtreme Scale parvient à se connecter aux entités de la grille et à y accéder. Pour un exemple, voir «Tutoriel : Démarrer avec WebSphere eXtreme Scale», à la page 1.
2. Générez le répertoire ou le fichier JAR de configuration du service REST d'eXtreme Scale. Pour les détails, voir dans «Installation du service de données REST», à la page 221 les explications sur le packaging et le déploiement.
3. Démarrez le serveur WebSphere Application Server Community Edition :
   a. Pour démarrer le serveur sans la sécurité Java SE activée, exécutez la commande suivante :

      UNIX Linux `racine_wasce/bin/startup.sh`

      Windows `racine_wasce/bin/startup.bat`
   b. Pour démarrer le serveur avec la sécurité Java SE activée, effectuez les étapes suivantes : UNIX Linux
      1) Ouvrez une ligne de commande ou une fenêtre de terminal et exécutez la commande de copie suivante (ou copiez le contenu du fichier de règles spécifié dans votre règle existante) : `cp base_servicerest/gettingstarted/wasce/geronimo.policy wasce_root/bin`
      2) Modifiez le fichier `racine_wasce/bin/setenv.sh`.
      3) Après la ligne qui contient `"WASCE_JAVA_HOME="`, ajoutez la ligne suivante : `export JAVA_OPTS="-Djava.security.manager -Djava.security.policy=geronimo.policy"`.

      Windows
      1) Ouvrez une fenêtre de ligne de commande et exécutez la commande de copie suivante ou copiez le contenu du fichier de règles défini vers la stratégie existante :

         ```
         copy base_servicerest\gettingstarted\wasce\geronimo.policy\bin
         ```
      2) Modifiez le fichier `racine_wasce/bin/setenv.bat`.
      3) Après la ligne qui contient "set WASCE_JAVA_HOME=", ajoutez la ligne suivante :

         ```
         set JAVA_OPTS="-Djava.security.manager
         -Djava.security.policy=geronimo.policy"
         ```
4. Ajoutez le fichier JAR d'exécution de client ObjectGrid au référentiel WebSphere Application Server Community Edition :
   a. Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition. L'URL par défaut est `http://localhost:8080/console`, l'ID utilisateur par défaut est `system` et le mot de passe, `manager`.
   b. Cliquez sur le lien **Référentiel** situé dans la partie gauche de la fenêtre de la console, dans le dossier **Services**.
   c. Dans la section **Ajouter une archive au référentiel**, entrez les éléments suivants dans les zones de texte :

Tableau 31. Ajout d'une archive au référentiel

Zone de texte	Valeur
Fichier	`base_wxs/lib/ogclient.jar`
Groupe	com.ibm.websphere.xs
Artefact	ogclient
Version	7.1
Type	JAR

d. Cliquez sur le bouton Installer

Reportez-vous à la note technique suivante pour des détails sur les différentes manières dont les dépendances de classes et de bibliothèques peuvent être configurées : Specifying external dependencies to applications running on WebSphere Application Server Community Edition.

5. Déployez vers le serveur WebSphere Application Server Community Edition le module du service de données REST, le fichier `wxsrestservice.war`, et démarrez-le.
   a. Copiez et éditez l'exemple de fichier XML de plan de déploiement : `base_servicerest/gettingstarted/wasce/geronimo-web.xml`, afin d'inclure les dépendances de chemin au répertoire ou au fichier JAR de configuration de votre service de données REST. Voir la section pour un exemple de définition du chemin d'accès aux classes afin d'y inclure votre fichier `wxsRestService.properties` ainsi que d'autres fichiers de configuration et classes de métadonnées.
   b. Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.

      Conseil : L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe, `manager`.
   c. Cliquez sur le lien **Déployer nouveau** situé dans la partie gauche de la fenêtre de la console.
   d. Entrez les valeurs suivantes dans les zones de texte de la page **Installer de nouvelles applications** :

Tableau 32. Installer de nouvelles applications

Zone de texte	Valeur
Archive	`base_servicerest/lib/wxsrestservice.war`
Plan	`base_servicerest/gettingstarted/wasce/geronimo-web.xml`

Conseil : Utilisez le chemin du fichier `geronimo-web.xml` que vous avez copié et édité au point 3.

   e. Cliquez sur le bouton Installer. La page de la console indique alors que l'application a été installée et démarrée.
   f. Examinez le journal de sortie du système WebSphere Application Server Community Edition ou la console pour déterminer si le service de données REST a démarré. Le message suivant doit apparaître :

      `CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.`

6. Démarrez le serveur WebSphere Application Server Community Edition en exécutant la commande suivante :

- UNIX Linux `racine_wasce/bin/startup.sh`
- Windows `racine_wasce/bin/startup.bat`

7. Installez sur le serveur WebSphere Application Server Community Edition le service de données REST d'eXtreme Scale et l'exemple fourni :
   a. Ajoutez le fichier JAR d'exécution de client ObjectGrid au référentiel WebSphere Application Server Community Edition :
      1) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition. L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe, `manager`.
      2) Cliquez sur le lien **Référentiel** à gauche de la fenêtre de la console, dans le dossier Services.
      3) Dans la section **Ajouter une archive au référentiel**, entrez les éléments suivants dans les zones de texte :

Tableau 33. Ajout d'une archive au référentiel

Zone de texte	Valeur
Fichier	base_wxs/lib/ogclient.jar
Groupe	com.ibm.websphere.xs
Artefact	ogclient
Version	7.1
Type	JAR

      4) Cliquez sur le bouton Installer.

         Conseil : Reportez-vous à la note technique suivante pour des détails sur les différentes manières dont les dépendances de classes et de bibliothèques peuvent être configurées : Specifying external dependencies to applications running on WebSphere Application Server Community Edition.

   b. Déployez le module de service de données REST, `wxsrestservice.war`, vers le serveur WebSphere Application Server Community Edition.
      1) Editez l'exemple de fichier XML de déploiement `base_servicerest/gettingstarted/wasce/geronimo-web.xml` pour inclure les dépendances de chemin d'accès dans les répertoires du chemin d'accès aux classes de l'exemple Mise en route :
         - Modifiez le chemin "classesDirs" des deux GBeans du client Mise en route :

         Le chemin de "classesDirs" pour le bean géré GettingStarted_Client_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.

         Le chemin de "classesDirs" pour le bean géré GettingStarted_Common_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.
      2) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.
      3) Cliquez sur le lien **Déployer nouveau** situé dans la partie gauche de la fenêtre de la console.
      4) Entrez les valeurs suivantes dans les zones de texte de la page **Installer de nouvelles applications** :

Tableau 34. Installer de nouvelles applications

Zone de texte	Valeur
Archive	*base_servicerest*/lib/wxsrestservice.war
Plan	*base_servicerest*/gettingstarted/wasce/geronimo-web.xml

5) Cliquez sur le bouton **Installer**.

   La page de la console indique alors que l'application a été installée et démarrée.

6) Recherchez le message ci-après sur la console ou dans le journal de sortie système de WebSphere Application Server Community Edition pour vérifier que le service de données REST a bien démarré :

   ```
   CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.
   ```

8. Vérifiez que le service de données REST fonctionne :

   Ouvrez un navigateur Web et accédez à l'URL `http://<hôte>:<port>/<racine_contexte>/restservice/<nom_grille>`

   Le port par défaut de WebSphere Application Server Community Edition est 8080 et il est défini à l'aide de la propriété HTTPPort dans le fichier `/var/config/config-substitutions.properties`.

   Par exemple : http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/

Résultats

Le document de service AtomPub est affiché.

Démarrage du service de données REST sur WebSphere Application Server Community Edition. : Java

Nous allons expliquer comment configurer et démarrer le service de données REST d'eXtreme Scale avec WebSphere Application Server Community Edition.

Avant de commencer

Vérifiez que la grille d'exemple est bien démarrée. Voir «Activation du service de données REST», à la page 436 pour savoir comment démarrer la grille.

Procédure

1. Téléchargez et installez WebSphere Application Server Community Edition version 2.1.1.3 ou plus récente dans le répertoire `racine_wasce`, par exemple le répertoire `/opt/IBM/wasce`.
2. Démarrez le serveur WebSphere Application Server Community Edition en exécutant la commande suivante :
   - Linux UNIX `racine_wasce/bin/startup.sh`
   - Windows `racine_wasce/bin/startup.bat`
3. Si la grille eXtreme Scale a été démarrée avec la sécurité d'eXtreme Scale activée, définissez les propriétés suivantes dans le fichier `base_servicerest/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=base_servicerest/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Installez sur le serveur WebSphere Application Server Community Edition le service de données REST d'eXtreme Scale et l'exemple fourni :
   a. Ajoutez le fichier JAR d'exécution du client ObjectGrid au référentiel WebSphere Application Server Community Edition :
      1) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.

         Conseil : L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe `manager`.
      2) Cliquez sur **Référentiel** dans le dossier Services.
      3) Dans la section **Ajouter une archive au référentiel**, entrez les éléments suivants dans les zones de texte :

Tableau 35. Archivage dans le référentiel

Zone de texte	Valeur
Fichier	base_wxs/lib/ogclient.jar
Groupe	com.ibm.websphere.xs
Artefact	ogclient
Version	7.0
Type	jar

      4) Cliquez sur le bouton Installer.

         Conseil : Reportez-vous à la note technique suivante pour des explications détaillées sur les différentes méthodes de configuration des dépendances de classes et des bibliothèques : Specifying external dependencies to applications running on WebSphere Application Server Community Edition.
   b. Déployez vers le serveur WebSphere Application Server Community Edition le module du service de données REST, le fichier `wxsrestservice.war`.
      1) Editez l'exemple de fichier XML de déploiement `base_servicerest/gettingstarted/wasce/geronimo-web.xml` pour inclure les dépendances de chemin d'accès dans les répertoires du chemin d'accès aux classes de l'exemple Mise en route :

         Modifiez le chemin classesDirs des deux GBeans du client Mise en route :
         - Le chemin de "classesDirs" pour le bean géré GettingStarted_Client_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.
         - Le chemin de "classesDirs" pour le bean géré GettingStarted_Common_SharedLib doit avoir la valeur `base_servicerest/gettingstarted/restclient/bin`.
      2) Ouvrez et connectez-vous à la console d'administration de WebSphere Application Server Community Edition.

         Conseil : L'URL par défaut est `http://localhost:8080/console`. L'ID utilisateur par défaut est `system` et le mot de passe `manager`.
      3) Cliquez sur **Déployer nouveau**.
      4) Entrez les valeurs suivantes dans les zones de texte de la page **Installer de nouvelles applications** :

Tableau 36. Valeurs d'installation

Zone de texte	Valeur
Archive	`base_servicerest/lib/wxsrestservice.war`
Plan	`base_servicerest/gettingstarted/wasce/geronimo-web.xml`

5) Cliquez sur le bouton Installer.

   La page de la console indique que l'application est installée et a démarré.

6) Recherchez le message ci-après sur la console ou dans le journal de sortie système de WebSphere Application Server Community Edition pour vérifier que le service de données REST a bien démarré :

   `CWOBJ4000I : Le service de données REST de WebSphere eXtreme Scale a été démarré.`

5. Vérifiez que le service de données REST fonctionne :
   a. Ouvrez le lien suivant dans une fenêtre de navigateur : `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`. Le document de service de la grille NorthwindGrid s'affiche.
   b. Ouvrez le lien suivant dans une fenêtre de navigateur : `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`. Le document Entity Model Data Extensions (EDMX) s'affiche.
6. Pour arrêter les processus de grille, utilisez `CTRL+C` dans les fenêtres de commande respectives.
7. Pour arrêter WebSphere Application Server Community Edition, utilisez la commande suivante :
   - UNIX Linux `racine_wasce/bin/shutdown.sh`
   - Windows `racine_wasce\bin\shutdown.bat`

   Conseil : L'ID utilisateur par défaut est `system` et le mot de passe `manager`. Si vous utilisez un port personnalisé, utilisez l'option `-port`.

Déploiement du service de données REST sur Apache Tomcat

Java

Cette rubrique explique comment configurer le service de données REST WebSphere eXtreme Scale sur Apache Tomcat Version 5.5 et les versions suivantes.

Pourquoi et quand exécuter cette tâche

- Un environnement JRE ou JDK Oracle IBM ou Version 6 , ou version suivante, est installé et une variable d'environnement JAVA_HOME est spécifiée.
- Apache Tomcat Version 5.5 ou ultérieure est installé. Voir Apache Tomcat pour savoir comment installer Tomcat.
- Installation autonome WebSphere eXtreme Scale.

Procédure

1. Si vous utilisez un JRE ou un JDK Oracle, installez l'ORB IBM dans Tomcat :
   a. Tomcat version 5.5 :

      Copiez tous les fichiers JAR depuis :

      le répertoire *base_wxs*`/lib/endorsed`

      vers :

le répertoire *racine_tomcat*/common/endorsed

b. Tomcat version 6.0 :

Créez un répertoire "endorsed" :

UNIX Linux `mkdir racine_tomcat/endorsed`

Windows `md racine_tomcat/endorsed`

Copiez tous les fichiers JAR de :

`base_wxs/lib/endorsed`

vers :

`racine_tomcat/common/endorsed`

2. Configurez et démarrez une grille de données.
   a. Pour plus d'informations sur la configuration d'une grille de données pour l'utiliser avec le service de données REST, voir Chapitre 6, «Configuration», à la page 269.
   b. Vérifiez qu'un client eXtreme Scale arrive à se connecter aux entités de la grille et à y accéder. Pour un exemple, voir «Configuration des services de données REST», à la page 435.
3. Générez le répertoire ou le fichier JAR de configuration du service REST d'eXtreme Scale. Pour les détails, voir dans «Installation du service de données REST», à la page 221 les explications sur le packaging et le déploiement.
4. Déployez le module du service de données REST : wxsrestservice.war sur le serveur Tomcat.

   Copiez le fichier `wxsrestservice.war` depuis :

   rép_base_servicerest/lib

   vers :

   racine_tomcat/webapps
5. Ajoutez le fichier JAR d'exécution du client ObjectGrid et le fichier JAR de l'application dans le chemin d'accès aux classes partagé, dans Tomcat :
   a. Editez le fichier *racine_tomcat*/conf/catalina.properties.
   b. Ajoutez les noms de chemin suivants à la fin de la propriété shared.loader, en les séparant par une virgule :
      - *rép_base_wxs*/lib/ogclient.jar
      - *rép_base_servicerest*/gettingstarted/restclient/bin
      - *rép_base_servicerest*/gettingstarted/common/bin
6. Si vous utilisez la sécurité Java 2, ajoutez les droits de sécurité au fichier de règles tomcat :
   - Si vous utilisez Tomcat version 5.5 :

     Fusionnez le contenu de l'exemple de fichier de règles catalina 5.5 qui se trouve dans

     rép_base_servicerest/gettingstarted/tomcat/catalina-5_5.policy avec le fichier *racine_tomcat*/conf/catalina.policy.
   - Si vous utilisez Tomcat version 6.0 :

     Fusionnez le contenu de l'exemple de fichier de règles catalina 6.0 qui se trouve dans

     rép_base_servicerest/gettingstarted/tomcat/catalina-6_0.policy avec le fichier *racine_tomcat*/conf/catalina.policy.
7. Démarrez le serveur Tomcat :

- **Si vous utilisez Tomcat 5.5 sous UNIX or Windows, ou la distribution ZIP de Tomcat 6.0 :**
  a. `cd racine_tomcat/bin`
  b. Démarrez le serveur :
     – Sans la sécurité Java 2 activée :

       UNIX Linux `./catalina.sh run`

       Windows `catalina.bat run`

     – Avec la sécurité Java 2 activée :

       UNIX Linux `./catalina.sh run -security`

       Windows `catalina.bat run -security`

  c. Les journaux d'Apache Tomcat sont affichés sur la console. Lorsque le service de données REST a correctement démarré, le message suivant est affiché dans la console d'administration :

     `CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.`

- **Si vous utilisez Tomcat 6.0 sous Windows à l'aide de la distribution du programme d'installation Windows :**
  a. `cd /bin`
  b. Démarrez l'outil de configuration d'Apache Tomcat 6 :

     `tomcat6w.exe`

  c. Pour activer la sécurité Java 2 : (facultatif) :

     Ajoutez les entrées suivantes aux options Java dans la page Java de la fenêtre des propriétés d'Apache Tomcat 6 :

     `-Djava.security.manager`

     `-Djava.security.policy=\conf\catalina.policy`

  d. Cliquez sur le bouton Démarrer de la fenêtre de propriétés d'Apache Tomcat 6 pour démarrer le serveur Tomcat.
  e. Consultez les journaux suivants pour vérifier que le serveur Tomcat a été correctement démarré :
     – *racine_tomcat*`/bin/catalina.log`

       Affiche le statut du moteur du serveur Tomcat
     – *racine_tomcat*`/bin/stdout.log`

       Affiche le journal de la sortie système.

  f. Si le service de données REST est correctement démarré, le message suivant est affiché dans le journal de la sortie système :

     `CWOBJ4000I: Le service de données REST de WebSphere eXtreme Scale a été démarré.`

8. Vérifiez que le service de données REST fonctionne bien.

   Ouvrez un navigateur Web et accédez à l'adresse URL suivante :

   `http://`*`host:port/racine_contexte`*`/restservice/`*`nom_grille`*

   Le port par défaut pour Tomcat est 8080 et il est configuré dans le fichier *racine_tomcat*`/conf/server.xml` dans l'élément <Connector>.

   Par exemple :

   `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/`

Résultats

Le document de service AtomPub est affiché.

Démarrage de services de données REST dans Apache Tomcat : Java

Cette rubrique explique comment configurer et démarrer le service de données REST d'eXtreme Scale avec Apache Tomcat version 5.5 ou plus récente.

Avant de commencer

Vérifiez que l'exemple de grille de données eXtreme Scale est démarré. Voir «Activation du service de données REST», à la page 436 pour plus d'informations sur le démarrage de la grille.

Procédure

1. Téléchargez et installez dans racine_tomcat Apache Tomcat version 5.5 ou ultérieure. Par exemple : `/opt/tomcat`
2. Installez sur le serveur Tomcat le service de données REST d'eXtreme Scale et l'exemple fourni :
   a. Si vous utilisez un JRE ou un JDK Oracle, vous devez installer l'ORB IBM dans Tomcat :
      - Pour Tomcat version 5.5

        Copiez tous les fichiers JAR de :

        `base_wxs/lib/endorsed`

        vers

        `racine_tomcat/common/endorsed`
      - Pour Tomcat version 6.0
        1) Créez un répertoire "validé".
           – UNIX Linux `mkdir racine_tomcat/endorsed`
           – Windows `md racine_tomcat/endorsed`
        2) Copiez tous les fichiers JAR de :

           `base_wxs/lib/endorsed`

           vers

           `racine_tomcat/endorsed`

   b. Déployez le module de service de données REST : wxsrestservice.war vers le serveur Tomcat.

      Copiez le fichier wxsrestservice.war depuis :

      `base_servicerest/lib`

      vers :

      `racine_tomcat/webapps`

   c. Ajoutez le fichier JAR d'exécution client ObjectGrid et le fichier JAR d'application au chemin d'accès aux classes dans Tomcat :
      1) Modifiez le fichier `racine_tomcat/conf/catalina.properties`.
      2) Ajoutez les noms de chemins suivants à la fin de la propriété shared.loader sous forme de liste séparée par des virgules :
         - base_wxs/lib/ogclient.jar
         - base_servicerest/gettingstarted/restclient/bin
         - base_servicerest/gettingstarted/common/bin

Important : Le séparateur de chemin doit être une barre **oblique**.

3. Si la grille de données eXtreme Scale a été démarrée avec la sécurité eXtreme Scale activée, définissez la propriété suivante dans le fichier `restservice_home/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=base_servicerest/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Démarrez le serveur Tomcat avec le service de données REST :
   - Si vous utilisez Tomcat 5.5 sous UNIX ou Windows, ou Tomcat 6.0 sous UNIX :
     a. `cd racine_tomcat/bin`
     b. Démarrez le serveur :
        - UNIX Linux `./catalina.sh run`
        - Windows `catalina.bat run`
     c. La console affiche ensuite les journaux d'Apache Tomcat. Au démarrage du service de données REST, le message suivant s'affiche dans la console d'administration :

        `CWOBJ4000I : Le service de données REST de WebSphere eXtreme Scale a été démarré.`
   - Si vous utilisez Tomcat 6.0 sous Windows :
     a. `cd racine_tomcat/bin`
     b. Démarrez l'outil de configuration Apache Tomcat 6 avec la commande suivante : `tomcat6w.exe`
     c. Dans la fenêtre des propriétés d'Apache Tomcat 6, cliquez sur le bouton Démarrer pour démarrer le serveur Tomcat.
     d. Révisez les journaux suivants pour vérifier que le serveur Tomcat a démarré correctement :
        - `racine_tomcat/bin/catalina.log`

          Affiche l'état du moteur du serveur Tomcat.
        - `racine_tomcat/bin/stdout.log`

          Affiche le journal de sortie système.
     e. Au démarrage du service de données REST, le message suivant s'affiche dans le journal de sortie système : `CWOBJ4000I : Le service de données REST de WebSphere eXtreme Scale a été démarré.`
5. Vérifiez que le service de données REST fonctionne :
   a. Ouvrez un navigateur et accédez à :

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid

      Le document de service pour NorthwindGrid s'affiche.
   b. Accédez à :

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Le document Entity Model Data Extensions (EDMX) s'affiche.
6. Pour arrêter les processus de grille de données, utilisez CTRL+C dans la fenêtre de commande correspondante.
7. Pour arrêter Tomcat, utilisez CTRL +C dans la fenêtre depuis laquelle vous l'avez démarré.

Configuration des navigateurs Web pour accéder aux flux ATOM du service de données REST

Java

Par défaut, le service de données REST d'eXtreme Scale crée des flux ATOM lorsqu'il utilise un navigateur Web. Le format des flux ATOM risque de ne pas être compatible avec les navigateurs plus anciens. Il y a également le risque d'une mauvaise interprétation des données qui ne seront pas affichées comme des données XML. Vous pouvez configurer Internet Explorer Version 8 et Firefox Version 3 pour afficher les flux ATOM et les données XML dans le navigateur.

Pourquoi et quand exécuter cette tâche

Par défaut, le service de données REST d'eXtreme Scale crée des flux ATOM lorsqu'il utilise un navigateur Web. Le format des flux ATOM risque de ne pas être compatible avec les autres navigateurs ou d'être interprété de sorte que les données ne puissent pas être consultées comme des données XML. Pour les anciens navigateurs, vous serez invité à sauvegarder les fichiers sur le disque. Une fois que les fichiers ont été téléchargés, utilisez votre lecteur XML favori pour consulter les fichiers. Le XML généré n'étant pas formaté pour être affiché, tout est imprimé sur une seule ligne. La plupart des programmes de lecture XML, tels qu'Eclipse, prennent en charge le reformatage du XML dans un format lisible.

Pour les navigateurs modernes, tels que Microsoft Internet Explorer Version 8 et Firefox Version 3, les fichiers XML ATOM peuvent être affichés de manière native dans le navigateur. Les rubriques ci-après fournissent des détails sur la manière de configurer Internet Explorer Version 8 et Firefox Version 3 pour afficher les flux ATOM et le XML dans le navigateur.

Procédure

Configuration d'Internet Explorer Version 8

- Pour permettre à Internet Explorer de lire les flux ATOM que le service de données REST génère, procédez comme suit :
  1. Cliquez sur **Outils** > **Options Internet**
  2. Sélectionnez l'onglet **Contenu**
  3. Cliquez sur le bouton **Paramètres** de la section **Flux et composants Web Slice**
  4. Désélectionnez la case "Activer le mode Lecture du flux"
  5. Cliquez sur **OK** pour retourner au navigateur.
  6. Redémarrez Internet Explorer.

Configuration de Firefox Version 3

- Firefox n'affiche pas automatiquement les pages avec le type de contenu suivant : application/atom+xml. La première fois qu'une page est affichée, Firefox vous invite à sauvegarder le fichier. Pour afficher la page, ouvrez le fichier avec Firefox, comme suit :
  1. Dans la boîte de dialogue de sélection de l'application, sélectionnez le bouton d'option "Ouvrir avec" et cliquez sur le bouton **Parcourir**.
  2. Accédez au répertoire d'installation de Firefox. Par exemple : `C:\Program Files\Mozilla Firefox`
  3. Sélectionnez `firefox.exe`, puis cliquez sur le bouton **OK**.

4. Cochez la case "Toujours utiliser ce programme pour ouvrir ce type de fichier".
5. Cliquez sur le bouton **OK**.
6. Firefox affiche ensuite la page XML ATOM dans une nouvelle fenêtre ou page de navigateur

- Firefox affiche automatiquement les flux ATOM dans un format lisible. Toutefois, les flux créés par le service de données REST incluent XML. Firefox ne peut pas afficher le XML à moins que vous ne désactiviez le présentateur de flux. Contrairement à Internet Explorer, dans Firefox, le plug-in d'affichage des flux ATOM doit être édité de manière explicite. Pour configurer Firefox afin qu'il puisse lire les flux ATOM comme des fichiers XML, procédez comme suit :
  1. Ouvrez le fichier suivant dans un éditeur de texte : `<firefoxInstallRoot>\components\FeedConverter.js`. Dans le chemin d'accès, `<firefoxInstallRoot>` correspond au répertoire principal dans lequel Firefox est installé.

     Pour les systèmes d'exploitation Windows, le répertoire par défaut est le suivant : `C:\Program Files\Mozilla Firefox`.
  2. Recherchez le fragment de code similaire au suivant :

     ```
     // montre la page de flux si elle n'a pas été reniflée
     et que nous avons un document,
     // ou un document, un titre et un lien ou un ID
     if (result.doc && (!this._sniffed ||
         (result.doc.title && (result.doc.link || result.doc.id)))) {
     ```
  3. Placez les deux lignes commençant par `if` et `result` en commentaire, en les précédant de deux barres obliques (`//`).
  4. Ajoutez l'instruction suivante au fragment de code : `if(0) {`.
  5. Le texte résultant doit ressembler au suivant :

     ```
     // montre la page de flux si elle n'a pas été reniflée
     et que nous avons un document,
     // ou un document, un titre et un lien ou un ID
     //if (result.doc && (!this._sniffed ||
     //    (result.doc.title && (result.doc.link || result.doc.id)))) {
     if(0) {
     ```
  6. Enregistrez le fichier.
  7. Redémarrez Firefox
  8. Firefox peut maintenant afficher automatiquement tous les flux dans le navigateur.
- Testez votre configuration en essayant quelques URL.

Exemple

Cette section décrit certains exemples d'URL qui peuvent être utilisés pour afficher les données qui ont été ajoutées par l'exemple d'initiation fourni avec le service de données REST. Avant d'utiliser les URL suivantes, ajoutez le fichier par défaut à l'exemple de grille de données eXtreme Scale en utilisant l'exemple de client Java ou l'exemple de client Visual Studio WCF Data.

Dans les exemples qui suivent, l'on part du principe que le port utilisé est le 8080, mais cela peut varier. Reportez-vous à la section pour des explications détaillées sur la manière de configurer le service de données REST sur différents serveurs d'applications.

- Visualiser un seul client dont l'ID est "ACME" :

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')
  ```
- Visualiser toutes les commandes du client "ACME" :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')/orders
```

- Visualiser le client "ACME" et les commandes :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')?$expand=orders
```

- Visualiser la commande 1000 du client "ACME" :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')
```

- Visualiser la commande 1000 du client "ACME" et le Customer qui lui est associé :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Order(orderId=1000,customer_customerId='ACME')?$expand=customer
```

- Visualiser la commande 1000 du client "ACME" et le Customer et les OrderDetails associés à ce client :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Order(orderId=1000,customer_customerId='ACME')?$expand=customer,orderDetails
```

- Visualiser toutes les commandes du client "ACME" pour le mois d'octobre 2009 (GMT) :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Customer(customerId='ACME')/orders?$filter=orderDate
ge datetime'2009-10-01T00:00:00'
and orderDate lt datetime'2009-11-01T00:00:00'
```

- Visualiser les trois premières commandes et les trois premiers orderDetails du client "ACME" pour le mois d'octobre 2009 (GMT) :

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
Customer(customerId='ACME')/orders?$filter=orderDate
ge datetime'2009-10-01T00:00:00'
and orderDate lt datetime'2009-11-01T00:00:00'
&$orderby=orderDate&$top=3&$expand=orderDetails
```

Utilisation d'un client Java avec les services de données REST

Java

L'application client Java utilise l'API EntityManager d'eXtreme Scale pour insérer des données dans la grille.

Pourquoi et quand exécuter cette tâche

Les sections précédentes expliquent comment créer une grille de données eXtreme Scale et configurer et démarrer le service de données eXtreme Scale REST. L'application client Java utilise l'API EntityManager d'eXtreme Scale pour insérer des données dans la grille. Elle ne montre pas comment utiliser les interfaces REST. Ce client a pour fonction de montrer comme l'API EntityManager est utilisée pour interagir avec la grille de données eXtreme Scale et de modifier les données dans la grille. Pour afficher des données dans la grille avec le service de données REST, utilisez un navigateur Web ou l'application client Visual Studio 2008.

Procédure

Pour ajouter rapidement du contenu à la grille de données eXtreme Scale, exécutez la commande suivante :

1. Ouvrez une ligne de commande ou une fenêtre de terminal, puis définissez la variable d'environnement JAVA_HOME :
   - Linux UNIX `export JAVA_HOME=base_java`
   - Windows `set JAVA_HOME=base_java`
2. `cd base_servicerest/gettingstarted`

3. Insérez des données dans la grille. Les données insérées seront extraites ultérieurement à l'aide d'un navigateur Web et du service de données REST.

   Si la grille de données a été démarrée *sans*eXtreme Scale la sécurité, utilisez les commandes suivantes.

   - UNIX Linux `./runclient.sh load default`
   - Windows `runclient.bat load default`

   Si la grille de données a été démarrée *sans*eXtreme Scale la sécurité, utilisez les commandes suivantes.

   - UNIX Linux `./runclient_secure.sh load default`
   - Windows `runclient_secure.bat load default`

   Pour un client Java, utilisez la syntaxe suivante :

   - UNIX Linux `runclient.sh` *`commande`*
   - Windows `runclient.bat` *`commande`*

   Les commandes suivantes sont disponibles :

   - `load default`

     Charge un ensemble prédéfini d'entités Customer, Category et Product dans la grille de données et crée un ensemble aléatoire de commandes pour chaque client.
   - `load category` *`IDcatégorie Nomcatégorie IDpremierproduit nbre_produits`*

     Crée une catégorie de produits et un nombre fixe d'entités Product dans la grille de données. Le paramètre firstProductId identifie le numéro d'identification du premier produit et chaque produit suivant se voit affecter l'identifiant suivant jusqu'à ce que le nombre spécifié de produits soit créé.
   - `load customer` *`companyCode contactNamecompanyName numOrders firstOrderIdshipCity maxItems discountPct`*

     Charge un nouveau client dans la grille de données et crée un groupe fixe d'entités Order pour un produit aléatoire chargé actuellement dans la grille. Le nombre de commandes est déterminé par le paramètre <numOrders>. Chaque commande sera dotée d'un nombre aléatoire d'entités OrderDetail jusqu'à <maxItems>
   - `display customer` *`companyCode`*

     Affiche une entité Customer et les entités Order et OrderDetail associées.
   - `display category` *`categoryId`*

     Affiche une entité de produit Category et les entités Product associées.

Résultats

- `runclient.bat load default`
- `runclient.bat load customer IBM "John Doe" "IBM Corporation" 5 5000 Rochester 5 0.05`
- `runclient.bat load category 5 "Household Items" 100 5`
- `runclient.bat display customer IBM`
- `runclient.bat display category 5`

Exécution et génération de l'exemple de données de la grille et du client Java avec Eclipse

L'échantillon de démarrage de service de données REST peut être mis à jour et amélioré à l'aide d'Eclipse. Pour plus d'informations sur la configuration de votre environnement Eclipse, voir le document : `base_servicerest/gettingstarted/ECLIPSE_README.txt`.

Une fois que le projet WXSRestGettingStarted a été importé dans Eclipse et que sa génération s'effectue correctement, l'échantillon se recompile automatiquement et les fichiers script utilisés pour démarrer le serveur de conteneur et le client sélectionnent automatiquement les fichiers de classes et les fichiers XML. Le service de données REST détecte automatiquement toute modification, car le serveur Web est configuré pour effectuer une lecture automatique des répertoires de construction Eclipse.

Important : En cas de modification des fichiers source ou de configuration, le serveur de conteneur eXtreme Scale et l'application du service de données REST doivent tous deux être redémarrés. Le serveur de conteneur eXtreme Scale doit être démarré avant l'application Web du service de données REST.

Client WCF de Visual Studio 2008 avec le service de données REST

L'exemple Mise en route du service de données REST d'eXtreme Scale inclut un client WCF Data Services qui peut interagir avec le service de données REST. L'exemple est écrit comme une application de ligne de commande dans C#.

Configuration logicielle requise

L'exemple de client WCF Data Services C# requiert la configuration suivante :

- Système d'exploitation
  - Microsoft Windows XP
  - Microsoft Windows Server 2003
  - Microsoft Windows Server 2008
  - Microsoft Windows Vista
- Microsoft Visual Studio 2008 avec Service Pack 1

  Conseil : Pour les configurations matérielle et logicielle supplémentaires requises, voir le lien précédent.
- Microsoft .NET Framework 3.5 Service Pack 1
- Microsoft Support : La mise à jour pour .NET Framework 3.5 Service Pack 1 est disponible

Génération et exécution du client Mise en route

L'exemple de client des services de données WCF inclut un projet et une solution Visual Studio 2008, ainsi que le code source permettant d'exécuter l'exemple. L'exemple doit être chargé dans Visual Studio 2008 et compilé dans un programme exécutable sous Windows pour pouvoir être exécuté. Pour générer et exécuter l'exemple, voir le document texte : `base_servicerest/gettingstarted/VS2008_README.txt`.

Syntaxe des commandes du client WCF Data Services C#

Windows `WXSRESTGettingStarted.exe <URL du service> <commande>`

L'<URL du service> est l'URL du service de données REST d'eXtreme Scale configuré dans la section .

Les commandes suivantes sont disponibles :

- `load default`

  Charge un ensemble prédéfini d'entités Customer, Category et Product dans la grille de données et crée un ensemble aléatoire de commandes pour chaque client.

- `load category <categoryId> <categoryName> <firstProductId> <numProducts>`

  Crée une catégorie de produits et un nombre fixe d'entités Product dans la grille de données. Le paramètre firstProductId identifie l'identificateur du premier produit et chaque produit suivant reçoit le prochain ID jusqu'à ce que le nombre de produits spécifié soit créé.

- `load customer <companyCode> <contactName> <companyName> <numOrders> <firstOrderId> <shipCity> <maxItems> <discountPct>`

  Charge un nouveau client dans la grille de données et crée un groupe fixe d'entités Order pour un produit aléatoire déjà chargé actuellement dans la grille. Le nombre de commandes est déterminé par le paramètre <numOrders>. Chaque commande contient un nombre aléatoire d'entités OrderDetail, inférieur à la valeur <maxItems>

- `display customer <companyCode>`

  Affiche un entité Customer et les entités Order et OrderDetail associées.

- `display category <categoryId>`

  Affiche l'entité Category d'un produit et les entités Product associées.

- `unload`

  Supprime toutes les entités chargées à l'aide de la commande "default load".

Les exemples suivants illustrent diverses commandes.

- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load default`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load customer`
- `IBM "John Doe" "IBM Corporation" 5 5000 Rochester 5 0.05`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load category 5 "Household Items" 100 5`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid display customer IBM`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid display category 5`

Déploiement d'une passerelle REST

Vous pouvez déployer et configurer la passerelle REST pour WebSphere eXtreme Scale dans WebSphere Application Server ou dans un serveur Profil Liberty.

Avant de commencer

Vérifiez qu'un serveur Profil Liberty est créé. Pour plus d'informations, voir Installation de Profil Liberty.

Pourquoi et quand exécuter cette tâche

La passerelle REST est un servlet qui est défini dans le fichier archive Web (WAR) `wxsRESTGateway.war`. Avec cette passerelle REST, vous utilisez un identificateur URI (Uniform Resource Identifier) pour accéder aux données dans la grille de données.

Procédure

1. Activez la fonction de passerelle REST en modifiant manuellement le fichier `server.xml` ou en utilisant Liberty Profile Developer Tools.
   - Activez la passerelle REST dans le fichier Profil Liberty `server.xml`.

     ```
     <featureManager>
       <feature>eXtremeScale.rest-1.1</feature>
     </featureManager>
     ```
   - Activez la passerelle REST dans le fichier Profil Liberty `server.xml` en utilisant Liberty Profile Developer Tools.
     - Démarrez IBMWebSphere Application Server Version 8.6 Liberty Profile Developer Tools. Pour plus d'informations, voir Scénario : Exécution de serveurs de grille dans le profil Liberty en utilisant des outils Eclipse.
     - Dans l'onglet **Design**, sélectionnez **Feature Manager**. Cliquez sur **Add** dans la section Feature Manager Details. Sélectionnez la fonction **eXtremeScale.rest-1.1** et ajoutez-la.
     - Feature Manager étant sélectionné, cliquez sur **Add** dans la section Feature Manager Details. Sélectionnez la fonction **servlet-3.0** et ajoutez-la.
     - Enregistrez le fichier `server.xml`.
   - Activez la passerelle REST dans WebSphere Application Server.
     - Installez WebSphere eXtreme Scale avec WebSphere Application Server. Pour plus d'informations, voir «Installation de WebSphere eXtreme Scale ou de WebSphere eXtreme Scale Client avec WebSphere Application Server», à la page 226.
     - Déployez le fichier *was_install_root*`\optionalLibraries\ObjectGrid\restgateway\wxsRESTGateway.war` sur WebSphere Application Server.
2. Configurez la passerelle REST.
   a. Configurez la passerelle REST dans le fichier `server.xml`. Entrez la ligne de code suivante :

      ```
      <xsREST contextRoot="myContextRoot" remoteDomain="myDomain"/>
      ```

      Avertissement : Les attributs, `contextRoot` et `remoteDomain`, sont facultatifs. La racine de contenu par défaut est `resources`.

   b. Configurez un serveur eXtreme Scale. Pour plus d'informations, voir Configuration de serveurs eXtreme Scale pour une exécution dans le profil Liberty.

   c. Configurez un serveur de conteneur.

      Les options suivantes sont disponibles pour configurer un service de conteneur :

      - Copiez un fichier valide `objectgrid.xml` (avec ou sans fichier `objectGridDeployment.xml` correspondant) dans le répertoire *wlp_home*`/usr/servers/`*server_name*`/grids`. Ce répertoire `grids` est

surveillé par le produit lors de l'exécution. Les modifications des fichiers dans ce répertoire génèrent des événements dans l'environnement d'exécution Profil Liberty. Par exemple, lorsqu'un nouveau fichier `objectgrid.xml` ou `objectGridDeployment.xml` ou ces deux nouveaux fichiers sont détectés, un serveur de conteneur est créé. Lorsque l'un de ces fichiers est supprimé, eXtreme Scale arrête ce serveur de conteneur. Lorsque les fichiers sont modifiés, eXtreme Scale arrête et redémarre le conteneur. Plusieurs conteneurs de fragment peuvent exister dans un même serveur eXtreme Scale, ce qui implique que des sous-répertoires existent dans le répertoire `grids`.

- Installez un ensemble eXtreme. Cet ensemble doit faire référence à un fichier `blueprint.xml` qui contient les médatadonnées de serveur. Cette méthode de lancement d'un serveur est similaire à la façon dont vous pouvez démarrer des serveurs dans les environnements OSGi dans WebSphere eXtreme Scale Version 7.1.1 sans le Profil Liberty. Dans la version 8.5, l'élément serveur n'est plus nécessaire dans le fichier `blueprint.xml`. Par conséquent, vous devez définir les métadonnées du serveur dans le fichier `server.xml`. Installez et démarrez les ensembles en les faisant glisser et en les déposant dans le répertoire `grids` comme vous le faites pour déposer des fichiers XML dans le répertoire `grids`.
- Utilisez l'API de serveur embarqué. Cette option revient à démarrer le serveur dans un environnement autonome. Avec le Profil Liberty, cependant, vous devez vérifier le code pour démarrer le serveur eXtreme Scale.

Pour plus d'informations, voir Démarrage et arrêt des serveurs dans le profil Liberty.

3. Démarrez le serveur Profil Liberty pour exécuter la passerelle client REST.

Que faire ensuite

Lorsque la passerelle REST est activée, un utilisateur ayant accès au servlet peut accéder aux données dans une grille de données. Par conséquent, vous devez utiliser la sécurité d'application Web dans WebSphere Application Server pour contrôler l'autorisation. Pour plus d'informations sur la sécurisation des applications Web qui utilisent cette passerelle REST, voir Securing web applications in the WebSphere Application Server dans le centre de documentation.

Le fichier `wxsRESTGateway.war`, qui contient le fichier `web.xml` pour la configuration de la sécurité, se trouve dans les emplacements suivants en fonction de votre installation :

- *`wlp_install_root`*`/wxs/web/rest`
- *`was_install_root`*`/optionalLibraries/ObjectGrid/restgateway`
- *`wxs_standalone_install_root`*`/ObjectGrid/restgateway`

Maintenant, vous pouvez utiliser le service de données Web dans le Profil Liberty pour communiquer avec la grille de données via un identificateur URI. Pour plus d'informations, voir Développement d'applications de grilles de données avec la passerelle REST.

Configuration des serveurs pour OSGi

Java

WebSphere eXtreme Scale inclut un ensemble OSGi de serveur qui permet de démarrer et de configurer les serveurs et les conteneurs dans une infrastructure OSGi. Les rubriques de configuration expliquent comment utiliser l'ensemble de serveur eXtreme Scale, le service OSGi Blueprint et la configuration eXtreme Scale pour exécuter des serveurs eXtreme Scale dans une infrastructure OSGi Eclipse Equinox.

Pourquoi et quand exécuter cette tâche

Vous devez exécuter les tâches suivantes pour démarrer un serveur eXtreme Scale dans Eclipse Equinox:

Procédure

1. Créez un ensemble OSGi qui stockera les plug-in eXtreme Scale en les exposant comme services et mettez à jour le fichier XML descripteur d'ObjectGrid pour référencer les services.
2. Configurez OSGi pour démarrer le serveur de conteneur eXtreme Scale.
3. Installez et redémarrez l'ensemble de serveur eXtreme Scale dans l'infrastructure OSGi.
4. Installez et démarrez l'ensemble OSGi qui contient les plug-in eXtreme Scale.

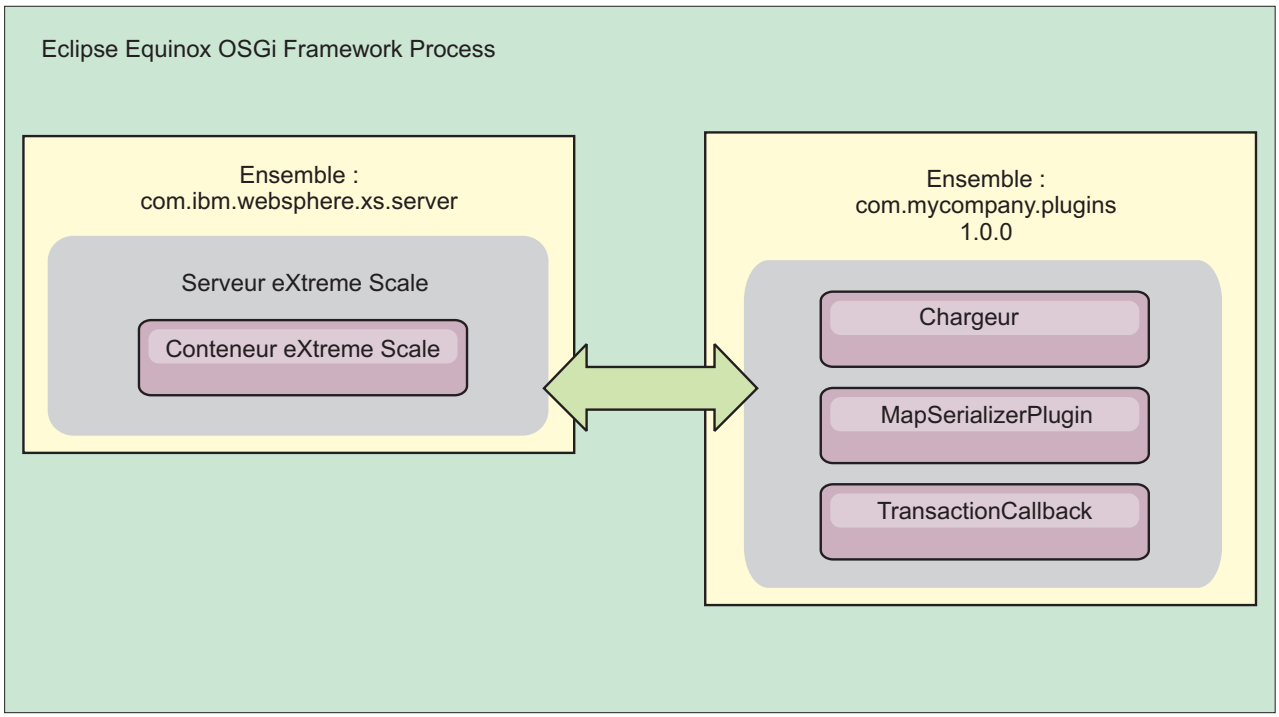


Figure 52. Processus Eclipse Equinox d'installation et de démarrage des ensembles OSGi avec des plug-in eXtreme Scale

Configuration des plug-in eXtreme Scale avec OSGi Blueprint

Java

Tous les plug-in eXtreme Scale ObjectGrid et BackingMap peuvent être définis comme beans et services OSGi en utilisant le service OSGi Blueprint disponible avec Eclipse Gemini ou Apache Aries.

Avant de commencer

Pour pouvoir configurer vos plug-in comme services OSGi, vous devez regrouper les plug-in dans un ensemble OSGi et connaître les concepts de base des plug-in requis. L'ensemble doit importer les modules client ou serveur WebSphere eXtreme Scale et d'autres packages dépendants nécessaires aux plug-in ou créer une dépendance d'ensemble dans les ensembles de serveur ou de client eXtreme Scale. Cette rubrique explique comment configurer le fichier XML Blueprint XML pour créer des beans de plug-in et les exposer comme services OSGi pour que eXtreme Scale les utilise.

Pourquoi et quand exécuter cette tâche

Les beans et services sont définis dans un fichier XML Blueprint et le conteneur Blueprint découvre, crée et interconnecte les beans et les expose comme services. Le processus rend les beans accessibles aux autres ensembles OSGi, y compris les ensembles de serveur et de client eXtreme Scale.

Lors de la création de services de plug-in personnalisés pour les utiliser avec eXtreme Scale, l'ensemble qui doit héberger les plug-in doit être configuré pour utiliser Blueprint. En outre, un fichier XML Blueprint doit être créé et stocké dans l'ensemble. Lisez la rubrique relative à la création d'applications OSGi avec la spécification Blueprint Container qui décrit de manière générale la spécification.

Procédure

1. Créez un fichier XML Blueprint. Attribuez-lui un nom de votre choix. Toutefois, vous devez inclure l'espace de nom Blueprint :

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
   ...
   </blueprint>
   ```

2. Créez des définitions de bean dans le fichier XML Blueprint pour chaque plug-in eXtreme Scale.

   Les beans sont définis en utilisant l'élément `<bean>`, ils peuvent être connectés à d'autres références de bean et ils peuvent inclure des paramètres d'initialisation.

   Important : Lors de la définition d'un bean, vous devez utiliser la portée correcte. Blueprint prend en charge les portées singleton et prototype. eXtreme Scale prend également en charge une portée de fragment personnalisée.

   Définissez la plupart des plug-in eXtreme Scale comme prototype ou beans à portée de fragment, car tous les beans doivent être uniques pour chaque fragment ObjectGrid ou instance BackingMap auquel ou à laquelle ils sont associés. Les beans à portée de fragment peuvent être utiles lorsque vous utilisez les beans dans d'autres contextes pour pouvoir extraire l'instance correcte.

   Pour définir un bean à portée prototype, utilisez l'attribut `scope="prototype"` sur le bean :

   ```
   <bean id="myPluginBean" class="com.mycompany.MyBean" scope="prototype">
   ...
   </bean>
   ```

   Pour définir un beans à portée de fragment, vous devez ajouter l'espace de nom `objectgrid` au schéma XML et utiliser l'attribut `scope="objectgrid:shard"` sur le bean :

```
<?xml version="1.0" encoding="UTF-8"?>

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
           xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"

    xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                http://www.ibm.com/schema/objectgrid/objectgrid.xsd">

    <bean id="myPluginBean" class="com.mycompany.MyBean"
   scope="objectgrid:shard">
    ...
    </bean>

...
```

3. Créez des définitions de bean PluginServiceFactory pour chaque bean de plug-in. Tous les beans eXtreme Scale doivent avoir un bean PluginServiceFactory défini pour que la portée de bean correcte puisse être appliquée. eXtreme Scale inclut une fabrique BlueprintServiceFactory que vous pouvez utiliser. Elle contient deux propriétés que vous devez définir. Vous devez affecter à la propriété blueprintContainer la référence `blueprintContainer` et attribuer à la propriété beanId le nom de l'identificateur du bean. Lorsque eXtreme Scale recherche le service pour instancier les beans appropriés, le serveur recherche l'instance du composant bean en utilisant le conteneur Blueprint.

   ```
   bean id="myPluginBeanFactory"
       class="com.ibm.websphere.objectgrid.plugins.osgi.BluePrintServiceFactory">
       <property name="blueprintContainer" ref="blueprintContainer"/>
       <property name="beanId" value="myPluginBean" />
   </bean>
   ```

4. Créez un gestionnaire de service pour chaque bean PluginServiceFactory. Chaque gestionnaire de service expose le bean PluginServiceFactory en utilisant l'élément `<service>`. L'élément de service identifie le nom à exposer à OSGi, la référence au bean PluginServiceFactory et l'interface à exposer, ainsi que le classement du service. eXtreme Scale utilise le classement du gestionnaire de service pour effectuer des mises à niveau de service lorsque la grille eXtreme Scale est active. Si le classement n'est pas défini, l'infrastructure OSGi utilise le classement 0 par défaut. Consultez la rubrique relative à la mise à jour des classements de service pour plus d'informations.

   Blueprint contient diverses options de configuration des gestionnaires de service. Pour définir un gestionnaire de service simple pour un bean PluginServiceFactory, créez un élément `<service>` pour chaque bean PluginServiceFactory :

   ```
   <service ref="myPluginBeanFactory"
       interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
       ranking="1">
   </service>
   ```

5. Stockez le fichier XML Blueprint dans l'ensemble de plug-in. Le fichier XML Blueprint doit être stocké dans le répertoire `OSGI-INF/blueprint` du conteneur Blueprint pour être découvert.

   Pour stocker le fichier XML Blueprint dans un répertoire différent, vous devez définir l'en-tête de manifeste Bundle-Blueprint suivant :

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

Résultats

Les plug-in eXtreme Scale sont maintenant configurés pour être exposés dans un conteneur OSGi Blueprint. En outre, le fichier XML descripteur ObjectGrid est configuré pour référencer les plug-in en utilisant le service OSGi Blueprint.

Configuration des serveurs avec OSGi Blueprint

Java

Vous pouvez configurer les serveurs de conteneur WebSphere eXtreme Scale en utilisant un fichier XML OSGi Blueprint qui permet de simplifier le regroupement et le développement d'ensembles de serveur autonomes.

Avant de commencer

Cette rubrique suppose que vous avez exécuté les tâches suivantes :

- L'infrastructure OSGi Eclipse Equinox a été installée et démarrée avec le conteneur Eclipse Gemini ou Apache Aries Blueprint.
- L'ensemble de serveur eXtreme Scale a été installé et démarré.
- L'ensemble de plug-in dynamiques eXtreme Scale a été créé.
- Le fichier XML descripteur eXtreme Scale ObjectGrid et le fichier XML de stratégie de déploiement ont été créés.

Pourquoi et quand exécuter cette tâche

Cette tâche explique comment configurer un serveur eXtreme Scale avec un conteneur en utilisant un fichier XML Blueprint. Le résultat de la procédure est un ensemble de conteneur. Lorsque l'ensemble de conteneur est démarré, l'ensemble de serveur eXtreme Scale suit l'ensemble, analyse le fichier XML de serveur et démarre un serveur et un conteneur.

Un ensemble de conteneur peut être éventuellement combiné à l'application et aux plug-in eXtreme Scale lorsque des mises à jour de plug-in dynamiques ne sont pas nécessaires ou que les plug-in ne prennent pas en charge la mise à jour dynamique.

Procédure

1. Créez un fichier XML Blueprint avec l'espace de nom `objectgrid` inclut. Vous pouvez affecter le nom de votre choix au fichier. Toutefois, il doit inclure l'espace de nom Blueprint :

```
<?xml version="1.0" encoding="UTF-8"?>

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
           xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
           xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                http://www.ibm.com/schema/objectgrid/objectgrid.xsd">
...
</blueprint>
```

2. Ajoutez la définition XML du serveur eXtreme Scale avec les propriétés de serveur appropriées. Voir le fichier XML descripteur Spring pour plus d'informations sur toutes les propriétés de configuration disponibles. Voir l'exemple suivant de définition de fichier XML :

```
<objectgrid:server id="xsServer" tracespec="ObjectGridOSGi=all=enabled"
tracefile="logs/osgi/wxsserver/trace.log" jmxport="1199" listenerPort="2909">
<objectgrid:catalog host="catserver1.mycompany.com" port="2809" />
<objectgrid:catalog host="catserver2.mycompany.com" port="2809" />
</objectgrid:server>
```

3. Ajoutez la définition XML du conteneur eXtreme Scale avec la référence à la définition de serveur et les fichiers XML descripteur d'ObjectGrid et de déploiement d'ObjectGrid regroupés dans l'ensemble. Par exemple :

```
<objectgrid:container id="container"
    objectgridxml="/META-INF/objectGrid.xml"
    deploymentxml="/META-INF/objectGridDeployment.xml"
    server="xsServer" />
```

4. Stockez le fichier XML Blueprint dans l'ensemble de conteneur. Le fichier XML Blueprint doit être stocké dans le répertoire `OSGI-INF/blueprint` du conteneur Blueprint pour être trouvé.

   Pour stocker le fichier XML Blueprint dans un répertoire différent, vous devez définir l'en-tête du manifeste Bundle-Blueprint. Par exemple :

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

5. Regroupez les fichiers dans un fichier JAR d'ensemble unique. Voir l'exemple suivant de hiérarchie de répertoires d'ensemble :

   ```
   MyBundle.jar
       /META-INF/manifest.mf
       /META-INF/objectGrid.xml
       /META-INF/objectGridDeployment.xml
       /OSGI-INF/blueprint/blueprint.xml
   ```

Résultats

Un ensemble de conteneur eXtreme Scale est maintenant créé et peut être installé dans Eclipse Equinox. Lorsque l'ensemble de conteneur est démarré, l'environnement d'exécution du serveur eXtreme Scale dans l'ensemble de serveur eXtreme Scale démarre automatiquement le serveur eXtreme Scale de singleton en utilisant les paramètres définis dans l'ensemble et démarre un serveur de conteneur. L'ensemble peut être arrêté et démarré, ce qui arrête et redémarre le conteneur. Le serveur est un singleton et ne s'arrête pas lorsque l'ensemble est démarré pour la première fois.

Configuration des serveurs avec l'administration de configuration OSGI

Java

Vous pouvez utiliser le service d'administration de configuration (config admin) OSGi pour configurer les serveurs de conteneur WebSphere eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Pour configurer un serveur, le PID (Persistent Identifier) ManagedService, com.ibm.websphere.xs.server, est défini pour faire référence au fichier des propriétés du serveur ObjectGrid dans le système de fichiers. Pour configurer un conteneur, le PID ManagedServiceFactory, com.ibm.websphere.xs.container, est défini pour faire référence au fichier XML de déploiement ObjectGrid et le fichier XML de stratégie de déploiement dans le système de fichiers.

Lorsque les deux PID sont définis dans le service config admin, le service de serveur eXtreme Scale initialise automatiquement le serveur et démarre le conteneur avec les fichiers de configuration spécifiés. Les PID config admin persistent dans le répertoire de configuration OSGi. Si la configuration n'est pas effacée, les paramètres sont conservés entre les redémarrages d'infrastructure.

Il existe plusieurs utilitaires tiers pour définir les propriétés config admin. Voici des exemples d'outils pris en charge par le produit :

- Le client de ligne de commande Luminis OSGi Configuration Admin permet d'effectuer la configuration depuis la ligne de commande.

- Apache Felix File Install permet de définir les paramètres PID config admin dans des fichiers de propriétés standard.

Pour configurer les serveurs de conteneur eXtreme Scale avec le client de ligne de commande OSGi Configuration Administration pour Luminis, procédez comme suit :

Procédure

1. Créez un PID de service géré pour le fichier de propriété de serveur ObjectGrid dans la console OSGi en exécutant les commandes suivantes :

   ```
   osgi> cm create com.ibm.websphere.xs.server
   osgi> cm put com.ibm.websphere.xs.server objectgrid.server.props /mypath/server.properties
   ```

2. Créez un PID (persistence identifier) de fabrique de service géré pour le conteneur ObjectGrid dans la console OSGi en exécutant les commandes suivantes.

   Avertissement : Utilisez le PID créé avec la commande **createf** config admin. Le PID utilisé dans l'exemple suivant n'est fourni qu'à titre d'exemple.

   ```
   osgi> cm createf com.ibm.websphere.xs.container
   PID: com.ibm.websphere.xs.container-123456789-0
   osgi> cm put com.ibm.websphere.xs.container-123456789-0 objectgridFile /mypath/objectGrid.xml
   osgi> cm put com.ibm.websphere.xs.container-123456789-0 deploymentPolicyFile /mypath/deployment.xml
   ```

Résultats

Maintenant, les serveurs de conteneur eXtreme Scale sont configurés pour démarrer dans une infrastructure OSGi Eclipse Equinox.

Que faire ensuite

Les serveurs de conteneur peuvent être également créés à l'aide d'un programme en utilisant l'ServerFactory et des activateurs de regroupement OSGi. Pour plus d'informations sur l'utilisation de l'API ServerFactory, voir la documentation de l'API.

Configuration d'une grille de données d'entreprise pour la mise en cache dynamique en utilisant un profil Liberty

Un serveur Profil Liberty peut héberger une grille de données qui met en mémoire cache les données des applications dont la mémoire cache dynamique est activée.

Avant de commencer

- Installez le Profil Liberty. Pour plus d'informations, voir Installation de Profil Liberty.
- Créez une application qui utilise le cache dynamique. Pour plus d'informations, voir «Configuration de l'instance de cache dynamique par défaut (baseCache)», à la page 401.

Pourquoi et quand exécuter cette tâche

Le Profil Liberty héberge la grille de données qui prend en charge les applications à cache dynamique. Cela implique que l'application s'exécute sur une installation traditionnelle de WebSphere Application Server. Pour que ces applications soient mises en cache par l'environnement d'exécution eXtreme Scale, vous devez configurer WebSphere Application Server pour utiliser les propriétés de service de domaine de catalogue et de serveur que vous définissez dans le Profil Liberty.

Procédure

1. Activez la fonction de cache dynamique WebSphere eXtreme Scale.
   a. Ajoutez la fonction de cache dynamique au fichier Profil Liberty `server.xml`. Par exemple, le fichier `server.xml` contient la section de code suivante :

```
<featureManager>
<feature>eXtremeScale.server-1.1</feature>
<feature>eXtremeScale.dynacacheGrid-1.1</feature>
</featureManager>
```

2. Facultatif : Définissez les propriétés dans l'élément `xsDynacacheGrid` dans le fichier `server.xml`. Vous changez les propriétés suivantes. Cependant, il est recommandé d'accepter les valeurs par défaut.

   globalIndexDisabled
   L'invalidation de l'index global améliore l'efficacité d'invalidation dans un grand environnement partitionné comportant, par exemple, plus de 40 partitions. Pour plus d'informations, voir «Invalidation des données», à la page 40. Valeur par défaut : `false`

   objectGridName
   Chaîne qui définit le nom de la grille de données. Valeur par défaut : `DYNACACHE_REMOTE`

   ojectGridTxTimeout
   Indique le délai d'exécution maximal autorisé pour une transaction. Si une transaction n'est pas terminée dans ce laps de temps, la transaction est marquée pour annulation et une exception `TransactionTimeoutException` est générée. Valeur par défaut : `30` (secondes)

   backingMapLockStrategy
   Indique si le gestionnaire de verrouillage interne est utilisé chaque fois qu'une transaction accède à une entrée de mappe. Spécifiez l'une des trois valeurs suivantes pour cet attribut : `OPTIMISTIC`, `PESSIMISTIC` ou `NONE`. Valeur par défaut : `PESSIMISTIC`

   backingMapCopyMode
   Indique si une opération get d'une entrée de l'instance BackingMap renvoie la valeur réelle, une copie de la valeur ou un proxy de la valeur. Si vous utilisez XDF (eXtreme data format) pour que Java et .NET puissent accéder à la même grille de données, la valeur par défaut et le mode de copie requis sont COPY_TO_BYTES. Autrement, le mode de copie COPY_ON_READ_AND_COMMIT est utilisé. Affectez à l'attribut CopyMode l'une des cinq valeurs suivantes :

      COPY_ON_READ_AND_COMMIT
      La valeur par défaut est `COPY_ON_READ_AND_COMMIT`. Spécifiez la valeur `COPY_ON_READ_AND_COMMIT` pour qu'une application ne fasse jamais référence à l'objet de valeur qui se trouve dans l'instance BackingMap. A la place, l'application utilise toujours une copie de la valeur qui se trouve dans l'instance BackingMap. (Facultatif).

      COPY_ON_READ
      Spécifiez la valeur `COPY_ON_READ` pour améliorer les performances par rapport à la valeur `COPY_ON_READ_AND_COMMIT` en éliminant la copie créée lors de la validation d'une transaction. Pour conserver l'intégrité des données de la mappe de sauvegarde, l'application s'engage à supprimer toutes les références à une entrée une fois que la transaction est validée. Si vous définissez cette valeur, une méthode ObjectMap.get renvoie une copie de la valeur au lieu

d'une référence à la valeur, ce qui garantit que les modifications apportées par l'application à la valeur n'affectent pas l'élément BackingMap tant que la transaction n'est pas validée.

COPY_ON_WRITE
Spécifiez la valeur COPY_ON_WRITE pour améliorer les performances par rapport à la valeur COPY_ON_READ_AND_COMMIT en éliminant la copie créée lors du premier appel de la méthode ObjectMap.get par une transaction pour une clé donnée. A la place, la méthode ObjectMap.get renvoie un proxy de la valeur au lieu d'une référence directe à l'objet de valeur. Le proxy garantit qu'aucune copie de la valeur n'est effectuée tant que l'application n'appelle pas de méthode set sur l'interface de la valeur.

NO_COPY
Spécifiez la valeur NO_COPY pour permettre à une application de ne jamais modifier d'objet de valeur obtenu à l'aide d'une méthode ObjectMap.get en échange de meilleures performances. Spécifiez la valeur NO_COPY pour les mappes associées aux entités de l'API EntityManager.

COPY_TO_BYTES
Spécifiez la valeur COPY_TO_BYTES pour améliorer l'encombrement mémoire des objets de type complexe et les performances lorsque la copie d'un objet s'appuie sur la sérialisation. Si un objet ne peut pas être cloné ou qu'aucune interface ObjectTransformer personnalisée avec une méthode copyValue efficace n'est fournie, le mécanisme de copie par défaut doit sérialiser et inflatez l'objet pour effectuer une copie. Avec le paramètre COPY_TO_BYTES, l'inflation n'est effectuée que lors d'une opération de lecture et la sérialisation, lors d'une validation.

Valeur par défaut : COPY_ON_READ_AND_COMMIT

backingMapNearCacheEnabled
Définissez la valeur true pour activer le cache local client. Pour utiliser un cache local, vous devez affecter à l'attribut **lockStrategy** la valeur NONE ou OPTIMISTIC. Valeur par défaut : false

mapSetNumberOfPartitions
Indique le nombre de partitions de l'élément mapSet. Valeur par défaut : 47

mapSetMinSyncReplicas
Indique le nombre minimal de fragments réplique synchrones de chaque partition du mapSet. Les fragments ne sont pas placés tant que le domaine ne peut pas prendre en charge le nombre minimal de fragments réplique synchrones. Pour pouvoir prendre en charge la valeur minSyncReplicas, vous devez augmenter le nombre de serveurs d'une unité par rapport à la valeur **minSyncReplicas**. Si le nombre de répliques synchrones tombe en dessous de la valeur **minSyncReplicas**, les transactions d'écriture ne sont plus autorisées pour la partition. Valeur par défaut : 0

mapSetMaxSyncReplicas
Indique le nombre maximal de fragments réplique synchrones de chaque partition du mapSet. Aucune autre réplique synchrone n'est placée pour une partition une fois qu'un domaine a atteint ce nombre de fragments réplique synchrones pour cette partition spécifique. L'ajout de serveurs de

conteneur qui peuvent prendre en charge cet ObjectGrid peut augmenter le nombre de répliques synchrones si la valeur **maxSyncReplicas** n'a pas été déjà atteinte. Valeur par défaut : `0`

mapSetNumInitialContainers
Indique le nombre de serveurs de conteneur requis pour le placement initial des fragments de cet élément mapSet. Cet attribut peut permettre d'économiser la bande passante des processus et du réseau lorsque vous mettez une grille de données en ligne à partir d'un démarrage à froid. Valeur par défaut : `1`

mapSetDevelopmentMode
Avec cet attribut, vous pouvez influencer le positionnement d'un fragment par rapport à ses fragments homologues. Si l'attribut developmentMode a la valeur `false`, deux fragments d'une même partition ne peuvent pas être placés sur un même ordinateur. Si l'attribut developmentMode a la valeur `true`, les fragments d'une même partition peuvent être placés sur une même machine. Dans les deux cas, deux fragments d'une même partition ne sont jamais placés dans le même serveur de conteneur. Valeur par défaut : `false`

mapSetReplicaReadEnabled
Si cet attribut est défini sur `true`, les demandes de lecture sont réparties entre le fragment primaire d'une partition et ses fragments réplique. Si l'attribut replicaReadEnabled est défini sur `false`, les demandes de lecture ne sont acheminées que vers le fragment primaire. Valeur par défaut : `false`

.

3. Configurez WebSphere Application Server pour pointer vers Profil Liberty.

   Vous pouvez connecter les applications Web à cache dynamique WebSphere eXtreme Scale à un domaine de services de catalogue exécuté dans une autre cellule WebSphere Application Server ou comme processus autonome. Comme les serveurs de catalogue configuré à distance ne démarrent pas automatiquement dans la cellule, vous devez démarrer manuellement les serveurs de catalogue configurés à distance.

   Lorsque vous configurez un domaine de services de catalogue distant, le nom de domaine doit correspondre au nom de domaine que vous avez défini lorsque vous démarrez les serveurs de catalogue distants. Le nom de domaine de services de catalogue par défaut des serveurs de catalogue autonome est `DefaultDomain`. Définissez un nom de domaine de services de catalogue avec la commande **startOgServer** ou **startXsServer** et le paramètre **-domain**, un fichier de propriétés de serveur ou avec l'API de serveur embarqué. Vous devez démarrer chaque processus de serveur de catalogue distant dans le domaine distant avec le même nom de domaine. Pour plus d'informations sur le démarrage des serveurs de catalogue, voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494.

Configuration des clients eXtreme Scale REST dans le profil Liberty

En tant qu'administrateur, vous pouvez définir plusieurs configurations de noeuds finaux de domaine client pouvant être utilisées par les applications client, telles que la passerelle WebSphere eXtreme Scale REST.

Pourquoi et quand exécuter cette tâche

En tant que développeur, vous pouvez écrire des applications sans connaître les serveurs de grille auxquels les applications vont se connecter. Par exemple, l'un de vos clients peut disposer de serveurs distincts pour les tests et la production. Un administrateur peut configurer l'environnement auquel l'application fait référence sans modifier le code.

Procédure

1. Définissez le domaine client dans le fichier `server.xml`. Dans l'exemple suivant, test est le domaine client par défaut. Si vous ne définissez pas l'attribut `remoteDomain`, le domaine par défaut est utilisé.

   ```
   <xsClientDomain default="test">
       <endpointConfig> test ; testHost1:2809,testHost2:2809 ; /home/testuser/client_security.props </endpointConfig>
       <endpointConfig> dev; localhost:2809,testHost2:2809 </endpointConfig>
   <!-- note that client security props file is optional -->
       <endpointConfig> production; prodHost1:2809,prodHost2:2809,prodHost3:2809 ;
   /home/testuser/client_security.props </endpointConfig>
   </xsClientDomain>
   ```

   L'élément `endpointConfig` permet de définir les données de noeud final d'une grille de données. Cet élément a la syntaxe suivante :

   nom noeud final; *liste de paires nom d'hôte:port séparées par une virgule* ; *chemin d'accès au fichier des propriétés de sécurité du client*

   Ce fichier est facultatif. Si vous n'en définissez pas, le client se connecte en supposant que la sécurité de grille de données est désactivée.
2. Configurez la passerelle REST maintenant que vous avez définie la configuration précédente. Par exemple :

   ```
   <xsREST remoteDomain="dev" />
   ```
3. Facultatif : En tant que développeur, vous pouvez accéder aux domaines client configurés. Par exemple :

   ```
   CatalogDomainManager catalogDomainManager = objectGridManager.getCatalogDomainManager();
   CatalogDomainInfo catalogDomainInfo = catalogDomainManager.getDomainInfo("dev");
   if (catalogDomainInfo == null) {
   catalogDomainInfo = catalogDomainManager.getDefaultDomainInfo();
   }
   ClientClusterContext ccc = objectGridManager.connect(catalogDomainInfo.getClientCatalogServerEndpoints(),
   catalogDomainInfo.getClientSecurityConfiguration(), null);
   ...
   ```

Chapitre 7. Administration

L'administration et l'exploitation de l'environnement du produit consiste à démarrer et arrêter des serveurs, gérer la disponibilité de la grille de données et récupérer à partir de scénarios de défaillance de centre de données. Une fois que vous avez configuré les serveurs de catalogue et les serveurs de conteneur, vous pouvez démarrer et arrêter les serveurs à l'aide de diverses méthodes. La méthode que vous utilisez pour démarrer et arrêter les serveurs varie selon que vous utilisez une topologie intégrée, une topologie autonome ou une topologie exécutée dans WebSphere Application Server.

Démarrage et arrêt des serveurs sécurisés

Vous pouvez démarrer et arrêter les serveurs de catalogue autonome et de conteneur avec les scripts ou l'API de serveur embarqué.

Avant de commencer

Si vous démarrez ou arrêtez les serveurs dans un environnement autonome qui utilise un fournisseur de sécurité client externe, vous devez définir la variable d'environnement *CLIENT_AUTH_LIB* avant d'exécuter les script de démarrage et d'arrêt. Pour plus d'informations sur la définition de cette variable, voir «Démarrage des serveurs sécurisés dans un environnement autonome», à la page 650.

8.6+ Pourquoi et quand exécuter cette tâche

Les scripts de démarrage et d'arrêt que vous utilisez pour l'environnement dépendent du type de mécanisme de transport que vous utilisez :

- Si vous disposez d'une grille de données d'entreprise qui utilise le transport IBM eXtremeIO (XIO), utilisez le script **startXsServer** **stopXsServer**.
- Si vous disposez d'applications Java uniquement utilisant le transport ORB (Object Request Broker), utilisez le script **startOgServer** ou **stopOgServer**.

Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Démarrage des serveurs autonomes (XIO)

Lorsque vous exécutez une configuration autonome, l'environnement se compose de serveurs de catalogue, de serveurs de conteneur et de processus client. Les serveurs WebSphere eXtreme Scale peuvent être également intégrés à des applications Java existantes en utilisant l'API Embedded Server. Vous devez manuellement configurer et démarrer ces processus.

Avant de commencer

Vous pouvez démarrer des serveurs WebSphere eXtreme Scale dans un environnement dans lequel WebSphere Application Server n'est pas installé. Si vous utilisez WebSphere Application Server, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 306.

Démarrage d'un service de catalogue autonome qui utilise le transport eXtremeIO IBM (XIO)

Vous devez démarrer le service de catalogue manuellement si vous utilisez un environnement WebSphere eXtreme Scale réparti qui n'est pas exécuté dans WebSphere Application Server.

Avant de commencer

- Si vous utilisez WebSphere Application Server, le service de catalogue démarre automatiquement dans les processus existants. Pour plus d'informations, voir «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 508.

Pourquoi et quand exécuter cette tâche

Démarrage du service de catalogue avec le script **startXsServer**. Lorsque vous appelez la commande de démarrage, utilisez le script **startXsServer.sh** sur les plateformes UNIX ou **startXsServer.bat** sous Windows.

Le service de catalogue peut être exécuté dans un seul processus ou il peut inclure plusieurs serveurs de catalogue afin de constituer un domaine de service de catalogue. Un domaine de service de catalogue est obligatoire dans un environnement de production pour la haute disponibilité. Pour plus d'informations, voir Service de catalogue à haute disponibilité. Vous pouvez également spécifier des paramètres supplémentaires pour le script pour lier le transport à un hôte et un port spécifiques, spécifier le domaine ou activer la sécurité.

Procédure

- **Démarrez un processus de serveur de catalogue.**

  Pour démarrer un serveur de catalogues, entrez les commandes suivantes à partir de la ligne de commande :

  1. Accédez au répertoire `bin`.

     ```
     cd rép_base_wxs/bin
     ```

  2. Exécutez la commande **startXsServer**.

     ```
     startXsServer.bat|sh catalogServer
     ```

  Pour la liste de tous les paramètres de ligne de commande disponibles, voir «Script **startXsServer** (XIO)», à la page 485. N'utilisez pas une seule machine virtuelle Java (JVM) pour exécuter le service de catalogue dans un environnement de production. Si le service de catalogue échoue, aucun nouveau client ne peut être acheminé vers l'instance eXtreme Scale déployée et aucune nouvelle instance ObjectGrid ne peut être ajoutée au domaine. Pour ces motifs, vous devez démarrer un ensemble de machines virtuelles Java pour pouvoir exécuter un domaine de service de catalogue.

- **Démarrez un domaine de service de catalogue constitué de plusieurs noeud finals.**

Pour démarrer un ensemble de serveurs afin d'exécuter un service de catalogue, vous devez utiliser l'option **-catalogServiceEndPoints** sur le script startXsServer. Cet argument accepte une liste de noeuds finals de services de catalogue dans le format *serverName*:*hostName*:*clientPort*:*peerPort*.L'exemple suivant indique comment démarrer la première des trois machines virtuelles Java pour héberger un service de catalogue :

1. Accédez au répertoire bin.

   ```
   cd rép_base_wxs/bin
   ```

2. Exécutez la commande **startXsServer**.

```
startXsServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

   Dans ce cas, le serveur cs1 de l'hôte MyServer1.company.com est démarré. Ce nom de serveur est le premier argument transmis au script. Lors de l'initialisation du serveur cs1, les paramètres **-catalogServiceEndpoints** sont examinés pour déterminer les ports alloués pour ce processus. La liste est également utilisée pour permettre au serveur cs1 d'accepter les connexions des autres serveurs : cs2 et cs3.

3. Pour démarrer les serveurs de catalogues restants de la liste, transmettez les arguments ci-après au script **startXsServer**. Démarrage du serveur cs2 sur l'hôte MyServer2.company.com.

```
startXsServer.bat|sh cs2 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

   Démarrage du serveur cs3 sur MyServer3.company.com :

```
startXsServer.bat|sh cs3 -catalogServiceEndPoints
cs3:MyServer3.company.com:6601:6602,cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602
```

   L'ordre de la liste pour le paramètre **-catalogServiceEndpoints** peut être différent pour les divers serveurs de catalogue, mais les serveurs contenus dans la liste doivent être identiques. N'insérez aucun espace dans la liste.

   Important : Démarrez au moins deux serveurs de catalogue en même temps.

   Vous devez démarrer les serveurs de catalogue qui se trouvent dans une grille de données en parallèle, car chaque serveur s'interrompt pour attendre que les autres serveurs de catalogue rejoignent le groupe central. Un serveur de catalogue qui est configuré pour une grille de données ne démarre pas tant qu'il n'a pas identifié les autres membres du groupe. Le serveur de catalogues arrive à expiration si aucun autre serveur ne devient disponible.

- **Liez le transport à un hôte et un port spécifiques.**

  En dehors des ports définis dans l'argument **catalogServiceEndpoints**, chaque service de catalogue utilise également un ORB (Object Request Broker) pour accepter les connexions des clients et des conteneurs. Par défaut, l'ORB écoute sur le port 2809 du système hôte local. Si vous souhaitez associer l'ORB à un hôte et un port spécifiques sur la machine virtuelle Java d'un service de catalogue, utilisez les arguments **-listenerHost** et **-listenerPort**. L'exemple suivant montre comment démarrer un serveur de catalogue JVM avec son transport associé au port 7000 sur MyServer1.company.com:

  ```
  startXsServer.sh catalogServer -listenerHost MyServer1.company.com
  -listenerPort 7000
  ```

  Chacun des conteneurs et clients eXtreme Scale doit être fourni avec des données de point de contact d'ORB de service de catalogue. Les clients n'ont

besoin que d'un sous-ensemble de ces données, mais vous devez utiliser au moins deux points de contact pour la haute disponibilité.

- Facultatif : **Nom du domaine de service de catalogue**

  Un nom de domaine de service de catalogue n'est pas requis lors du démarrage d'un service de catalogue. Toutefois, si vous utilisez la réplication multimaître ou plusieurs domaines de service de catalogue dans un même ensemble de processus, vous devez définir un nom de domaine de service de catalogue unique. Le nom de domaine par défaut est `DefaultDomain`. Pour affecter un nom à votre domaine, utilisez l'option **`-domain`**. L'exemple ci-après montre comment démarrer la machine virtuelle Java d'un service de catalogue avec le nom de domaine `myDomain`.

  ```
  startXsServer.sh catalogServer -domain myDomain
  ```

  Pour plus d'informations sur la configuration de la réplication multimaître, voir «Configuration de plusieurs topologies de centres de données», à la page 340.

- **Démarrez un service de catalogue sécurisé.** Pour plus d'informations, voir «Démarrage des serveurs sécurisés dans un environnement autonome», à la page 650.

- **Démarrez le service de catalogue à l'aide d'un programme.**

  Tout paramètre JVM qui est marqué par la méthode CatalogServerProperties.setCatalogServer peut héberger le service de catalogue pour eXtreme Scale. Cette méthode indique à l'environnement d'exécution du serveur eXtreme Scale d'instancier le service de catalogue lorsque le serveur est démarré. Le code qui suit montre comment instancier le serveur de catalogue eXtreme Scale :

  ```
  CatalogServerProperties catalogServerProperties =
   ServerFactory.getCatalogProperties();
  catalogServerProperties.setCatalogServer(true);

  //La méthode getInstance() démarre le service de catalogue.
  Server server = ServerFactory.getInstance();
  ```

  Pour plus d'informations sur le démarrage des serveurs à l'aide d'un programme, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.

Démarrage des serveurs de conteneur qui utilisent le transport IBM eXtremeIO (XIO)

Vous pouvez démarrer les serveurs de conteneurs depuis la ligne de commande en utilisant une topologie de déploiement ou un fichier `server.properties`.

Pourquoi et quand exécuter cette tâche

Pour démarrer un processus de conteneur, vous avez besoin d'un fichier ObjectGrid XML. Ce fichier spécifie quels serveurs eXtreme Scale sont hébergés par le conteneur. Vérifiez que le conteneur est équipé pour héberger la grille de données dans le fichier XML que vous lui transmettez. Toutes les classes que ces grilles de données requièrent doivent se trouver dans le chemin d'accès aux classes du conteneur. Pour plus d'informations sur le fichier XMLObjectGrid, voir Fichier XML du descripteur d'ObjectGrid.

Procédure

- **Démarrez le conteneur de serveur depuis la ligne de commande.**

  1. Depuis la ligne de commande, accédez au répertoire `bin` :

     ```
     cd racine_install_wxs/bin
     ```

2. Exécutez la commande suivante :

```
startXsServer.bat|sh c0 -objectGridFile ../xml/companyGrid.xml
-catalogServiceEndPoints MyServer1.company.com:2809
```

Important : Sur le conteneur, l'option **-catalogServiceEndPoints** est utilisée pour faire référence à l'hôte et au port d'écoute sur le service de catalogue. Le service de catalogue utilise les options **-listenerHost** et **-listenerPort** pour spécifier l'hôte et le port de la liaison d'écoute ou accepte la liaison par défaut. Lorsque vous démarrez un conteneur, utilisez l'option **-catalogServiceEndPoints** pour référencer les valeurs transmises aux options **-listenerHost** et **-listenerPort** sur le service de catalogue. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées quand le service de catalogue est démarré, le transport XIO se lie au port 2809 sur le système hôte local du service de catalogue. N'utilisez pas l'option **-catalogServiceEndPoints** pour référencer les hôtes et les ports transmis à l'option **-catalogServiceEndPoints** sur le service de catalogue. Sur le service de catalogue, l'option **-catalogServiceEndPoints** est utilisée pour spécifier les ports nécessaires pour une configuration de serveur statique.
Ce processus est identifié par c0, le premier argument transmis au script. Utilisez le fichier `companyGrid.xml` pour démarrer le conteneur. Si le transport XIO du serveur de catalogue est exécuté sur un hôte différent du conteneur ou qu'il utilise un autre port que le port par défaut, vous devez utiliser l'argument **-catalogServiceEndPoints** pour vous connecter au transport XIO. Pour cet exemple, partez du principe qu'un unique service de catalogue est exécuté sur le port `2809` sur `MyServer1.company.com`

- **Démarrez le conteneur à l'aide d'une règle de déploiement.**

  Sans être nécessaire, une stratégie est recommandée pendant le démarrage du conteneur. La règle de déploiement est utilisée pour configurer le partitionnement et la réplication pour eXtreme Scale. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement. Comme l'exemple précédent ne fournit pas de fichier de règle de déploiement, l'exemple reçoit toutes les valeurs par défaut en ce qui concerne la réplication, le partitionnement et le positionnement. Donc, les mappes dans le CompanyGrid se trouvent dans un mapSet. Le mapSet n'est ni partitionné ni répliqué. Pour plus d'informations sur les fichiers de règle de déploiement, voir Fichier XML du descripteur de la règle de déploiement. L'exemple suivant utilise le fichier `companyGridDpReplication.xml` pour démarrer le serveur de conteneur c0 :

  1. A partir de la ligne debvommande, accédez au répertoire bin :

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande suivante :

     ```
     startXsServer.bat|sh c0 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Remarque : Si vous disposez de classes Java stockées dans un répertoire spécifique ou que vous utilisez un chargeur ou un agent, au lieu de modifier le script startXsServer, vous pouvez lancer le serveur avec des arguments, comme suit : `-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar` . Dans le fichier `companyGridDpReplication.xml`, un seul groupe de mappes contient toutes les mappes. Ce mapSet est divisé en 10 partitions. Chaque partition a une réplique synchrone et aucune réplique asynchrone. Tout conteneur utilisant la règle de déploiement `companyGridDpReplication.xml` combinée au fichier XML ObjectGrid `companyGrid.xml` est également capable d'héberger des fragments de CompanyGrid. Démarrez une autre machine virtuelle Java de conteneur, c1 :

1. A partir de la ligne debvommande, accédez au répertoire bin :
   ```
   cd racine_install_wxs/bin
   ```
2. Exécutez la commande suivante :
   ```
   startXsServer.sh c1 -objectGridFile ../xml/companyGrid.xml
   -deploymentPolicyFile ../xml/companyGridDpReplication.xml
   -catalogServiceEndPoints MyServer1.company.com:2809
   ```

Chaque règle de déploiement contient au moins un élément objectgridDeployment. Quand un conteneur est démarré, il publie sa règle de déploiement sur le service de catalogue. Le service de catalogue examine chaque élément objectgridDeployment. Si l'attribut objectgridName correspond à l'attribut objectgridName d'un élément objectgridDeployment précédemment reçu, l'élément objectgridDeployment le plus récent est ignoré. Le premier élément objectgridDeployment reçu pour un attribut objectgridName spécifique est utilisé comme élément maître. Par exemple, partons du principe que la machine virtuelle Java c2 utilise une règle de déploiement qui divise le mapSet en nombre différent de partitions :

companyGridDpReplicationModified.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd"
    xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="CompanyGrid">
        <mapSet name="mapSet1" numberOfPartitions="5"
            minSyncReplicas="1" maxSyncReplicas="1"
            maxAsyncReplicas="0">
            <map ref="Customer" />
            <map ref="Item" />
            <map ref="OrderLine" />
            <map ref="Order" />
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Vous pouvez maintenant démarrer une troisième machine virtuelle Java, c2 :

1. A partir de la ligne debvommande, accédez au répertoire bin :
   ```
   cd racine_install_wxs/bin
   ```
2. Exécutez la commande suivante :
   ```
   startXsServer.sh c2 -objectGridFile ../xml/companyGrid.xml
   -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
   -catalogServiceEndPoints MyServer1.company.com:2809
   ```

Le conteneur sur la machine virtuelle Java c2 est démarré avec une règle de déploiement qui spécifie 5 partitions pour le mapSet1. Cependant, le service de catalogue contient déjà la copie maître de l'objectgridDeployment pour le CompanyGrid. Quand la machine virtuelle Java c0 a été démarrée, elle a spécifié que 10 partitions existent pour ce mapSet. Comme il s'agit du premier conteneur à démarrer et publier sa règle de déploiement, cette dernière devient la stratégie maître. En conséquence, toute valeur d'attribut objectgridDeployment égale à `CompanyGrid` dans une règle de déploiement suivante est ignorée.

- **Démarrez un conteneur à l'aide d'un fichier de propriétés de serveur.**

  Vous pouvez utiliser un fichier de propriétés de serveur pour configurer la fonction de trace et la sécurité sur un conteneur. Exécutez les commandes suivantes pour démarrer un conteneur c3 avec un fichier de propriétés de serveur.

1. A partir de la ligne debvommande, accédez au répertoire bin :
   ```
   cd racine_install_wxs/bin
   ```
2. Exécutez la commande suivante :
   ```
   startXsServer.sh c3 -objectGridFile ../xml/companyGrid.xml
   -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
   -catalogServiceEndPoints MyServer1.company.com:2809
   -serverProps ../serverProps/server.properties
   ```

   Voici un exemple de fichier `server.properties` :

   ```
   server.properties
   workingDirectory=
   traceSpec=*=all=disabled
   systemStreamToFileEnabled=true
   enableMBeans=true
   memoryThresholdPercentage=50
   ```

   Il s'agit d'un fichier de propriétés de serveur de base dans lequel la sécurité n'est pas activée. Pour plus d'informations concernant le fichier `server.properties`, voir Fichier de propriétés du serveur.

- **Démarrez un serveur de conteneur à l'aide d'un programme.**

  Pour plus d'informations sur le démarrage des serveurs de conteneur à l'aide d'un programme, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.

Script `startXsServer` (XIO)

Le script **`startXsServer`** démarre les serveurs de conteneur et de catalogue en utilisant le mécanisme de transport eXtremeIO IBM (XIO). Vous devez utiliser **`startXsServer`** lorsque vous voulez utiliser une grille de données d'entreprise. Vous pouvez utiliser divers paramètres lorsque vous démarrez vos serveurs pour activer la trace, spécifiez des numéros de port, etc.

Rôle

Vous pouvez utiliser le script **`startXsServer`** pour démarrer les serveurs.

Placement

Le script **`startXsServer`** se trouve dans le répertoire `bin` du répertoire root, par exemple :

```
cd racine_install_wxs/bin
```

Remarque : Si vous disposez de classes Java stockées dans un répertoire spécifique ou que vous utilisez un chargeur ou un agent, au lieu de modifier le script **`startXsServer`**, vous pouvez lancer le serveur avec des arguments, comme suit :
`-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`

.

Syntaxe des serveurs de catalogue

Pour démarrer un serveur de catalogues :

Windows

```
startXsServer.bat <server> [options]
```

UNIX

```
startXsServer.sh <server>[options]
```

Pour démarrer un serveur de catalogues configuré par défaut, utilisez les commandes suivantes :

Windows

```
startXsServer.bat catalogServer
```

UNIX

```
startXsServer.sh catalogServer
```

Options de démarrage des serveurs de catalogue

Les paramètres suivants sont tous facultatifs.

Paramètres de démarrage d'un serveur de catalogues :

-catalogServiceEndPoints <serverName:hostName:clientPort:peerPort>
Indique une liste de serveurs de catalogue à lier ensemble dans un domaine de services de catalogue. Chaque attribut est défini comme suit :

serverName
Définit le nom du serveur de catalogue.

hostName
Spécifie le nom d'hôte de l'ordinateur sur lequel le serveur est lancé.

clientPort
Spécifie le port utilisé pour la communication de service de catalogue homologue.

peerPort
Cette value est identique à haManagerPort. Spécifie le port utilisé pour la communication de service de catalogue homologue.

L'exemple suivant démarre le serveur de catalogue, cs1, qui se trouve dans le même domaine de services de catalogue que les serveurs cs2 et cs3 :

```
startXsServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

Si vous démarrez des serveurs de catalogue supplémentaires, ils doivent inclure les mêmes serveurs dans l'argument **-catalogServiceEndPoints**. L'ordre de la liste peut être différent, mais les serveurs contenus dans la liste doivent être les mêmes pour chaque serveur de catalogue. N'insérez aucun espace dans la liste.

-clusterSecurityFile <fichier_xml_sécurité_cluster>
Indique le fichier `objectGridSecurity.xml` sur le disque dur, qui décrit les propriétés de sécurité communes à tous les serveurs (y compris les serveurs de catalogue et les serveurs de conteneur). L'un des exemples de propriété est la configuration de l'authentificateur qui représente le registre d'utilisateurs et le mécanisme d'authentification.

Exemple : `/opt/xs/ogsecurity.xml`

-clusterSecurityUrl <URL du xml de la sécurité du cluster>
Indique le fichier `objectGridSecurity.xml` comme URL du fichier sur le disque dur ou sur le réseau, qui décrit les propriétés de sécurité communes à tous les serveurs, y compris les serveurs de catalogue et les serveurs de conteneur. L'un

des exemples de propriété est la configuration de l'authentificateur qui représente le registre d'utilisateurs et le mécanisme d'authentification.

Exemple :`file:///opt/xs/ogsecurity.xml`

-domain <nom de domaine>
Indique le nom du domaine de services de catalogue du serveur de catalogue. Le domaine de services de catalogue crée un groupe de serveurs de catalogue à haute disponibilité. Chaque serveur de catalogue pour un seul domaine doit spécifier la même valeur pour le paramètre **-domain**.

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

-haManagerPort <port>
Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, un port libre est choisi. Cette propriété est ignorée dans les environnements WebSphere Application Server.

-JMXServicePort <port>
Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)

Valeur par défaut : `1099` pour les serveurs de catalogue

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel le transport ORB (Object Request Broker) ou eXtremeIO (XIO) se lie pour les communications. La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que le mécanisme de transport dans la machine JVM connaisse l'adresse IP de liaison. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

-quorum true|false

Active le quorum pour le service de catalogue. Le quorum sert à garantir qu'une majorité du domaine de service de catalogue est disponible avant le déplacement des partitions sur les serveurs de conteneur disponibles. Pour activer le quorum, spécifiez la valeur `true` ou `enabled`. La valeur par défaut est `disabled`. Cette valeur s'applique au service de catalogue uniquement. Pour plus d'informations, voir Quorums de serveurs de catalogue.

-script <fichier script>
Indique l'emplacement d'un script personnalisé pour les commandes que vous spécifiez pour démarrer les serveurs de catalogue ou les conteneurs, puis définir des paramètres ou effectuer des modifications en fonction des besoins.

-serverProps <fichier de propriétés du serveur>
Indique le fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques au serveur. Le nom de fichier spécifié pour cette propriété correspond simplement à un chemin classique, tel que `c:/tmp/og/catalogserver.props`.

-timeout <secondes>
Indique un nombre de secondes avant que le démarrage du serveur n'arrive à expiration.

-traceFile <fichier de trace>

Indique le nom du fichier dans lequel les informations de trace seront consignées. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Exemple :** `../logs/c4Trace.log`

-traceSpec <spécification de la trace>

Active la trace et la chaîne de spécification de trace du serveur de conteneur. La trace est désactivée par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. Exemples :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

8.6+ **-transport <transport type>**

Indique le type de transport à utiliser pour tous les serveurs dans le domaine de service de catalogue Vous pouvez définir la valeur `XIO` ou `ORB`.

Le script **startXsServer** définit le type de transport `XIO` par défaut.

Si vous avez défini le paramètre **-transport** et la propriété de serveur **transport** sur un serveur de catalogue, la valeur du paramètre **-transport** est utilisée.

Syntaxe des serveurs de conteneur Windows

```
startXsServer.bat <server> -objectgridFile <xml file>
-deploymentPolicyFile <fichier xml> [options]
```

Windows

```
startXsServer.bat <server> -objectgridUrl <xml URL>
-deploymentPolicyUrl <URL du xml> [options]
```

UNIX

```
startXsServer.sh <server> -objectgridFile <xml file>
-deploymentPolicyFile <fichier xml> [options]
```

UNIX

```
 startXsServer.sh <server> -objectgridUrl <xml URL>
-deploymentPolicyUrl <URL du xml> [options]
```

Options des serveurs de conteneur

-catalogServiceEndPoints<nomHôte:port,nomHôte:port>
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port dans le service de catalogue.

Valeur par défaut : localhost:2809

-deploymentPolicyFile <fichier xml de la règle de déploiement>
Indique le chemin d'accès au fichier de la stratégie de déploiement sur le disque dur. La stratégie de déploiement est utilisée pour configurer le partitionnement et la réplication. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement.

Exemple : `../xml/SimpleDP.xml`

-deploymentPolicyUrl <url de la règle de déploiement>
Indique l'URL du fichier de la stratégie de déploiement sur le disque dur ou sur le réseau. La stratégie de déploiement est utilisée pour configurer le partitionnement et la réplication. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement.

Exemple : `file://xml/SimpleDP.xml`

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

-JMXServicePort <port>

Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)**Valeur par défaut :** `1099`

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel le transport ORB (Object Request Broker) ou eXtremeIO (XIO) se lie pour les communications. La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que le mécanisme de transport dans la machine JVM connaisse l'adresse IP de liaison. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.**Valeur par défaut :** 2809

-objectgridFile <Fichier XML du descripteur d'ObjectGrid>
Indique le chemin d'accès au fichier du descripteur d'ObjectGrid. Le fichier XML ObjectGrid spécifie les serveurs eXtreme Scale hébergés par le conteneur.

-objectgridUrl <URL du descripteur d'ObjectGrid>
Spécifie une URL pour le fichier descripteur ObjectGrid. Le fichier XML ObjectGrid spécifie les serveurs eXtreme Scale hébergés par le conteneur.

-script <fichier script>
Indique l'emplacement d'un script personnalisé pour les commandes que vous spécifiez pour démarrer les serveurs de catalogue ou les conteneurs, puis définir des paramètres ou effectuer des modifications en fonction des besoins.

-serverProps <fichier de propriétés du serveur>
Indique le chemin d'accès au fichier de propriétés du serveur.

Exemple :../security/server.props

-timeout <secondes>
Indique un nombre de secondes avant que le démarrage du serveur n'arrive à expiration.

-traceFile <fichier de trace>

Indique le nom du fichier dans lequel les informations de trace seront consignées. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Exemple :** ../logs/c4Trace.log

-traceSpec <spécification de la trace>

Active la trace et la chaîne de spécification de trace du serveur de conteneur. La trace est désactivée par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. Exemples :

- ObjectGrid=all=enabled
- ObjectGrid*=all=enabled

-zone <nom de zone>
Indique la zone à utiliser pour tous les conteneurs du serveur. Voir les «Routage par zone préférée», à la page 294 informations sur les zones dans *Présentation du produit* pour plus d'informations sur la configuration de zones.

Arrêt des serveurs autonomes qui utilisent le transport IBM eXtremeIO

Vous pouvez utiliser le script stopXsServer pour arrêter les processus serveur eXtreme Scale.

Pourquoi et quand exécuter cette tâche

Exécutez le script **stopXsServer** en accédant au répertoire bin :

```
cd racine_install_wxs/bin
```

Procédure

- **Arrêtez un serveur de conteneur.**

  Exécutez le script **stopXsServer** pour arrêter le serveur de conteneur. Utilisez cette commande uniquement lorsque vous arrêtez un seul serveur de conteneur. Si vous exécutez la commande d'arrêt du serveur de catalogue unique sur plusieurs serveurs de conteneur à la suite, des problèmes de performances et de désabonnement se produisent pour le placement des fragments.

  ```
  stopXsServer containerServer -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Avertissement : L'option **-catalogServiceEndPoints** doit correspondre à la valeur de l'option **-catalogServiceEndPoints** utilisée pour démarrer le conteneur. Si aucune option **-catalogServiceEndPoints** n'a été utilisée pour démarrer le conteneur, les valeurs par défaut sont vraisemblablement localhost ou le nom d'hôte et 2809 pour le port d'écoute pour la connexion au service de catalogue. Autrement, utilisez les valeurs envoyées à **-listenerHost** et **-listenerPort** dans le service de catalogue. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées quand le service de catalogue est démarré, le port d'écoute se lie au port 2809 sur l'hôte local pour le service de catalogue.

- **Arrêtez plusieurs serveurs de conteneur.**

  Pour éviter les problèmes de //désabonnement et performances pour le placement des fragments lorsque vous voulez arrêter plusieurs serveurs de conteneur en même temps, utilisez la syntaxe de commande suivante. Séparez les serveurs de conteneur avec une virgule :

  ```
  stopXsServer containerServer0,containerServer1,containerServer2
  -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Si vous souhaitez arrêter tous les conteneurs dans une zone ou un hôte, vous pouvez utiliser le paramètre **-teardown**. Pour plus d'informations, voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 508.

- **Arrêtez les serveurs de catalogue.**

  Exécutez le script **stopXsServer** pour arrêter le serveur de catalogue.

  ```
  stopXsServer.sh catalogServer -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Avertissement : Lorsque vous arrêtez un service de catalogue, utilisez l'option **-catalogServiceEndPoints** pour référencer l'hôte et le port dans le service de catalogue. Le service de catalogue utilise les options **-listenerHost** et **-listenerPort** pour spécifier l'hôte et le port pour la liaison ou accepte la liaison par défaut. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées lors du démarrage du service de catalogue, le transport XIO se lie au port 2809 sur localhost pour le service de catalogue. L'option **-catalogServiceEndPoints** pour arrêter un service de catalogue est différente de l'option que vous avez utilisée pour le démarrer.

  Démarrer un service de catalogue requiert des ports d'accès homologues et des ports d'accès clients si les ports par défaut n'ont pas été utilisés. L'arrêt d'un service de catalogue ne requiert que le port d'écoute.

- **Arrêtez le serveur de la console Web.** Pour arrêter le serveur de la console Web, exécutez le script **stopConsoleServer.bat|sh**. Ce script se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 547.

- **Activez la fonction de trace pour le processus d'arrêt du serveur.**

Si un conteneur ne parvient pas à s'arrêter, vous pouvez activer la fonction de trace pour vous aider dans le débogage du problème. Pour activer la fonction de trace au cours de l'arrêt d'un serveur, ajoutez les paramètres **-traceSpec** et **-traceFile** pour les commandes d'arrêt. Le paramètre **-traceSpec** spécifie le type de trace et le paramètre **-traceFile** spécifie le chemin d'accès et le nom du fichier à créer et à utiliser pour les données de trace.

1. A partir de la ligne de commande, accédez au répertoire bin.
   ```
   cd racine_install_wxs/bin
   ```
2. Exécutez le script **stopXsServer** avec la fonction de trace activée.
   ```
   stopXsServer.sh c4 -catalogServiceEndPoints MyServer1.company.com:2809
    -traceFile ../logs/c4Trace.log -traceSpec ObjectGrid=all=enabled
   ```

   Une fois la trace obtenue, recherchez les erreurs relatives aux conflits de ports, aux classes manquantes, aux fichiers XML manquants ou incorrects, ou toute trace de pile. Suggestions de spécifications de trace au démarrage :
   - `ObjectGrid=all=enabled`
   - `ObjectGrid*=all=enabled`

   Pour connaître toutes les options de spécification de trace, voir «Options de trace du serveur», à la page 669.

- **Arrêtez les serveurs embarqués à l'aide d'un programme.**
  Pour plus d'informations sur cette opération, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.

Script stopXsServer (XIO)

Le script **stopXsServer** arrête les serveurs de catalogue et de conteneur.

Rôle

Utilisez le script **stopXsServer** pour arrêter un serveur. Vous devez indiquer le nom du serveur et ses noeuds finals de service de catalogue.

Placement

Le script **stopXsServer** se trouve dans le répertoire `bin` du répertoire `root`, par exemple :

```
cd racine_install_wxs/bin
```

Utilisation

Pour arrêter un serveur de catalogue ou de conteneur : Windows

```
stopXsServer.bat <server_name> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

UNIX

```
stopXsServer.sh <server_name> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

Options

-catalogServiceEndPoints <csHost:csListenerPort, csHost:csListenerPort...>
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port.

Pour les serveurs de conteneur : la liste des noeuds finals du service de catalogue doit être identique à la liste qui a été utilisé pour démarrer le serveur

de conteneur. Si vous n'avez pas spécifié cette option lorsque vous avez démarré le serveur de conteneur, utilisez la valeur par défaut `localhost:2809`.

Pour les serveurs de catalogue : si vous arrêtez le service de catalogue, utilisez les valeurs que vous avez indiquées pour les options **-listenerHost** et **-listenerPort** lorsque vous avez démarré le service de catalogue. Si vous n'avez pas spécifié ces options lorsque vous avez démarré le serveur de catalogue, utilisez la valeur par défaut `localhost:2809`. La valeur **-catalogServiceEndPoints** que vous utilisez lorsque vous arrêtez le service de catalogue est différente lorsque vous démarrez le service de catalogue.

-clientSecurityFile <fichier de propriétés du serveur>
Indique le chemin d'accès au fichier de propriétés client qui définit les propriétés de sécurité du client. Voir Fichier de propriétés du client pour plus d'informations sur les paramètres de sécurité dans ce fichier.

-traceSpec <spécification de la trace>

Active la trace et la chaîne de spécification de trace du serveur de conteneur. La trace est désactivée par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. Exemples :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <fichier de trace>

Indique le nom du fichier dans lequel les informations de trace seront consignées. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Exemple :** `../logs/c4Trace.log`

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

Démarrage des serveurs autonomes qui utilisent le transport ORB

(Obsolète) Lorsque vous exécutez une configuration autonome, l'environnement se compose de serveurs de catalogue, de serveurs de conteneur et de processus client. Les serveurs WebSphere eXtreme Scale peuvent être également intégrés à des applications Java existantes en utilisant l'API Embedded Server. Vous devez manuellement configurer et démarrer ces processus.

Avant de commencer

Vous pouvez démarrer des serveurs WebSphere eXtreme Scale dans un environnement dans lequel WebSphere Application Server n'est pas installé. Si vous utilisez WebSphere Application Server, voir «Configuration de WebSphere eXtreme Scale avec WebSphere Application Server», à la page 306.

Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO

(XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Démarrage d'un service de catalogue autonome qui utilise le transport ORB

(Obsolète) Vous devez démarrer le service de catalogue manuellement si vous utilisez un environnement WebSphere eXtreme Scale réparti qui n'est pas exécuté dans WebSphere Application Server.

Avant de commencer

Obsolète :  **8.6+** Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

- Si vous utilisez WebSphere Application Server, le service de catalogue démarre automatiquement dans les processus existants. Pour plus d'informations, voir «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 508.

Pourquoi et quand exécuter cette tâche

Démarrage du serveur de catalogue avec le script **startOgServer**. Lorsque vous appelez la commande de démarrage, utilisez le script **startOgServer.sh** sur les plateformes UNIX ou **startOgServer.bat** sous Windows.

Le service de catalogue peut être exécuté dans un seul processus ou il peut inclure plusieurs serveurs de catalogue afin de constituer un domaine de service de catalogue. Un domaine de service de catalogue est obligatoire dans un environnement de production pour la haute disponibilité. Pour plus d'informations, voir Service de catalogue à haute disponibilité. Vous pouvez également spécifier des paramètres supplémentaires au script pour associer l'ORB (Object Request Broker) à un hôte et un port spécifiques, spécifier le domaine ou activer la sécurité.

Procédure

- **Démarrez un processus de serveur de catalogue.**

  Pour démarrer un serveur de catalogues, entrez les commandes suivantes à partir de la ligne de commande :

  1. Accédez au répertoire `bin`.

     ```
     cd objectgridRoot/bin
     ```

  2. Exécutez la commande **startOgServer**.

     ```
     startOgServer.bat|sh catalogServer
     ```

  Pour la liste de tous les paramètres de ligne de commande disponibles, voir «Script **startOgServer** (ORB)», à la page 499. N'utilisez pas une seule machine virtuelle Java (JVM) pour exécuter le service de catalogue dans un environnement de production. Si le service de catalogue échoue, aucun nouveau client ne peut être acheminé vers l'instance eXtreme Scale déployée et aucune nouvelle instance ObjectGrid ne peut être ajoutée au domaine. Pour ces motifs, vous devez démarrer un ensemble de machines virtuelles Java pour pouvoir exécuter un domaine de service de catalogue.

- **Démarrez un domaine de service de catalogue constitué de plusieurs noeud finals.**

  Pour démarrer un ensemble de serveurs afin d'exécuter un service de catalogue, vous devez utiliser l'option **-catalogServiceEndPoints** sur le script startOgServer. Cet argument accepte une liste de noeuds finals de services de catalogue dans le format *serverName*:*hostName*:*clientPort*:*peerPort*.L'exemple suivant indique comment démarrer la première des trois machines virtuelles Java pour héberger un service de catalogue :

  1. Accédez au répertoire bin.

     cd *racine_install_wxs*/bin

  2. Exécutez la commande **startOgServer**.

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

     Dans ce cas, le serveur cs1 de l'hôte MyServer1.company.com est démarré. Ce nom de serveur est le premier argument transmis au script. Lors de l'initialisation du serveur cs1, les paramètres **-catalogServiceEndpoints** sont examinés pour déterminer les ports alloués pour ce processus. La liste est également utilisée pour permettre au serveur cs1 d'accepter les connexions des autres serveurs : cs2 et cs3.

  3. Pour démarrer les serveurs de catalogues restants de la liste, transmettez les arguments ci-après au script **startOgServer**. Démarrage du serveur cs2 sur l'hôte MyServer2.company.com.

```
startOgServer.bat|sh cs2 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

     Démarrage du serveur cs3 sur MyServer3.company.com :

```
startOgServer.bat|sh cs3 -catalogServiceEndPoints
cs3:MyServer3.company.com:6601:6602,cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602
```

     L'ordre de la liste pour le paramètre **-catalogServiceEndpoints** peut être différent pour les divers serveurs de catalogue, mais les serveurs contenus dans la liste doivent être identiques. N'insérez aucun espace dans la liste.

     Important : Démarrez au moins deux serveurs de catalogue en même temps.

     Vous devez démarrer les serveurs de catalogue qui se trouvent dans une grille de données en parallèle, car chaque serveur s'interrompt pour attendre que les autres serveurs de catalogue rejoignent le groupe central. Un serveur de catalogue qui est configuré pour une grille de données ne démarre pas tant qu'il n'a pas identifié les autres membres du groupe. Le serveur de catalogues arrive à expiration si aucun autre serveur ne devient disponible.

- **Liez l'ORB à un hôte et un port spécifiques.**

  En dehors des ports définis dans l'argument **catalogServiceEndpoints**, chaque service de catalogue utilise également un ORB (Object Request Broker) pour accepter les connexions des clients et des conteneurs. Par défaut, l'ORB écoute sur le port 2809 du système hôte local. Si vous souhaitez associer l'ORB à un hôte et un port spécifiques sur la machine virtuelle Java d'un service de catalogue, utilisez les arguments **-listenerHost** et **-listenerPort**. L'exemple suivant montre comment démarrer le serveur de catalogue d'une machine virtuelle Java unique avec son ORB associé au port 7000 sur MyServer1.company.com :

```
startOgServer.sh catalogServer -listenerHost MyServer1.company.com
-listenerPort 7000
```

Chacun des conteneurs et clients eXtreme Scale doit être fourni avec des données de point de contact d'ORB de service de catalogue. Les clients n'ont besoin que d'un sous-ensemble de ces données, mais vous devez utiliser au moins deux points de contact pour la haute disponibilité.

- Facultatif : **Nom du domaine de service de catalogue**

  Un nom de domaine de service de catalogue n'est pas requis lors du démarrage d'un service de catalogue. Toutefois, si vous utilisez la réplication multimaître ou plusieurs domaines de service de catalogue dans un même ensemble de processus, vous devez définir un nom de domaine de service de catalogue unique. Le nom de domaine par défaut est `DefaultDomain`. Pour affecter un nom à votre domaine, utilisez l'option **`-domain`**. L'exemple ci-après montre comment démarrer la machine virtuelle Java d'un service de catalogue avec le nom de domaine `myDomain`.

  ```
  startOgServer.sh catalogServer -domain myDomain
  ```

  Pour plus d'informations sur la configuration de la réplication multimaître, voir «Configuration de plusieurs topologies de centres de données», à la page 340.

- **Démarrez un service de catalogue sécurisé.** Pour plus d'informations, voir «Démarrage des serveurs sécurisés dans un environnement autonome», à la page 650.
- **Démarrez le service de catalogue à l'aide d'un programme.**

  Tout paramètre JVM qui est marqué par la méthode CatalogServerProperties.setCatalogServer peut héberger le service de catalogue pour eXtreme Scale. Cette méthode indique à l'environnement d'exécution du serveur eXtreme Scale d'instancier le service de catalogue lorsque le serveur est démarré. Le code qui suit montre comment instancier le serveur de catalogue eXtreme Scale :

  ```
  CatalogServerProperties catalogServerProperties =
   ServerFactory.getCatalogProperties();
  catalogServerProperties.setCatalogServer(true);

  //La méthode getInstance() démarre le service de catalogue.
  Server server = ServerFactory.getInstance();
  ```

  Pour plus d'informations sur le démarrage des serveurs à l'aide d'un programme, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.

Démarrage des serveurs de conteneur qui utilisent le transport ORB

(Obsolète) Vous pouvez démarrer les serveurs de conteneur depuis la ligne de commande en utilisant une topologie de déploiement ou un fichier `server.properties`.

8.6+ Avant de commencer

Obsolète : 8.6+ Les commandes **`startOgServer`** et **`stopOgServer`** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **`startXsServer`** et **`stopXsServer`** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Pourquoi et quand exécuter cette tâche

Pour démarrer un processus de conteneur, vous avez besoin d'un fichier ObjectGrid XML. Ce fichier spécifie quels serveurs eXtreme Scale sont hébergés par le conteneur. Vérifiez que votre conteneur est équipé pour héberger chaque ObjectGrid dans le fichier XML que vous lui transmettez. Toutes les classes que ces ObjectGrids requièrent doivent se trouver dans le chemin d'accès aux classes pour le conteneur. Pour plus d'informations sur le fichier XMLObjectGrid, voir Fichier `objectGrid.xsd`.

Procédure

- **Démarrez le serveur de conteneur depuis la ligne de commande.**
  1. A partir de la ligne de commande, accédez au répertoire bin :
     ```
     cd racine_install_wxs/bin
     ```
  2. Exécutez la commande suivante :
     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

     Important : Sur le serveur de conteneur, l'option **-catalogServiceEndPoints** permet de référencer l'hôte et le port ORB (Object Request Broker) dans le service de catalogue. Le service de catalogue utilise les options **-listenerHost** et **-listenerPort** pour spécifier l'hôte et le port de la fonction ORB ou accepte la liaison par défaut. Lorsque vous démarrez un conteneur, utilisez l'option **-catalogServiceEndPoints** pour référencer les valeurs transmises aux options **-listenerHost** et **-listenerPort** sur le service de catalogue. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées quand le service de catalogue est démarré, la fonction ORB est liée au port 2809 sur le système hôte local pour le service de catalogue. N'utilisez pas l'option **-catalogServiceEndPoints** pour référencer les hôtes et les ports transmis à l'option **-catalogServiceEndPoints** sur le service de catalogue. Sur le service de catalogue, l'option **-catalogServiceEndPoints** est utilisée pour spécifier les ports nécessaires pour une configuration de serveur statique.
     Ce processus est identifié par c0, le premier argument transmis au script. Utilisez le fichier `companyGrid.xml` pour démarrer le conteneur. Si votre fonction ORB de serveur de catalogue est exécutée sur un hôte différent que celui du conteneur ou qu'elle utilise un autre port que celui par défaut, vous devez utiliser l'argument **-catalogServiceEndPoints** pour vous connecter à la fonction ORB. Pour cet exemple, partez du principe qu'un unique service de catalogue est exécuté sur le port `2809` sur `MyServer1.company.com`
- **Démarrez le conteneur à l'aide d'une règle de déploiement.**

  Sans être nécessaire, une stratégie est recommandée pendant le démarrage du conteneur. La règle de déploiement est utilisée pour configurer le partitionnement et la réplication pour eXtreme Scale. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement. Comme l'exemple précédent ne fournit pas de fichier de règle de déploiement, l'exemple reçoit toutes les valeurs par défaut en ce qui concerne la réplication, le partitionnement et le positionnement. Donc, les mappes dans le CompanyGrid se trouvent dans un mapSet. Le mapSet n'est ni partitionné ni répliqué. Pour plus d'informations sur les fichiers de règle de déploiement, voir Fichier XML du descripteur de la règle de déploiement. L'exemple suivant utilise le fichier `companyGridDpReplication.xml` pour démarrer une machine virtuelle Java de conteneur, c0 :
  1. A partir de la ligne de commande, accédez au répertoire bin :
     ```
     cd racine_install_wxs/bin
     ```

2. Exécutez la commande suivante :

```
startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
-deploymentPolicyFile ../xml/companyGridDpReplication.xml
-catalogServiceEndPoints MyServer1.company.com:2809
```

Remarque : Si vous avez des classes Java stockées dans un répertoire spécifique, ou si vous utilisez un chargeur ou un agent, au lieu de modifier le script StartOgServer, vous pouvez lancer le serveur avec des arguments, comme suit : `-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`
. Dans le fichier `companyGridDpReplication.xml`, un seul groupe de mappes contient toutes les mappes. Ce mapSet est divisé en 10 partitions. Chaque partition a une réplique synchrone et aucune réplique asynchrone. Tout conteneur utilisant la règle de déploiement `companyGridDpReplication.xml` combinée au fichier XML ObjectGrid `companyGrid.xml` est également capable d'héberger des fragments de CompanyGrid. Démarrez une autre machine virtuelle Java de conteneur, c1 :

1. A partir de la ligne de commande, accédez au répertoire bin :

```
cd racine_install_wxs/bin
```

2. Exécutez la commande suivante :

```
startOgServer.sh c1 -objectGridFile ../xml/companyGrid.xml
-deploymentPolicyFile ../xml/companyGridDpReplication.xml
-catalogServiceEndPoints MyServer1.company.com:2809
```

Chaque règle de déploiement contient au moins un élément objectgridDeployment. Quand un conteneur est démarré, il publie sa règle de déploiement sur le service de catalogue. Le service de catalogue examine chaque élément objectgridDeployment. Si l'attribut objectgridName correspond à l'attribut objectgridName d'un élément objectgridDeployment précédemment reçu, l'élément objectgridDeployment le plus récent est ignoré. Le premier élément objectgridDeployment reçu pour un attribut objectgridName spécifique est utilisé comme élément maître. Par exemple, partons du principe que la machine virtuelle Java c2 utilise une règle de déploiement qui divise le mapSet en nombre différent de partitions :

companyGridDpReplicationModified.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd"
    xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="CompanyGrid">
        <mapSet name="mapSet1" numberOfPartitions="5"
            minSyncReplicas="1" maxSyncReplicas="1"
            maxAsyncReplicas="0">
            <map ref="Customer" />
            <map ref="Item" />
            <map ref="OrderLine" />
            <map ref="Order" />
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>
```

Vous pouvez maintenant démarrer une troisième machine virtuelle Java, c2 :

1. A partir de la ligne de commande, accédez au répertoire bin :

```
cd racine_install_wxs/bin
```

2. Exécutez la commande suivante :

```
startOgServer.sh c2 -objectGridFile ../xml/companyGrid.xml
-deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
-catalogServiceEndPoints MyServer1.company.com:2809
```

Le conteneur sur la machine virtuelle Java c2 est démarré avec une règle de déploiement qui spécifie 5 partitions pour le mapSet1. Cependant, le service de catalogue contient déjà la copie maître de l'objectgridDeployment pour le CompanyGrid. Quand la machine virtuelle Java c0 a été démarrée, elle a spécifié que 10 partitions existent pour ce mapSet. Comme il s'agit du premier conteneur à démarrer et publier sa règle de déploiement, cette dernière devient la stratégie maître. En conséquence, toute valeur d'attribut objectgridDeployment égale à `CompanyGrid` dans une règle de déploiement suivante est ignorée.

- **Démarrez un conteneur à l'aide d'un fichier de propriétés de serveur.**

  Vous pouvez utiliser un fichier de propriétés de serveur pour configurer la fonction de trace et la sécurité sur un conteneur. Exécutez les commandes suivantes pour démarrer un conteneur c3 avec un fichier de propriétés de serveur.

  1. A partir de la ligne de commande, accédez au répertoire bin :

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez la commande suivante :

     ```
     startOgServer.sh c3 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     -serverProps ../serverProps/server.properties
     ```

  Voici un exemple de fichier `server.properties` :

  ```
  server.properties
  workingDirectory=
  traceSpec=*=all=disabled
  systemStreamToFileEnabled=true
  enableMBeans=true
  memoryThresholdPercentage=50
  ```

  Il s'agit d'un fichier de propriétés de serveur de base dans lequel la sécurité n'est pas activée. Pour plus d'informations concernant le fichier `server.properties`, voir Fichier de propriétés du serveur.

- **Démarrez un serveur de conteneur à l'aide d'un programme.**

  Pour plus d'informations sur le démarrage des serveurs de conteneur à l'aide d'un programme, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.

Script `startOgServer` (ORB)

(Obsolète) Le script **`startOgServer`** démarre les serveurs de conteneur et de catalogue qui utilisent le mécanisme de transport ORB (Object Request Broker). Vous pouvez utiliser divers paramètres lorsque vous démarrez vos serveurs pour activer la trace, spécifiez des numéros de port, etc.

Rôle

Vous pouvez utiliser le script **`startOgServer`** pour démarrer les serveurs.

Obsolète : 8.6+ Les commandes **`startOgServer`** et **`stopOgServer`** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **`startXsServer`** et **`stopXsServer`** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Placement

Le script **startOgServer** se trouve dans le répertoire `bin` du répertoire root, par exemple :

```
cd racine_install_wxs/bin
```

Remarque : Si vous disposez de classes Java stockées dans un répertoire spécifique ou que vous utilisez un chargeur ou un agent, au lieu de modifier le script **startOgServer**, vous pouvez lancer le serveur avec des arguments, comme suit :
`-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`

.

Syntaxe des serveurs de catalogue

Pour démarrer un serveur de catalogues :

Windows

```
startOgServer.bat <server> [options]
```

UNIX

```
startOgServer.sh <server>[options]
```

Pour démarrer un serveur de catalogues configuré par défaut, utilisez les commandes suivantes :

Windows

```
startOgServer.bat catalogServer
```

UNIX

```
startOgServer.sh catalogServer
```

Options de démarrage des serveurs de catalogue

Les paramètres suivants sont tous facultatifs.

Paramètres de démarrage d'un serveur de catalogues :

-catalogServiceEndPoints <serverName:hostName:clientPort:peerPort>
Définit la liste des serveurs de catalogue à lier ensemble dans un domaine de services de catalogue. Chaque attribut est défini comme suit :

serverName
Définit le nom du serveur de catalogue.

hostName
Spécifie le nom d'hôte de l'ordinateur sur lequel le serveur est lancé.

clientPort
Spécifie le port utilisé pour la communication de service de catalogue homologue.

peerPort
Cette value est identique à haManagerPort. Spécifie le port utilisé pour la communication de service de catalogue homologue.

L'exemple suivant démarre le serveur de catalogue, cs1, qui se trouve dans le même domaine de services de catalogue que les serveurs cs2 et cs3 :

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,cs2:MyServer2.company.com:6601:6602,cs3:MyServer3.company.com:6601:6602
```

Si vous démarrez des serveurs de catalogue supplémentaires, ils doivent inclure les mêmes serveurs dans l'argument **-catalogServiceEndPoints**. L'ordre de la liste peut être différent, mais les serveurs contenus dans la liste doivent être les mêmes pour chaque serveur de catalogue. N'insérez aucun espace dans la liste.

-clusterSecurityFile <fichier_xml_sécurité_cluster>
Indique le fichier `objectGridSecurity.xml` sur le disque dur, qui décrit les propriétés de sécurité communes à tous les serveurs (y compris les serveurs de catalogue et les serveurs de conteneur). L'un des exemples de propriété est la configuration de l'authentificateur qui représente le registre d'utilisateurs et le mécanisme d'authentification.

Exemple :`/opt/xs/ogsecurity.xml`

-clusterSecurityUrl <URL du xml de la sécurité du cluster>
Indique le fichier `objectGridSecurity.xml` comme URL du fichier sur le disque dur ou sur le réseau, qui décrit les propriétés de sécurité communes à tous les serveurs, y compris les serveurs de catalogue et les serveurs de conteneur. L'un des exemples de propriété est la configuration de l'authentificateur qui représente le registre d'utilisateurs et le mécanisme d'authentification.

Exemple :`file:///opt/xs/ogsecurity.xml`

-domain <nom de domaine>
Indique le nom du domaine de services de catalogue du serveur de catalogue. Le domaine de services de catalogue crée un groupe de serveurs de catalogue à haute disponibilité. Chaque serveur de catalogue pour un seul domaine doit spécifier la même valeur pour le paramètre **-domain**.

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

-haManagerPort <port>
Indique le numéro de port utilisé par le gestionnaire de haute disponibilité. Si cette propriété n'est pas définie, un port libre est choisi. Cette propriété est ignorée dans les environnements WebSphere Application Server.

-JMXServicePort <port>
Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)

Valeur par défaut : `1099` pour les serveurs de catalogue

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual

machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel le transport ORB (Object Request Broker) ou eXtremeIO (XIO) se lie pour les communications. La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que le mécanisme de transport dans la machine JVM connaisse l'adresse IP de liaison. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** `localhost`

-listenerPort <port>

Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.**Valeur par défaut :** `2809`

-quorum true|false

Active le quorum pour le service de catalogue. Le quorum sert à garantir qu'une majorité du domaine de service de catalogue est disponible avant le déplacement des partitions sur les serveurs de conteneur disponibles. Pour activer le quorum, spécifiez la valeur `true` ou `enabled`. La valeur par défaut est `disabled`. Cette valeur s'applique au service de catalogue uniquement. Pour plus d'informations, voir Quorums de serveurs de catalogue.

-script <fichier script>

Indique l'emplacement d'un script personnalisé pour les commandes que vous spécifiez pour démarrer les serveurs de catalogue ou les conteneurs, puis définir des paramètres ou effectuer des modifications en fonction des besoins.

-serverProps <fichier de propriétés du serveur>

Indique le fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques au serveur. Le nom de fichier spécifié pour cette propriété correspond simplement à un chemin classique, tel que `c:/tmp/og/catalogserver.props`.

-traceSpec <spécification de la trace>

Active la trace et la chaîne de spécification de trace du serveur de conteneur. La trace est désactivée par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. Exemples :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <fichier de trace>

Indique le nom du fichier dans lequel les informations de trace seront consignées. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Exemple :** `../logs/c4Trace.log`

-timeout <secondes>

Indique un nombre de secondes avant que le démarrage du serveur n'arrive à expiration.

8.6+ **-transport <transport type>**

Indique le type de transport à utiliser pour tous les serveurs dans le domaine de service de catalogue Vous pouvez définir la valeur XIO ou ORB.

Le script **startOgServer** définit le type de transport ORB par défaut.

Si vous avez défini le paramètre **-transport** et la propriété de serveur **transport** ssur un serveur de catalogue, la valeur du paramètre **-transport** est utilisée.

Syntaxe des serveurs de conteneur Windows

```
startOgServer.bat <serveur> -objectgridFile <fichier xml>
-deploymentPolicyFile <fichier xml> [options]
```

Windows

```
startOgServer.bat <serveur> -objectgridUrl <URL du xml>
-deploymentPolicyUrl <URL du xml> [options]
```

UNIX

```
startOgServer.sh <server> -objectgridFile <fichier xml>
-deploymentPolicyFile <fichier xml> [options]
```

UNIX

```
 startOgServer.sh <serveur> -objectgridUrl <URL du xml>
-deploymentPolicyUrl <URL du xml> [options]
```

Options des serveurs de conteneur

-catalogServiceEndPoints<nomHôte:port,nomHôte:port>
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port dans le service de catalogue.

Valeur par défaut : localhost:2809

-deploymentPolicyFile <fichier xml de la règle de déploiement>
Indique le chemin d'accès au fichier de la stratégie de déploiement sur le disque dur. La stratégie de déploiement est utilisée pour configurer le partitionnement et la réplication. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement.

Exemple : `../xml/SimpleDP.xml`

-deploymentPolicyUrl <url de la règle de déploiement>
Indique l'URL du fichier de la stratégie de déploiement sur le disque dur ou sur le réseau. La stratégie de déploiement est utilisée pour configurer le partitionnement et la réplication. La règle de déploiement peut également être utilisée pour influencer le comportement de positionnement.

Exemple : `file://xml/SimpleDP.xml`

-JMXConnectorPort <port>
Définit le port SSL (Secure Sockets Layer) auquel se connecte le service Java Management Extensions (JMX).

-JMXServicePort <port>

Spécifie le numéro du port sur lequel le serveur MBean écoute les communications avec Java Management Extensions (JMX). La propriété JMXServicePort indique le port non SSL pour JMX. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre

configuration. Si vous voulez utiliser JMX/RMI, définissez explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. (Requis pour les environnements autonomes uniquement.)**Valeur par défaut :** 1099

-jvmArgs <arguments de la machine virtuelle Java>
Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs -Xms256M -Xmx1G

-listenerHost <nom d'hôte>

Indique le nom d'hôte auquel le transport ORB (Object Request Broker) ou eXtremeIO (XIO) se lie pour les communications. La valeur doit être un nom qualifié complet de domaine ou une adresse IP. Si la configuration implique plusieurs cartes réseau, configurez l'hôte du programme d'écoute et le port d'écoute pour que le mécanisme de transport dans la machine JVM connaisse l'adresse IP de liaison. Si vous ne définissez pas l'adresse IP à utiliser, des symptômes (délais de connexion, défaillances inhabituelles d'API et clients qui semblent se bloquer) apparaissent.**Valeur par défaut :** localhost

-listenerPort <port>
Indique le numéro de port auquel se connecte le transport Object Request Broker (ORB) ou eXtremeIO (XIO) . Ce paramètre configure les conteneurs et les clients pour communiquer avec le service de catalogue. Dans WebSphere Application Server, le port d'écoute est hérité par BOOTSTRAP_ADDRESS port (si vous utilisez le transport ORB ou le port XIO_address (si vous utilisez le transport XIO). Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue.**Valeur par défaut :** 2809

-objectgridFile <Fichier XML du descripteur d'ObjectGrid>
Indique le chemin d'accès au fichier du descripteur d'ObjectGrid. Le fichier XML ObjectGrid spécifie les serveurs eXtreme Scale hébergés par le conteneur.

-objectgridUrl <URL du descripteur d'ObjectGrid>
Spécifie une URL pour le fichier descripteur ObjectGrid. Le fichier XML ObjectGrid spécifie les serveurs eXtreme Scale hébergés par le conteneur.

-script <fichier script>
Indique l'emplacement d'un script personnalisé pour les commandes que vous spécifiez pour démarrer les serveurs de catalogue ou les conteneurs, puis définir des paramètres ou effectuer des modifications en fonction des besoins.

-serverProps <fichier de propriétés du serveur>
Indique le chemin d'accès au fichier de propriétés du serveur.

Exemple :../security/server.props

-timeout <secondes>
Indique un nombre de secondes avant que le démarrage du serveur n'arrive à expiration.

-traceFile <fichier de trace>

Indique le nom du fichier dans lequel les informations de trace seront consignées. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Exemple :** ../logs/c4Trace.log

-traceSpec <spécification de la trace>

Active la trace et la chaîne de spécification de trace du serveur de conteneur. La trace est désactivée par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. Exemples :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-zone <nom de zone>
Indique la zone à utiliser pour tous les conteneurs du serveur. Voir les «Routage par zone préférée», à la page 294 informations sur les zones dans *Présentation du produit* pour plus d'informations sur la configuration de zones.

Arrêt des serveurs autonomes qui utilisent le transport ORB

(Obsolète) Vous pouvez utiliser le script `stopOgServer` pour arrêter les processus serveur eXtreme Scale.

8.6+ Avant de commencer

Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Pourquoi et quand exécuter cette tâche

Exécutez le script **stopOgServer** en accédant au répertoire `bin` :

```
cd racine_install_wxs/bin
```

Procédure

- **Arrêtez un serveur de conteneur.**

  Utilisez le script **stopOgServer** pour arrêter le serveur de conteneur. Utilisez cette commande uniquement lorsque vous arrêtez un seul serveur de conteneur. Si vous exécutez la commande d'arrêt du serveur de catalogue unique sur plusieurs serveurs de conteneur à la suite, des problèmes de performances et de désabonnement se produisent pour le placement des fragments.

  ```
  stopOgServer containerServer -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Avertissement : L'option **-catalogServiceEndPoints** doit correspondre à la valeur de l'option **-catalogServiceEndPoints** utilisée pour démarrer le conteneur. S'il n'a pas été fait usage de **-catalogServiceEndPoints** pour démarrer le conteneur, les valeurs par défaut seront probablement localhost ou le nom d'hôte et 2809 pour le port ORB de connexion au service de catalogue. Autrement, utilisez les valeurs envoyées à **-listenerHost** et **-listenerPort** dans le service de catalogue. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées lors du démarrage du service de catalogue, l'ORB se lie au port 2809 sur localhost pour le service de catalogue.

- **Arrêtez plusieurs serveurs de conteneur.**

  Pour éviter les problèmes de //désabonnement et performances pour le placement des fragments lorsque vous voulez arrêter plusieurs serveurs de conteneur en même temps, utilisez la syntaxe de commande suivante. Séparez les serveurs de conteneur avec une virgule :

```
stopOgServer containerServer0,containerServer1,containerServer2
-catalogServiceEndPoints MyServer1.company.com:2809
```

Si vous souhaitez arrêter tous les conteneurs dans une zone ou un hôte, vous pouvez utiliser le paramètre **-teardown**. Pour plus d'informations, voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 508.

- **Arrêtez les serveurs de catalogue.**

  Exécutez le script **stopOgServer** pour arrêter le serveur de catalogue.

  ```
  stopOgServer.sh serveurCatalogue -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Avertissement : Lorsque vous arrêtez un service de catalogue, utilisez l'option **-catalogServiceEndPoints** pour référencer l'hôte ORB (Object Request Broker) et le port dans le service de catalogue. Le service de catalogue utilise les options **-listenerHost** et **-listenerPort** pour spécifier l'hôte et le port pour la liaison ORB ou accepte la liaison par défaut. Si les options **-listenerHost** et **-listenerPort** ne sont pas utilisées quand le service de catalogue est démarré, la fonction ORB est liée au port 2809 sur le système hôte local pour le service de catalogue. L'option **-catalogServiceEndPoints** pour arrêter un service de catalogue est différente de l'option que vous avez utilisée pour le démarrer.

  Démarrer un service de catalogue requiert des ports d'accès homologues et des ports d'accès clients si les ports par défaut n'ont pas été utilisés. En revanche, l'arrêt d'un service de catalogue ne requiert que le port de l'ORB.

- **Arrêtez le serveur de la console Web.** Pour arrêter le serveur de la console Web, exécutez le script **stopConsoleServer.bat|sh**. Ce script se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 547.

- **Activez la fonction de trace pour le processus d'arrêt du serveur.**

  Si un conteneur ne parvient pas à s'arrêter, vous pouvez activer la fonction de trace pour vous aider dans le débogage du problème. Pour activer la fonction de trace au cours de l'arrêt d'un serveur, ajoutez les paramètres **-traceSpec** et **-traceFile** pour les commandes d'arrêt. Le paramètre **-traceSpec** spécifie le type de trace et le paramètre **-traceFile** spécifie le chemin d'accès et le nom du fichier à créer et à utiliser pour les données de trace.

  1. A partir de la ligne de commande, accédez au répertoire bin.

     ```
     cd racine_install_wxs/bin
     ```

  2. Exécutez le script **stopOgServer** avec la fonction de trace activée.

     ```
     stopOgServer.sh c4 -catalogServiceEndPoints MyServer1.company.com:2809
      -traceFile ../logs/c4Trace.log -traceSpec ObjectGrid=all=enabled
     ```

     Une fois la trace obtenue, recherchez les erreurs relatives aux conflits de ports, aux classes manquantes, aux fichiers XML manquants ou incorrects, ou toute trace de pile. Suggestions de spécifications de trace au démarrage :

     - `ObjectGrid=all=enabled`
     - `ObjectGrid*=all=enabled`

     Pour connaître toutes les options de spécification de trace, voir «Options de trace du serveur», à la page 669.

- **Arrêtez les serveurs embarqués à l'aide d'un programme.**

  Pour plus d'informations sur cette opération, voir «Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510.

Script stopOgServer (ORB)

(Obsolète) Le script **stopOgServer** arrête les serveurs de catalogue et de conteneur.

Rôle

Utilisez le script **stopOgServer** pour arrêter un serveur. Vous devez indiquer le nom du serveur et ses noeuds finals de service de catalogue.

Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Emplacement

Le script **stopOgServer** se trouve dans le répertoire `bin` du répertoire `root` , par exemple :

```
cd racine_install_wxs/bin
```

Syntaxe

Pour arrêter un serveur de catalogue ou de conteneur : Windows

```
stopOgServer.bat <server_name> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

UNIX

```
stopOgServer.sh <server_name> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

Options

-catalogServiceEndPoints <csHost:csListenerPort, csHost:csListenerPort...>
Spécifie l'hôte ORB (Object Request Broker) et le numéro de port.

Pour les serveurs de conteneur : la liste des noeuds finals du service de catalogue doit être identique à la liste qui a été utilisé pour démarrer le serveur de conteneur. Si vous n'avez pas spécifié cette option lorsque vous avez démarré le serveur de conteneur, utilisez la valeur par défaut `localhost:2809`.

Pour les serveurs de catalogue : si vous arrêtez le service de catalogue, utilisez les valeurs que vous avez indiquées pour les options **-listenerHost** et **-listenerPort** lorsque vous avez démarré le service de catalogue. Si vous n'avez pas spécifié ces options lorsque vous avez démarré le serveur de catalogue, utilisez la valeur par défaut `localhost:2809`. La valeur **-catalogServiceEndPoints** que vous utilisez lorsque vous arrêtez le service de catalogue est différente lorsque vous démarrez le service de catalogue.

-clientSecurityFile <fichier de propriétés du serveur>
Indique le chemin d'accès au fichier de propriétés client qui définit les propriétés de sécurité du client. Voir Fichier de propriétés du client pour plus d'informations sur les paramètres de sécurité dans ce fichier.

-traceSpec <spécification de la trace>

Active la trace et la chaîne de spécification de trace du serveur de conteneur. La trace est désactivée par défaut. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. Exemples :

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <fichier de trace>

Indique le nom du fichier dans lequel les informations de trace seront consignées. Cette propriété s'applique à la fois au serveur de conteneur et au service de catalogue. **Exemple :** `../logs/c4Trace.log`

-jvmArgs <arguments de la machine virtuelle Java>

Indique un ensemble d'arguments de machine virtuelle Java. Chaque option après l'option **-jvmArgs** est utilisée pour démarrer la machine JVM (Java virtual machine) du serveur. Si le paramètre **-jvmArgs** est utilisé, vérifiez qu'il s'agit du dernier argument de script facultatif spécifié.

Exemple :-jvmArgs `-Xms256M -Xmx1G`

Arrêt propre des serveurs avec l'utilitaire `xscmd`

Vous pouvez utiliser l'utilitaire **inattendue** avec la commande **-c teardown** pour arrêter une liste ou un groupe de serveurs de catalogue et de conteneur. Cette commande simplifie l'arrêt d'une partie ou de la totalité d'une grille de données en évitant que le service de catalogue ait a exécuter des actions de placement et de récupération inutiles qui sont généralement exécutées lorsque des processus sont arrêtés.

Procédure

- Arrêtez les serveurs d'une liste de serveurs.

  Fournissez une liste de serveurs après le paramètre **-teardown** :

  ```
  xscmd –c teardown -sl catalogServer1,catalogServer2,containerServer1
  ```

- Arrêter tous les serveurs dans une zone donnée.

  Utilisez le paramètre **-z** et fournissez le nom de la zone. Le serveur de catalogue détermine les serveurs qui sont en cours d'exécution dans la zone et l'utilitaire **xscmd** demande une liste de serveurs dans la zone sélectionnée pour arrêter les serveurs :

  ```
  xscmd –c teardown –z zone_name
  ```

- Arrêter tous les serveurs sur un hôte.

  Utilisez le paramètres **-hf** et fournissez le nom de l'hôte. Par exemple, pour arrêter tous les serveurs sur `myhost.mycompany.com`, entrez `-hf myhost.mycompany.com`. Le serveur de catalogue détermine les serveurs qui sont en cours d'exécution sur l'hôte et l'utilitaire **xscmd** demande une liste de serveurs sur hôte sélectionné pour arrêter les serveurs :

  ```
  xscmd –teardown –hf <host_name>
  ```

Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server

Les serveurs de catalogue et de conteneur peuvent être démarrés automatiquement dans un environnement WebSphere Application Server ou WebSphere Application Server Network Deployment.

Avant de commencer

Configurer les serveurs de catalogue et les serveurs de conteneur pour s'exécuter sur WebSphere Application Server:

- «Configuration du service de catalogue dans WebSphere Application Server», à la page 306
- «Configuration des serveurs de conteneur dans WebSphere Application Server», à la page 333

Pourquoi et quand exécuter cette tâche

Le cycle de vie des serveurs de catalogue et de conteneur dans WebSphere Application Server est lié au processus dans lequel ces serveurs s'exécutent.

Procédure

- **Démarrez les services de catalogue dans WebSphere Application Server** :

  Le cycle de vie d'un serveur de catalogue est lié au processus WebSphere Application Server. Après avoir configuré le domaine de service de catalogue dans WebSphere Application Server, redémarrez chaque serveur que vous avez défini pour ce domaine. Le service de catalogue démarre automatiquement sur les serveurs que vous avez associés au domaine de service de catalogue. Le service de catalogue peut également démarrer automatiquement dans les scénarios suivants, en fonction de l'édition WebSphere Application Server :

  – **Base WebSphere Application Server** : vous pouvez configurer votre application pour démarrer automatiquement un serveur de conteneur et le service de catalogue. Cette fonctionnalité simplifie le test des unités dans les environnements de développement comme Rational Application Developer car vous n'avez pas besoin de démarrer explicitement un service de catalogue. Pour plus d'informations, voir «Configuration des applications WebSphere Application Server pour démarrer automatiquement les serveurs de conteneur», à la page 333.

  – **WebSphere Application Server Network Deployment** : le service de catalogue démarre automatiquement dans le processus du gestionnaire de déploiement si le noeud du gestionnaire de déploiement a WebSphere eXtreme Scale est installé et que le profil du gestionnaire de déploiement est étendu. Pour plus d'informations, voir «Configuration du service de catalogue dans WebSphere Application Server», à la page 306..

- **Démarrez les serveurs de conteneur dans WebSphere Application Server** :

  Le cycle de vie d'un serveur de catalogue est lié à l'application WebSphere Application Server. Lorsque vous démarrez l'application configurée, les serveurs de conteneur démarrent également.

- **Arrêtez la totalité de la grille de données des serveurs** :

  Vous pouvez arrêter les serveurs de catalogue et de conteneur en arrêtant les applications et les serveurs d'applications associés. Toutefois, vous pouvez également arrêter l'ensemble d'une grille de données avec l'utilitaire **xscmd** ou des beans gérés :

  – **Dans l'utilitaire xscmd :**

    Voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 508 pour plus d'informations sur l'arrêt de l'ensemble d'une grille de données.

  – **Avec des beans gérés :**

    utilisez l'opération tearDownServers sur le bean géré PlacementServiceMBean.

Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs

Java

Avec WebSphere eXtreme Scale, vous pouvez utiliser une interface de programmation d'application pour gérer le cycle de vie des conteneurs et serveurs imbriqués. Vous pouvez configurer le serveur à l'aide d'un programme avec les options que vous pouvez également configurer avec la ligne de commande ou les propriétés de serveur incluses dans un fichier. Vous pouvez configurer le serveur imbriqué pour en faire un serveur de conteneur et/ou un service de catalogue.

Avant de commencer

- Vous devez disposer d'une méthode permettant d'exécuter du code depuis une machine virtuelle Java existante. les classes eXtreme Scale doivent être disponibles dans l'arborescence du chargeur de classe.
- Si les serveurs de conteneur utilisent IBM eXtremeMemory, vous devez d'abord configurer les bibliothèques natives. Pour plus d'informations, voir «Configuration d'IBM eXtremeMemory», à la page 357.

Pourquoi et quand exécuter cette tâche

Vous pouvez effectuer de nombreuses tâches d'administration à l'aide de l'API d'administration. L'API est couramment utilisée comme serveur interne pour stocker l'état d'une application Web. Le serveur Web peut démarrer un serveur WebSphere eXtreme Scale imbriqué et signaler le serveur de conteneur au service de catalogue. Le serveur est ensuite ajouté comme membre d'une grille répartie plus importante. Cette utilisation peut offrir des possibilité d'évolution et une haute disponibilité à un fichier de données qui reste sinon volatile.

Vous pouvez contrôler à l'aide d'un programme le cycle de vie complet d'un serveur eXtreme Scale imbriqué. Les exemples sont aussi génériques que possible et n'illustrent que des exemples de code spécifiques aux étapes présentées.

Procédure

1. Procurez-vous l'objet ServerProperties de la classe ServerFactory et configurez les options nécessaires.

   Chaque serveur eXtreme Scale possède un ensemble de propriétés configurables. Lorsqu'un serveur est démarré à partir de la ligne de commande, ces propriétés reçoivent les valeurs par défaut, mais vous pouvez remplacer plusieurs propriétés en fournissant un fichier ou une source externe. Dans la portée imbriquée, vous pouvez directement définir les propriétés avec un objet ServerProperties. Vous devez définir ces propriétés avant d'obtenir une instance de serveur de la classe ServerFactory. L'exemple de fragment de code ci-après obtient un objet ServerProperties, définit la zone CatalogServiceBootStrap et initialise plusieurs paramètres de serveur facultatifs. Pour une liste des paramètres configurables, reportez-vous à la documentation de l'API.

   ```
   ServerProperties props = ServerFactory.getServerProperties();
   props.setCatalogServiceBootstrap("host:port"); // requis pour se connecter à un service
   de catalogue spécifique
   props.setServerName("ServerOne"); // nommez le serveur
   props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled");
   // Définit la spécification de trace
   ```

2. Si vous souhaitez que le serveur soit un service de catalogue, procurez-vous l'objet CatalogServerProperties.

Chaque serveur imbriqué peut être un service de catalogue, un serveur conteneur ou un serveur de conteneur et un service de catalogue. L'exemple ci-après obtient l'objet CatalogServerProperties, active l'option de service de catalogue et configure divers paramètres de service de catalogue.

```
CatalogServerProperties catalogProps = ServerFactory.getCatalogProperties();
catalogProps.setCatalogServer(true); // false par défaut ; doit être défini
comme service de catalogue
catalogProps.setQuorum(true); // active/désactive le quorum
```

3. Procurez-vous une instance Server à partir de la classe ServerFactory. L'instance Server est un singleton de portée processus chargé de gérer l'appartenance dans la grille. Une fois que cette instance a été instanciée, ce processus est connecté et devient hautement disponible pour les autres serveurs de la grille. L'exemple suivant montre comment créer l'instance Server :

   ```
   Server server = ServerFactory.getInstance();
   ```

   Si nous considérons l'exemple précédent, la classe ServerFactory fournit une méthode statique qui renvoie une instance Server. La classe ServerFactory est prévue pour être la seule interface permettant d'obtenir une instance Server. Par conséquent la classe garantit que l'instance est un singleton ou une instance pour chaque machine virtuelle Java ou chargeur de classe isolé. La méthode getInstance initialise l'instance Server. Vous devez configurer toutes les propriétés du serveur avant d'initialiser l'instance. La classe Server est chargée de créer des instances Container. Vous pouvez utiliser à la fois la classe ServerFactory et la classe Server pour gérer le cycle de vie de l'instance Server imbriquée.

4. Démarrez une instance Container à l'aide de l'instance Server.

   Pour que des fragments puissent être positionnées sur un serveur imbriqué, vous devez créer un conteneur sur le serveur. L'interface Server contient une méthode createContainer et accepte un argument DeploymentPolicy. L'exemple ci-après utilise l'instance de serveur que vous avez obtenue pour créer un conteneur à l'aide du fichier de règles de déploiement. Notez que les conteneurs requièrent un chargeur de classe pour lequel les fichiers binaires de l'application sont disponibles, à des fins de sérialisation. Vous pouvez rendre ces fichiers binaires disponibles en appelant la méthode createContainer avec comme chargeur de classe du contexte de l'unité d'exécution, celui que vous souhaitez utiliser.

   ```
   DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(new
      URL("file://urltodeployment.xml"),
    new URL("file://urltoobjectgrid.xml"));
   Container container = server.createContainer(policy);
   ```

5. Retirez et nettoyez un serveur de conteneur.

   Vous pouvez supprimer et nettoyer un serveur de conteneur en exécutant la méthode teardown sur l'instance Container obtenue. L'exécution de la méthode teardown sur un conteneur nettoie ce dernier de manière appropriée et supprime le conteneur du serveur imbriqué.

   La procédure de nettoyage du conteneur inclut le déplacement et le démontage de tous les fragments positionnés dans ce conteneur. Chaque serveur peut contenir plusieurs conteneurs et fragments. Le nettoyage d'un conteneur n'affecte pas le cycle de vie de l'instance Server parent. L'exemple ci-après montre comment exécuter la méthode teardown sur un serveur. La méthode teardown est rendue accessible via l'interface ContainerMBean. En utilisant l'interface ContainerMBean, si vous n'avez plus accès à ce conteneur à l'aide d'un programme, vous pouvez toujours le supprimer et le nettoyer avec son bean géré. Une méthode terminate existe également dans l'interface Container ; ne l'utilisez pas, sauf si cela est indispensable. Cette méthode est plus puissante et ne coordonne pas un déplacement et un nettoyage des fragments appropriés.

```
container.teardown();
```

6. Arrêtez le serveur imbriqué.

   Lorsque vous arrêtez un serveur imbriqué, vous arrêtez également les conteneurs et les fragments en cours d'exécution sur ce serveur. Lorsque vous arrêtez un serveur imbriqué, vous devez nettoyer toutes les connexions ouvertes et déplacer ou démonter tous les fragments. L'exemple ci-après illustre comment arrêter un serveur et utiliser la méthode waitFor sur l'instance Server pour s'assurer que cette dernière s'arrête complètement. Comme pour l'exemple de conteneur, la méthode stopServer est rendue accessible via l'interface ServerMBean. A l'aide de cette interface, vous pouvez arrêter un serveur avec le bean géré (MBean) correspondant.

```
ServerFactory.stopServer();  // Utilise la fabrique pour arrêter le singleton du serveur
 // ou
server.stopServer(); // Utilise directement l'instance Server
server.waitFor(); // Est renvoyé une fois que le serveur a correctement terminé
ses procédures d'arrêt
```

   Exemple de code complet :

```
 import java.net.MalformedURLException;
import java.net.URL;

import com.ibm.websphere.objectgrid.ObjectGridException;
import com.ibm.websphere.objectgrid.deployment.DeploymentPolicy;
import com.ibm.websphere.objectgrid.deployment.DeploymentPolicyFactory;
import com.ibm.websphere.objectgrid.server.Container;
import com.ibm.websphere.objectgrid.server.Server;
import com.ibm.websphere.objectgrid.server.ServerFactory;
import com.ibm.websphere.objectgrid.server.ServerProperties;

public class ServerFactoryTest {

    public static void main(String[] args) {

        try {

            ServerProperties props = ServerFactory.getServerProperties();
            props.setCatalogServiceBootstrap("catalogservice-hostname:catalogservice-port");
            props.setServerName("ServerOne"); // name server
            props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // TraceSpec

            /*
             * Dans la plupart des cas, le serveur ne sert que de serveur conteneur
             * et se connecte à un service de catalogue externe. Cette utilisation
             * favorise davantage la haute disponibilité. L'extrait de code commenté
             * ci-après permet à ce serveur de devenir un service de catalogue.
             *
             *
             * CatalogServerProperties catalogProps =
             * ServerFactory.getCatalogProperties();
             * catalogProps.setCatalogServer(true); // activez le service de catalogue
             * catalogProps.setQuorum(true); // activez le quorum
             */

            Server server = ServerFactory.getInstance();

            DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy
     (new URL("url to deployment xml"),  new URL("url to objectgrid xml file"));
            Container container = server.createContainer(policy);

            /*
             * Le fragment est maintenant positionné sur ce conteneur si les exigences
             * de déploiement sont satisfaites.
             * Cela englobe la création du serveur et du conteneur imbriqués.
             *
             * Les lignes ci-après illustrent simplement l'appel des méthodes de nettoyage
             */

            container.teardown();
            server.stopServer();
            int success = server.waitFor();

        } catch (ObjectGridException e) {
            // Container failed to initialize
        } catch (MalformedURLException e2) {
            // invalid url to xml file(s)
        }
```

```
        }
    }
```

API de serveurs intégrés

Java

WebSphere eXtreme Scale comprend des interfaces de programmes d'application (API) et des interfaces de programmation de système permettant d'intégrer des serveurs et des clients eXtreme Scale dans les applications Java existantes.

Instanciation du serveur eXtreme Scale

Plusieurs propriétés permettent de configurer l'instance du serveur eXtreme Scale, qu'il est possible d'extraire de la méthode ServerFactory.getServerProperties. L'objet ServerProperties étant un singleton, chaque appel à la méthode getServerProperties extrait la même instance.

Vous créez un serveur en utilisant le code suivant.

```
Server server = ServerFactory.getInstance();
```

Toutes les propriétés définies avant le premier appel de la méthode getInstance sont utilisées pour initialiser le serveur.

Définition des propriétés du serveur

Vous pouvez définir les propriétés du serveur jusqu'au premier appel de la méthode ServerFactory.getInstance. Le premier appel de la méthode getInstance instancie le serveur eXtreme Scale et lit toutes les propriétés configurées. Les propriétés définies après la création n'ont aucun effet. L'exemple suivant montre comment définir les propriétés avant d'instancier une instance Serveur.

```
// L'on obtient les propriétés du serveur associées à ce processus.
ServerProperties serverProperties = ServerFactory.getServerProperties();

// L'on définit le nom du serveur pour ce processus.
serverProperties.setServerName("EmbeddedServerA");

// L'on définit le nom de la zone dans laquelle est contenu ce processus.
serverProperties.setZoneName("EmbeddedZone1");

// L'on définit les informations de point de contact requises
pour l'amorçage du service de catalogue.
serverProperties.setCatalogServiceBootstrap("localhost:2809");

// Définition du nom d'hôte d'écoute à utiliser pour la liaison.
serverProperties.setListenerHost("host.local.domain");

// Définition du port d'écoute à utiliser pour la liaison.
serverProperties.setListenerPort(9010);

// L'on désactive tous les beans gérés pour ce processus.
serverProperties.setMBeansEnabled(false);

Server server = ServerFactory.getInstance();
```

Incorporation du service de catalogue

Tout paramètre JVM qui est marqué par la méthode CatalogServerProperties.setCatalogServer peut héberger le service de catalogue

pour eXtreme Scale. Cette méthode indique à l'environnement d'exécution du serveur eXtreme Scale d'instancier le service de catalogue lorsque le serveur est démarré. Le code qui suit montre comment instancier le serveur de catalogue eXtreme Scale :

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
```

Incorporation d'un serveur de conteneur

Exécutez la méthode Server.createContainer pour qu'une JVM héberge plusieurs serveurs de conteneur eXtreme Scale. Le code qui suit montre comment instancier un serveur de conteneur :

```
Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);
```

Processus serveur autonome

Vous pouvez démarrer ensemble tous les services, ce qui est utilise aussi bien en phase de développement qu'en production. En démarrant les services ensemble, le même processus se charge de toutes les tâches suivantes : démarrage du service de catalogue, démarrage d'un ensemble de conteneurs et exécution de la logique de connexion client. Démarrer les services de cette manière résout les problèmes de programmation avant le déploiement dans un environnement réparti. Le code qui suit montre comment instancier un serveur eXtreme Scale autonome :

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);
```

Incorporation de eXtreme Scale dans WebSphere Application Server

La configuration pour eXtreme Scale est définie automatiquement lorsque vous installez eXtreme Scale dans un environnement WebSphere Application Server. Vous n'êtes pas obligé de définir des propriétés avant d'accéder au serveur pour créer un conteneur. Le code suivant montre comment instancier un serveur eXtreme Scale server inWebSphere Application Server:

```
Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL);
Container container = server.createContainer(policy);
```

«Utilisation de l'API de serveur embarqué pour démarrer et arrêter les serveurs», à la page 510 contient un exemple expliquant pas à pas comment démarrer par programmation un service de catalogue et un conteneur intégrés.

Administration avec l'utilitaire xscmd

Utilisez l'utilitaire **xscmd** pour effectuer des tâches d'administration dans l'environnement , telles qu'établir des liens de réplication multimaître, remplacer un quorum et arrêter des groupes de serveurs avec la commande teardown.

Avant de commencer

- Les serveurs de catalogue et les serveurs de conteneur doivent être démarrés. Si les serveurs de catalogue se trouvent dans un domaine de service de catalogue, au moins deux serveurs de catalogue doivent être démarrés.
- Vérifiez que la variable d'environnement *JAVA_HOME* est définie pour utiliser l'environnement d'exécution installé avec le produit. Si vous utilisez la version d'évaluation du produit, vous devez définir la variable d'environnement *JAVA_HOME*.

Pourquoi et quand exécuter cette tâche

L'utilitaire **xscmd** remplace l'exemple d'utilitaire **xsadmin** comme outils de surveillance et d'administration complètement pris en charge. Vous pouvez exécuter des opérations similaires avec l'outil **xsadmin**, mais cet outil n'est pas pris en charge. L'exemple **xsadmin** fournit une méthode pour effectuer l'analyse syntaxique et la détection des données de déploiement actuelles et peut servir de modèle pour l'écriture d'utilitaires personnalisés. Si vous utilisiez l'outil **xsadmin** pour la surveillance et l'administration, mettez à jour vos scripts pour utiliser l'utilitaire **inattendue**. Pour plus d'informations sur le mappage des commandes **xsadmin** à la nouvelle commande **xscmd**, voir «Migration de l'outil **xsadmin** vers l'outil **xscmd**», à la page 260.

Procédure

1. Facultatif : Si l'authentification de client est activée : Ouvrez une fenêtre de ligne de commande. Sur la ligne de commande, définissez les variables d'environnement appropriées.
2. Accédez au répertoire *rép_base_wxs*/bin.

   ```
   cd rép_base_wxs/bin
   ```
3. Affichez l'aide des différentes options **xscmd**.
   - Pour afficher l'aide générale, exécutez la commande suivante :
     - UNIX `./xscmd.sh -h`
     - Windows `xscmd.bat -h`
   - Pour afficher la liste de toutes les commandes, exécutez la commande suivante :
     - UNIX `./xscmd.sh -lc`
     - Windows `xscmd.bat -lc`
   - Pour afficher l'aide d'une commande, exécutez la commande suivante :
     - UNIX `./xscmd.sh -h command_name`
     - Windows `xscmd.bat -h command_name`
   - Pour afficher une liste des groupes de commandes, exécutez la commande suivante :
     - UNIX `./xscmd.sh -lcg`
     - Windows `xscmd.bat -lcg`

- Pour afficher la liste des commandes dans un groupe de commandes, exécutez la commande suivante :
  - UNIX `./xscmd.sh -lc command_group_name`
  - Windows `xscmd.bat -lc command_group_name`

4. Exécutez les commandes de connexion à de serveurs de catalogue spécifiques. Par défaut, **xscmd** se connecte au serveur de catalogue sur l'hôte local en utilisant le nom d'hôte et le port `localhost:2809`. Vous pouvez également fournir la liste des noms d'hôte et des ports à la commande pour que vous puissiez vous connecter aux serveurs de catalogue sur d'autres hôtes. Dans la liste, l'utilitaire **xscmd** se connecte à un hôte aléatoire. La liste des hôtes que vous fournissez doit se trouver dans le même domaine de services de catalogue.
   - Fournissez la liste des serveurs de catalogue autonomes auxquels vous voulez vous connecter :
     - UNIX `./xscmd.sh -c <command_name> -cep hostname:port(,hostname:port)`
     - Windows `xscmd.bat -c <command_name> -cep hostname:port(,hostname:port)`

     Dans les commandes précédentes, *command_name* est le nom de la commande que vous exécutez. La valeur *hostname:port* est le nom d'hôte du serveur de catalogue et le port d'écoute. La valeur de port d'écoute sur un serveur de catalogue autonome est définie lorsque vous exécutez la commande **startOgServer** ou **startXsServer**.
   - Fournissez la liste des serveurs de catalogue WebSphere Application Server auxquels vous voulez vous connecter. Vous ne pouvez pas vous connecter aux serveurs de catalogue qui s'exécutent sur WebSphere Application Server avec la valeur localhost par défaut :
     - UNIX `./xscmd.sh -c <command_name> -cep was_hostname:port(,hostname:port)`
     - Windows `xscmd.bat -c <command_name> -cep was_hostname:port(,hostname:port)`

     Dans les commandes précédentes, *command_name* est le nom de la commande que vous exécutez. La valeur *was_hostname* est le nom d'hôte du serveur de catalogue dans la cellule WebSphere Application Server. La valeur *port* est le port d'écoute.

     8.6+ La valeur de port d'écoute dans WebSphere Application Server est héritée :
     - Si vous utilisez le transport d'ORB, la valeur **BOOTSTRAP_ADDRESS** pour chaque serveur d'applications WebSphere Application Server est utilisée. La valeur par défaut est `9809` si le serveur de catalogue s'exécute dans le gestionnaire de déploiement.
     - Si vous utilisez le transport IBM eXtremeIO, la valeur **XIO_ADDRESS** est utilisée. La valeur par défaut est `4809` si le serveur de catalogue s'exécute dans le gestionnaire de déploiement.

     Si vous exécutez le serveur de catalogue sur un serveur d'applications, vérifiez la configuration de port BOOTSTRAP_ADDRESS ou XIO_ADDRESS du serveur d'applications pour déterminer le numéro de port.

     Important : Si les serveurs de conteneur s'exécutent dans un environnement sécurisé WebSphere Application Server, exécutez l'utilitaire **xscmd** à partir de

l'installation WebSphere eXtreme Scale Client dans l'environnement WebSphere Application Server, par exemple, depuis le répertoire `/opt/IBM/WebSphere/AppServer/bin`.

5. 8.6+ Facultatif : Définissez une valeur de délai d'attente lorsque vous exécutez vos commandes. Vous pouvez utiliser l'option **-to** ou **--timeout** comme paramètre global dans n'importe quelle commande. Cette valeur définit le nombre de secondes avant l'expiration du délai lorsque vous vous connectez à des serveurs de catalogue dans la commande. Si vous vous connectez à un serveur de catalogue qui risque d'être indisponible à la suite de l'expiration du système d'exploitation ou réseau, cette option peut être utile pour réduire l'attente.

   La valeur d'expiration par défaut est 30 secondes.

Contrôle du placement

Vous pouvez utiliser différentes options pour contrôler quand les fragments sont placés sur les différents serveurs de la configuration. Lors du démarrage, vous pouvez décider de retarder le placement des fragments. Lorsque vous exécutez tous les serveurs de conteneur, il peut être nécessaire de suspendre ou de changer le placement pendant dans la gestion des serveurs.

Procédure

Contrôle du placement lors du démarrage
Vous pouvez contrôler quand les fragments commencent à être placés lors du démarrage de l'environnement. Il existe un contrôle par défaut. Si vous n'exécutez aucune action pour contrôler le placement des fragments, ce dernier commence immédiatement. Dans ce cas, les fragments peuvent ne pas être placés uniformément lorsque les serveurs de conteneur suivants démarrent et que d'autres opérations de placement sont exécutées pour équilibrer la répartition.

- Suspendez temporairement l'équilibrage des fragments pour que les fragments ne soient pas placés immédiatement lorsque les serveurs de conteneur démarrent.

  La suspension de l'équilibrage des fragments évite le placement inégal des fragments. Avant de démarrer les serveurs de conteneur, utilisez la commande **xscmd -c suspendBalancing** pour arrêter l'équilibrage des fragments pour une grille de données et un groupe de mappes donnés. Après que les serveurs de conteneur ont démarré, vous pouvez utiliser la commande **xscmd -c resumeBalancing** pour commencer à placer les fragments sur le serveurs de conteneur.

- Définissez la propriété **placementDeferralInterval** pour réduire le nombre de cycles de placement de fragments sur les serveurs de conteneur. Le placement des fragments se déclenche à la fréquence définie.

  Définissez la propriété **placementDeferralInterval** dans le fichier des propriétés du serveur de catalogue. Si vous utilisez l'API de serveur intégré, utilisez la méthode setPlacementDeferralInterval sur l'interface CatalogServerProperties. Cette propriété définit le délai en millisecondes qui précède le placement des fragments sur les serveurs de conteneur. La valeur par défaut de la propriété est 15 secondes. Avec cette valeur, lorsqu'un serveur de conteneur démarre, le placement ne démarre pas tant que le délai défini dans la propriété n'est pas écoulé. Si plusieurs serveurs de conteneur démarrent consécutivement, le chronomètre de report est réinitialisé si un nouveau serveur de conteneur démarre dans le délai défini. Si, par exemple, un deuxième conteneur démarre 10 secondes après le premier, le placement a lieu 15 secondes après le

démarrage du deuxième serveur de conteneur. Toutefois, si un troisième serveur de conteneur démarre 20 secondes après le deuxième, le placement a déjà commencé sur les deux premiers serveurs de conteneur.

Lorsqu'un serveur de conteneur devient indisponible, le placement a lieu dès que le serveur de catalogue a connaissance de l'événement pour que la récupération ait lieu aussi rapidement que possible.

Suivez les conseils ci-dessous pour déterminer si la valeur de report du placement correspond au délai correct :

- Lorsque vous démarrez les serveurs simultanément, consultez les messages CWOBJ1001 dans le fichier `SystemOut.log` de chaque serveur de conteneur. L'horodatage de ces messages dans chaque fichier journal de serveur de conteneur indique l'heure de début du serveur de conteneur. Vous pouvez envisager d'ajuster la propriété **placementDeferralInterval** pour inclure plus de démarrages de serveur de conteneur. Par exemple, si le premier serveur de conteneur démarre 90 secondes avant le dernier serveur de conteneur, vous pouvez affecter la valeur 90 secondes à la propriété.
- Notez le délai d'apparition des messages CWOBJ1511 après les messages CWOBJ1001. Ce délai peut indiquer si le report a abouti.
- Si vous utilisez un environnement de développement, tenez compte du délai lorsque vous testez l'application.

- Définissez l'attribut **numInitialContainers**.

  Si vous avez déjà utilisé l'attribut **numInitialContainers**, vous pouvez continuer de l'utiliser. Toutefois, il est préférable d'utiliser les commandes **xscmd -c suspendBalancing** et **xscmd -c resumeBalancing** suivies de **placementDeferralInterval** que l'attribut **numInitialContainers** pour contrôler le placement. L'attribut **numInitialContainers** indique le nombre de serveurs de conteneur nécessaires avant le placement initial des fragments dans cet élément mapSet. L'attribut **numInitialContainers** se trouve dans le fichier XML descripteur de stratégie de déploiement. Si **numInitialContainers** et **placementDeferralInterval** sont définis, aucun placement n'a lieu jusqu'à ce que la valeur **numInitialContainers** soit atteinte, quelle que soit la valeur de la propriété **placementDeferralInterval**.

Contrôle du placement après le démarrage initial

- Forcez le placement.

  Utilisez la commande **xscmd -c triggerPlacement -g my_OG -ms my_Map_Set**, où *my_OG* et *my_Map_Set* sont affectés de valeur pour la grille de données et le groupe de mappes, pour forcer le placement à un moment où il n'aurait pas lieu. Par exemple, vous pouvez exécuter cette commande lorsque le délai défini par la propriété **placementDeferralInterval** n'a pas encore été transmise ou lorsque l'équilibrage est suspendu.

- Réaffectez un fragment primaire.

  Utilisez la commande **xscmd -c swapShardWithPrimary** pour affecter un fragment de réplique comme fragment primaire. Le fragment primaire antérieur devient une réplique.

- Rééquilibrez les fragments primaire et de réplique.

  Utilisez la commande **xscmd -c balanceShardTypes** pour ajuster le taux des fragments primaire et de réplique uniformément entre les serveurs de conteneur de la configuration. Le taux est cohérent dans un fragment sur chaque serveur de conteneur.

- Suspendez ou relancez le placement.

  Utilisez la commande **xscmd -c suspendBalancing** ou **xscmd -c resumeBalancing** pour arrêter et démarrer l'équilibrage des fragments d'une grille de données ou

d'un groupe de mappes. Lorsque l'équilibrage est suspendu, les actions de placement suivantes peuvent toujours s'exécuter :

- La promotion de fragment peut avoir lieu lorsque les serveurs de conteneur sont défaillants.
- La permutation de rôle de fragment avec la commande **xscmd -c swapShardWithPrimary**.
- L'équilibrage déclenché par le placement de fragments avec la commande **xscmd -c triggerPlacement -g myOG -ms myMapSet**.

- **8.6+** Réactivez les conteneurs de fragments désactivés pour le placement des fragments.

  En cas de problème de placement des fragments dans un conteneur de fragments, le conteneur de fragments est placé dans la liste de désactivation du placement des fragments. Les conteneurs de fragments de cette liste ne peuvent pas être utilisés pour le placement jusqu'à ce que vous les réactiviez ou que la machine virtuelle Java qui héberge le conteneur de fragment soit recyclée. Lorsque la machine JVM est arrêtée, le conteneur de fragments est supprimé. Lorsque la machine JVM est redémarrée, le nombre de conteneurs augmente et un nouveau nom est utilisé pour le conteneur de fragments pour une grille de données spécifique. Problèmes pouvant entraîner la désactivation d'un conteneur de fragments : longs cycles de récupération de place qui affectent la santé de la machine JVM, problèmes de configuration DNS ou de dénomination, indisponibilité intermittentes du réseau, etc. Les fragments qui ont été placés avec succès dans le conteneur de fragments ne sont pas retirés du fragment de conteneur. Il est possible que les clients puissent accéder à un fragment, mais les communications entre les fragments de conteneur ou les serveurs de catalogue et de conteneur ne fonctionnent pas.

  Les conteneurs de fragments qui se trouvent dans la liste de désactivation du placement des fragment considérés comme NON AFFECTES. Si la machine JVM du conteneur de fragments n'est pas recyclée ou qu'un autre conteneur de fragments n'est pas arrêté ou démarré, les fragments restent non affectés si vous n'utilisez pas la commande **xscmd -c triggerPlacement**. Le cycle d'équilibre ne s'exécute pas automatiquement lorsqu'un conteneur de fragments est désactivé, car le fragment concerné (ou les données qui s'y trouvent) peut générer un problème. Pour éviter de propager les fragments vers les autres conteneurs de fragments, le cycle d'équilibre ne s'exécute pas automatiquement. Vous devez analyser le problème et exécuter la commande **xscmd -c triggerPlacement** avant toute modification du cycle de vie d'un conteneur.

  Pour afficher la liste des conteneurs de fragments désactivés, utilisez la commande **xscmd -c listDisabledForPlacement**.

  Les conteneurs de fragments de cette liste ne peuvent pas être utilisés pour le placement jusqu'à ce que vous les réactiviez. Résolvez les problèmes éventuels liés au conteneur de fragments, puis exécutez la commande **xscmd -c enableForPlacement -ct <shard_container>**.

Que faire ensuite

Vous pouvez surveiller le placement de votre environnement à l'aide de la commande **xscmd -c placementServiceStatus**.

Gestion de la disponibilité ObjectGrid

L'état de disponibilité d'une instance ObjectGrid détermine les requêtes pouvant être traitées à tout moment. Vous pouvez utiliser l'interface de StateManager pour définir et extraire l'état d'une instance ObjectGrid.

Pourquoi et quand exécuter cette tâche

Il existe quatre états de disponibilité pour une instance ObjectGrid.

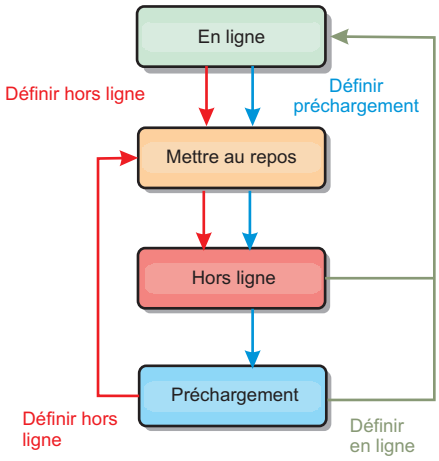


Figure 53. Etats de disponibilité d'une instance ObjectGrid

ONLINE
L'état ONLINE est l'état de disponibilité par défaut d'un ObjectGrid. Un ObjectGrid en ligne est capable de traiter n'importe quelle requête d'un client eXtreme Scale typique. Toutefois, les requêtes d'un client de préchargement sont rejetées lorsque l'ObjectGrid est en ligne.

QUIESCE
L'état QUIESCE est transitionnel. Un ObjectGrid qui a l'état QUIESCE passe rapidement à l'état OFFLINE. Lorsqu'un ObjectGrid a l'état QUIESCE, il peut traiter les transactions en attente. Toutefois, les nouvelles transactions sont rejetées. Un ObjectGrid peut rester au repos jusqu'à 30 secondes, Ensuite, l'état de disponibilité devient OFFLINE.

OFFLINE
L'état OFFLINE provoque le rejet de toutes les transactions envoyées à ObjectGrid.

PRELOAD
L'état PRELOAD (préchargement) peut servir à charger des données dans un ObjectGrid à partir d'un client de préchargement. Lorsque l'ObjectGrid est à l'état de préchargement, seul un client de préchargement peut valider des transactions par rapport à cet ObjectGrid. Toutes les autres transactions sont rejetées.

Une demande est rejetée si un ObjectGrid n'a pas l'état de disponibilité approprié pour la demande. Dans ce cas, une exception AvailabilityException est générée.

Procédure

1. Définissez l'état initial d'un ObjectGrid avec le fichier XML de configuration de l'ObjectGrid.

   Vous pouvez utiliser l'attribut **initialState** sur ObjectGrid pour indiquer son état au démarrage. Normalement, lorsqu'un ObjectGrid termine son initialisation, il devient disponible pour le routage. L'état peut ensuite être changé de manière à empêcher l'acheminement du trafic vers l'ObjectGrid. Si l'ObjectGrid doit être initialisé, mais pas immédiatement disponible, vous pouvez utiliser l'attribut **initialState**.

L'attribut initialState est défini dans le fichier XML de configuration de l'ObjectGrid. L'état par défaut est ONLINE. Les valeurs admises sont les suivantes :

- ONLINE (par défaut)
- PRELOAD
- OFFLINE

Voir Fichier XML du descripteur d'ObjectGrid pour plus d'informations sur l'attribut **initialState**.

Si l'attribut initialState est défini dans un ObjectGrid, l'état doit être explicitement ramené à Online, car sinon l'ObjectGrid reste indisponible. Une exception AvailabilityException se produit si l'ObjectGrid n'a pas l'état.

Voir la AvailabilityState documentation d'API pour plus d'informations.

Utilisation de l'attribut initialState pour le préchargement

Si l'ObjectGrid est préchargé avec des données, un laps de temps est susceptible de s'écouler entre le moment où il est disponible et le moment où il passe à l'état de préchargement permettant de bloquer le trafic client. Pour éviter ce laps de temps, l'état initial d'un ObjectGrid peut être défini comme PRELOAD. L'ObjectGrid effectue toujours l'initialisation requise, mais bloque le trafic jusqu'au changement d'état et permet au préchargement d'avoir lieu.

Les états PRELOAD et OFFLINE bloquent le trafic, mais seul l'état PRELOAD permet de lancer un préchargement.

Basculement et équilibrage

Si une grille de données de réplique est promue pour être une grille de données primaire, la réplique n'utilise pas le paramètre **initialState**. Si la grille de données primaire est déplacée pour cause de rééquilibrage, le paramètre **initialState** n'est pas utilisé, car les données sont copiées vers le nouvel emplacement primaire avant la fin du transfert Si la réplication n'est pas configurée, le fragment primaire passe à l'état **initialState** en cas de basculement et un nouveau fragment primaire doit être placé.

2. Changez l'état de disponibilité avec l'interface StateManager.

   Utilisez l'interface de StateManager pour définir l'état de disponibilité d'un ObjectGrid. Pour définir l'état de disponibilité d'un ObjectGrid exécuté sur les serveurs, transmettez un client ObjectGrid correspondant à l'interface de StateManager. Le code suivant démontre comment changer l'état de disponibilité d'un ObjectGrid.

   ```
   Client ClientClusterContext = ogManager.connect("localhost:2809", null, null);
   ObjectGrid myObjectGrid = ogManager.getObjectGrid(client, "myObjectGrid");
   StateManager stateManager = StateManagerFactory.getStateManager();
   stateManager.setObjectGridState(AvailabilityState.OFFLINE, myObjectGrid);
   ```

   Chaque fragment de l'ObjectGrid passe à l'état à appliquer lorsque la méthode setObjectGridState est appelée sur l'interface de StateManager. Lorsque la méthode est renvoyée, tous les fragments de l'ObjectGrid doivent être définis sur l'état adéquat.

   Utilisez un plug-in ObjectGridEventListener pour changer l'état de disponibilité d'un ObjectGrid côté serveur. Changez l'état de disponibilité d'un ObjectGrid côté serveur seulement lorsque ce dernier présente une partition unique. Si l'ObjectGrid présente plusieurs partitions, la méthode shardActivated est appelée sur chaque partition principale, ce qui entraîne des appels superflus pour le changement d'état de l'ObjectGrid

   ```
   public class OGListener implements ObjectGridEventListener,
      ObjectGridEventGroup.ShardEvents {
       public void shardActivated(ObjectGrid grid) {
   ```

```
            StateManager stateManager = StateManagerFactory.getStateManager();
            stateManager.setObjectGridState(AvailabilityState.PRELOAD, grid);
        }
    }
```

L'état QUIESCE étant transitionnel, vous ne pouvez pas utiliser l'interface de StateManager pour définir l'état d'un ObjectGrid sur QUIESCE. L'ObjectGrid passe par cet état avant d'être défini sur l'état OFFLINE.

3. Extrayez l'état de disponibilité.

   Utilisez la méthode getObjectGridState de l'interface de StateManager pour récupérer l'état de disponibilité d'un ObjectGrid.

   ```
   StateManager stateManager = StateManagerFactory.getStateManager();
   AvailabilityState state = stateManager.getObjectGridState(inventoryGrid);
   ```

   La méthode getObjectGridState choisit une partition principale de l'ObjectGrid au hasard et renvoie son état de disponibilité. Tous les fragments ObjectGrid doivent présenter le même état ou être en transition vers le même état. C'est pour cela que cette méthode propose un résultat acceptable pour l'état de disponibilité actuel de l'ObjectGrid.

Gestion des incidents du centre de données

Lorsque le centre de données entre un scénario d'échec, envisagez de remplacer de quorum de sorte que les événements du serveur de conteneur ne soient pas ignorés. Vous pouvez utiliser l'utilitaire **xscmd** pour obtenir des informations sur les tâches de quorum et exécuter ces tâches, telles que l'état du quorum et le remplacement du quorum.

Avant de commencer

- Configurez le mécanisme de quorum qui doit être identique dans tous les serveurs de catalogue. Pour plus d'informations, voir «Configuration du mécanisme de quorum», à la page 327.
- Le quorum est le nombre minimum de serveurs de catalogue nécessaires à l'exécution des opérations de positionnement pour la grille de données et représente l'ensemble des serveurs de catalogue, sauf si vous définissez un nombre inférieur. WebSphere eXtreme Scale s'attend à perdre le quorum pour les raisons suivantes :
  - défaillance d'un membre machine virtuelle Java du service de catalogue
  - microcoupure réseau
  - perte de centre de données

  Le message suivant indique que le quorum a été perdu. Recherchez la présence éventuelle de ce message dans les journaux de vos services de catalogue.

  ```
  CWOBJ1254W: Le service de catalogue attend un quorum.
  ```

Pourquoi et quand exécuter cette tâche

Remplacement du quorum dans un scénario de défaillance du centre de données uniquement. Lorsque vous remplacez le quorum, n'importe quelle instance de serveur de catalogue restante peut être utilisée. Tous les survivants sont notifiés lorsque l'un d'entre eux reçoit l'injonction de redéfinir le quorum.

Procédure

- Interrogation de l'état du quorum avec l'utilitaire **xscmd**.

  ```
  xscmd -c showQuorumStatus -cep cathost:2809
  ```

Utilisez cette option pour afficher l'état du quorum d'une instance de service de catalogue.

8.6+ Si vous le souhaitez, vous pouvez utiliser l'option **-to** ou **--timeout** dans la commande pour réduire le délai d'attente pour éviter d'attendre les délais d'attente de système d'exploitation ou réseau au cours d'un arrêt réseau ou d'une perte système. La valeur d'expiration par défaut est de 30 secondes.L'une des sorties suivantes s'affiche :

- `Le quorum est désactivé` : les serveurs de catalogue s'exécutent en mode Quorum désactivé. Ce mode est un mode de développement ou un mode de centre de données unique. Ne l'utilisez pas pour plusieurs configurations de centre de données.
- `Le quorum est activé et le serveur de catalogue a le quorum` : le quorum est activé et le système fonctionne normalement.
- `Le quorum est activé et le serveur de catalogue attend un quorum` : le quorum est activé et le quorum a été perdu.
- `Le quorum est activé et le quorum est remplacé` : le quorum est activé et le quorum n'a pas été remplacé.
- `L'état du quorum est proscrit` : lorsqu'une microcoupure se produit, le service de catalogue est scindé en deux partitions A et B. Le serveur de catalogue A a redéfini le quorum. La partition réseau se résout et le serveur dans la partition B est proscrit, nécessitant un redémarrage des machines virtuelles Java. Cela se produit également si la machine virtuelle Java du catalogue redémarre pendant la microcoupure et que cette dernière se termine.

- Remplacez le quorum avec l'utilitaire **xscmd**.

```
xscmd -c overrideQuorum -cep cathost:2809
```

L'exécution de cette commande oblige le serveur de catalogue restant à rétablir un quorum.

- Diagnostiquez un quorum avec l'utilitaire **xscmd**.
  - **Affichage de la liste des groupes centraux :**

    Utilisez l'option **-c listCoreGroups** pour afficher la liste de tous les groupes principaux du serveur de catalogue.

    ```
    xscmd -c listCoreGroups -cep cathost:2809
    ```

  - **Désassemblage des serveurs :**

    Utilisez l'option **–c teardown** pour supprimer manuellement un serveur de la grille de données. Le retrait d'un serveur de la grille est généralement inutile. Les serveurs sont automatiquement supprimés lorsqu'ils sont déclarés comme étant défaillants, mais la commande est fournie pour être utilisée sous le contrôle du support IBM. Voir «Arrêt propre des serveurs avec l'utilitaire **xscmd**», à la page 508 pour plus d'informations sur l'utilisation de cette commande.

    ```
    xscmd –c teardown server1,server2,server3 -cep cathost:2809 –g Grid
    ```

  - **Affichage de la table de routage :**

    Utilisez l'option **-c routetable** pour afficher la table de routage en cours en simulant une nouvelle connexion client à la grille de données. Elle valide également la table de routage en confirmant que tous les serveurs de conteneur reconnaissent bien leur rôle dans la table (par exemple, quel type de fragment pour quelle partition).

    ```
    xscmd -c routetable -cep cathost:2809 –g myGrid
    ```

  - **Vérification des tailles de mappes :**

Utilisez l'option **-c showMapSizes** pour vérifier que la distribution des clés est uniforme sur les fragments dans la clé. Si certains serveurs de conteneur ont plusieurs clés que d'autres, il est probable que la distribution de la fonction de hachage sur les objets key est incorrecte.

```
xscmd -c showMapSizes -cep cathost:2809 -g myGrid -ms myMapSet
```

– **Définition des chaînes de trace :**

Utilisez l'option **-c setTraceSpec** pour définir les paramètres de trace de toutes les machines virtuelles Java qui correspondent au filtre spécifié pour la commande **xscmd**. Ce paramètre modifie les paramètres de trace uniquement, jusqu'à ce qu'une autre commande soit utilisée ou que les machines virtuelles modifiées soient défaillantes ou s'arrêtent.

```
xscmd -c setTraceSpec -spec ObjectGrid*=event=enabled -cep cathost:1099
–g myGrid –hf host1
```

Cette chaîne permet de tracer toutes les machines virtuelles Java sur le serveur avec le nom d'hôte indiqué, `host1`, en l'occurrence.

– **Affichage des fragments non affectés :**

Utilisez l'option **-c showPlacement -sf U** pour afficher la liste des fragments qui ne peuvent pas être placés sur la grille de données. Les fragments ne peuvent pas être placés lorsque le service de placement a une contrainte qui empêche le placement. Par exemple, si vous démarrez des machines virtuelles sur un serveur physique unique en mode de production, seuls seront placés les fragments primaires. Les répliques ne sont pas affectées jusqu'à ce que les machines JVM démarrent sur un second serveur physique. Le service de placement place les répliques uniquement sur les machines JVM avec des adresses IP différentes de celles qui hébergent les fragments primaires. L'absence de machines virtuelles Java dans une zone peut également provoquer la non-attribution de fragments.

```
xscmd -c showPlacement -sf U  -cep cathost:2809 –g myGrid
```

Demande, affichage et invalidation des données

Vous pouvez utiliser les interfaces de requête dans la console de surveillance et dans l'utilitaire **xscmd** pour extraire de petits ensembles de clés et de valeurs à partir d'une mappe et invalider des ensembles de données.

Avant de commencer

- Si vous utilisez la console Web pour interroger afficher et invalider des données, configurez au préalable la console de surveillance. Pour plus d'informations, voir «Surveillance à l'aide de la console Web», à la page 547.
- Si vous utilisez **xscmd** pour interroger afficher et invalider des données, configurez l'utilitaire **xscmd**. Pour plus d'informations, voir «Administration avec l'utilitaire **xscmd**», à la page 515.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser la console ou l'utilitaire **xscmd** pour interroger le contenu d'une grille de données. Vous pouvez interroger les données en exécutant une expression régulière sur la clé de données. Vous pouvez ensuite utiliser la même requête pour invalider les données. Pour des exemples d'expressions régulières voir Syntaxe d'expression régulière.

Procédure

- Demandez affichez ou invalidez des données avec la console.

1. Accédez à la page de requête dans la console. Dans la console Web, cliquez sur **Gestion** > **Contenu de la grille de données de requête**. Choisissez la mappe sur laquelle vous souhaitez effectuer le filtrage.
2. Recherchez ou filtrez les données dans la mappe. Vous pouvez utiliser l'une des options suivantes pour rechercher ou filtrer les données :
   - Entrez une expression régulière et cliquez sur le bouton **Rechercher** (). Une liste de clés correspondant à l'expression régulière s'affiche. La liste des données peut être un sous-ensemble de toutes les données correspondantes.
   - Pour filtrer les résultats sur un jeu de partitions, cliquez sur le bouton **Filtrer** (). Vous pouvez ensuite entrer une expression régulière et choisir une plage de partitions pour laquelle vous souhaitez filtrer les résultats.
3. 8.6+ Affichez les valeurs des clés affichées. Sélectionnez **Afficher les valeurs**. Les valeurs s'affichent dans le tableau. Si la valeur est trop longue à afficher, des points de suspension (**...**) tronquent la valeur. Cliquez sur la valeur pour afficher toute la zone. Les valeurs sont renvoyées sous forme de chaînes de texte. Certaines valeurs peuvent ne pas être converties en chaînes explicites et des valeurs hexadécimales s'affichent.

   Important : L'application peut stocker des valeurs d'objet pour lesquelles la classe Java n'est pas reconnue par le serveur. Si l'application utilise eXtreme Data Format (XDF), ces valeurs s'affichent. Si XDF n'est pas utilisé et que la classe Java n'est pas reconnue par le serveur, un message indique que la classe de l'objet n'est pas disponible pour le serveur.
4. Invalidation des données. Lorsque vous invalidez les données, les données sont supprimées de façon définitive de la grille de données.

   Clés sélectionnées
   Vous pouvez sélectionner dans la table les clés à invalider. Vous pouvez alors cliquer individuellement sur les entrées ou cocher la case Tout sélectionner qui permet de sélectionner un maximum de 500 entrées dans la table. Une fois les entrées à supprimer sélectionnées, cliquez sur **Invalider** > **Clés sélectionnées**.

   Toutes les clés correspondant à la requête
   Vous pouvez aussi invalider toutes les données qui correspondent à votre expression régulière. Cette option permet de supprimer toutes les données de la grille de données qui correspondent à l'expression régulière, pas simplement les 500 entrées maximum qui sont affichées dans la console. Pour invalider les entrées associées à l'expression régulière sélectionnée, cliquez sur **Invalider** > **Toutes les clés correspondant à la requête**.
5. 8.6+ Supprimez tout le contenu de la mappe. Cliquez sur **Effacer la mappe**. Vous devez confirmer la suppression de toutes les entrées dans la mappe sélectionnée.

- Demandez affichez ou invalidez des données avec l'utilitaire **xscmd**.

  Interrogation de données :

  ```
  xscmd.sh –c findbykey –g <grille_données> -m <mappe>
  -fs <chaîne_recherche> [-fp <id_partition>]
  ```

  Vous devez inclure la grille de données, la mappe et l'expression régulière pour la valeur de la chaîne de recherche. Vous pouvez

également appliquer un filtrage par ID partition. Le résultat renvoie un sous-ensemble de la totalité de la requête.

Invalidation de données :

Incluez l'argument **-inv** dans la commande pour invalider les données sélectionnées par la requête.

```
xscmd –c findbykey –g <grille_données> -m <mappe>
-fs <chaîne_recherche> [-fp <id_partition>] -inv
```

Vous devez inclure la grille de données, la mappe et l'expression régulière pour la valeur de la chaîne de recherche. Vous pouvez également appliquer un filtre par ID partition. Lors de l'exécution de l'invalidation, toutes les valeurs correspondantes sont invalidées, et pas seulement le petit ensemble renvoyé par la requête.

8.6+ Affichez les valeurs des données demandées :

Incluez l'argument **-rv** dans la commande pour afficher les valeurs des données sélectionnées par la requête.

```
xscmd.sh –c findbykey –g <grille_données> -m <mappe>
-fs <find_string> -rv
```

Vous devez inclure la grille de données, la mappe et l'expression régulière pour la valeur de la chaîne de recherche. Vous pouvez également appliquer un filtre par ID partition. Le résultat renvoie un sous-ensemble de la totalité de la requête et inclut les valeurs de chaque clé.

Important : UNIX Linux Si votre expression régulière commence par les caractères `.*`, il se peut que ces caractères ne soient pas traités correctement lors de l'exécution de la commande. Pour résoudre ce problème, mettez en forme votre expression régulière d'une des manières suivantes :

- Placez votre expression régulière entre des apostrophes : `-fs '.*'`
- Utilisez une barre oblique inversée comme caractère d'échappement pour l'astérisque : `-fs .\*`

Exemple :

L'exemple suivant recherche toutes les entrées de la grille de données `Grid` et de la mappe `Map1`.

```
xscmd -c findbykey -g Grid -m Map1 -fs ".*"
```

La commande renvoie les résultats suivants :

```
3 matching keys were found.

Partition Key
--------- ---
2         keyghi
4         keydef
6         keyabc
```

Extraction des informations d'environnement eXtreme Scale avec l'utilitaire `xscmd`

Vous pouvez utiliser l'utilitaire **xscmd** avec la commande **-c showinfo** pour afficher des détails importants concernant les serveurs qui s'exécutent dans l'environnement WebSphere eXtreme Scale, notamment les serveurs WebSphere eXtreme Scale, machines virtuelles Java, et (le cas échéant) les serveurs qui

s'exécutent avec WebSphere Application Server. Exécutez cette commande pour extraire les informations de version et de nom, le nom d'hôte et l'adresse IP, ainsi que les répertoires d'installation de ces serveurs. La commande **-c showinfo** vous permet d'extraire ces détails sans avoir à vérifier les fichiers journaux et les répertoires, ni à utiliser des applications de tierce partie .

Procédure

- Assurez-vous qu'au moins un des serveurs de catalogue est en cours d'exécution. Si vous souhaitez extraire des détails sur l'environnement pour tout le domaine eXtreme Scale, assurez-vous que tous les serveurs sont actifs.

  Pour extraire des informations sur l'environnement pour tout le domaine eXtreme Scale, exécutez la commande suivante : Windows

  ```
  xscmd.bat –c showinfo
  ```

  UNIX

  ```
  ./xscmd.sh –c showinfo
  ```

  La commande renvoie toutes les informations relatives aux serveurs qui s'exécutent dans votre environnement.
- Pour extraire des informations sur un serveur particulier, utilisez le paramètre **-s** et indiquez le nom du serveur.

  Windows

  ```
  xscmd.bat –c showinfo –s <nom_serveur>
  ```

  UNIX

  ```
  ./xscmd.sh –c showinfo –s <nom_serveur>
  ```
- Pour afficher la liste des serveurs, utilisez le paramètre **-sl**.

  Windows

  ```
  xscmd.bat –c showinfo –sl  <nom_serveur>[,<nom_serveur>]
  ```

  UNIX

  ```
  ./xscmd.sh –c showinfo –sl  <nom_serveur>[,<nom_serveur>]
  ```
- Pour extraire des informations d'environnement concernant un ensemble spécifique de serveurs qui s'exécutent sur un hôte donné, utilisez le paramètre **-hf** et indiquez le nom de l'hôte. Windows

  ```
  xscmd.bat –c showinfo –hf <nom_hôte>
  ```

  UNIX

  ```
  ./xscmd.sh –c showinfo –hf <nom_hôte>
  ```

Démarrage des serveurs eXtreme Scale en utilisant l'infrastructure OSGi Eclipse Equinox

Les serveurs de conteneur WebSphere eXtreme Scale peuvent être démarrés dans une infrastructure OSGi Eclipse Equinox en utilisant plusieurs méthodes.

Avant de commencer

Pour pouvoir démarrer un conteneur eXtreme Scale, vous devez exécuter les tâches suivantes :

1. L'ensemble de serveur WebSphere eXtreme Scale doit être installé dans Eclipse Equinox.
2. L'application doit être placée dans un ensemble OSGi.
3. Les plug-in WebSphere eXtreme Scale (s'il en existe) doivent être placés dans un ensemble OSGi. Ils peuvent se trouver dans le même ensemble que l'application ou dans des ensembles séparés.
4. Si les serveurs de conteneur utilisent IBM eXtremeMemory, vous devez d'abord configurer les bibliothèques natives. Pour plus d'informations, voir «Configuration d'IBM eXtremeMemory», à la page 357.

Pourquoi et quand exécuter cette tâche

Cette tâche explique comment démarrer un serveur de conteneur eXtreme Scale dans une infrastructure OSGi Eclipse Equinox. Vous pouvez utiliser n'importe laquelle des méthodes suivantes pour démarrer les serveurs de conteneur en utilisant l'implémentation Eclipse Equinox :

- Service OSGi Blueprint

  Vous pouvez inclure toute la configuration et toutes les métadonnées dans un ensemble OSGi. Voir l'illustration suivante pour comprendre le processus Eclipse Equinox de cette méthode :

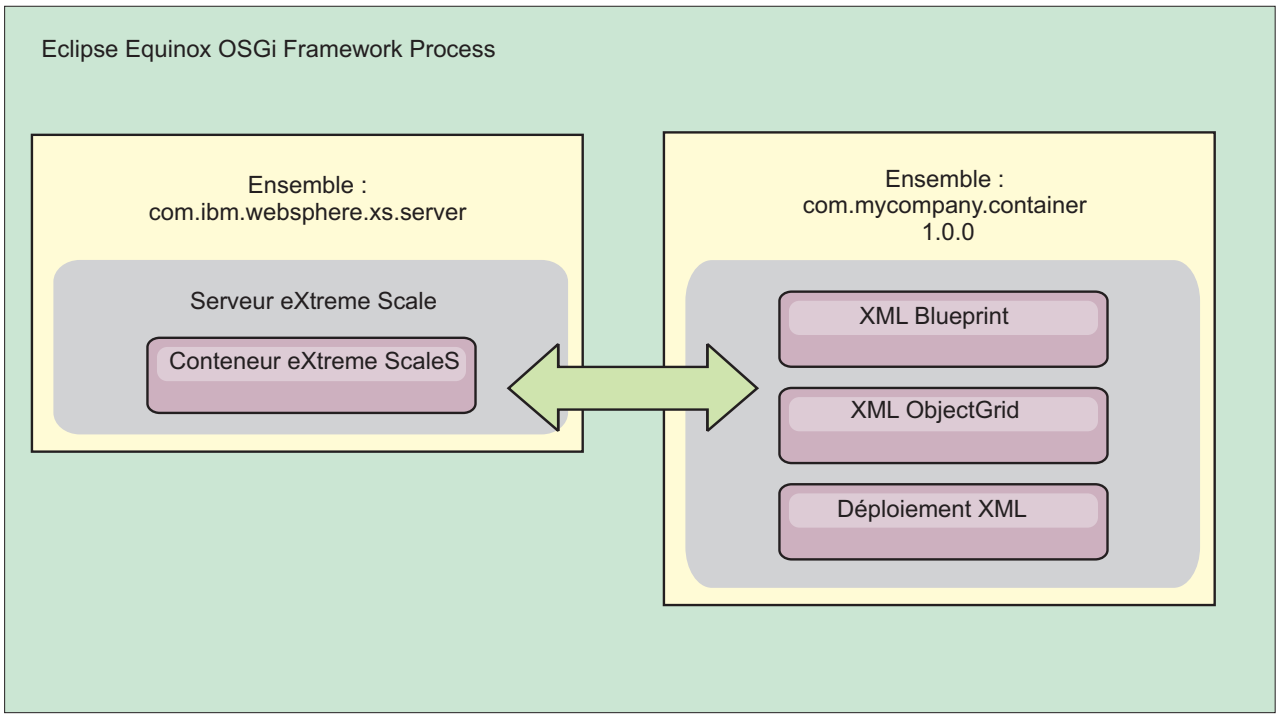


Figure 54. Processus Eclipse Equinox pour inclure toute la configuration et toutes les métadonnées dans un ensemble OSGi

- Service Admin de configuration OSGi

  Vous pouvez définir la configuration et les métadonnées en dehors d'un ensemble OSGi. Voir l'image suivante pour comprendre le processus Eclipse Equinox pour cette méthode :

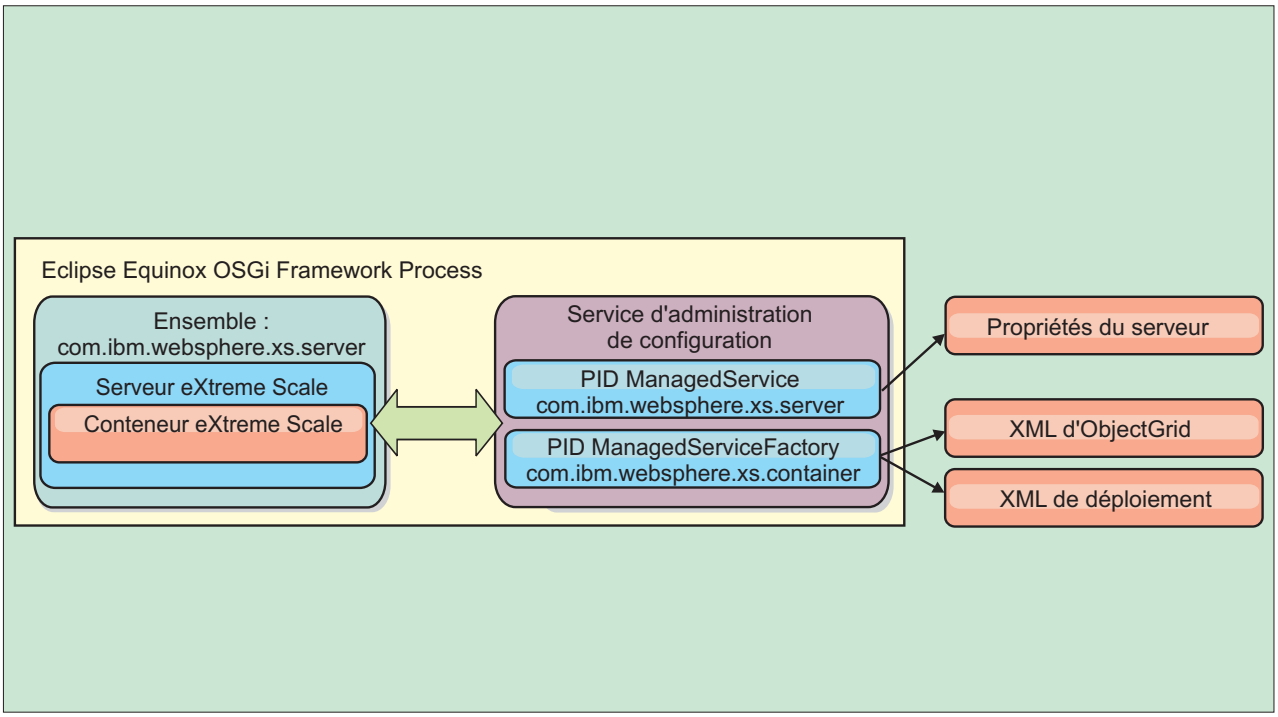


Figure 55. Processus Eclipse Equinox pour définir la configuration et les métadonnées en dehors d'un ensemble OSGi

- A l'aide d'un programme

  Prend en charge les solutions de configuration personnalisées.

Dans chaque cas, un singleton de serveur eXtreme Scale est configuré et un ou plusieurs conteneurs sont configurés.

L'ensemble de serveur eXtreme Scale, `objectgrid.jar`, contient toutes les bibliothèques nécessaires pour démarrer et exécuter un conteneur de grille eXtreme Scale dans une infrastructure OSGi. L'environnement d'exécution du serveur communique avec les plug-in fournis par l'utilisateur et les objets de données en utilisant le gestionnaire de service OSGi.

Important : Après que l'ensemble de serveur eXtreme Scale a été démarré et le serveur eXtreme Scale initialisé, il ne peut pas être redémarré. Le processus Eclipse Equinox doit être redémarré pour redémarrer le serveur eXtreme Scale.

Vous pouvez utiliser le support eXtreme Scale pour l'espace de nom Spring pour configurer les serveurs de conteneur eXtreme Scale dans un fichier XML Blueprint. Lorsque les éléments XML de serveur et de conteneur sont ajoutés au fichier XML Blueprint, le gestionnaire d'espace de nom eXtreme Scale démarre automatiquement un serveur de conteneur en utilisant les paramètres définis dans le fichier XML Blueprint lors du démarrage de l'ensemble. Le gestionnaire arrête le conteneur lorsque l'ensemble s'arrête.

Pour configurer les serveurs de conteneur eXtreme Scale avec XML Blueprint, procédez comme suit :

Procédure

- Démarrez un serveur de conteneur eXtreme Scale en utilisant OSGi Blueprint.
  1. Créez un ensemble de conteneur.

2. Installez l'ensemble de conteneur dans l'infrastructure OSGi Eclipse Equinox. Voir «Installation et démarrage des plug-in OSGi».
3. Démarrez l'ensemble de conteneur.

- Démarrez un serveur de conteneur eXtreme Scale en utilisant l'administrateur de configuration OSGi.
  1. Configurez le serveur et le conteneur en utilisant l'administrateur de configuration.
  2. Lorsque l'ensemble de serveur eXtreme Scale est démarré ou que les PID (persistent identifier) sont créé avec l'administrateur de configuration, le serveur et le conteneur démarrent automatiquement.
- Démarrez un serveur de conteneur eXtreme Scale en utilisant l'API ServerFactory. Voir la documentation d'API de serveur.
  1. Créez une classe d'activateur d'ensemble OSGi et utilisez l'API eXtreme Scale ServerFactory pour démarrer un serveur.

Installation et démarrage des plug-in OSGi

Dans cette tâches, vous installez l'ensemble de plug-in dynamique dans l'infrastructure OSGi, puis vous démarrez le plug-in.

Avant de commencer

Cette rubrique suppose que vous avez exécuté les tâches suivantes :

- Vous avez installé l'ensemble serveur ou client eXtreme Scale dans l'infrastructure OSGi Eclipse Equinox. Voir «Installation des ensembles eXtreme Scale», à la page 224.
- Vous avez implémenté un ou plusieurs plug-in dynamiques BackingMap ou ObjectGrid. Voir Génération de plug-in dynamiques eXtreme Scale.
- Vous avez regroupé les plug-in dynamiques comme services OSGi dans des ensembles OSGi.

Pourquoi et quand exécuter cette tâche

Cette tâche explique comment installer l'ensemble en utilisant la console Eclipse Equinox. L'ensemble peut être installé en utilisant plusieurs méthodes différentes, y compris en modifiant le fichier de configuration `config.ini`. Les produits qui intègrent Eclipse Equinox incluent des méthodes alternatives de gestion des ensembles. Pour plus d'informations sur l'ajout d'ensembles dans le fichier `config.ini` dans Eclipse Equinox, voir les options d'exécution Eclipse.

OSGi permet de démarrer les ensembles ayant des services dupliqués. WebSphere eXtreme Scale utilise le dernier classement de service. Lors du démarrage de plusieurs infrastructures OSGi dans une grille de données eXtreme Scale, vous devez veiller à démarrer les classements de service corrects sur chaque serveur afin que la grille ne soit pas démarrée en utilisant une combinaison de versions différentes.

Pour identifier les versions utilisées par la grille de données, utilisez l'utilitaire xscmd pour vérifier les classements en cours et disponibles. Pour plus d'informations sur les classements de service disponibles, voir «Mise à jour des services OSGi pour les plug-in eXtreme Scale avec **xscmd**», à la page 535.

Procédure

Installez l'ensemble de plug-in dans l'infrastructure OSGi Eclipse Equinox en utilisant la console OSGi.

1. Démarrez l'infrastructure Eclipse Equinox avec la console activée, par exemple :
   ```
   <java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
2. Installez l'ensemble de plug-in dans la console Equinox.
   ```
   osgi> install file:///<path to bundle>
   ```
   Equinox affiche l'ID du nouvel ensemble installé :
   ```
   Bundle id is 17
   ```
3. Entrez la ligne suivante pour démarrer l'ensemble dans la console Equinox, où `<id>` est l'ID d'ensemble affecté lors de l'installation de l'ensemble :
   ```
   osgi>  start <id>
   ```
4. Extrayez l'état du service dans la console Equinox pour vérifier que l'ensemble a démarré :
   ```
   osgi> ss
   ```
   Lorsque l'ensemble a démarré correctement, il affiche l'état ACTIVE, par exemple :
   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Installez l'ensemble de plug-in dans l'infrastructure OSGi Eclipse Equonix en utilisant le fichier config.ini file.

5. Copiez l'ensemble de plug-in dans le répertoire Eclipse Equinox plug-in, par exemple :
   ```
   <equinox_root>/plugins
   ```
6. Modifiez le fichier de configuration Eclipse Equinox `config.ini` et ajoutez l'ensemble à la propriété osgi.bundles, par exemple :
   ```
   osgi.bundles=\
   org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
   org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
   org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
   com.mycompany.plugin.bundle_VRM.jar@1:start
   ```
   Important : Vérifiez qu'il existe une ligne blanche après le dernier nom d'ensemble. Chaque ensemble est séparé par une virgule.
7. Démarrez l'infrastructure Eclipse Equinox avec la console activée, par exemple :
   ```
   <java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
8. Extrayez l'état de service dans la console Equinox pour vérifier que l'ensemble est démarré. Par exemple :
   ```
   osgi> ss
   ```
   Une fois l'ensemble démarré, il affiche l'état ACTIVE. Par exemple :
   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Résultats

L'ensemble de plug-in est maintenant installé et démarré. Le conteneur ou le client peut être maintenant démarré eXtreme Scale. Pour plus d'informations sur le développement des plug-in eXtreme Scale, voir la rubrique API système et plug-in.

Administration des services OSGi en utilisant l'utilitaire xscmd

Vous pouvez utiliser l'utilitaire **xscmd** pour exécuter des tâches d'administration, telles qu'afficher les serveurs et leurs classements utilisés par chaque conteneur, et mettre à niveau l'environnement d'exécution pour utiliser les nouvelles versions des ensembles.

Pourquoi et quand exécuter cette tâche

Avec l'infrastructure Eclipse Equinox OSGi, vous pouvez installer plusieurs versions d'un même ensemble et vous pouvez mettre à jour ces ensemble lors de l'exécution. WebSphere eXtreme Scale est un environnement distribué qui exécute les serveurs de conteneur dans une multitude d'instances de l'infrastructure OSGi.

Les administrateurs doivent copier, installer et démarrer manuellement les ensembles dans l'infrastructure OSGi. eXtreme Scale contient un personnaliseur ServiceTrackerCustomizer OSGi pour suivre les services identifiés comme plug-in eXtreme Scale dans le fichier XML descripteur. Utilisez l'utilitaire **xscmd** pour valider la version utilisée du plug-in, les versions pouvant être utilisées et exécuter des mises à niveau d'ensemble.

eXtreme Scale utilise le numéro de classement de service pour identifier la version de chaque service. Lorsque au moins deux services sont chargés avec la même référence, eXtreme Scale utilise automatiquement le service ayant le classement le plus élevé.

Procédure

- Exécutez la commande **osgiCurrent** et vérifiez que chaque serveur eXtreme Scale utilise le classement de service de plug-in correct.

  Comme eXtreme Scale choisit automatiquement la référence de service ayant le classement le plus élevé, il se peut que la grille de données démarre avec plusieurs classements d'un service de plug-in.

  Si la commande détecte une discordance de classements ou qu'elle ne trouve pas un service, un niveau d'erreur différent de zéro est défini. Si la commande aboutit, le niveau d'erreur 0 est défini.

  L'exemple suivant montre la sortie de la commande **osgiCurrent** lorsque deux plus-ins sont installés dans une grille sur quatre serveurs. Le plug-in loaderPlugin utilise le classement 1 et le plug-in txCallbackPlugin, le classement 2.

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- --------------- --------------- ----------- -----------
loaderPlugin      1               MyGrid          MapSetA     server1
loaderPlugin      1               MyGrid          MapSetA     server2
loaderPlugin      1               MyGrid          MapSetA     server3
loaderPlugin      1               MyGrid          MapSetA     server4
txCallbackPlugin  2               MyGrid          MapSetA     server1
txCallbackPlugin  2               MyGrid          MapSetA     server2
txCallbackPlugin  2               MyGrid          MapSetA     server3
txCallbackPlugin  2               MyGrid          MapSetA     server4
```

  L'exemple suivant montre la sortie de la commande **osgiCurrent** lorsque le serveur 2 a été démarré avec un nouveau classement du plug-in loaderPlugin :

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- --------------- --------------- ----------- -----------
loaderPlugin      1               MyGrid          MapSetA     server1
loaderPlugin      2               MyGrid          MapSetA     server2
loaderPlugin      1               MyGrid          MapSetA     server3
loaderPlugin      1               MyGrid          MapSetA     server4
```

```
txCallbackPlugin  2              MyGrid          MapSetA     server1
txCallbackPlugin  2              MyGrid          MapSetA     server2
txCallbackPlugin  2              MyGrid          MapSetA     server3
txCallbackPlugin  2              MyGrid          MapSetA     server4
```

- Exécutez la commande **osgiAll** pour vérifier que les services de plug-in ont été correctement démarrés sur chaque serveur de conteneur eXtreme Scale.

  Lorsque des ensembles contenant des services référencés par une configuration ObjectGrid démarrent, l'environnement d'exécution eXtreme Scale suit le plug-in, mais il ne l'utilise pas immédiatement. La commande **osgiAll** montre les plug-in disponibles pour chaque serveur.

  Lorsqu'elle est exécutée sans paramètres, tous les services de toutes les grilles et de tous les serveurs sont indiqués. Des filtres supplémentaires, notamment le filtre **-serviceName** <service_name>, peuvent être définis pour limiter la sortie à un seul service ou sous-ensemble de la grille de données.

  L'exemple suivant montre la sortie de la commande **osgiAll** lorsque deux plug-in sont démarrés sur deux serveurs. Les classements 1 et 2 du plug-in loaderPlugin sont démarrés et le classement 1 du plug-in txCallbackPlugin est démarré. Le résumé à la fin de la sortie indique que les deux serveurs voient les mêmes classements de service :

```
Server: server1
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1

Server: server2
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1

Summary - All servers have the same service rankings.
```

  L'exemple suivant montre la sortie de la commande **osgiAll** lorsque l'ensemble qui contient le plug-in loaderPlugin avec le classement 1 est arrêté sur le serveur 1. Le résumé à la fin de la sortie indique que le serveur n'a pas le plug-in loaderPlugin avec le classement 1 :

```
Server: server1
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        2
   txCallbackPlugin    1

Server: server2
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1

Summary - The following servers are missing service rankings:
   Server  OSGi Service Name Missing Rankings
   ------  ----------------- ----------------
   server1 loaderPlugin      1
```

  L'exemple suivant montre la sortie si le nom de service est défini avec l'argument **-sn** et que le service n'existe pas.

```
Server: server2
   OSGi Service Name Available Rankings
   ----------------- ------------------
   invalidPlugin     No service found

Server: server1
```

```
OSGi Service Name Available Rankings
---------------- ------------------
invalidPlugin     No service found

Summary - All servers have the same service rankings.
```

- Exécutez la commande **osgiCheck** pour vérifier les groupes de services de plug-in et de classements s'ils sont disponibles.

  La commande **osgiCheck** accepte un ou plusieurs groupes de classements de service de la manière suivante `-serviceRankings <service name>;<ranking>[,<serviceName>;<ranking>]`

  Lorsque les classements sont tous disponibles, la méthode retourne un niveau d'erreur 0. Si un ou plusieurs classements sont indisponibles, un niveau d'erreur différent de zéro est défini. Une table de tous les serveurs qui ne contiennent pas les classements de service définis s'affiche. Des filtres supplémentaires peuvent être utilisés pour limiter la vérification des services à un sous-ensemble des serveurs disponibles dans le domaine eXtreme Scale.

  Par exemple, si le classement ou le service est absent, le message suivant s'affiche :

```
Server  OSGi Service Unavailable Rankings
------  ------------ --------------------
server1 loaderPlugin 3
server2 loaderPlugin 3
```

- Exécutez la commande **osgiUpdate** pour mettre à jour le classement d'un ou de plusieurs plug-in pour tous les serveurs dans un seul ObjectGrid et MapSet dans une seule opération.

  La commande accepte un ou plusieurs groupes de classements de service de la manière suivante : `-serviceRankings <service name>;<ranking>[,<serviceName>;<ranking>] -g <grid name> -ms <mapset name>`

  Avec cette commande, vous pouvez exécuter les opérations suivantes :

  – Vérifier que les services spécifiés sont disponibles pour la mise à niveau sur chacun des serveurs.
  – Mettre la grille hors ligne en utilisant l'interface StateManager. Pour plus d'informations, voir «Gestion de la disponibilité ObjectGrid», à la page 519. Ce processus met au repos la grille et attend la fin des transactions en cours en interdisant le démarrage de nouvelles transactions. Ce processus indique également aux programmes d'écoute ObjectGridLifecycleListener et BackingMapLifecycleListener d'arrêter toute activité transactionnelle. Voir Plug-in de programme d'écoute d'événement pour plus d'informations sur les plug-in de programme d'écoute.
  – Mettre à jour chaque conteneur eXtreme Scale exécuté dans une infrastructure OSGi pour utiliser les nouvelles versions de service.
  – Mettre la grille en ligne pour reprendre l'exécution des transactions.

  Le processus de mise à jour est idempotent de sortie que si un client n'exécute pas une tâche, l'opération est annulée. Si un client ne peut pas exécuter l'annulation ou qu'il est interrompu pendant la mise à jour, la même commande peut être réexécutée et elle reprend à l'étape appropriée.

  Si le client ne peut pas continuer et que le processus est redémarré depuis un autre client, utilisez l'option `-force` pour permettre au client d'exécuter la mise à jour. La commande **osgiUpdate** empêche plusieurs clients de mettre à jour simultanément un même groupe de mappes. Pour plus d'informations sur la commande **osgiUpdate**, voir «Mise à jour des services OSGi pour les plug-in eXtreme Scale avec **xscmd**», à la page 535.

Mise à jour des services OSGi pour les plug-in eXtreme Scale avec `xscmd`

WebSphere eXtreme Scale prend en charge la mise à niveau des ensembles de plug-in de serveur de conteneur lorsque la grille est active. Ainsi, les administrateurs peuvent mettre à jour les applications et effectuer des ajouts sans avoir à démarrer les processus de la grille.

Avant de commencer

Procédez comme suit avant de mettre à jour les ensembles eXtreme Scale OSGi vers une nouvelle version :

1. Démarrez les serveurs eXtreme Scale dans une infrastructure OSGi compatible.
2. Divisez tous les plug-in eXtreme Scale en ensembles ; ils doivent utiliser les classements de service pour identifier chaque version des plug-in.
3. Définissez les objets cache comme types primitifs Java, tels que `byte[]`, Integer ou String ou bien ils doivent être stockés en utilisant un plug-in MapSerializerPlugin. Les objets données sont stockés dans l'ensemble eXtreme Scale et ne sont pas mis à niveau. Seuls les plug-in qui interagissent avec les données sont mis à jour.
4. Créez des données d'objet cache compatibles avec la version. Les nouveaux plug-in doivent pouvoir interagir avec les données créées par les anciens plug-in.
5. Créez des plug-in pour écouter les événements ObjectGridLifecycle et BackingMapLifecycle et pour régénérer les références aux autres plug-in ou métadonnées dans ces plug-in. Ainsi, les plug-in référencés sont régénérés lorsque le plug-in principal est mis à jour.
6. Le processus de mise à jour OSGi eXtreme Scale affecte uniquement les serveurs. Vous devez mettre à jour de manière indépendante les clients qui utilisent les plug-in.

Pourquoi et quand exécuter cette tâche

Sans l'activation d'OSGi, si un administrateur doit mettre à jour les plug-in d'application ou les objets cache, chaque noeud de grille doit être mis à jour un par un, ce qui affecte le réseau, la mémoire et l'utilisation du processeur. Cette opération est nécessaire, car les plug-in et les objets Java cache sont directement stockés dans la grille. Lorsque les classes sont mises à jour sans redémarrer les processus, les plug-in de grille génèrent des conflits, car chaque classe un chargeur ClassLoader différent.

Le produit eXtreme Scale contient l'utilitaire xscmd et des beans gérés qui permettent aux administrateurs d'afficher tous les ensembles de plug-in installés dans l'infrastructure OSGi d'hébergement de chaque conteneur de la grille et de choisir la révision à utiliser. Lorsque vous utilisez l'utilitaire xscmd pour mettre à jour les plug-in vers un nouveau classement, la grille est mise au repos et toutes les transactions sont arrêtées, les plug-in sont mis à jour et la grille est réactivée. En cas d'erreur lors de la mise à jour, le processus est annulé et l'ancien classement est restauré.

Procédure

1. Créez une version de l'ensemble en augmentant le numéro de version dans le manifeste de l'ensemble et le classement de chaque service de plug-in eXtreme

Scale. Si la version de l'ensemble d'origine est `Bundle-Version: 1.0.0`, la version suivante peut être `Bundle-Version: 1.1.0`.

Si le classement de service d'origine est `ranking="1"`, le classement suivant peut être `ranking="2"`.

Important : Les classements de service OSGi doivent être des entiers.

2. Copiez le nouvel ensemble vers chaque noeud de l'infrastructure OSGi qui héberge un serveur de conteneur eXtreme Scale.
3. Installez le nouvel ensemble dans l'infrastructure OSGi. L'ensemble est affecté d'un identificateur, par exemple :

   ```
   osgi> install <URL to bundle>
   ```

4. Démarrez le nouvel ensemble en utilisant l'identificateur affecté, par exemple :

   ```
   osgi> start <id>
   ```

   Une fois le nouvel ensemble démarré, le programme de suivi de service OSGi eXtreme Scale détecte l'ensemble et le rend disponible pour la mise à jour.
5. Utilisez la commande **xscmd -c osgiAll** pour vérifier que chaque serveur de conteneur voit le nouvel ensemble. La commande **osgiAll** interroge tous les conteneurs dans la grille pour tous les services qui sont référencés dans le fichier XML descripteur ObjectGrid et affiche tous les classements disponibles, par exemple :

   ```
   xscmd -c osgiAll

   Server: server1
      OSGi Service Name          Available Rankings
      -----------------          ------------------
      myLoaderServiceFactory     1, 2
      mySerializerServiceFactory 1, 2

   Server: server2
      OSGi Service Name          Available Rankings
      -----------------          ------------------
      myLoaderServiceFactory     1, 2
      mySerializerServiceFactory 1, 2

   Summary - All servers have the same service rankings.
   ```

6. Utilisez la commande **xscmd -c osgiCheck** pour vérifier qu'un ou plusieurs classements de service sont des cibles de mise à jour valides. Par exemple :

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```

7. Si la commande **osgiCheck** ne trouve pas des erreurs résultantes, suspendez l'équilibreur du service de placement pour éviter les mouvements de fragments en cas d'erreur lors de la mise à jour. Pour suspendre le placement, utilisez la commande **xscmd -c suspendBalancing** pour chaque grille d'objets et chaque groupe de mappes affectés par la mise à jour, par exemple :

   ```
   xscmd -c suspendBalancing -g MyGrid -ms MyMapSet
   ```

8. Lorsque l'équilibrage a été suspendu pour chaque grille d'objets et groupe de mappes, utilisez la commande **xscmd -c osgiCheck** de nouveau pour vérifier qu'un ou plusieurs classements de service sont des cibles de mise à jour valides. Par exemple :

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```

9. Lorsque l'équilibrage a été suspendu pour chaque grille d'objets et groupe de mappes, utilisez la commande **osgiUpdate** pour mettre à jour le service sur tous les serveurs pour une grille d'objets et un groupe de mappes. Par exemple :

   ```
   xscmd -c osgiUpdate  -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2 -g MyGrid -ms MyMapSet
   ```

10. Vérifiez que la mise à niveau a abouti. Par exemple :

    ```
    La mise à jour abouti pour les classements de services suivants :
    Service                    Ranking
    -------                    -------
    mySerializerServiceFactory  2
    myLoaderServiceFactory      2
    ```

11. Après avoir vérifié que le classement a été mis à jour, activez de nouveau l'équilibrage en utilisant la commande **xscmd -c resumeBalancing**. Par exemple, :

    ```
    xscmd -c resumeBalancing -g MyGrid -ms MyMapSet
    ```

12. Arrêtez et désinstallez l'ancien ensemble dans chaque infrastructure OSGi qui héberge le conteneur eXtreme Scale. Par exemple, entrez le code suivant dans la console Eclipse Equinox :

    ```
    osgi> stop <id>
    osgi> uninstall <id>
    ```

Résultats

L'ensemble eXtreme Scale a été mis à jour vers une nouvelle version.

Administration avec les beans gérés (MBeans)

Java

Vous pouvez utiliser plusieurs types de beans gérés JMX (Java Management Extensions) différents pour administrer et surveiller les déploiements. Chaque bean géré fait référence à une entité spécifique, une mappe, une grille de données, un serveur ou un service.

Interfaces MBean JMX et WebSphere eXtreme Scale

Chaque bean géré contient des méthodes get qui représentent des valeurs d'attribut. Ces méthodes get ne peuvent pas être appelées directement à partir de votre programme. La spécification JMX traite les attributs différemment des opérations. Vous pouvez afficher les attributs à l'aide de la console JMX d'un fournisseur et effectuer des opérations dans votre programme ou à l'aide de la console JMX d'un fournisseur.

Package com.ibm.websphere.objectgrid.management

Voir la documentation d'API pour une présentation et les spécifications détaillées pour la programmation de tous les beans gérés utilisables : Package com.ibm.websphere.objectgrid.management .

Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin

Java

Vous pouvez utiliser l'utilitaire wsadmin fourni dans WebSphere Application Server pour accéder aux informations des beans gérés (MBean).

Procédure

Exécutez l'outil wsadmin depuis le répertoire `bin` dans votre installation WebSphere Application Server. L'exemple suivant restaure une vue de la position actuelle du fragment dans un logiciel eXtreme Scale dynamique. Vous pouvez exécuter l'outil wsadmin depuis n'importe quelle installation où eXtreme Scale est en cours d'exécution. Vous n'avez pas besoin d'exécuter l'outil wsadmin sur le service de catalogue.

```
$ wsadmin.sh -lang jython
wsadmin>placementService = AdminControl.queryNames
 ("com.ibm.websphere.objectgrid:*,type=PlacementService")
wsadmin>print AdminControl.invoke(placementService,
 "listObjectGridPlacement","library ms1")

<objectGrid name="library" mapSetName="ms1">
  <container name="container-0" zoneName="DefaultDomain"
  hostName="host1.company.org" serverName="server1">
     <shard type="Primary" partitionName="0"/>
     <shard type="SynchronousReplica" partitionName="1"/>
  </container>
  <container name="container-1" zoneName="DefaultDomain"
  hostName="host2.company.org" serverName="server2">
     <shard type="SynchronousReplica" partitionName="0"/>
     <shard type="Primary" partitionName="1"/>
  </container>
  <container name="UNASSIGNED" zoneName="_ibm_SYSTEM"
  hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
  </container>
</objectGrid>
```

Accès aux beans gérés (MBeans) à l'aide d'un programme

Java

Vous pouvez vous connecter aux beans gérés avec des applications Java. Ces applications utilisent les interfaces dans le package com.ibm.websphere.objectgrid.management.

Pourquoi et quand exécuter cette tâche

Les méthodes d'accès à l'aide d'un programme aux beans gérés varient en fonction du type de serveur auquel vous vous connectez.

- Connexion à un serveur MBean de service de catalogue
- Connexion à un serveur MBean de conteneur
- Connexion à un serveur MBean de service de catalogue hébergé dans WebSphere Application Server
- Connexion à un serveur Mbean de service de catalogue avec la sécurité activée

Procédure

- **Connectez-vous à un serveur MBean de service de catalogue autonome :**

  L'exemple de programme suivant se connecte à un serveur MBean de service de catalogue autonome et renvoie une chaîne formatée XML qui répertorie chaque serveur de conteneur avec ses fragments alloués pour ObjectGrid et un MapSet

donné.

```
package com.ibm.websphere.sample.xs.admin;

import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collecte les informations de placement à partir du serveur de catalogue pour un ObjectGrid donné.
 */
public final class CollectPlacementPlan {
    private static String hostName = "localhost";

    private static int port = 1099;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Se connecte au service de catalogue ObjectGrid pour extraire les informations de placement et
    * les affiche.
     *
     * @param args
     * @throws Exception
     *             If there is a problem connecting to the catalog service MBean server.
     */
    public static void main(String[] args) throws Exception {
        String serviceURL = "service:jmx:rmi:///jndi/rmi://" + hostName + ":" + port +
                    "/objectgrid/MBeanServer";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames
(new ObjectName("com.ibm.websphere.objectgrid"
             + ":*,type=PlacementService"), null);
            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            Object placementXML = catalogServerConnection.invoke(placementService,
                "listObjectGridPlacement", new Object[] {
                objectGridName, mapSetName }, new String[] { String.class.getName(),
 String.class.getName() });
            System.out.println(placementXML);
        } catch (Exception e) {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }

}
```

Figure 56. CollectPlacementPlan.java

Quelques remarques concernant l'exemple de programme :

– La valeur **JMXServiceURL** pour le service de catalogue a toujours le format suivant : `service:jmx:rmi:///jndi/rmi://<host>:<port>/objectgrid/MBeanServer`, où `<host>` est l'hôte sur lequel le service de catalogue est exécuté

et <port> est le port du service JMX fourni avec l'option **-JMXServicePort** lors du démarrage du service de catalogue. Si aucun port n'est défini, la valeur par défaut est 1099.

– Pour activer les statistiques ObjectGrid ou de mappe, vous devez définir la propriété suivantes dans le fichier des propriétés du serveur lorsque vous démarrez un conteneur ObjectGrid : statsSpec=all=enabled

– Pour désactiver les beans gérés exécutés dans les serveurs de conteneur, définissez la propriété suivante dans le fichier des propriétés du serveur : enableMBeans=false.

Exemple de sortie. Cette sortie indique que deux serveurs de conteneur sont actifs. Le serveur de conteneur Container-0 héberge quatre fragments primaires. Le serveur de conteneur Container-1 héberge une réplique synchrone pour chaque fragment primaire sur le serveur de conteneur Container-0. Dans cette configuration, deux répliques synchrones et une réplique asynchrone sont configurées. Par conséquent, le serveur de conteneur Unassigned dispose des fragments restants. Si plus de deux serveurs sont démarrés, le serveur de conteneur Unassigned n'est pas affiché.

```
<objectGrid name="library" mapSetName="ms1">
  <container name="Container-1" zoneName="DefaultZone"
    hostName="myhost.mycompany.com" serverName="ogserver">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
    <shard type="SynchronousReplica" partitionName="2"/>
    <shard type="SynchronousReplica" partitionName="3"/>
  </container>
  <container name="Container-0" zoneName="DefaultZone"
    hostName="myhost.mycompany.com" serverName="ogserver">
    <shard type="Primary" partitionName="0"/>
    <shard type="Primary" partitionName="1"/>
    <shard type="Primary" partitionName="2"/>
    <shard type="Primary" partitionName="3"/>
  </container>
  <container name="library:ms1:_UnassignedContainer_" zoneName="_ibm_SYSTEM"
    hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
    <shard type="SynchronousReplica" partitionName="2"/>
    <shard type="SynchronousReplica" partitionName="3"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="1"/>
    <shard type="AsynchronousReplica" partitionName="2"/>
    <shard type="AsynchronousReplica" partitionName="3"/>
  </container>
</objectGrid>
```

- **Connectez-vous à un serveur MBean de conteneur :**

  Les serveurs de conteneur hébergent des beans gérés pour obtenir des informations sur les mappes et les instances ObjectGrid individuelles exécutées dans le serveur de conteneur. L'exemple de programme suivant affiche l'état de chaque serveur de conteneur hébergé par le serveur de catalogue avec l'adresse JMX localhost:1099:

```
package com.ibm.websphere.sample.xs.admin;

import java.util.List;
import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectInstance;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collecte l'état du placement directement depuis les conteneurs disponibles.
 */
public final class CollectContainerStatus {
    private static String hostName = "localhost";

    private static int port = 1099;

    /**
     * @param args
     */
    public static void main(String[] args) throws Exception {
        String serviceURL = "service:jmx:rmi:///jndi/rmi://" + hostName + ":" + port + "/objectgrid/MBeanServer";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName("com.ibm.websphere.objectgrid"
                    + ":*,type=PlacementService"), null);

            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            List<String> containerJMXAddresses = (List<String>) catalogServerConnection.invoke(placementService,
                    "retrieveAllServersJMXAddresses", new Object[0], new String[0]);
            for (String address : containerJMXAddresses) {
                JMXServiceURL containerJMXURL = new JMXServiceURL(address);
                JMXConnector containerConnector = JMXConnectorFactory.connect(containerJMXURL);
                MBeanServerConnection containerConnection = containerConnector.getMBeanServerConnection();
                Set<ObjectInstance> containers = containerConnection.queryMBeans(
                        new ObjectName("*:*,type=ObjectGridContainer"), null);
                for (ObjectInstance container : containers) {
                    System.out.println(containerConnection.getAttribute(container.getObjectName(), "Status"));
                }
            }
        } finally {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }
}
```

Figure 57. CollectContainerStatus.java

L'exemple de programme affiche l'état du serveur de chaque conteneur. Ci-après, un exemple de sortie :

```
<container name="Container-0" zoneName="DefaultZone" hostName="descartes.rchland.ibm.com"
  serverName="ogserver">
  <shard type="Primary" partitionName="1"/>
  <shard type="Primary" partitionName="0"/>
  <shard type="Primary" partitionName="3"/>
  <shard type="Primary" partitionName="2"/>
</container>
```

- **Connectez-vous à un serveur MBean de serveur de catalogue hébergé dans WebSphere Application Server :**

  La méthode d'accès à l'aide d'un programme aux beans gérés dans MBeans WebSphere Application Server diffère légèrement de la méthode d'accès aux beans gérés dans une configuration autonome.

1. Créez et compilez un programme Java pour vous connecter au serveur MBean. Exemple de programme :

```
package com.ibm.websphere.sample.xs.admin;

import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collecte les informations de placement à partir du serveur de catalogue exécuté
dans un gestionnaire de déploiement pour un ObjectGrid donné.
 */
public final class CollectPlacementPlanWAS {
    private static String hostName = "localhost";

    private static int port = 9809;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Connects to the catalog service to retrieve placement information and prints it out.
     *
     * @param args
     * @throws Exception
     *             If there is a problem connecting to the catalog service MBean server.
     */
    public static void main(String[] args) throws Exception {

        // connect to bootstrap port of the deployment manager
        String serviceURL = "service:jmx:iiop://" + hostName + ":" + port + "/jndi/JMXConnector";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName("com.ibm.websphere.objectgrid"
             + ":*,type=PlacementService"), null);

            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            Object placementXML = catalogServerConnection.invoke(placementService,
                        "listObjectGridPlacement", new Object[] {
                    objectGridName, mapSetName }, new String[] { String.class.getName(), String.class.getName() });
            System.out.println(placementXML);
        } finally {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }

}
```

Figure 58. CollectPlacementPlan.java

2. Exécutez la commande suivante.

```
"$JAVA_HOME/bin/java" "$WAS_LOGGING" -Djava.security.auth.login.config="$app_server_root/properties/wsjaas_client.conf" \
 -Djava.ext.dirs="$JAVA_HOME/jre/lib/ext:$WAS_EXT_DIRS:$WAS_HOME/plugins:$WAS_HOME/lib/WMQ/java/lib" \
 -Djava.naming.provider.url=<an_IIOP_URL_or_a_corbaloc_URL_to_your_application_server_machine_name> \
 -Djava.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory \
 -Dserver.root="$WAS_HOME" "$CLIENTSAS" "$CLIENTSSL" $USER_INSTALL_PROP \
 -classpath "$WAS_CLASSPATH":<list_of_your_application_jars_and_classes> \
 <fully_qualified_class_name_to_run> <your_application_parameters>
```

Cette commande suppose que le script *racine_was*/bin/setupCmdLine.sh a été exécuté pour définir correctement les variables.

`corbaloc:iiop:1.0@<host>:<port>/NameService` est un exemple de format de valeur de propriété java.naming.provider.url.

- **Connectez-vous à un serveur MBean de service de catalogue avec la sécurité activée :**

  Pour plus d'informations sur la connexion au bean géré du service de catalogue avec la sécurité activée, voir «Sécurité JMX (Java Management Extensions)», à la page 635.

Que faire ensuite

Pour d'autres exemples sur la manière d'afficher les statistiques et effectuer des opérations d'administration avec des beans gérés, voir l'exemple d'application **xsadmin**. Vous pouvez analyser le code source de l'exemple d'application xsadmin dans le fichier *rép_base_wxs*/samples/xsadmin.jar d'une installation autonome ou dans le fichier *rép_base_wxs*/xsadmin.jar dans une installation WebSphere Application Server. Voir Exemple : utilitaire **xsadmin** pour plus d'informations sur les opérations que vous pouvez exécuter avec l'exemple d'application **xsAdmin**.

Des informations supplémentaires sont également disponibles sur les beans gérés dans le package com.ibm.websphere.objectgrid.management.

Administration de connexions client J2C

Java

La fabrique de connexions WebSphere eXtreme Scale comporte une connexion client eXtreme Scale qui peut être partagée entre plusieurs applications et se maintenir au fil des redémarrages des applications.

Pourquoi et quand exécuter cette tâche

La connexion client comporte un bean de gestion qui fournit des informations sur le statut des connexions et sur les opérations de gestion du cycle de vie.

Procédure

Gérez les connexions client. Une fois la première connexion obtenue auprès de l'objet de fabrique de connexions XSConnectionFactory, une connexion client eXtreme Scale est établie vers la grille de données distante et le bean géré ObjectGridJ2CConnection est créé. La connexion client est maintenue pendant la durée de vie du processus. Pour mettre fin à une connexion client, faites appel à l'un des événements suivants :

- Arrêtez l'adaptateur de ressources. Un adaptateur de ressources peut être arrêté, par exemple lorsqu'il est imbriqué dans une application et que cette dernière est arrêtée.
- Appelez l'opération de bean géré resetConnection sur le bean géré ObjectGridJ2CConnection. Lorsque la connexion est réinitialisée, toutes les connexions sont invalidées, les transactions terminées et la connexion client ObjectGrid détruite. Les appels suivants vers les méthodes getConnection dans la fabrique de connexions génèrent une nouvelle connexion client.

WebSphere Application Server fournit également d'autres beans de gestion pour gérer les connexions J2C et surveiller les pools de connexions et les performances.

Chapitre 8. Contrôle

Vous pouvez utiliser la console de surveillance, les API, les beans gérés, les journaux et les utilitaires inclus pour surveiller les performances de votre environnement d'application.

Présentation des statistiques

Les statistiques dans WebSphere eXtreme Scale sont basés sur une arborescence interne de statistiques. L'API StatsAccessor, les modules PMI (Performance Monitoring Infrastructure) et l'API MBean sont générés à partir de l'arborescence interne.

L'illustration suivante montre la configuration générale des statistiques pour WebSphere eXtreme Scale.

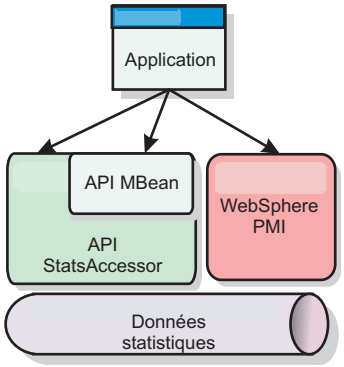


Figure 59. Présentation des statistiques

Toutes ces API permettent de visualiser l'arborescence des statistiques, mais chacune d'entre elles possède une fonction spécifique :

- **API Statistics**: L'API Statistics permet aux développeurs d'accéder directement aux statistiques pour des solutions flexibles et personnalisables d'intégration de statistiques, tels que des beans gérés personnalisés ou la consignation.
- **API MBean :** cette API est un mécanisme de surveillance basé sur une spécification. Elle utilise l'API Statistics et s'exécute de manière locale sur la machine virtuelle Java du serveur. Les structures de l'API et des beans gérés sont conçues pour s'intégrer aisément à des utilitaires tiers. Utilisez l'API MBean lorsque vous exécutez une grille d'objets répartie.
- **WebSphere Application Server Modules PMI :** utilisez ces modules si vous exécutez WebSphere eXtreme Scale dans WebSphere Application Server. Ces modules permettent de visualiser l'arborescence interne des statistiques.

API Statistics

A l'instar d'une mappe d'arborescence, il existe un chemin et une clé correspondants qui permettent d'extraire un module spécifique ou, dans ce cas, le niveau de granularité ou d'agrégation. Par exemple, supposons que l'arborescence contienne toujours un noeud racine arbitraire et que les statistiques soient regroupées pour une mappe appelée "payroll" appartenant à une instance

d'ObjectGrid appelée "accounting". Par exemple, pour accéder au module en fonction du niveau d'agrégation ou de granularité d'une mappe, vous pouvez insérer un paramètre String[] des chemins. Dans ce cas, vous obtenez String[] {root, "accounting", "payroll"}, chaque paramètre String représentant le chemin du noeud. Cette structure a pour avantage de permettre à l'utilisateur de spécifier le tableau dans un noeud quelconque du chemin et d'obtenir le niveau d'agrégation du noeud en question. L'insertion du paramètre String[] {root, "accounting"} vous permet d'obtenir les statistiques de mappe, sauf pour la grille entière de "accounting". L'utilisateur peut ainsi spécifier les types de statistiques à surveiller, ainsi que le niveau d'agrégation nécessaire pour l'application.

WebSphere Application ServerModules PMI

WebSphere eXtreme Scale inclut des modules de statistiques à utiliser avec l'infrastructure PMI WebSphere Application Server. Lorsqu'une instance de WebSphere eXtreme Scale est ajoutée à un profil WebSphere Application Server, les scripts d'ajout intègrent automatiquement les modules WebSphere eXtreme Scale dans les fichiers de configuration WebSphere Application Server. PMI vous permet d'activer et de désactiver les modules de statistiques, d'assembler automatiquement les statistiques selon différents niveaux de granularité et même de représenter les données sous forme de graphiques à l'aide du logiciel pré-intégré Tivoli Performance Viewer. Pour plus d'informations, voir «Surveillance à l'aide de la fonction PMI de WebSphere Application Server», à la page 573.

Intégration de produits tiers avec les beans gérés (MBean)

Les API eXtreme Scale et les beans gérés sont conçus pour faciliter l'intégration d'applications de surveillance tierces. JConsole et MC4J sont des exemples de consoles Java Management Extensions (JMX) légères qui permettent d'analyser les informations relatives à une topologie eXtreme Scale. Vous avez également la possibilité d'utiliser les API de programmation pour écrire des implémentations d'adaptateur afin de créer des instantanés ou d'effectuer un suivi des performances d'eXtreme Scale. WebSphere eXtreme Scale inclut un exemple d'application de surveillance qui permet d'effectuer la surveillance dès l'installation et qui peut servir de modèle pour créer des utilitaires de surveillance plus avancés.

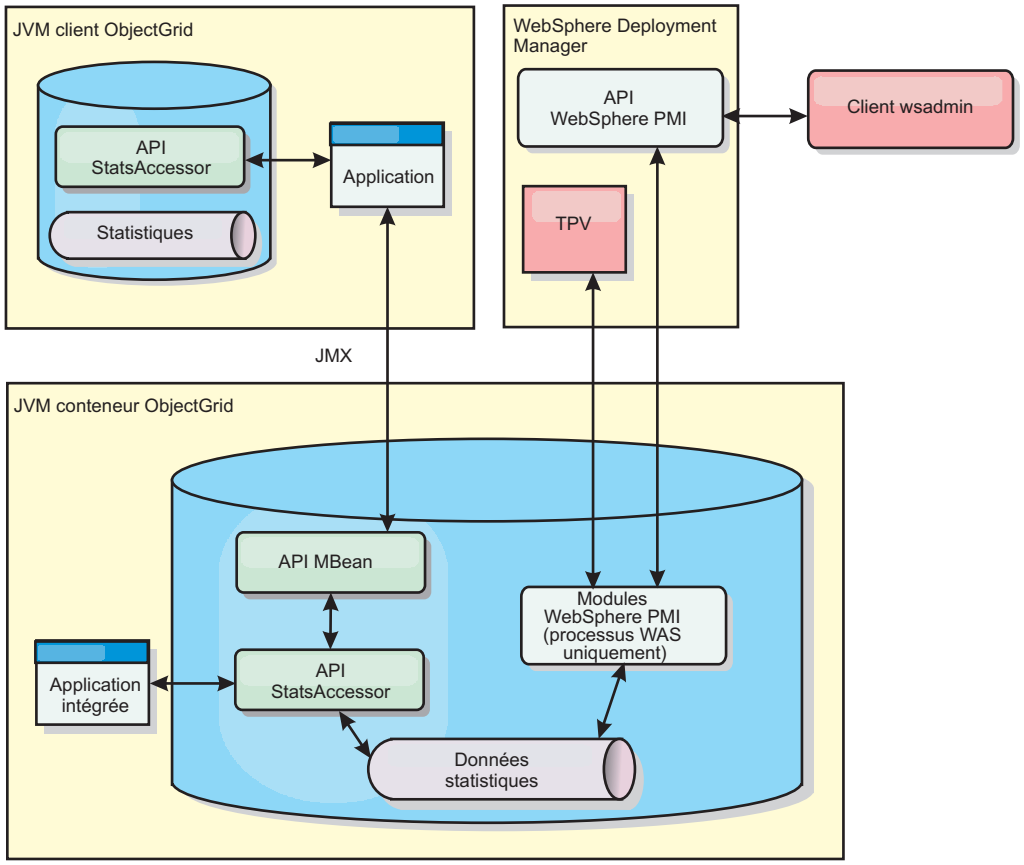


Figure 60. Présentation de l'API de bean géré

Pour plus d'informations, voir Exemple : utilitaire **xsadmin**. Pour plus d'informations sur l'intégration d'applications tierces spécifiques, voir les rubriques suivantes :

- Surveillance d'eXtreme Scale à l'aide d'un agent de surveillance IBM Tivoli
- «Surveillance d'eXtreme Scale à l'aide de Hyperic HQ», à la page 597
- «Surveillance des applications eXtreme Scale à l'aide de CA Wily Introscope», à la page 594

Surveillance à l'aide de la console Web

Avec la console Web, vous pouvez générer des graphiques des statistiques actuelles et historiques. Cette console fournit un certain nombre de graphiques préconfigurés pour des présentations générales et elle comporte une page de rapports personnalisés que vous pouvez utiliser pour élaborer des graphiques à partir des statistiques disponibles. Les fonctionnalités graphiques de la console de surveillance de WebSphere eXtreme Scale permettent de visualiser les performances globales des grilles des données présentes dans votre environnement.

Démarrage et consignation sur la console Web

Démarrez le serveur de la console en exécutant la commande **startConsoleServer** et en vous connectant au serveur en utilisant l'ID utilisateur et le mot de passe par défaut.

Avant de commencer

- **Navigateurs Web requis**

  Utilisez l'un des navigateurs suivants avec la console Web :

  – Mozilla Firefox, version 3.5.x et versions ultérieures
  – Mozilla Firefox, version 3.6.x et versions ultérieures
  – Microsoft Internet Explorer version 7 ou 8

Procédure

1. Facultatif : Si vous voulez exécuter votre serveur de console sur un port autre que le port par défaut, éditez le fichier *racine_install_wxs*/ObjectGrid/console/config/zero.config. Le port par défaut pour le serveur de la console est 7080 pour HTTP et 7443 pour HTTPS. Vous pouvez éditer les propriétés suivantes pour modifier les valeurs par défaut :

   ```
   /config/http/port = 7080
   /config/https/port = 7443
   ```

   Si vous éditez ces valeurs une fois que le serveur de la console est démarré, redémarrez le serveur pour utiliser les nouveaux numéros de port.
2. Démarrez le serveur de la console. Le script **startConsoleServer.bat|sh** de démarrage du serveur de la console se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation.
3. Connectez-vous à la console.
   a. Dans votre navigateur Web, accédez à https://your.console.host:7443, en remplaçant your.console.host par le nom de l'hôte du serveur sur lequel vous avez installé la console.
   b. Connectez-vous à la console.
      - **ID utilisateur :** admin
      - **Mot de passe :** admin

      La page d'accueil de la console s'affiche.
4. Modifiez la configuration de la console. Cliquez sur **Paramètres** > **Configuration** pour afficher la configuration de la console. La configuration de la console comprend ce type d'informations :
   - la chaîne de trace pour le client WebSphere eXtreme Scale, comme *=all=disabled
   - le nom et le mot de passe de l'administrateur
   - son adresse e-mail

Que faire ensuite

- Connectez vos serveurs de catalogue à la console Web pour démarrer le suivi des statistiques. Pour plus d'informations, voir «Connexion de la console Web aux serveurs de catalogue».
- Si vous avez besoin d'arrêter le serveur de la console Web, exécutez le script **stopConsoleServer.bat|sh**. Ce script se trouve dans le répertoire *racine_install_wxs*/ObjectGrid/bin de votre installation.

Connexion de la console Web aux serveurs de catalogue

Pour démarrer les statistiques d'affichage dans la console Web, vous devez d'abord vous connecter aux serveurs de catalogue que vous voulez surveiller. Des étapes supplémentaires sont requises si la sécurité est activée sur les serveurs de catalogue.

Avant de commencer

- Le serveur de console Web doit être actif. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 547.
- Vous devez disposer d'au moins un serveur de catalogue en cours d'exécution auquel vous voulez vous connecter. Voir «Démarrage d'un service de catalogue autonome qui utilise le transport ORB», à la page 494 or «Démarrage d'un service de catalogue autonome qui utilise le transport eXtremeIO IBM (XIO)», à la page 480 pour plus d'informations.

Procédure

1. Si SSL (Secure Sockets Layer) est activé sur vos serveurs de catalogue, vous devez configurer un fichier de clés, un fichier de clés certifiées et un fichier de propriétés client. Vous activez SSL pour un serveur de catalogue en affectant à l'attribut TransportType la valeur `SSL-Required` dans Fichier de propriétés du serveur.
   a. Configurez un fichier de clés et un fichier de clés certifiées, puis échangez ou les certificats publics ou importez-les. Par exemple, vous pouvez copier le fichier de clés et le fichier de clés certifiées vers un emplacement sur le serveur qui exécute la console Web.
   b. Editez le fichier de propriétés du client sur le serveur de la console Web pour inclure les propriétés de configuration SSL. Par exemple, vous pouvez éditez le fichier *racine_install_wxs*`/ObjectGridProperties/sampleclient.properties`. Les propriétés suivantes sont requises pour les connexions SSL sortantes à partir de la console Web :

```
#------------------------------------------------------------------------------
# SSL Configuration
#
# - contextProvider     (IBMJSSE2, IBMJSSE, IBMJSSEFIPS, etc.)
# - protocol            (SSL, SSLv2, SSLv3, TLS, TLSv1, etc.)
# - keyStoreType        (JKS, JCEK, PKCS12, etc.)
# - trustStoreType      (JKS, JCEK, PKCS12, etc.)
# - keyStore            (fully qualified path to key store file)
# - trustStore          (fully qualified path to trust store file)
# - alias               (string specifying ssl certificate alias to use from keyStore)
# - keyStorePassword    (string specifying password to the key store - encoded or not)
# - trustStorePassword  (string specifying password to the trust store - encoded or not)
#
# Uncomment these properties to set the SSL configuration.
#------------------------------------------------------------------------------
#alias=clientprivate
#contextProvider=IBMJSSE
#protocol=SSL
#keyStoreType=JKS
#keyStore=etc/test/security/client.private
#keyStorePassword={xor}PDM2OjErLyg\=
#trustStoreType=JKS
#trustStore=etc/test/security/server.public
#trustStorePassword={xor}Lyo9MzY8
```

   Important : Windows Si vous utilisez Windows, vous devez définir littéralement la barre oblique inverse (\) dans le chemin. Par exemple, si vous souhaitez utiliser le chemin d'accès `C:\opt\ibm`, entrez `C:\\opt\\ibm` dans le fichier des propriétés.
2. Créez et maintenez des connexions aux serveurs de catalogue que vous voulez surveiller. Répétez les étapes suivantes pour ajouter chaque serveur de catalogue à la configuration.
   a. Cliquez sur **Paramètres** > **Serveurs de catalogue eXtreme Scale**.
   b. Ajoutez un nouveau serveur de catalogue.
      1) Cliquez sur l'icône Ajouter () pour enregistrer un serveur de catalogue existant.

2) Fournissez des informations, telles que le nom d'hôte et le port d'écoute. Voir «Planification des ports réseau», à la page 59 pour plus d'informations sur la configuration des ports et les valeurs par défaut.
3) Cliquez sur **OK**.
4) Vérifiez que le serveur de catalogue a bien été ajouté à l'arborescence de navigation.

3. Regroupez les serveurs de catalogue que vous avez créés dans un domaine de service de catalogue. Vous devez créer un domaine de service de catalogue lorsque la sécurité est activée dans vos serveurs de catalogue, car les paramètres de sécurité sont configurés dans le domaine de service de catalogue.
   a. Cliquez sur la page **Paramètres** > **Domaines eXtreme Scale**.
   b. Ajoutez un nouveau service de catalogue.

1) Cliquez sur l'icône Ajouter ( ) pour enregistrer un service de catalogue existant. Entrez le nom du domaine de service de catalogue.
2) Une fois que vous avez créé le domaine de service de catalogue, vous pouvez modifier les propriétés. Les propriétés du domaine de service de catalogue sont les suivantes :

   Nom Indique le nom d'hôte du domaine, attribué par l'administrateur.

   Serveurs de catalogue
   Liste un ou plusieurs catalogues qui appartiennent au domaine sélectionné. Vous pouvez ajouter les serveurs de catalogue que vous avez créés dans l'étape précédente.

   Classe de génération
   Indique le nom de la classe qui implémente l'interface CredentialGenerator. Cette classe utilisée pour obtenir les données d'identification des clients. Si vous définissez une valeur dans cette zone, la valeur remplace la propriété **crendentialGeneratorClass** dans le fichier `client.properties`.

   Propriétés du générateur
   Spécifie les propriétés de la classe d'implémentation CredentialGenerator. Les propriétés correspondent à l'objet avec la méthode setProperties(String). La valeur credentialGeneratorprops n'est utilisée que si la valeur de la propriété credentialGeneratorClass n'est pas null. Si vous définissez une valeur dans cette zone, la valeur remplace la propriété **credentialGeneratorProps** dans le fichier `client.properties`.

   Chemin des propriétés du client eXtreme Scale
   Indique le chemin d'accès au fichier de propriétés client que vous avez édité pour inclure les propriétés de sécurité dans une étape précédente. Par exemple, vous pouvez indiquer le fichier `c:\ObjectGridProperties\sampleclient.properties`. Si vous souhaitez empêcher la console de tenter d'utiliser les connexions sécurisées, vous pouvez supprimer la valeur dans cette zone. Après avoir défini le chemin, la console utilise une connexion non sécurisée.

3) Cliquez sur **OK**.
4) Vérifiez que le domaine a bien été ajouté à l'arborescence de navigation.

Pour afficher les informations concernant un domaine de service de catalogue existant, cliquez sur le nom du domaine de service de catalogue dans l'arborescence de navigation sur la page **Paramètres** > **Domaines eXtreme Scale**.

4. Visualisez le statut de la connexion La zone **Domaine en cours** indique le nom du domaine de service de catalogue qui est actuellement utilisé pour afficher des informations dans la console Web. L'état de la connexion s'affiche en regard du nom du domaine de service de catalogue.

Affichage des statistiques avec la console Web

Vous pouvez surveiller les statistiques et d'autres informations de performances avec la console Web.

Avant de commencer

Pour pouvoir afficher les statistiques avec la console Web, vous devez procéder comme suit :

1. Démarrez le serveur de console Web. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 547.
2. Connectez vos serveurs de catalogue au serveur de la console Web. Pour plus d'informations, voir «Connexion de la console Web aux serveurs de catalogue», à la page 548.
3. Exécutez les grilles et les applications de données actives dans les serveurs qui sont gérés par votre domaine de service de catalogue.

Pourquoi et quand exécuter cette tâche

Une fois les grilles de données créées et les applications configurées pour utiliser ces grilles, laissez aux statistiques le temps d'être générées. Par exemple, avec une grille de données de caches dynamiques, les statistiques ne sont pas disponibles jusqu'à ce qu'un WebSphere Application Server qui exécute un cache dynamique se connecte au cache dynamique. En général, il suffit d'attendre environ une minute après l'apport d'une modification de configuration importante pour observer le changement au niveau des statistiques.

Conseil : Pour afficher des informations spécifiques sur un point de données ou un graphique, placez le pointeur de la souris sur l'élément.

Procédure

- Pour afficher les statistiques en cours du serveur, cliquez sur **Surveiller** > **Vue d'ensemble du serveur**.
- Pour afficher les performances de toutes vos grilles de données, cliquez sur **Surveiller** > **Vue d'ensemble des domaines de grilles de données**.
- Pour afficher des grilles de données individuelles, cliquez sur **Surveiller** > **Présentation de la grille de données** > *data_grid_name*. Cette page propose un récapitulatif incluant le nombre d'entrées en cache, la durée moyenne des transactions et le débit moyen.
- Pour afficher d'autres détails sur une grille de données spécifique, cliquez sur **Surveiller** > **Détails d'une grille de données**. Une arborescence affiche toutes les grilles de données de votre configuration. Vous explorer cette arborescence en aval et accéder à une grille de données spécifique afin d'afficher les mappes appartenant à cette grille de données. Vous pouvez cliquer sur le nom d'une grille de données ou d'une mappe pour obtenir plus d'informations.

- Pour choisir les statistiques à placer dans votre rapport personnalisé, cliquez sur **Contrôler** > **Rapports personnalisés**.

  Cette vue permet d'élaborer des graphiques détaillés à partir des diverses statistiques. L'arborescence permet d'explorer les grilles de données et les serveurs disponibles, ainsi que leurs statistiques. Un menu s'affiche lorsqu'on clique ou que l'on appuie sur Entrée sur un noeud qui référence des données pouvant être représentées dans un graphique. Vous pouvez créer un nouveau graphique contenant les statistiques ou, si elles sont compatibles, ajouter ces statistiques à celles d'un graphique existant. Pour plus d'informations, voir «Surveillance à l'aide de rapports personnalisés», à la page 557.

Statistiques de la console Web

En fonction de la vue que vous utilisez dans la console Web, vous pouvez afficher différentes statistiques relatives à votre configuration. Ces statistiques incluent la mémoire utilisée, les grilles de données les plus utilisées et le nombre d'entrées en mémoire cache.

Présentation du domaine de grille de données

Les statistiques de présentation du domaine de la grille de données figurent dans la page **Surveiller** > **Présentation du domaine de la grille de données**. Cliquez sur l'un des onglets suivants pour plus d'informations sur le domaine de la grille de données :

Onglet Capacité utilisée
: Le graphique de **distribution de la capacité utilisée de la grille de données en cours** contient une image du **pool total** et les **plus importants consommateurs de capacité utilisés**. Ne sont affichées que les 25 grilles de données les plus consommatrices. Le graphique de **capacité utilisé dans le temps** indique le nombre d'octets consommés par la grille affichée.

Onglet Débit moyen
: Le graphique des **cinq grilles de données les plus actives par temps moyen de transaction en millisecondes** contient la liste des cinq premiers caches de données organisés en fonction du temps de transaction moyen. Le graphique **Débit moyen au fil du temps** affiche les débits moyens, maximum et minimum au cours de la dernière heure, du dernier jour et de la dernière semaine.

Onglet Délai de transaction moyen
: Le graphique des **cinq grilles de données les plus lentes** contient des données sur les grilles de données les plus lentes. Le graphique du **temps de transaction moyen dans le temps** indique les temps de transaction moyen, maximum et minimum au cours de la dernière heure, du dernier jour et de la dernière semaine.

Présentation de la grille de données

Pour afficher des statistiques sur une grille de données, cliquez sur **Surveiller** > **Présentation de la grille de données** > *data_grid_name*.

Récapitulatif en cours sur les 30 dernières secondes
Affiche le nombre actuel d'entrées de cache, le temps de transaction moyen, le débit moyen, et le nombre de réussites en mémoire de la grille de données sélectionnée.

Onglet Capacité utilisée
Le graphique **Récapitulatif en cours au cours des 30 seconds dernières minutes** affiche le nombre d'entrées de cache et de la capacité utilisée en octets au cours d'un intervalle de temps spécifié.

Onglet Utilisation du cache
L'onglet **Utilisation de cache** permet de visualiser le nombre de demandes ayant abouti dans le cache, et affiche les tentatives de cache, de réussites en mémoire cache, et le taux de réussite en mémoire cache au cours d'un intervalle de temps spécifié.

Onglet Débit moyen
Le graphique **Débit moyen vs. Durée moyenne de transaction** affiche le temps de transaction et le débit au cours d'une période donnée.

Informations une grille de données

Les statistiques de grille de données figurent dans la page **Surveiller** > **Détails de la grille de données**. Vous pouvez consulter les données pour une grille sélectionnée et les mappes qui se trouvent dans cette grille.

Récapitulatif en cours sur les 30 seconds dernières secondes
Affiche la capacité utilisée actuelle, le nombre d'entrées de cache, le temps de transaction moyen et le débit moyen de la grille de données sélectionnée.

Répartition des capacités actuelles utilisées par les mappes de grilles d'objets eXtreme Scale
Permet de visualiser un pool total, ce qui inclut les capacités par zone ainsi que les capacités totales dans chaque zone. Ne sont affichées que les 25 plus importantes mappes ObjectGrid. Vous pouvez également afficher les principaux consommateurs de capacité utilisés en fonction de chaque mappe.

Répartition de la capacité utilisée pour la zone actuelle
Affiche un pool total qui inclut le pool total et les principaux consommateurs de capacité utilisés dans la zone de la grille de données sélectionnée. Vous pouvez également afficher les principaux consommateurs de capacité utilisés en fonction de chaque zone.

Statistiques de mappe :

Récapitulatif en cours sur les 30 seconds dernières secondes
Affiche la capacité utilisée actuelle, le nombre d'entrées de cache, le temps de transaction moyen et le débit moyen de la grille de données sélectionnée.

Répartition de la capacité utilisée pour la partition actuelle
Vue d'une partition qui contient le pool total et les principaux consommateurs de capacité utilisés. Ne sont affichées que les 25 partitions

les plus consommatrices. Vous pouvez également afficher les principaux consommateurs de capacité utilisés en fonction de chaque partition.

Présentation du serveur

Les statistiques du serveur figurent dans la page **Surveiller** > **Présentation du serveur**

Répartition actuelle de la mémoire utilisée du serveur
Ce graphique est composé de deux vues. **Pool total** affiche la quantité actuelle de mémoire utilisée (réelle) dans l'environnement d'exécution du serveur. **Plus grands consommateurs de mémoire utilisés** indique la mémoire utilisée par serveur, mais seuls les 25 premiers serveurs qui utilisent le plus de mémoire sont indiqués.

Total de la mémoire utilisée dans le temps
Affiche l'utilisation de la mémoire réelle dans l'environnement d'exécution du serveur.

Mémoire utilisée au fil du temps
Affiche la quantité de mémoire utilisée dans l'environnement d'exécution du serveur.

Rapports personnalisés : statistiques du domaine de services de catalogue

Vous pouvez afficher des statistiques de domaine de services de catalogue en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Délai de transaction moyen (ms)
Affiche la durée moyenne que met une transaction à s'effectuer dans ce domaine.

Débit de transaction moyen (trans/s)
Affiche le nombre moyen de transactions par seconde dans ce domaine.

Délai maximum de transaction (ms)
Affiche le temps *maximum* qu'a mis une transaction pour s'exécuter dans ce domaine.

Délai minimum de transaction (ms)
Affiche le temps *minimum* qu'a mis une transaction pour s'exécuter dans ce domaine.

Délai de transaction total (ms)
Affiche le temps total passé à des transactions dans ce domaine depuis l'initialisation de ce dernier.

Rapports personnalisés : statistiques du serveur de conteneur

Vous pouvez afficher des statistiques de domaine de serveur de catalogue en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Délai de transaction moyen (ms)
Affiche pour ce serveur de catalogue la durée moyenne que met une transaction à s'effectuer.

Débit de transaction moyen (trans/s)
Affiche le nombre moyen de transactions par seconde pour ce serveur de catalogue.

Délai maximum de transaction (ms)
Affiche pour ce serveur de catalogue le temps *maximum* qu'a mis une transaction pour s'exécuter.

Délai minimum de transaction (ms)
Affiche pour ce serveur de catalogue le temps *minimum* qu'a mis une transaction pour s'exécuter.

Délai de transaction total (ms)
Affiche pour ce serveur de catalogue le temps total passé à des transactions depuis l'initialisation du serveur.

Nombre total d'entrées en cache
Affiche le nombre actuel d'objets mis en cache appartenant à des grilles supervisées par ce serveur de catalogue.

Taux de réussites (pourcentage)
Affiche le taux de réussites pour la grille de données sélectionnée. Un taux élevé est souhaitable. Ce taux indique le degré d'efficacité de la grille pour éviter d'accéder au stockage de persistance.

Octets utilisés
Affiche la consommation de la mémoire par cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre minimal d'octets utilisés
Affiche le point bas de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre maximal d'octets utilisés
Affiche le point haut de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre total de réussites
Affiche le nombre total de fois où les données demandées ont été trouvées dans la mappe, dispensant de devoir accéder au stockage de persistance.

Nombre total de demandes get
Affiche le nombre total de fois où la mappe a dû accéder au stockage de persistance pour obtenir des données.

Segments de mémoire disponibles (Mo)
Affiche la quantité effective de segments mémoire disponibles pour la machine virtuelle Java en cours d'utilisation par le serveur de catalogue.

Total des segments de mémoire
Affiche la quantité effective de segments mémoire disponibles pour la machine virtuelle Java en cours d'utilisation par ce serveur de catalogue.

Nombre de processeurs disponibles
Affiche le nombre de processeurs qui sont disponibles pour ce service de catalogue et ses mappes. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Taille maximale des segments de mémoire (Mo)
Affiche la quantité maximale de segments mémoire disponibles pour la machine virtuelle Java en cours d'utilisation par ce serveur de catalogue.

Mémoire utilisée
Affiche la mémoire utilisée dans la machine virtuelle Java en cours d'utilisation par ce serveur de catalogue.

Rapports personnalisés : statistiques de la grille de données

Vous pouvez afficher des statistiques de grille de données en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Délai de transaction moyen (ms)
Affiche la durée moyenne que mettent pour s'effectuer des transactions impliquant cette grille.

Débit de transaction moyen (trans/s)
Affiche le nombre moyen de transactions effectuées par seconde par cette grille.

Délai maximum de transaction (ms)
Affiche le temps *maximum* qu'a mis une transaction effectuée par cette grille.

Délai minimum de transaction (ms)
Affiche le temps *minimum* qu'a mis une transaction effectuée par cette grille.

Délai de transaction total (ms)
Affiche le temps total de traitement des transactions pour cette grille.

Rapports personnalisés : statistiques de mappe

Vous pouvez afficher des statistiques en créant un rapport personnalisé. Cliquez sur **Surveiller** > **Rapports personnalisés**.

Nombre total d'entrées en cache
Affiche le nombre d'objets de la mappe actuellement mis en cache.

Taux de réussites (pourcentage)
Affiche le taux de réussites pour la mappe sélectionnée. Un taux élevé est souhaitable. Ce taux indique le degré d'efficacité de la mappe pour éviter d'accéder au stockage de persistance.

Octets utilisés
Affiche la consommation de la mémoire par cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre minimal d'octets utilisés
Affiche la consommation minimum en octets pour cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre maximal d'octets utilisés
Affiche la consommation maximum en octets pour cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

Nombre total de réussites
Affiche le nombre total de fois où les données demandées ont été trouvées dans la mappe, dispensant de devoir accéder au stockage de persistance.

Nombre total de demandes get
Affiche le nombre total de fois où la mappe a dû accéder au stockage de persistance pour obtenir des données.

Segments de mémoire disponibles (Mo)
Affiche la quantité effective de segments mémoire disponibles pour cette mappe dans la machine virtuelle Java en cours d'utilisation par le serveur de catalogue.

Total des segments de mémoire (Mo)
Affiche la quantité totale de segments mémoire disponibles pour cette mappe dans la machine virtuelle Java en cours d'utilisation par le serveur de catalogue. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Nombre de processeurs disponibles
Affiche le nombre de processeurs disponibles pour cette mappe. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Taille maximale des segments de mémoire (Mo)
Affiche la quantité maximum de segments mémoire disponibles pour cette mappe dans la machine virtuelle Java en cours d'utilisation par le serveur de catalogue.

Mémoire utilisée (Mo)
Affiche la quantité de mémoire utilisée dans cette mappe.

Surveillance à l'aide de rapports personnalisés

Vous pouvez générer des rapports personnalisés pour enregistrer les divers graphiques qui contiennent des statistiques sur les domaines de service de catalogue, les grilles de données et les serveurs de conteneur dans votre environnement. Vous pouvez enregistrer les rapports personnalisés et les charger pour les consulter ultérieurement.

Avant de commencer

Pour pouvoir afficher les statistiques avec la console Web, vous devez procéder comme suit :

1. Démarrez le serveur de console Web. Pour plus d'informations, voir «Démarrage et consignation sur la console Web», à la page 547.
2. Connectez vos serveurs de catalogue au serveur de la console Web. Pour plus d'informations, voir «Connexion de la console Web aux serveurs de catalogue», à la page 548.
3. Exécutez les grilles et les applications de données actives dans les serveurs qui sont gérés par votre domaine de service de catalogue

Procédure

- Créez un rapport personnalisé.
  1. Cliquez sur **Contrôler** > **Rapports personnalisés**. Les domaines eXtreme Scale que vous avez définis sont répertoriés dans un format d'arborescence. Vous pouvez développer chacun de ces domaines pour afficher les statistiques disponibles que vous pouvez ajouter au rapport personnalisé.
  2. Ajoutez des graphiques avec les statistiques à suivre. Les statistiques disponibles sont signalées par l'icône de graphique (). Cliquez sur l'une des statistiques à suivre. Choisissez **Ajouter au nouveau graphique** ou **Ajouter au graphique existant**. En fonction de votre sélection, la statistique sélectionnée s'affiche dans un nouvel onglet de graphique ou dans le graphique sélectionné. Vous pouvez ajouter une métrique à un graphique existant uniquement si les métriques déjà sur le graphique et la nouvelle métrique utilisent la même unité.
- Enregistrez un rapport personnalisé. La sauvegarde du rapport personnalisé enregistre les statistiques dans tous les onglets que vous avez créés. Pour sauvegarder le rapport, cliquez sur **Sauvegarder**.
- Chargez un rapport personnalisé. Cliquez sur **Charger** et choisissez le rapport personnalisé enregistré à afficher.

Surveillance de la santé de l'environnement

Message Center fournit une vue agrégée des notifications d'événements des messages de l'outil de diagnostic de premier niveau. Vous pouvez afficher ces notifications d'événements avec Messages Center dans la console Web, l'utilitaire **xscmd** ou à l'aide d'un programme avec des beans gérés.

Présentation de Message Center

Message Center agrège les événements d'état de santé depuis tous les serveurs de conteneur et de catalogue dans un domaine de service de catalogue en temps réel. Lorsque Message Center est configuré, vous pouvez afficher une présentation en cours des événements critiques qui surviennent dans différents serveurs sans collecter les journaux de chaque serveur.

Implémentation de Message Center

Les déploiements de grille de données peuvent impliquer des douzaines ou des centaines de processus serveur répartis. Si un incident se produit, vous pouvez ouvrir le fichier journal du serveur de conteneur concerné pour analyser le problème en détail.

Message Center est constitué des composants suivants :

Agrégation d'événements

Lorsque vous configurez la surveillance de la santé sur un serveur de catalogue, vous recevez kes événements agrégés qui affectent la santé de l'ensemble du domaine de service de catalogue. La structure inclut l'indication de la source et la gravité des types d'événements suivants :

- Tous les événements de l'outil de diagnostic de premier niveau
- Toutes les entrées de journal WARNING ou SEVERE
- La liste de toutes les entrées de journal, notamment INFO, WARNING et SEVERE

- Les opération de démarrage et d'arrêt du serveur
- Perte ou récupération du quorum

Message Center dans la console Web
Message Center dans la console Web affiche les enregistrements d'événement agrégés. Ces événements incluent les événements récents et les notifications de mise à jour en temps réel qui se produisent après l'ouverture de la console.

Evénements dans l'utilitaire `xscmd`
Vous pouvez également afficher la liste des événements récents à l'aide de l'utilitaire **`xscmd`**. Lorsque des événements se produisent, vous pouvez rediriger les enregistrements d'événement pour créer des utilitaires de scriptage automatique.

Beans géré pour l'intégration dans d'autres logiciels de surveillance
Vous pouvez également utiliser les beans gérés de gestion disponibles pour enficher Message Center dans vos autres logiciels de surveillance JMX (Java Management Extensions). La documentation de ces beans gérés est incluse dans la documentation des API.

Message Center et Analyseur de journal

L'analyseur de journal est un autre outil d'analyse d'un groupe de messages de journal. Cet outil impose de collecter manuellement les journaux de divers serveurs dans votre environnement. Ensuite, vous pouvez exécuter l'outil pour créer des rapports sur les conditions des problèmes. Utilisez l'analyseur de journal pour exécuter une analyse post-mortem des journaux lorsque vous devez analyser un nombre de messages supérieur au sous-ensemble de 1 000 messages que vous pouvez afficher dans Message Center. Utilisez Message Center pour surveiller en temps réel l'état de la grille de données pour identifier rapidement les problèmes qui apparaissent. Ensuite, vous pouvez vérifier les fichiers journaux du serveur de conteneur associé ou utiliser l'analyseur de journal pour analyser plus en détail le problème.

Configuration et architecture de la surveillance de la santé

Vous pouvez activer Message Center en configurant un ou plusieurs serveurs de catalogue comme concentrateur. Chaque concentrateur dispose de ses propres abonnements et historiques d'événements. Chaque événement de l'historique porte un numéro de séquence. Les historiques d'événements sur les serveurs de catalogue ne sont pas synchronisés et sont différents. Les serveurs de catalogue peuvent s'abonner aux événements de journal et de l'outil de diagnostic de premier niveau des autres serveurs de catalogue.

Configuration de Message Center

Pour utiliser Message Center, vous devez configurer les serveurs de catalogue comme concentrateurs de messagerie.

Procédure

1. Activez le serveur de catalogue comme concentrateur pour l'infrastructure de surveillance de santé. Tous les serveurs sont activés comme concentrateurs par défaut. Vous pouvez activer ou désactiver ce paramètre avec la propriété suivante dans le fichier `server.properties` du serveur de catalogue :

   8.6+ enableManagementConcentrator
   Indique si le serveur de catalogue est un concentrateur pour Message

Center. Cette propriété est activée par défaut. Pour désactiver le concentrateur, définissez la valeur `false`.

Valeur par défaut : `true`

2. Facultatif : Si vous souhaitez consigner les messages de journal INFO, vous devez spécifier une expression régulière qui filtre ces messages. Définissez l'expression régulière avec la propriété suivante dans le fichier `server.properties` du serveur de catalogue :

   8.6+ logNotificationFilter
   Définit une expression régulière qui filtre tous les messages, notamment les message de journal de niveau INFO. Ce filtre détermine les messages qui génèrent des événements de surveillance de la santé. Si vous ne spécifiez pas d'expression régulière, les messages de journal au niveau INFO ne sont pas publiés via le canevas de surveillance de la santé. Par défaut, seul les messages WARNING et SEVERE génèrent des événements de surveillance de la santé.

   Exemple : `logNotificationFilter=.*DYNACACHE.*`

3. Lorsque vous modifiez les propriétés du serveur, vous devez redémarrer le serveur de catalogue. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.

Que faire ensuite

Une fois que les serveurs de catalogue sont activés comme concentrateur pour l'infrastructure de surveillance de santé, vous pouvez utiliser Message Center dans la console Web ou l'utilitaire **xscmd** pour afficher les notifications d'événements de santé.

Affichage des notifications d'événement de santé dans Message Center

Vous pouvez utiliser Messages Center dans la console Web pour évaluer la santé en temps réel de la grille de données complète et du domaine de service de catalogue. Les événements qui s'affichent dans Message Center sont un sous-ensemble d'événements qui sont filtrés pour afficher les questions les plus critiques.

Avant de commencer

- Configurez les concentrateurs de notification de santé sur les serveurs de catalogue. Pour plus d'informations, voir «Configuration de Message Center», à la page 559.
- Démarrez la console Web et connectez-la au domaine de serveur de catalogue. Pour plus d'informations, voir «Surveillance à l'aide de la console Web», à la page 547.

Procédure

- Affichez les erreurs graves, les messages de l'outil de diagnostic de premier niveau et démarrez et arrêtez les événements serveur via les notifications d'événements dans la console Web. Ces notifications s'affichent automatiquement lorsque vous êtes connecté à une page dans la console Web.
- Affichez les messages dans Message Center. Dans la console Web, cliquez sur **Surveillance** > **Message Center**. Message Center affiche les 1 000 derniers messages critiques qui ont été envoyés via le concentrateur de messages du serveur de catalogue.

Le concentrateur de messages du serveur de catalogue filtre les messages qui s'affichent et il affiche donc les 1 000 messages qui s'affichent. Par conséquent, Message Center contient un sous-ensemble de tous les événements critiques qui surviennent sur les serveurs de catalogue. Si de nouveaux messages sont disponibles alors que la page est ouverte, un message d'information avec l'option d'actualisation de la page s'affiche dans la partie supérieure de la page.

- Filtrez les messages affichés dans Message Center. Vous pouvez ajouter jusqu'à trois règles de filtrage. Une règle est constituée d'une colonne, d'une condition et d'une valeur.
  1. Dans la console Web, cliquez sur **Surveillance** > **Message Center**.
  2. Cliquez sur le bouton de filtrage ().
  3. Ajoutez une règle.

     a. Cliquez sur le bouton d'ajout ().
     b. Dans Message Center, choisissez la colonne à filtrer :

        ID ID d'événement généré par Message Center.

        Type Le type de message qui indique la gravité du message. Les valeurs admises sont : Grave, Avertissement, Erreur, et Information.

        Date Date et heure de génération du message.

        Source
        Serveur d'origine du message.

        Message
        Texte du message de l'événement de message.

     c. Sélectionnez la condition à laquelle vous voulez appliquer le filtre. La liste suivante de conditions s'appliquent à pratiquement toutes les colonnes, à l'exception de la colonne de date et de type :
        - Contient
        - Est
        - Commence par
        - Se termine par

     d. Entrez la valeur de filtrage de la colonne.

     Exemple : pour afficher les messages de server1 uniquement, sélectionnez la colonne **Source**. Sélectionnez la condition **Est**. Pour la valeur, tapez `server1`.
  4. Vous pouvez choisir d'établir une correspondance avec certaines des règles que vous avez définies ou toutes les règles.
  5. Cliquez sur **Filtrer** pour appliquer les filtres définis à la sortie de Message Center.

Que faire ensuite

Si vous constatez que des événements critiques se produisent sur l'un des serveurs de conteneur, ouvrez le fichier journal du serveur de conteneur pour les analyser. Pour plus d'informations, voir «Activation de la consignation», à la page 665.

Affichage des informations de santé avec l'utilitaire xscmd

Vous pouvez afficher les notifications d'événements en cours, l'historique de notification d'événement et afficher et définir des filtres de notification à partir du centre de message avec l'utilitaire **xscmd**.

Avant de commencer

- Configurez les concentrateurs de notification de santé sur les serveurs de catalogue. Pour plus d'informations, voir «Configuration de Message Center», à la page 559.
- Démarrez l'utilitaire **xscmd** et connectez-le au domaine de service de catalogue. Pour plus d'informations, voir «Administration avec l'utilitaire **xscmd**», à la page 515.

Procédure

- Affichez l'historique des notifications d'événements avec l'utilitaire **xscmd**. La sortie apparaît dans un tableau.

  ```
  xscmd -c showNotificationHistory -cep hostname:port(,hostname:port)
  ```

- Ecoutez les nouvelles notifications.

  ```
  xscmd -c listenForNotifications -cep hostname:port(,hostname:port)
  ```

  La sortie apparaît dans un format brut et s'exécute jusqu'à ce que vous arrêtiez la commande. Vous pouvez écrire des scripts supplémentaires pour analyser la sortie.

- Créez une liste filtrée de toutes les entrées de journal, y compris les entrées INFO, WARNING et SEVERE. Par défaut, Message Center et les commandes affichent uniquement les erreurs WARNING et SEVERE et les événements. Vous pouvez définir le filtre pour tous les serveurs dans l'environnement ou sur un serveur unique.

  ```
  xscmd -c setNotificationFilter -fs <regular expression>  [-server <servername>]
  ```

- Affichez les filtres de notification en cours pour tous les serveurs de l'environnement ou un serveur unique.

  ```
  xscmd -c getNotificationFilter [-s servername]
  ```

Surveillance à l'aide de fichiers CSV

Vous pouvez activer la surveillance des données à écrire dans des fichiers CSV (comma-separated values). Ces fichiers CSV peuvent contenir des informations sur la machine JVM, la mappe ou l'instance ObjectGrid.

Pourquoi et quand exécuter cette tâche

En activant l'écriture des données de surveillance dans des fichiers CSV, vous pouvez télécharger et analyser les données d'historique de chaque serveur de conteneur. Les données sont collectées lorsque vous démarrez le serveur avec les propriétés serveur qui activent les fichiers CSV. Vous pouvez ensuite télécharger les fichiers CSV à tout moment et utiliser les fichiers comme vous le désirez.

Procédure

1. Mettez à jour le fichier des propriétés du serveur avec les propriétés suivantes qui sont liées à l'activation des fichiers CSV.

   ```
   parameter=default value
   jvmStatsLoggingEnabled=true
   maxJVMStatsFiles=5
   ```

```
maxJVMStatsFileSize=100
jvmStatsFileName=jvmstats
jvmStatsWriteRate=10

mapStatsLoggingEnabled=true
maxMapStatsFiles=5
maxMapStatsFileSize=100
mapStatsFileName=mapstats
mapStatsWriteRate=10

ogStatsLoggingEnabled=true
maxOGStatsFiles=5
maxOGStatsFileSize=100
ogStatsFileName=ogstats
ogStatsWriteRate=10
```

Pour plus d'informations sur ces propriétés, voir Fichier de propriétés du serveur.

2. Redémarrez le serveur pour sélectionner les modifications dans le fichier des propriétés du serveur.
3. Téléchargez le fichier CSV. Le fichier CSV est écrit dans le répertoire *server_name*/logs.

   8.6+ Chaque fichier CSV contient une en-tête qui nomme chacune des colonnes. Chaque colonne est séparée par une virgule.
4. Importez le fichier CSV dans le programme que vous utilisez pour traiter les données, par exemple, une feuille de calcul.

Que faire ensuite

Pour plus d'informations sur les données qui figurent dans les fichiers CSV, voir «Définition des statistiques des fichiers CSV».

Définition des statistiques des fichiers CSV

Les fichiers CSV que vous pouvez télécharger pour un serveur comprennent des statistiques que vous pouvez utiliser pour créer des diagrammes d'historique ou d'autres informations.

Journal des statistiques JVM (Java virtual machine)

TimeStamp (colonne 1)
Indique la date et l'heure de l'image instantanée des statistiques, prise pour la machine JVM.

ServerName (colonne 2)
Indique le nom du serveur de la machine JVM.

Hostname (colonne 3)
Indique le nom de la machine JVM.

FreeMemory (colonne 4)
Indique le nombre d'octets disponibles pour la machine JVM.

MaxMemory (colonne 5)
Indique le nombre maximal d'octets qui peut être attribué pour la machine JVM.

TotalMemory (colonne 6)
Affiche l'utilisation de la mémoire réelle dans l'environnement d'exécution du serveur.

AvailProcs (colonne 7)
Affiche le nombre de processeurs qui sont disponibles pour ce service de catalogue et ses mappes. Pour une stabilité optimale, vous devez exécuter les serveurs à 60 % du chargement de processeur et les segments de mémoire des machines virtuelles Java à 60 % du chargement des segments de mémoire. Les pics peuvent alors pousser l'utilisation du processeur à 80-90 %, mais ce ne doit pas être le niveau habituel d'exécution de vos serveurs.

Journal des statistiques de mappe

TimeStamp (colonne 1)
Indique la date et l'heure de l'image instantanée des statistiques, prise pour la mappe.

MapName (colonne 2)
Indique le nom de la mappe.

OgName (colonne 3)
Indique le nom de la grille de données à laquelle appartient la mappe.

PartitionId (colonne 4)
Indique l'ID de la partition.

MapSetName (colonne 5)
Indique le groupe de mappes auquel appartient la mappe.

HitRate (colonne 6)
Affiche le taux de réussites pour la mappe sélectionnée. Un taux élevé est souhaitable. Le taux de réussite indique la manière dont la grille de données contribue à éviter d'accéder au stockage de persistance.

Count (colonne 7)
Indique le nombre d'échantillons de données collectés depuis le démarrage du serveur. Par exemple, la valeur 100 indique que l'entrée est le 100ème échantillon collecté depuis le démarrage du serveur.

TotalGetCount (colonne 8)
Affiche le nombre total de fois où la mappe a dû accéder au stockage de persistance pour obtenir des données.

TotalHitCount (colonne 9)
Affiche le nombre total de fois où les données demandées ont été trouvées dans la mappe, dispensant de devoir accéder au stockage de persistance.

StartTime (colonne 10)
Indique l'heure à laquelle l'appel a commencé à partir de la dernière réinitialisation des compteurs. Les réinitialisations se produisent lorsque le serveur démarre ou redémarre.

LastCount (colonne 11)
Indique la durée écoulée depuis le dernier échantillon de données.

LastTotalGetCount (colonne 12)
Indique le nombre total actuel d'opérations d'extraction à partir de la mémoire cache moins le nombre d'opérations d'extraction dans la période précédente.

LastTotalHitCount (colonne 13)
Indique le nombre total actuel d'opérations d'extraction à partir de la mémoire cache moins le nombre d'opérations d'extraction dans la période précédente.

UsedBytes (colonne 14)
Affiche la consommation de la mémoire par cette mappe. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

MinUsedBytes (colonne 15)
Affiche le point bas de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

MaxUsedBytes (colonne 16)
Affiche le point haut de la consommation de mémoire par ce service de catalogue et ses mappes. Les statistiques d'octets utilisés sont exactes uniquement lorsque vous utilisez des objets simples ou le mode de copie COPY_TO_BYTES.

LastUsedBytes (colonne 17)
Indique la valeur UsedBytes en cours moins la valeur UsedBytes à partir de la période de collecte des statistiques précédentes.

SampleLen (colonne 18)
Indique la durée, en millisecondes, de la période d'échantillonnage des données.

Journal de statistiques ObjectGrid

TimeStamp (colonne 1)
Indique la date et l'heure de l'image instantanée des statistiques, prise pour la grille de données.

OgName (colonne 2)
Indique le nom de la grille de données.

PartitionId (colonne 3)
Indique l'ID de la partition.

Count (colonne 4)
Indique le nombre d'échantillons de données collectés depuis le démarrage du serveur. Par exemple, la valeur 100 indique que l'entrée est le 100ème échantillon collecté depuis le démarrage du serveur.

Hostname (colonne 5)
Indique le nom d'hôte.

DomainName (colonne 6)
Indique le domaine du service de catalogue auquel la grille de données appartient.

MaxTime (colonne 7)
Affiche pour ce serveur le temps *maximum* qu'a mis une transaction pour s'exécuter.

MinTime (colonne 8)
Affiche pour ce serveur le temps *minimum* qu'a mis une transaction pour s'exécuter.

MeanTime (colonne 9)
Indique le temps moyen passé sur une transaction.

TotalTime (colonne 10)
Affiche pour ce serveur le temps total passé à des transactions depuis l'initialisation du serveur.

AvgTransTime (colonne 11)
Affiche pour ce serveur la durée moyenne que met une transaction pour s'exécuter.

AvgThroughPut (colonne 12)
Affiche le nombre moyen de transactions par seconde pour ce serveur.

SumOfSquares (colonne 13)
Spécifie la somme des carrés pour le temps de transaction. Cette valeur mesure l'écart par rapport à la moyenne à un moment donné.

SampleLen (colonne 14)
Indique la durée, en millisecondes, de la période d'échantillonnage des données.

LastDataSample (colonne 15)
Indique la durée écoulée depuis le dernier échantillon de données.

LastTotalTime (colonne 16)
Indique le temps total actuel moins le temps total précédent de l'échantillonnage de données.

StartTime (colonne 17)
Indique l'heure à laquelle les statistiques ont commencé à être collectées depuis la dernière réinitialisation des données. Les données sont réinitialisées lorsque le serveur redémarre.

Activation des statistiques

WebSphere eXtreme Scale utilise un modèle de statistiques interne pour suivre et filtrer les données. Toutes les vues de données se basent sur cette structure sous-jacente pour assembler des instantanés des statistiques. Vous pouvez extraire des informations des modules de statistiques à l'aide de plusieurs méthodes.

Pourquoi et quand exécuter cette tâche

Pour obtenir la liste de tous les modules sur lesquels vous pouvez activer les statistiques, voir Classe StatsSpec.

Procédure

- Activez les statistiques à l'aide du fichier de propriétés du serveur. Vous pouvez utiliser la propriété **statsSpec** du fichier de propriétés du serveur pour que le serveur de conteneur définisse la spécification des statistiques au démarrage du serveur. Pour plus d'informations, voir Fichier de propriétés du serveur.
- Activez les statistiques à l'aide de l'utilitaire **xscmd**. Vous pouvez utiliser la commande **-c setStatsSpec** pour définir la spécification des statistiques lors de l'exécution. Pour plus d'informations, voir «Administration avec l'utilitaire **xscmd**», à la page 515.
- Activez les statistiques par voie de programmation à l'aide de l'interface StatsSpec. Pour plus d'informations, voir «Surveillance à l'aide de l'API Statistics», à la page 568.
- Activez les statistiques à l'aide de JMX en utilisant l'opération setStatsSpec dans le DynamicServerMBean. Pour plus d'informations, voir DynamicServerMBean de l'interface.

Exemple

Voici quelques exemples des chaînes statsSpec que vous pouvez spécifier à l'aide du fichier de propriétés, de l'utilitaire **xscmd** ou de l'interface StatsSpec :

Activez toutes les statistiques pour tous les modules :

```
all=enabled
```

Désactivez toutes les statistiques pour tous les modules :

```
all=disabled
```

Activez les statistiques pour toutes les statistiques dans le module OGStatsModule :

```
og.all=enabled
```

Activez les statistiques pour toutes les statistiques dans les modules OGStatsModule et MapStatsModule :

```
og.all=enabled;map.all=enabled
```

Activez les statistiques uniquement pour les statistiques Map Used bytes et désactivez tout le reste :

```
all=disabled;map.usedbytes=enabled
```

Modules des statistiques

WebSphere eXtreme Scale utilise un modèle de statistiques interne pour suivre et filtrer les données. Toutes les vues de données se basent sur cette structure sous-jacente pour assembler des instantanés des statistiques.

Présentation

Dans WebSphere eXtreme Scale, les statistiques sont suivies et stockées dans des composants StatsModules. Le modèle de statistiques contient plusieurs types de modules :

OGStatsModule
: Fournit des statistiques sur une instance ObjectGrid, notamment le temps de réponse des transactions.

MapStatsModule
: Fournit des statistiques sur une mappe unique, notamment le nombre d'entrées et le taux de réussite.

QueryStatsModule
: Fournit des statistiques sur les requêtes, notamment la création de plan et les temps d'exécution.

AgentStatsModule
: Fournit des statistiques sur les agents d'API DataGrid, notamment les temps de sérialisation et d'exécution.

HashIndexStatsModule
: Fournit des statistiques sur la requête HashIndex et les temps d'exécution de maintenance.

SessionStatsModule
: Fournit des statistiques sur le plug-in du gestionnaire de sessions HTTP.

Pour de plus amples informations sur les modules de statistiques, voir l'le package com.ibm.websphere.objectgrid.stats dans la documentation de l'API.

Statistiques dans un environnement local

Le modèle est structuré comme un arbre n-aire (une arborescence dont tous les noeuds sont au même degré) contenant tous les types de modules de statistiques répertoriés dans la liste précédente. Du fait de cette structure, tous les noeuds de l'arborescence sont représentés par l'interface StatsFact. L'interface StatsFact peut représenter un seul module ou un groupe de modules à des fins d'agrégation. Par exemple, si plusieurs noeuds terminaux de l'arborescence représentent des objets MapStatsModule spécifiques, le noeud StatsFact parent de ces noeuds contient les statistiques agrégés pour tous les modules enfants. Une fois l'objet StatsFact extrait, vous pouvez extraire le module de statistiques correspondant à l'aide de l'interface.

A l'instar d'une mappe d'arborescence, vous pouvez utiliser un chemin ou une clé correspondante pour extraire un objet StatsFact spécifique. Le chemin est une valeur String[] qui contient tous les noeuds du chemin de l'objet demandé. Par exemple, supposons que vous avez créé un objet ObjectGrid appelé ObjectGridA, qui contient deux mappes : MapA et MapB. Le chemin du module de statistiques de MapA se présente comme suit : `[ObjectGridA, MapA]`. Le chemin des statistiques agrégées des deux mappes se présente comme suit : `[ObjectGridA]`.

Statistiques dans un environnement réparti

Dans un environnement réparti, les modules de statistiques sont extraits à l'aide d'un chemin différent. Un serveur pouvant contenir plusieurs partitions, l'arborescence de statistiques doit suivre la partition à laquelle chaque module appartient. Le chemin de recherche d'un objet StatsFact spécifique est donc différent. A l'aide de l'exemple précédent, en indiquant que les mappes se trouvent dans la partition 1, utilisez le chemin `[1, ObjectGridA, MapA]` afin d'extraire cet objet StatsFact pour MapA.

Surveillance à l'aide de l'API Statistics

Java

L'API Statistics est l'interface directe avec l'arborescence interne des statistiques. Les statistiques sont désactivées par défaut, mais peuvent être activées en définissant une interface StatsSpec. Une interface StatsSpec définit la manière dont WebSphere eXtreme Scale doit surveiller les statistiques.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'API locale StatsAccessor pour interroger les données et accéder aux statistiques d'une instance ObjectGrid qui se trouve sur la même machine virtuelle Java (JVM) que le code en cours d'exécution. Pour plus d'informations sur les interfaces spécifiques, voir la documentation de l'API. Utilisez les étapes ci-après pour activer la surveillance de l'arborescence des statistiques interne.

Procédure

1. Extrayez l'objet StatsAccessor. L'interface StatsAccessor suit le modèle des singletons. Par conséquent, en dehors des problèmes liés au chargeur de classe, il doit exister une instance StatsAccessor pour chaque JVM. Cette classe sert d'interface principale pour toutes les opérations sur les statistiques locales. Le

code ci-après illustre l'extraction de la classe de l'accesseur. Appelez cette opération avant tout autre appel ObjectGrid.

```
public class LocalClient
{

   public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

   }

}
```

2. Définissez l'interface StatsSpec de grille de données. Définissez cette JVM de sorte qu'elle ne collecte toutes les statistiques qu'au niveau d'ObjectGrid. Vous devez vérifier qu'une application active toutes les statistiques qui peuvent être requises avant de commencer des transactions. L'exemple ci-après définit l'interface StatsSpec à l'aide d'une zone de constante statique et d'une chaîne de spécification. L'utilisation d'une zone de constante statique est plus simple car la zone a déjà défini la spécification. Toutefois, en utilisant une chaîne de spécification, vous pouvez autoriser toutes les combinaisons de statistiques requises.

```
public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      // Définissez la spécification via la chaîne de spécification
      StatsSpec spec = new StatsSpec("og.all=enabled");
      accessor.setStatsSpec(spec);

}
```

3. Envoyez des transactions à la grille pour force la collecte des données en vue de la surveillance. Pour collecter des données utiles pour les statistiques, vous devez envoyer des transactions à la grille de données. L'extrait de code suivant insère un enregistrement dans MapA, qui se trouve dans ObjectGridA. Les statistiques se trouvant au niveau d'ObjectGrid, toute mappe dans l'ObjectGrid renvoie les mêmes résultats.

```
public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      ObjectGridManager manager =
    ObjectGridmanagerFactory.getObjectGridManager();
      ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
      Session session = grid.getSession();
      Map map = session.getMap("MapA");

      // Effectuez une insertion
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();
}
```

4. Interrogez un objet StatsFact à l'aide de l'API StatsAccessor. Tous les chemins d'accès aux statistiques sont associés à une interface StatsFact. L'interface StatsFact est une marque de réservation générique permettant d'organiser et d'inclure un objet StatsModule. Pour que vous puissiez accéder au véritable module de statistiques, l'objet StatsFact doit être extrait.

```
public static void main(String[] args)
{

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      ObjectGridManager manager =
    ObjectGridManagerFactory.getObjectGridManager();
      ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
      Session session = grid.getSession();
      Map map = session.getMap("MapA");

      // Effectuez une insertion
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // Extrayez StatsFact

      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

}
```

5. Interagissez avec l'objet StatsModule. L'objet StatsModule est contenu dans l'interface StatsFact. Vous pouvez obtenir une référence au module à l'aide de l'interface StatsFact. L'interface StatsFact étant une interface générique, vous devez transtyper le module renvoyé dans le type StatsModule attendu. Cette tâche collectant des statistiques eXtreme Scale, l'objet StatsModule renvoyé est transtypé dans un type OGStatsModule. Une fois que le module est transtypé, vous avez accès à toutes les statistiques disponibles.

```
public static void main(String[] args) {

      // extrayez un descripteur de StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Définissez la spéc via la zone statique
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      ObjectGridManager manager =
    ObjectGridmanagerFactory.getObjectGridManager();
      ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
      Session session = grid.getSession();
      Map map = session.getMap("MapA");

      // Effectuez une insertion
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // Extrayez StatsFact
      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

      // Extrayez le module et l'heure
```

```
        OGStatsModule module = (OGStatsModule)fact.getStatsModule();
        ActiveTimeStatistic timeStat =
      module.getTransactionTime("Default", true);
        double time = timeStat.getMeanTime();

}
```

Surveillance avec l'utilitaire xscmd

L'utilitaire **xscmd** remplace l'exemple d'utilitaire **xsadmin** comme outil de surveillance et d'administration complètement pris en charge. Avec l'utilitaire **xscmd**, vous pouvez afficher des informations sous forme de texte à propos de la topologie WebSphere eXtreme Scale..

Avant de commencer

- Pour que l'utilitaire **xscmd** affiche des résultats, vous devez avoir créé votre topologie de grille. Les serveurs de catalogue et les serveurs de conteneur doivent être démarrés. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.
- Voir «Administration avec l'utilitaire **xscmd**», à la page 515 pour plus d'informations sur le démarrage de l'utilitaire **xscmd**.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'utilitaire **xscmd** pour afficher la structure et l'état actuels de la grille de données (par exemple, le contenu de la grille. Dans cet exemple, la structure de grille de données dans cette tâche est une grille de données simple *ObjectGridA* avec la mappe *MapA* qui appartient au groupe de mappes *MapSetA*. Cet exemple montre comment afficher tous les conteneurs dans une grille de données et afficher des mesures filtrées concernant la taille de la mappe *MapA*. Pour afficher toutes les options de la commande, exécutez l'utilitaire **xscmd** sans arguments ou avec l'option **-help**.

Procédure

1. Surveillez l'environnement avec l'utilitaire **xscmd**.
   - Pour activer les statistiques pour tous les serveurs, exécutez la commande suivante :
     - UNIX `./xscmd.sh -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
     - Windows `xscmd.bat -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
   - Pour afficher tous les serveurs de conteneur en ligne pour une grille de données, exécutez la commande suivante :
     - UNIX `./xscmd.sh -c showPlacement -g ObjectGridA -ms MapSetA`
     - Windows `xscmd.bat -c showPlacement -g ObjectGridA -ms MapSetA`

     Toutes les informations sur les conteneurs s'affichent.

     Avertissement : Pour obtenir ces informations lorsque le protocole TLS/SSL (Transport Layer Security/Secure Sockets Layer) est activé, vous devez démarrer les serveurs de catalogue et de conteneur avec le port de service JMX défini. Pour définir le port du service JMX, vous pouvez utiliser l'option **-JMXServicePort** dans le script **startOgServer** ou **startXsServer** ou bien appeler la méthode setJMXServicePort dans l'interface ServerProperties.
   - Pour afficher des informations sur les mappes de la grille de données ObjectGridA, exécutez la commande suivante :

- UNIX `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA`
- Windows `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA`

- Pour vous connecter au service de catalogue et afficher des informations sur la mappe MapA pour l'ensemble du domaine de services de catalogue, exécutez la commande suivante :
  - UNIX `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA -m MapA -cep CatalogMachine:6645`
  - Windows `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA -m MapA -cep CatalogMachine:6645`

  L'utilitaire **xscmd** se connecte au serveur MBean qui s'exécute dans un serveur de catalogue. En vous connectant à un seul serveur de catalogue, vous pouvez extraire des informations sur l'ensemble du domaine de services de catalogue. Un serveur de catalogue peut s'exécuter comme processus autonome, processus WebSphere Application Server ou être intégré dans un processus d'application personnalisé. Utilisez l'option **-cep** pour définir le nom d'hôte et le port du service de catalogue. Si vous incluez une liste de serveurs de catalogue pour l'option **-cep**, les serveurs de catalogue doivent se trouver dans le même domaine de services de catalogue. Vous pouvez extraire des statistiques pour un seul domaine de services de catalogue à la fois.
- Pour afficher le placement configuré et d'exécution de votre configuration, exécutez la commande suivante :
  - `xscmd -c placementServiceStatus`
  - `xscmd -c placementServiceStatus -g ObjectGridA -ms MapSetA`
  - `xscmd -c placementServiceStatus -ms MapSetA`
  - `xscmd -c placementServiceStatus -g ObjectGridA`

  Vous pouvez définir la portée de la commande pour afficher les informations de placement de l'intégralité de la configuration, une grille de données unique, un groupe de mappes unique ou une combinaison de grille de données et de groupe de mappes

2. 8.6+ Affichez les résumés des états de réplication dans l'environnement.
   - Afficher le résumé des révisions en attente de chaque serveur de conteneur. Vous pouvez exécuter la commande sur un serveur de conteneur spécifique avec l'argument **-ct** ou sur tous les serveurs de conteneur si vous n'incluez pas d'argument.
     - UNIX `./xscmd.sh -c showReplicationState -ct container1`
     - Windows `xscmd.bat -c showReplicationState -ct container1`

     Les informations contenues dans la sortie de cette commande inclut la réplication sortante et la réplication entrante. La réplication sortante contient les modifications qui doivent être extraites du fragment principal sur le serveur de conteneur et placées dans ses fragments réplique sur les autres serveurs de conteneur. La réplication entrante contient les modifications qui doivent être extraites des fragments principaux sur les autres serveur de conteneur et placées dans les répliques sur le serveur de conteneur. Ces données statistiques peuvent donner une idée de la santé de réplication. Si le nombre de révisions en suspens sur un serveur de conteneur augmente considérablement, des problèmes au niveau du conteneur peuvent exister.
   - Affichez le résumé des révisions en attente des fragments entre les domaines de service de catalogue. Vous pouvez exécuter la commande sur un serveur

de conteneur spécifique et le domaine de services de catalogue, ou l'intégralité de votre configuration si vous n'incluez pas d'argument.

– UNIX `./xscmd.sh -c showDomainReplicationState -dom domainA -ct container1`

– Windows `xscmd.bat -c showDomainReplicationState -dom domainA -ct container1`

Les informations contenues dans la sortie de cette commande comprend un récapitulatif des révisions en attente de chaque serveur de conteneur pour chaque domaine de service de catalogue lié. La commande renvoie les modifications à répliquer entre chaque fragment primaire et les fragments primaires distants correspondants qui se trouvent dans un autre domaine de service de catalogue.

Surveillance à l'aide de la fonction PMI de WebSphere Application Server

WebSphere eXtreme Scale prend en charge PMI (Performance Monitoring Infrastructure) lorsqu'il est exécuté dans un serveur d'applications WebSphere Application Server ou WebSphere Extended Deployment. PMI collecte des données de performances relatives aux applications exécutables et offre des interfaces permettant aux applications externes de surveiller les données de performances. Vous pouvez utiliser la console d'administration ou l'outil wsadmin pour accéder aux données de surveillance.

Avant de commencer

Vous pouvez utiliser PMI pour surveiller votre environnement lorsque vous utilisez WebSphere eXtreme Scale avec WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

WebSphere eXtreme Scale utilise la fonction PMI personnalisée de WebSphere Application Server pour ajouter sa propre instrumentation de PMI. Avec cette approche, vous pouvez activer et désactiver la fonction PMI de WebSphere eXtreme Scale à l'aide de la console d'administration ou des interfaces JMX (Java Management Extensions) de l'outil wsadmin. En outre, vous pouvez accéder aux statistiques de WebSphere eXtreme Scale à l'aide des interfaces PMI et JMX standard utilisées par les outils de surveillance et notamment Tivoli Performance Viewer.

Procédure

1. Activez la fonction PMI de eXtreme Scale. Vous devez activer PMI pour afficher les statistiques PMI. Pour plus d'informations, voir «Activation de PMI», à la page 574.
2. Extrayez les statistiques PMI de eXtreme Scale. Affichez les performances de vos applications eXtreme Scale à l'aide de Tivoli Performance Viewer. Pour plus d'informations, voir «Récupération des statistiques PMI», à la page 576.

Que faire ensuite

Pour plus d'informations sur l'outil wsadmin, voir «Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin», à la page 537.

Activation de PMI

Vous pouvez utiliser l'infrastructure PMI (Performance Monitoring Infrastructure) de WebSphere Application Server pour activer ou désactiver les statistiques à tout niveau. Par exemple, vous pouvez choisir d'activer les statistiques du nombre d'occurrences d'une mappe donnée, mais non le nombre de statistiques en entrée ou les statistiques de durée de mise à jour par lots du chargeur. Vous pouvez activer PMI dans la console d'administration ou à l'aide de scripts.

Avant de commencer

Votre serveur d'applications doit être démarré et une application compatible eXtreme Scale doit y être installée. Pour activer PMI à l'aide de scripts, vous devez pouvoir vous connecter et utiliser l'outil wsadmin. Pour plus d'informations sur l'outil wsadmin, reportez-vous à la rubrique Outil wsadmin, dans le Centre de documentation de WebSphere Application Server.

Pourquoi et quand exécuter cette tâche

Utilisez l'infrastructure PMI de WebSphere Application Server pour fournir un mécanisme granulaire à l'aide duquel vous pouvez activer ou désactiver les statistiques à tout niveau. Par exemple, vous pouvez choisir d'activer les statistiques du nombre d'occurrences d'une mappe donnée, mais non le nombre d'entrées ou les statistiques de durée de mise à jour par lots du chargeur. Cette section montre comment utiliser la console d'administration et les scripts wsadmin pour activer l'infrastructure PMI d'ObjectGrid.

Procédure

- **Activez PMI dans la console d'administration.**
  1. Dans la console d'administration, cliquez sur **Contrôle et réglage** > **Performance Monitoring Infrastructure** > *nom_serveur*.
  2. Vérifiez que la case Activer l'infrastructure PMI (Performance Monitoring Infrastructure) est cochée. Ce paramètre est activé par défaut. S'il ne l'est pas, cochez la case et redémarrez le serveur.
  3. Cliquez sur **Personnalisé**. Dans l'arborescence de configuration, sélectionnez l'ObjectGrid et le module Mappes d'ObjectGrid. Activez les statistiques de chaque module.

  La catégorie des types de transaction des statistiques ObjectGrid est créée lors de la phase d'exécution. Vous ne pouvez voir que les sous-catégories des statistiques ObjectGrid et des statistiques de mappe dans la page **Exécution**.
- **Activez PMI à l'aide de scripts.**
  1. Ouvrez une invite de ligne de commande. Accédez au répertoire *racine_was*/bin. Entrez **wsadmin** pour démarrer l'outil de ligne de commande wsadmin.
  2. Modifiez la configuration de l'environnement d'exécution de l'infrastructure PMI d'eXtreme Scale. Vérifiez que PMI est activé pour le serveur à l'aide des commandes suivantes :

     ```
     wsadmin>set s1 [$AdminConfig getid /Cell:CELL_NAME/Node:NODE_NAME/
         Server:APPLICATION_SERVER_NAME/]
     wsadmin>set pmi [$AdminConfig list PMIService $s1]
     wsadmin>$AdminConfig show $pmi.
     ```

     Si PMI n'est pas activé, exécutez les commandes suivantes pour activer PMI :

     ```
     wsadmin>$AdminConfig modify $pmi {{enable true}}
     wsadmin>$AdminConfig save
     ```

Si vous avez besoin d'activer PMI, redémarrez le serveur.

3. Définissez des variables pour modifier l'ensemble de statistiques en ensemble personnalisé à l'aide des commandes suivantes :

```
wsadmin>set perfName [$AdminControl completeObjectName type=Perf,
process=APPLICATION_SERVER_NAME,*]
wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
wsadmin>set params [java::new {java.lang.Object[]} 1]
wsadmin>$params set 0 [java::new java.lang.String custom]
wsadmin>set sigs [java::new {java.lang.String[]} 1]
wsadmin>$sigs set 0 java.lang.String
```

4. Spécifiez un ensemble de statistiques personnalisé à l'aide de la commande suivante :

```
wsadmin>$AdminControl invoke_jmx $perfOName setStatisticSet $params $sigs
```

5. Définissez des variables pour activer les statistiques de l'infrastructure PMI d'objectGridModule à l'aide des commandes suivantes :

```
wsadmin>set params [java::new {java.lang.Object[]} 2]
wsadmin>$params set 0 [java::new java.lang.String objectGridModule=1]
wsadmin>$params set 1 [java::new java.lang.Boolean false]
wsadmin>set sigs [java::new {java.lang.String[]} 2]
wsadmin>$sigs set 0 java.lang.String
wsadmin>$sigs set 1 java.lang.Boolean
```

6. Définissez la chaîne des statistiques à l'aide de la commande suivante :

```
wsadmin>set params2 [java::new {java.lang.Object[]} 2]
wsadmin>$params2 set 0 [java::new java.lang.String mapModule=*]
wsadmin>$params2 set 1 [java::new java.lang.Boolean false]
wsadmin>set sigs2 [java::new {java.lang.String[]} 2]
wsadmin>$sigs2 set 0 java.lang.String
wsadmin>$sigs2 set 1 java.lang.Boolean
```

7. Définissez la chaîne des statistiques à l'aide de la commande suivante :

```
wsadmin>$AdminControl invoke_jmx $perfOName setCustomSetString $params2 $sigs2
```

Ces étapes activent l'infrastructure PMI de l'environnement d'exécution d'eXtreme Scale, mais ne modifient pas la configuration de l'infrastructure PMI. Si vous redémarrez l'application, les paramètres PMI sont perdus, exceptée l'activation principale de PMI.

Exemple

Vous pouvez effectuer les étapes suivantes pour activer les statistiques PMI de l'exemple d'application :

1. Lancez l'application à l'aide de l'adresse Web `http://hôte:port/ObjectGridSample`, hôte et port correspondant au nom d'hôte et au numéro de port HTTP du serveur où l'exemple est installé.
2. Dans l'exemple d'application, cliquez sur ObjectGridCreationServlet, puis sur les boutons d'action 1, 2, 3, 4 et 5 pour générer des actions sur l'ObjectGrid et les mappes. Ne fermez pas tout de suite cette page de servlet.
3. Dans la console d'administration, cliquez sur **Contrôle et réglage** > **Performance Monitoring Infrastructure** > *nom_serveur*. Cliquez sur l'onglet **Exécution**.
4. Cliquez sur le bouton d'option **Personnalisé**.
5. Développez le module Mappes d'ObjectGrid dans l'arborescence d'exécution, puis cliquez sur le lien clusterObjectGrid. Le groupe Mappes d'ObjectGrid contient une instance ObjectGrid appelée clusterObjectGrid et le groupe clusterObjectGrid contient quatre mappes : counters, employees, offices, et sites. Dans l'instance ObjectGrids se trouve une instance clusterObjectGrid et sous cette instance, le type de transaction DEFAULT.

6. Vous pouvez activer les statistiques de votre choix. Par exemple, vous pouvez activer le nombre d'entrées de mappe pour la mappe des employés et le temps de réponse des transactions pour le type de transaction DEFAULT.

Que faire ensuite

Une fois que PMI est activé, vous pouvez afficher les statistiques PMI à l'aide de la console d'administration ou de scripts.

Récupération des statistiques PMI

En récupérant les statistiques PMI, vous pouvez voir les performances de vos applications eXtreme Scale.

Avant de commencer

- Activez la fonction de suivi des statistiques PMI pour votre environnement. Pour plus d'informations, voir «Activation de PMI», à la page 574.
- Les chemins dans cette tâche partent du principe que vous récupérez les statistiques pour l'exemple d'application, mais vous pouvez utiliser ces statistiques pour toute autre application avec des étapes similaires.
- Si vous utilisez la console d'administration, vous devez être capable de vous y connecter. Si vous utilisez un script, vous devez être capable de vous connecter à wsadmin.

Pourquoi et quand exécuter cette tâche

Vous pouvez récupérer les statistiques PMI pour les afficher dans Tivoli Performance Viewer en suivant les étapes dans la console d'administration ou par script.

- Etapes de la console d'administration
- Etapes du script

Pour plus d'informations concernant les statistiques qui peuvent être récupérées, voir «Modules PMI», à la page 577.

Procédure

- Récupérez les statistiques PMI dans la console d'administration.
  1. Dans la console d'administration, cliquez sur **Contrôle et réglage** > **Performance viewer** > **Activité actuelle**
  2. Sélectionnez le serveur que vous voulez contrôler à l'aide de Tivoli Performance Viewer, puis activez le contrôle.
  3. Cliquez sur le serveur pour afficher la page Performance viewer.
  4. Développez l'arborescence de configuration. Cliquez sur **ObjectGrid Maps** > **clusterObjectGrid**, sélectionnez **employés**. Développez **ObjectGrids** > **clusterObjectGrid** et sélectionnez **DEFAULT**.
  5. Dans le modèle d'application ObjectGrid, accédez au servlet ObjectGridCreationServlet, cliquez sur le bouton 1 et remplissez les mappes. Vous pouvez afficher les statistiques dans l'afficheur.
- Récupérez les statistiques PMI avec un script.
  1. Dans une invite de ligne de commande, accédez au répertoire *racine_was*`/bin`. Entrez `wsadmin` pour lancer l'outil wsadmin.
  2. Définissez les variables pour l'environnement à l'aide des commandes suivantes :

```
wsadmin>set perfName [$AdminControl completeObjectName type=Perf,*]
wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
wsadmin>set mySrvName [$AdminControl completeObjectName type=Server,
 name=APPLICATION_SERVER_NAME,*]
```

3. Définissez les variables pour obtenir les statistiques de mapModule à l'aide des commandes suivantes :

```
wsadmin>set params [java::new {java.lang.Object[]} 3]
wsadmin>$params set 0 [$AdminControl makeObjectName $mySrvName]
wsadmin>$params set 1 [java::new java.lang.String mapModule]
wsadmin>$params set 2 [java::new java.lang.Boolean true]
wsadmin>set sigs [java::new {java.lang.String[]} 3]
wsadmin>$sigs set 0 javax.management.ObjectName
wsadmin>$sigs set 1 java.lang.String
wsadmin>$sigs set 2 java.lang.Boolean
```

4. Obtenez les statistiques de mapModule à l'aide de la commande suivante :

```
wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params $sigs
```

5. Définissez les variables pour obtenir les statistiques d'objectGridModule à l'aide des commandes suivantes :

```
wsadmin>set params2 [java::new {java.lang.Object[]} 3]
wsadmin>$params2 set 0 [$AdminControl makeObjectName $mySrvName]
wsadmin>$params2 set 1 [java::new java.lang.String objectGridModule]
wsadmin>$params2 set 2 [java::new java.lang.Boolean true]
wsadmin>set sigs2 [java::new {java.lang.String[]} 3]
wsadmin>$sigs2 set 0 javax.management.ObjectName
wsadmin>$sigs2 set 1 java.lang.String
wsadmin>$sigs2 set 2 java.lang.Boolean
```

6. Obtenez les statistiques d'objectGridModule à l'aide de la commande suivante :

```
wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params2 $sigs2
```

Résultats

Vous pouvez afficher les statistiques dans Tivoli Performance Viewer.

Modules PMI

Vous pouvez surveiller les performances de vos applications avec les modules PMI (Performance Monitoring Infrastructure).

objectGridModule

Le module objectGridModule contient une statistique de durée : le temps de réponse des transactions. Une transaction est définie comme la durée entre l'appel de méthode Session.begin et l'appel de méthode Session.commit. Cette durée est suivie comme temps de réponse des transactions. L'élément racine de la structure objectGridModule, "root", sert de point d'entrée aux statistiques de WebSphere eXtreme Scale. Cet élément racine contient des ObjectGrids comme éléments enfant et ces derniers possèdent des types de transaction comme éléments enfant. Les statistiques de temps de réponse sont associées à chaque type de transaction.

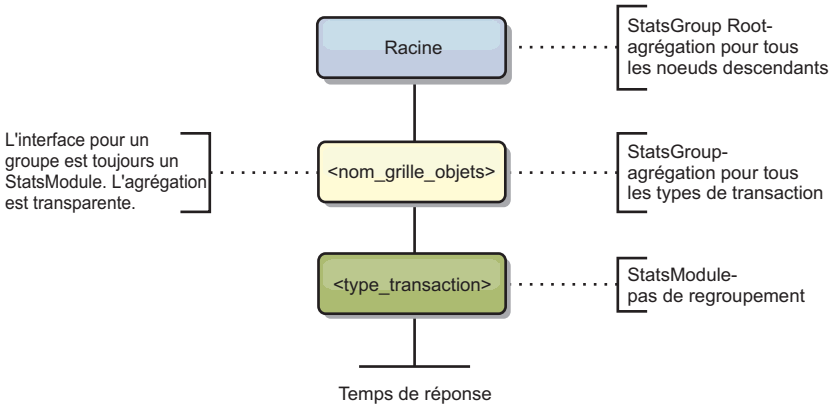


Figure 61. Structure de module ObjectGridModule

Le diagramme ci-après illustre un exemple de structure ObjectGridModule. Dans cet exemple, il existe deux instances ObjectGrid sur le système : ObjectGrid A et ObjectGrid B. L'instance ObjectGrid A possède deux types de transaction : A et default. L'instance ObjectGrid B ne possède que le type de transaction default.

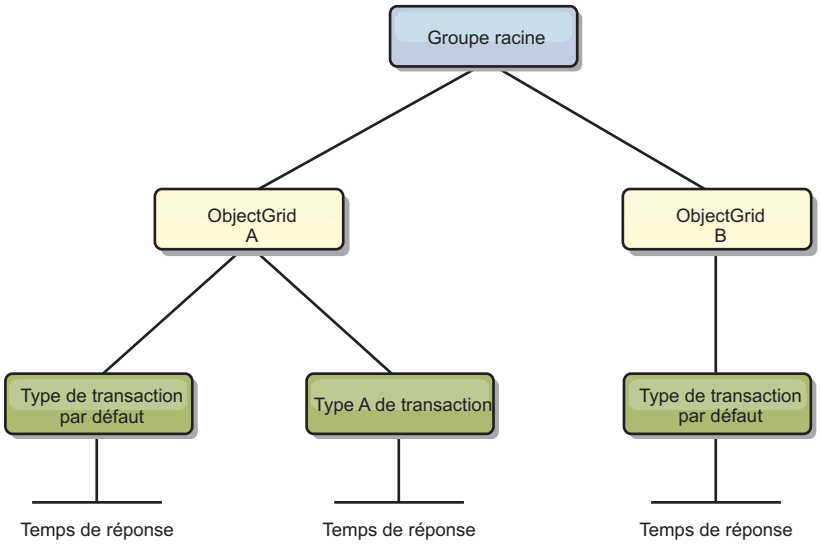


Figure 62. Exemple de structure de module ObjectGridModule

Les types de transaction sont définis par les développeurs d'applications car ils savent quels types de transaction sont utilisés par leurs applications. Le type de transaction est défini à l'aide de la méthode Session.setTransactionType(String) suivante :

```
/**
* Définit le type de transaction des transactions futures.
*
* Une fois que cette méthode a été appelée, toutes les transactions futures sont de
* même type jusqu'à ce qu'un autre type de transaction ait été défini. Si aucun type
* de transaction n'est défini, le type de transaction TRANSACTION_TYPE_DEFAULT
* par défaut est utilisé.
*
* Les types de transaction sont principalement utilisés à des fins de suivi des
* données statistiques.
* Les utilisateurs peuvent prédéfinir les types des transactions qui sont exécutées
* dans une application. L'idée consiste à regrouper les transactions de mêmes
* caractéristiques
* dans une même catégorie (type), afin qu'une statistique de temps de réponse
* des transactions puisse être utilisée pour rechercher chaque type de transaction.
*
* Ce suivi est utile si votre application possède différents types de
* transactions.
* Parmi eux, certains types de transaction, comme les transactions de mise à
* jour, possèdent un délai de traitement supérieur à celui d'autres
```

```
* transactions, telles que les transactions en lecture seule. Si le type de
* transaction est utilisé, les différentes transactions sont recherchées par des
* statistiques différentes, afin que ces dernières puissent être plus utiles.
*
* @param tranType Type de transaction des transactions futures.
*/
void setTransactionType(String tranType);
```

L'exemple suivant spécifie `updatePrice` comme type de transaction :

```
// Spécifiez le type de transaction updatePrice
// La durée entre session.begin() et session.commit() fait l'objet d'un suivi
// dans les statistiques de durée de "updatePrice".
session.setTransactionType("updatePrice");
session.begin();
map.update(stockId, new Integer(100));
session.commit();
```

La première ligne indique que le type de transaction suivant est updatePrice. Il existe une statistiques updatePrice sous l'instance ObjectGrid qui correspond à la session de l'exemple. A l'aide d'interfaces JMX (Java Management Extensions), vous pouvez obtenir le temps de réponse des transactions updatePrice. Vous pouvez également extraire les statistiques agrégées de tous les types de transaction sur l'instance ObjectGrid spécifiée.

mapModule

La structure mapModule contient trois statistiques sur les mappes eXtreme Scale :

- **Nombre d'occurrences de mappe** - *BoundedRangeStatistic* : Recherche le nombre d'occurrences d'une mappe. Le nombre d'occurrences est une valeur flottante comprise entre 0 et 100 compris, qui représente le pourcentage d'occurrences de mappe en relation avec les opérations d'extraction de mappe.
- **Nombre d'entrées** - *CountStatistic* : Recherche le nombre d'entrées dans la mappe.
- **Temps de réponse de la mise à jour par lots du chargeur** - *TimeStatistic* : Recherche le temps de réponse utilisé pour l'opération de mise à jour par lots du chargeur.

L'élément racine de la structure mapModule, "root", sert de point d'entrée aux statistiques des mappes ObjectGrid. Cet élément racine contient des ObjectGrids comme éléments enfant et ces derniers possèdent des mappes comme éléments enfant. Trois statistiques sont répertoriées pour chaque instance de mappe. La structure mapModule est illustrée dans le diagramme suivant :

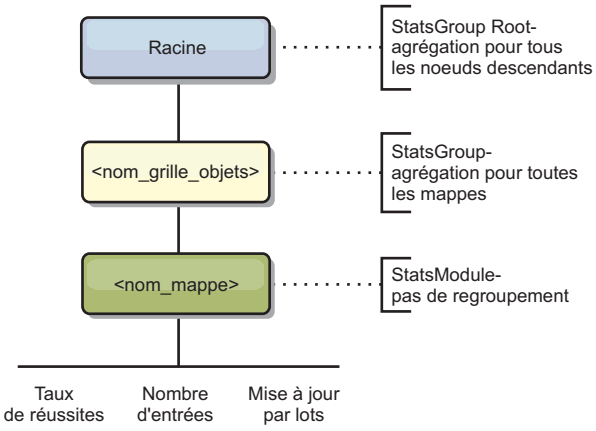
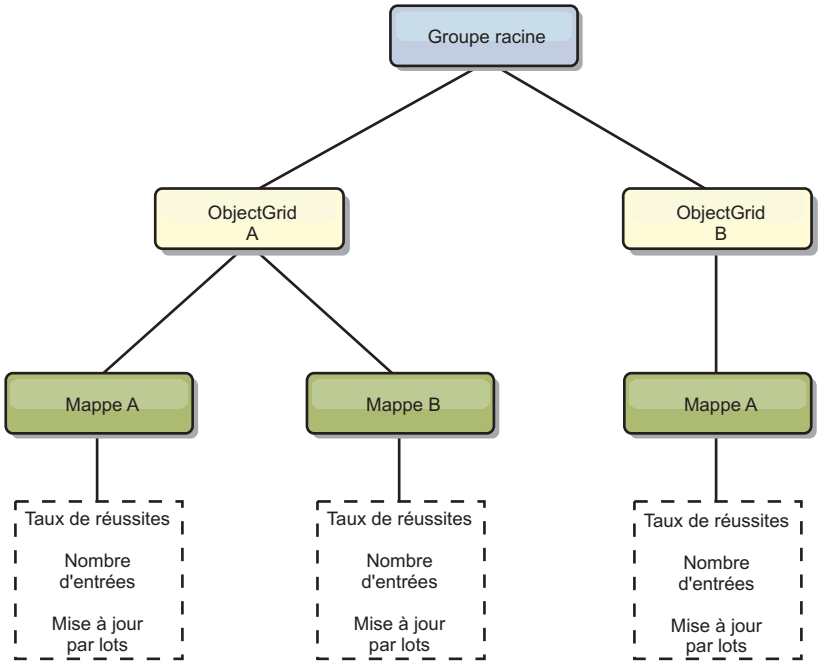


Figure 63. structure mapModule

Le diagramme suivant illustre un exemple de structure mapModule :

Figure 64. Exemple de structure de module mapModule



hashIndexModule

La structure hashIndexModule contient les statistiques suivantes sur les index de niveau mappe :

- **Nombre de recherches** - *CountStatistic* : Nombre d'appels de l'opération de recherche d'index.
- **Nombre de collisions** - *CountStatistic* : Nombre de collisions de l'opération de recherche.
- **Nombre d'échecs** - *CountStatistic* : Nombre d'échecs pour l'opération de recherche.
- **Nombre de résultats** - *CountStatistic* : Nombre de clés renvoyées par l'opération de recherche.
- **Nombre de mises à jour par lots** - *CountStatistic* : Nombre de mises à jour par lots sur cet index. Si la mappe correspondante es modifié qu'une quelconque manière, la méthode doBatchUpdate() de l'index est appelée. Cette statistiques indique la fréquence à laquelle votre index est modifié ou mis à jour.
- **Durée de recherche** - *TimeStatistic* : Temps que prend l'opération de recherche pour s'exécuter

L'élément racine de la structure hashIndexModule, "root", sert de point d'entrée aux statistiques de HashIndex. Cet élément racine contient des ObjectGrids comme éléments enfant, les ObjectGrids contiennent des mappes comme éléments enfant et enfin, ces mappes contiennent des instances HashIndex comme éléments enfant et les noeuds terminaux de l'arborescence. Trois statistiques sont répertoriées pour chaque instance HashIndex. La structure hashIndexModule est illustrée dans le diagramme suivant :

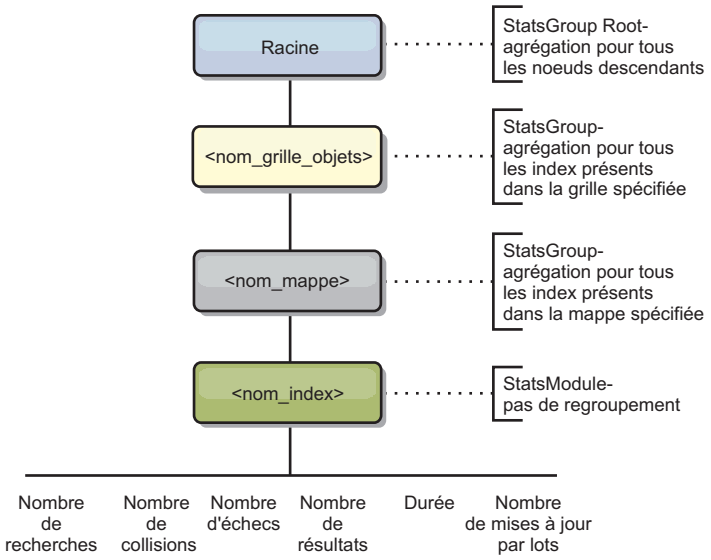


Figure 65. structure de module hashIndexModule

Le diagramme suivant illustre un exemple de structure hashIndexModule :

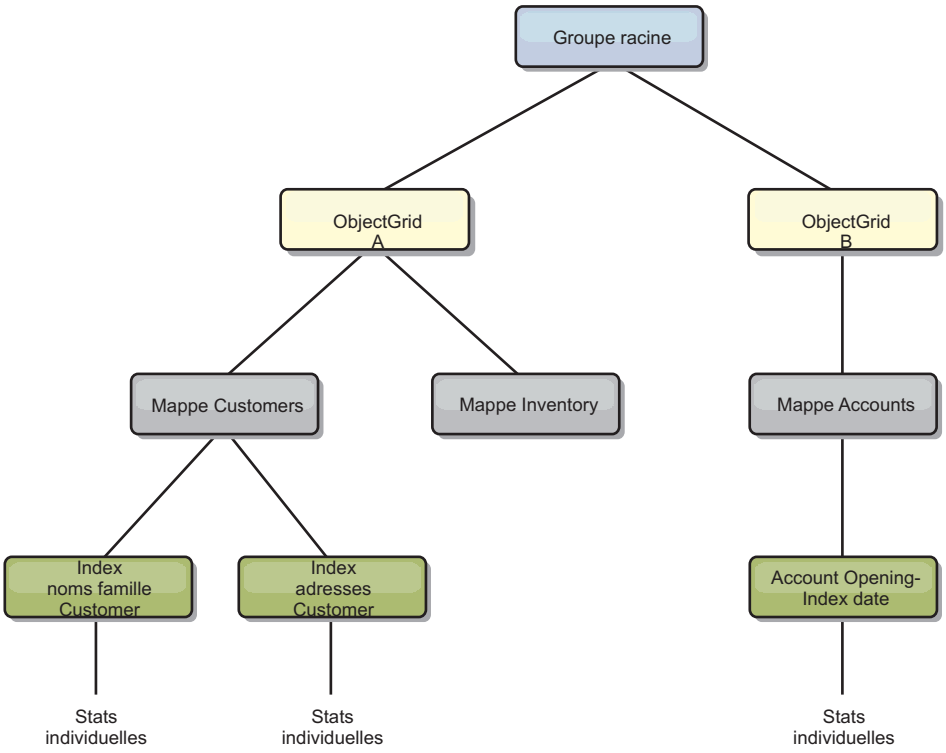


Figure 66. Exemple de structure de module hashIndexModule

agentManagerModule

La structure agentManagerModule contient les statistiques sur les agents de niveau mappe :

- **Temps de réduction** : *TimeStatistic* - Durée nécessaire pour que l'agent termine l'opération de réduction.

- **Durée totale** : *TimeStatistic* - Durée totale nécessaire à l'agent pour effectuer toutes les opérations.
- **Temps de sérialisation de l'agent** : *TimeStatistic* - Durée nécessaire pour sérialiser l'agent.
- **Temps d'inflation de l'agent** : *TimeStatistic* - Durée nécessaire pour l'inflation de l'agent sur le serveur.
- **Temps de sérialisation des résultats** : *TimeStatistic* - Durée nécessaire pour sérialiser les résultats de l'agent.
- **Temps d'inflation des résultats** : *TimeStatistic* - Durée nécessaire pour l'inflation des résultats de l'agent.
- **Nombre d'échecs** : *CountStatistic* - Nombre de fois que l'agent a échoué.
- **Nombre d'appels** : *CountStatistic* - Nombre d'appels d'AgentManager.
- **Nombre de partitions** : *CountStatistic* - Nombre de partitions vers lesquelles l'agent est envoyé.

L'élément racine de la structure agentManagerModule, "root", sert de point d'entrée aux statistiques d'AgentManager. Cet élément racine contient des ObjectGrids comme éléments enfant, les ObjectGrids contiennent des mappes comme éléments enfant et enfin, ces mappes contiennent des instances AgentManager comme éléments enfant et les noeuds terminaux de l'arborescence. Chaque instance AgentManager a des statistiques.

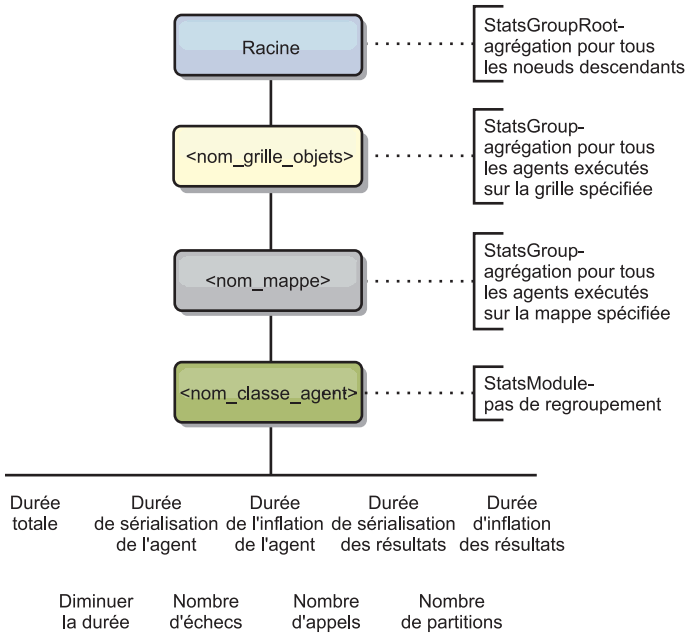


Figure 67. Structure agentManagerModule

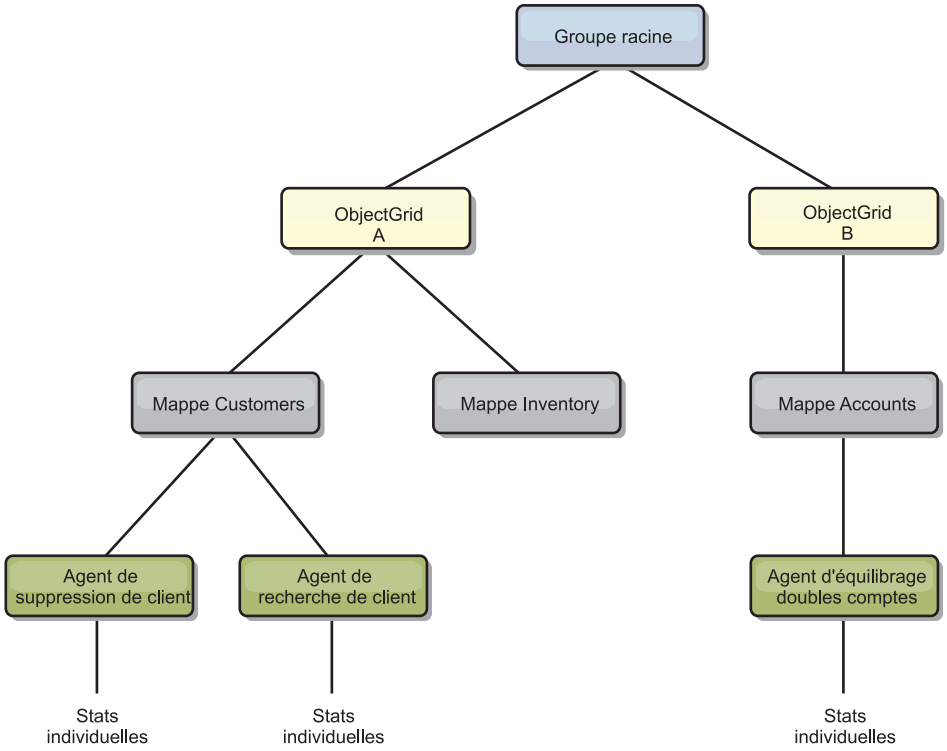


Figure 68. Exemple de structure agentManagerModule

queryModule

La structure queryModule contient les statistiques sur les requêtes eXtreme Scale :

- **Temps de création du plan** : *TimeStatistic* - Durée nécessaire pour créer le plan de requête.
- **Temps d'exécution** : *TimeStatistic* - Durée nécessaire pour exécuter la requête.
- **Nombre d'exécutions** : *CountStatistic* - Nombre de fois que la requête a été exécutée.
- **Nombre de résultats** : *CountStatistic* - Nombre de résultats pour chaque ensemble de résultats de chaque exécution de requête.
- **Nombre d'échecs** : *CountStatistic* - Nombre de fois que la requête a échoué.

L'élément racine de la structure queryModule, "root", sert de point d'entrée aux statistiques des requêtes. Cet élément racine contient des ObjectGrids comme éléments enfant et ces derniers possèdent des objets de requête comme éléments enfant et les noeuds terminaux de l'arborescence. Trois statistiques sont répertoriées pour chaque instance de requête.

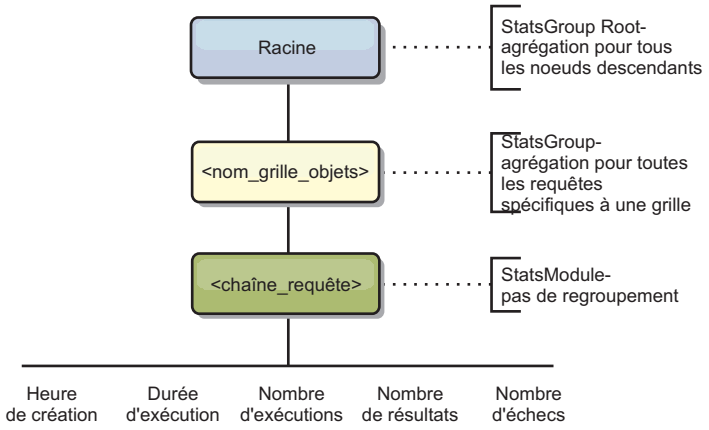


Figure 69. structure queryModule

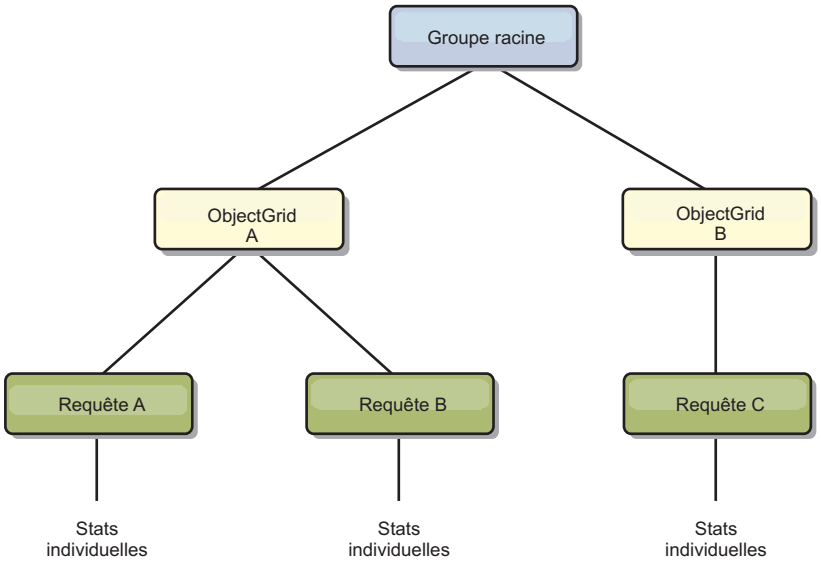


Figure 70. Exemple de structure queryModule QueryStats.jpg

Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin

Java

Vous pouvez utiliser l'utilitaire wsadmin fourni dans WebSphere Application Server pour accéder aux informations des beans gérés (MBean).

Procédure

Exécutez l'outil wsadmin depuis le répertoire `bin` dans votre installation WebSphere Application Server. L'exemple suivant restaure une vue de la position actuelle du fragment dans un logiciel eXtreme Scale dynamique. Vous pouvez exécuter l'outil wsadmin depuis n'importe quelle installation où eXtreme Scale est en cours d'exécution. Vous n'avez pas besoin d'exécuter l'outil wsadmin sur le service de catalogue.

```
$ wsadmin.sh -lang jython
wsadmin>placementService = AdminControl.queryNames
 ("com.ibm.websphere.objectgrid:*,type=PlacementService")
wsadmin>print AdminControl.invoke(placementService,
```

```
 "listObjectGridPlacement","library ms1")

<objectGrid name="library" mapSetName="ms1">
  <container name="container-0" zoneName="DefaultDomain"
  hostName="host1.company.org" serverName="server1">
     <shard type="Primary" partitionName="0"/>
     <shard type="SynchronousReplica" partitionName="1"/>
  </container>
  <container name="container-1" zoneName="DefaultDomain"
  hostName="host2.company.org" serverName="server2">
     <shard type="SynchronousReplica" partitionName="0"/>
     <shard type="Primary" partitionName="1"/>
  </container>
  <container name="UNASSIGNED" zoneName="_ibm_SYSTEM"
  hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
  </container>
</objectGrid>
```

Surveillance des statistiques de serveur avec des beans gérés (MBeans)

Java

Vous pouvez utiliser des beans gérés (MBeans) pour effectuer le suivi des statistiques dans votre environnement.

Avant de commencer

Pour que les attributs puissent être enregistrés, vous devez activer les statistiques. Vous pouvez activer les statistiques sur le serveur ou activer les statistiques de session HTTP pour suivre les attributs dans l'application client. Pour plus d'informations sur l'activation des statistiques de session HTTP, voir xref.

Vous pouvez activer les statistiques de l'une des manières suivantes :

- **A l'aide du fichier de propriétés du serveur :**

  Vous pouvez activer les statistiques dans le fichier de propriétés du serveur avec l'entrée clé-valeur `statsSpec=<SpécStats>`. Voici quelques exemples de paramètres possibles :

  - pour activer toutes les statistiques, utilisez `statsSpec=all=enabled`
  - pour n'activer que les statistiques d'ObjectGrid, utilisez `statsSpec=og.all=enabled` Pour une description de toutes les spécifications de statistiques possibles, voir l'API StatsSpec dans la documentation de l'API.

  Pour plus d'informations sur le fichier de propriétés du serveur, voir Fichier de propriétés du serveur.

- **Avec un bean géré :**

  Vous pouvez activer les statistiques à l'aide de l'attribut StatsSpec dans le bean géré ObjectGrid. Pour plus d'informations, voir l'API StatsSpec dans la documentation d'API.

- **Par programmation :**

  Vous pouvez également programmer l'activation des statistiques avec l'interface StatsAccessor, qui est extraite avec la classe StatsAccessorFactory. Utilisez cette interface dans un environnement client ou lorsque vous devez surveiller une grille de données qui s'exécute dans le processus en cours.

Procédure

- **Accédez aux statistiques de bean géré en utilisant l'outil wsadmin.**

  Pour plus d'informations, voir «Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin», à la page 537.
- **Accédez aux statistiques de bean géré à l'aide d'un programme.**

  Pour plus d'informations, voir «Accès aux beans gérés (MBeans) à l'aide d'un programme», à la page 538.

Exemple

Pour obtenir un exemple d'utilisation des beans gérés, voir Exemple : utilitaire **xsadmin**.

Surveillance des statistiques de session HTTP client

8.6+

Vous pouvez surveiller l'activité des sessions utilisateur des applications Web qui s'exécutent sur votre serveur.

Avant de commencer

Activez la gestion de session HTTP dans WebSphere eXtreme Scale. Pour plus d'informations, voir Gestion des sessions HTTP.

Pourquoi et quand exécuter cette tâche

Vous pouvez surveiller l'activité de session HTTP dans les types suivants d'installations :

- Installation autonome dans laquelle WebSphere eXtreme Scale utilise un autre serveur d'application
- Installation intégrée dans laquelle WebSphere eXtreme Scale utilise WebSphere Application Server comme serveur d'applications

Selon la manière dont vous avez déployé WebSphere eXtreme Scale, vous pouvez surveiller différents types d'activités de session :

- Lorsque vous utilisez WebSphere eXtreme Scale avec un autre serveur d'applications, vous pouvez surveiller les compteurs suivants :

Tableau 37. Types de statistiques de session HTTP

Nom	Description
createCount	Nombre de sessions créées.
invalidateCount	Nombre de sessions ayant été invalidées.
activeCount	Nombre de sessions simultanément actives. Une session est active si WebSphere Application Server traite une demande qui utilise cette session.
liveCount	Nombre de sessions locales actuellement stockées en cache depuis le moment où cette valeur a été activée
cacheDiscardCount	Nombre d'objets session qui ont été supprimés de la mémoire cache. Un algorithme LRU (least recently used) supprime les anciennes entrées pour faire de la place aux nouvelles sessions et aux échecs en mémoire cache. Ne s'applique qu'aux sessions persistantes.

Tableau 37. Types de statistiques de session HTTP (suite)

Nom	Description
affinityBreakCount	Nombre de demandes reçues pour les dernières sessions auxquelles une autre application Web a accédé. Cette valeur peut indiquer un traitement de basculement ou une configuration de plug-in endommagée.
timeoutInvalidationCount	Nombre de sessions qui sont invalidées par une expiration de délai.
activateNonExistSessionCount	Nombre de demandes pour une session qui n'existe plus, parce qu'elle est vraisemblablement arrivée à expiration. Ce compteur aide à déterminer si le délai d'expiration est trop court.

- Lorsque vous utilisez WebSphere Application Server, vous pouvez surveiller les compteurs suivants : Compteurs de session de servlet.

Procédure

Selon la manière dont vous avez déployé WebSphere eXtreme Scale, vous pouvez activer les statistiques de session client HTTP de l'une des manières suivantes :

- Si vous avez installé WebSphere eXtreme Scale dans un environnement autonome, activez les statistiques de session client HTTP avec le fichier `splicer.properties`.
  - Affectez à la propriété de statistique `enableSessionStats=true` la valeur true à la place de la valeur false.
  - Affectez à la propriété de statistique `sessionStatsSpec=session.all=enabled` la valeur all.

  Pour plus d'informations, voir «Fichier `splicer.properties`», à la page 397. Pour la description de toutes les spécifications de statistiques possibles, voir API StatsSpec.
- Si vous avez installé WebSphere eXtreme Scale pour la réplication de session sur WebSphere Application Server, activez l'activité de surveillance de session HTTP avec le service PMI (Performance Monitoring Infrastructure) dans WebSphere Application Server. Pour plus d'informations, voir Activation de PMI en utilisant la console d'administration.

  Remarque : Même si vous envisagez d'utiliser la console PMI pour surveiller l'activité de la session dans WebSphere Application Server, vous pouvez également activer l'activité de session HTTP dans WebSphere eXtreme Scale pour vous assurer que vous êtes en mesure de surveiller les types de statistiques de session dans les deux produits.

Que faire ensuite

Une fois que vous avez activé les statistiques de session HTTP dans WebSphere eXtreme Scale, vous pouvez afficher les statistiques via le bean géré, `com.ibm.websphere.objectgrid:type=Session,name=webAppContextRoot`

Vous pouvez afficher le bean géré en utilisant l'un des outils suivants :

- **Accédez aux statistiques de bean géré en utilisant l'outil wsadmin.**

  Pour plus d'informations, voir «Accès aux beans gérés (MBeans) à l'aide de l'outil wsadmin», à la page 537.
- **Accédez aux statistiques de bean géré à l'aide d'un programme.**

  Pour plus d'informations, voir «Accès aux beans gérés (MBeans) à l'aide d'un programme», à la page 538.

- **Accédez aux statistiques de bean géré avec des outils tels que JConsole (Java Monitoring and Management Console).**

Surveillance à l'aide d'outils fournis par une tierce partie

WebSphere eXtreme Scale peut être surveillé à l'aide de plusieurs solutions de surveillance d'entreprise couramment utilisées. Des agents de plug-in sont intégrés pour IBM Tivoli Monitoring et Hyperic HQ dont le rôle consiste à surveiller WebSphere eXtreme Scale à l'aide de beans de gestion accessibles publiquement. CA Wily Introscope utilise l'instrumentation de méthode Java pour capturer les statistiques.

Surveillance à l'aide d'IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale

L'agent IBM Tivoli Enterprise Monitoring est une solution de surveillance riche en fonctions que vous pouvez utiliser pour surveiller les bases de données, les systèmes d'exploitation et les serveurs dans des environnements hôte et répartis. WebSphere eXtreme Scale inclut un agent personnalisé que vous pouvez utiliser pour introspecter les beans de gestion d'eXtreme Scale. Cette solution fonctionne correctement pour les déploiements eXtreme Scale autonomes et les déploiements WebSphere Application Server.

Avant de commencer

- Installez WebSphere eXtreme Scale Version 7.0.0 ou ultérieure.

  Par ailleurs, les statistiques doivent être activées pour permettre la collecte de données statistiques à partir des serveurs WebSphere eXtreme Scale. Les diverses options d'activation des statistiques sont décrites dans «Surveillance des statistiques de serveur avec des beans gérés (MBeans)», à la page 585 et dans Exemple : utilitaire **xsadmin**.
- Installez IBM Tivoli Version 6.2.1 avec le Fix Pack 2 ou ultérieur.
- Installez l'agent du système d'exploitation Tivoli sur chaque serveur ou hôte sur lequel des serveurs eXtreme Scale sont exécutés.
- Installez l'agent WebSphere eXtreme Scale, que vous pouvez télécharger gratuitement à partir du site IBM Open Process Automation Library (OPAL).

Effectuez les étapes suivantes pour installer et configurer Tivoli Monitoring Agent :

Procédure

1. Installez Tivoli Monitoring Agent for WebSphere eXtreme Scale.

   Téléchargez l'image d'installation de Tivoli et extrayez ses fichiers dans un répertoire temporaire.
2. Installez les fichiers du support d'application d'eXtreme Scale.

   Installez le support d'application d'eXtreme Scale sur chacun des déploiements ci-après.
   - Tivoli Enterprise Portal Server (TEPS)
   - Client Enterprise Desktop (TEPD)
   - Tivoli Enterprise Monitoring Server (TEMS)

   a. Dans le répertoire temporaire que vous avez créé, démarrez une nouvelle fenêtre de commande et exécutez le fichier exécutable approprié pour votre plateforme. Le script d'installation détecte automatiquement votre type de déploiement Tivoli (TEMS, TEPD ou TEPS). Vous pouvez installer tout type

de déploiement sur un ou plusieurs hôtes ; ces trois types de déploiement requièrent l'installation des fichiers du support d'application de l'agent eXtreme Scale.

b. Dans la fenêtre **Programme d'installation**, vérifiez que la sélection des composants Tivoli déployés est correcte. Cliquez sur **Suivant**.

c. Si vous y êtes invité, soumettez votre nom d'hôte et vos données d'identification administratives. Cliquez sur **Suivant**.

d. Sélectionnez **Monitoring Agent for WebSphere eXtreme Scale**. Cliquez sur **Suivant**.

e. Le système vous indique les actions d'installation à effectuer. Cliquez sur **Suivant** ; vous pouvez voir la progression de l'installation jusqu'à sa fin.

Une fois que vous avez terminé cette procédure, tous les fichiers du support d'application requis par l'agent WebSphere eXtreme Scale sont installés.

3. Installez l'agent sur chacun des noeuds eXtreme Scale.

   Vous installez un agent de système d'exploitation Tivoli sur chacun des ordinateurs. Vous n'avez pas besoin de configurer ou de démarrer cet agent. Utilisez l'image d'installation de l'étape précédente pour exécuter le fichier exécutable spécifique à la plateforme.

   Vous n'avez besoin d'installer qu'un seul agent par hôte. Chaque agent peut prendre en charge plusieurs instances de serveur eXtreme Scale. Pour de meilleures performances, utilisez une instance d'agent pour surveiller environ 50 serveurs eXtreme Scale.

   a. Dans l'écran de bienvenue de l'assistant d'installation, cliquez sur **Suivant** pour ouvrir l'écran et spécifier les informations sur le chemin d'installation.

   b. Dans la zone **Répertoire d'installation d'IBM Tivoli Monitoring**, entrez ou recherchez `C:\IBM\ITM` (ou `/opt/IBM/ITM`). Ensuite, dans la zone de l'**emplacement du support installable**, vérifiez que la valeur affichée est correcte et cliquez sur **Suivant**.

   c. Sélectionnez les composants à ajouter, tels que **Effectuer une installation locale de la solution**, et cliquez sur **Suivant**.

   d. Sélectionnez les applications pour lesquelles vous souhaitez ajouter le support en les sélectionnant (par exemple, **Monitoring Agent for WebSphere eXtreme Scale**), puis en cliquant sur **Suivant**.

   e. La progression s'affiche jusqu'à ce que la prise en charge de l'application ait été ajoutée.

   Remarque : Répétez ces étapes sur chacun des noeuds eXtreme Scale. Vous pouvez également utiliser une installation en mode silencieux. Pour plus d'informations sur l'installation en mode silencieux, voir le Centre de documentation d'IBM Tivoli Monitoring.

4. Configurez l'agent WebSphere eXtreme Scale.

   Chacun des agents installés doit être configuré pour surveiller un serveur de catalogues et/ou un serveur eXtreme Scale.

   Les étapes de configuration des plateformes Windows et UNIX sont différentes. La configuration de la plateforme Windows s'exécute avec l'interface utilisateur **Manage Tivoli Monitoring Services**. La configuration des plateformes UNIX s'effectue depuis la ligne de commande.

   Windows Procédez comme suit pour configurer initialement l'agent sous Windows.

a. Dans la fenêtre **Gérer les services Tivoli Enterprise Monitoring**, cliquez sur **Démarrer** > **Tous les programmes** > **IBM Tivoli Monitoring** > **Gérer les services Tivoli Monitoring**.
b. Cliquez à l'aide du bouton droit de la souris sur **Monitoring Agent for WebSphere eXtreme Scale** et sélectionnez **Configure using default**, qui ouvre une fenêtre permettant de créer une instance unique de l'agent.
c. Choisissez un nom unique (par exemple, `instance1` et cliquez sur **Suivant**).

- Si vous prévoyez de surveiller des serveurs eXtreme Scale autonomes, effectuez les étapes suivantes :
  a. Mettez à jour les paramètres Java et assurez-vous que la valeur **Java Home** est correcte. Les arguments JVM peuvent rester vides. Cliquez sur **Suivant**.
  b. Sélectionnez le type **Type de connexion de serveur MBean** et utilisez `Serveur conforme à JSR-160` pour les serveurs eXtreme Scale autonomes. Cliquez sur **Suivant**.
  c. Si la sécurité est activée, mettez à jour les valeurs **ID utilisateur** et **Mot de passe**. Ne modifiez pas la valeur d'**URL de service JMX**. Vous remplacerez cette valeur ultérieurement. Ne modifiez pas la zone **Informations de chemin de classes JMX**. Cliquez sur **Suivant**.

  Pour configurer les serveurs pour l'agent sous Windows, procédez comme suit :
  a. Configurez des instances de sous-noeud des serveurs eXtreme Scale dans la sous-fenêtre **Serveurs de grille WebSphere eXtreme Scale**. S'il n'existe pas de serveurs de catalogue sur votre ordinateur, cliquez sur **Suivant** pour passer à la sous-fenêtre du service de catalogue.
  b. S'il existe plusieurs serveurs de conteneur eXtreme Scale sur votre ordinateur, configurez l'agent pour qu'il surveille chacun d'eux.
  c. Vous pouvez ajouter autant de serveurs eXtreme Scale que nécessaire, si leurs noms et ports sont uniques, en cliquant sur **Nouveau**. (Si un serveur eXtreme Scale est démarré, une valeur JMXPort doit être spécifiée.)
  d. Une fois que vous avez configuré les serveurs de conteneur, cliquez sur **Suivant** pour accéder à la sous-fenêtre **Serveurs de catalogue WebSphere eXtreme Scale**.
  e. En l'absence de serveurs de catalogue, cliquez sur **OK**. Si vous possédez des serveurs de catalogue, ajoutez une nouvelle configuration pour chaque serveur, comme pour les serveurs de conteneur. Choisissez de nouveau un nom unique, de préférence, celui utilisé au démarrage du service de catalogue. Cliquez sur **OK** pour terminer.
- Si vous prévoyez de surveiller les serveurs de l'agent sur des serveurs eXtreme Scale imbriqués dans un processus WebSphere Application Server, procédez comme suit :
  a. Mettez à jour les paramètres Java et assurez-vous que la valeur **Java Home** est correcte. Les arguments JVM peuvent rester vides. Cliquez sur **Suivant**.
  b. Sélectionnez le **type de connexion du serveur MBean**. Sélectionnez la version WebSphere Application Server qui convient pour votre environnement. Cliquez sur **Suivant**.
  c. Vérifiez que les informations WebSphere Application Server du panneau sont correctes. Cliquez sur **Suivant**.
  d. N'ajoutez qu'une définition de sous-noeud. Nommez cette définition de sous-noeud, mais ne mettez pas à jour la définition du port. Dans un

environnement WebSphere Application Server, les données peuvent être collectées sur tous les serveurs d'applications gérés par l'agent de noeud exécuté sur l'ordinateur. Cliquez sur **Suivant**.

e. S'il n'existe pas de serveur de catalogue dans l'environnement, cliquez sur **OK**. S'il en existe, ajoutez une nouvelle configuration pour chaque serveur de catalogues, comme pour les serveurs de conteneur. Choisissez un nom unique pour le service de catalogue, de préférence, celui que vous avez utilisé au démarrage du service de catalogue. Cliquez sur **OK** pour terminer.

Remarque : Les serveurs de conteneur n'ont pas besoin d'être regroupés avec le service de catalogue.

Maintenant que l'agent et les serveurs sont configurés et prêts, dans la fenêtre qui suit, cliquez à l'aide du bouton droit de la souris sur `instance1` pour démarrer l'agent.

UNIX Pour configurer l'agent sur la plateforme UNIX sur la ligne de commande, procédez comme suit :

Un exemple est illustré ci-après pour les serveurs autonomes qui utilisent un type de connexion compatible JSR160. Cet exemple illustre trois conteneurs eXtreme Scale sur l'hôte unique (rhea00b02) et les adresses du programme d'écoute JMX sont respectivement 15000, 15001 et 15002. Il n'existe pas de serveur de catalogue.

La sortie de l'utilitaire de configuration est affichée en *italiques à espacement fixe*, tandis que la réponse de l'utilisateur est en **gras à espacement fixe**. (Si aucune réponse utilisateur n'est requise, la valeur par défaut est sélectionnée en appuyant sur la touche Entrée.)

```
rhea00b02 # ./itmcmd config -A xt
Configuration de l'agent démarrée...
Entrez un nom d'instance (la valeur par défaut est : ) : inst1
Modifiez les paramètres "Agent de surveillance pour WebSphere eXtreme Scale" ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :
Modifier les paramètres 'Java' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :
Répertoire de base Java (la sélection par défaut est C:\Program Files\IBM\Java50) : /opt/OG61/java
Niveau de trace Java [ 1=Erreur, 2=Avertissement, 3=Information, 4=Débogage minimum, 5=Débogage moyen, 6=Débogage maximum,
    7=Tous ] (la sélection par défaut est 1) :
Arguments JVM (la sélection par défaut est ):
Modifier les paramètres 'Connexion' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :
Type de connexion de serveur MBean [ 1=Serveur conforme à JSR-160,
2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server
version 7.0 ] (la sélection par défaut est 1) : 1
Modifier les paramètres 'Serveur conforme à JSR-160' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :
ID utilisateur JMX (la sélection par défaut est ) :
Entrez Mot de passe JMX (la sélection par défaut est ) :
Entrez de nouveau : Mot de passe JMX (la sélection par défaut est ) :
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:port/objectgrid/MBeanServer) :
----------------------------------------
Informations de chemin de classe JMX
Chemins de base JMX (la sélection par défaut est ) :
Chemin de classes JMX (la sélection par défaut est ) :
Répertoires JAR JMX  (la sélection par défaut est ) :
Modifier les paramètres 'Service de catalogue WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) : 2
Modifier les paramètres 'Serveurs de grille WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) : 1
Aucun paramètre 'Serveurs de grille WebSphere eXtreme Scale' disponible
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier,
3=Supprimer, 4=Suivant, 5=Quitter] (la sélection par défaut est : 4) : 1
WebSphere eXtreme Scale Grid Servers (la valeur par défaut est ): rhea00b02_c0
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :
service:jmx:rmi:///jndi/rmi://localhost:15000/objectgrid/MBeanServer

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' :  WebSphere eXtreme Scale Grid Servers=ogx
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier,
3=Supprimer, 4=Suivant, 5=Quitter] (la sélection par défaut est : 4) : 1
Serveurs de grille WebSphere eXtreme Scale (la sélection par défaut est ) : rhea00b02_c1
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :
```

```
service:jmx:rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' :  WebSphere eXtreme Scale Grid Servers= rhea00b02_c1
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier,
3=Supprimer, 4=Suivant, 5=Quitter] (la sélection par défaut est : 4) : 1
Serveurs de grille WebSphere eXtreme Scale (la sélection par défaut est ) : rhea00b02_c2
Adresse URL de service JMX (la valeur par défaut est : service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :
service:jmx:rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' :  WebSphere eXtreme Scale Grid Servers= rhea00b02_c2
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier,
3=Supprimer, 4=Suivant, 5=Quitter] (la sélection par défaut est : 4) : 5

Cet agent se connectera-t-il à un TEMS ? [1=OUI, 2=NON]
(la sélection par défaut est 1) :
Nom d'hôte TEMS (la sélection par défaut est rhea00b00) :

Protocole de réseau [ip, sna, ip.pipe ou ip.spipe] (la sélection par défaut est ip.pipe) :

     Choisissez maintenant le prochain numéro de protocole parmi l'un des suivants :
     - ip
     - sna
     - ip.spipe
     - 0 pour aucun
Protocole de réseau 2 (la sélection par défaut est 0) :
Numéro de port IP.PIPE (la sélection par défaut est 1918) :
Entrez le nom de KDC_PARTITION (la sélection par défaut est null) :

Configurer la connexion TEMS secondaire ? [1=OUI, 2=NON] (la sélection par défaut est 2) :
Entrez Nom du réseau primaire optionnel ou 0 pour "aucun" (la sélection par défaut est 0) :
Configuration de l'agent terminée...
```

L'exemple précédent crée une instance d'agent appelée “inst1” et met à jour les paramètres de Java Home. Les serveurs de conteneur eXtreme Scale sont configurés, mais le service de catalogue n'est pas configuré.

Remarque : La procédure précédente crée un fichier texte au format suivant dans le répertoire : `<install_ITM>/config/<hôte>_xt_<nom de l'instance>.cfg`.

Exemple : `rhea00b02_xt_inst1.cfg`

Il est recommandé d'éditer ce fichier à l'aide de l'éditeur de texte en clair de votre choix. Voci un exemple de contenu d'un tel fichier :

```
INSTANCE=inst2 [SECTION=KQZ_JAVA [ { JAVA_HOME=/opt/OG61/java }  { JAVA_TRACE_LEVEL=ERROR }  ]
SECTION=KQZ_JMX_CONNECTION_SECTION [ { KQZ_JMX_CONNECTION_PROPERTY=KQZ_JMX_JSR160_JSR160 }  ]
SECTION=KQZ_JMX_JSR160_JSR160 [ { KQZ_JMX_JSR160_JSR160_CLASS_PATH_TITLE= }
{ KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:rmi:///jndi/rmi://localho
st:port/objectgrid/MBeanServer }  { KQZ_JMX_JSR160_JSR160_CLASS_PATH_SEPARATOR= }  ]
SECTION=OGS:rhea00b02_c1 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer }  ]
SECTION=OGS:rhea00b02_c0 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer }  ]
SECTION=OGS:rhea00b02_c2 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer }  ]]
```

Voici un exemple illustrant une configuration sur un déploiement WebSphere Application Server :

```
rhea00b02 # ./itmcmd config -A xt
Configuration de l'agent démarrée...
Entrez un nom d'instance (la valeur par défaut est : ) : inst1
Modifiez les paramètres "Agent de surveillance pour WebSphere eXtreme Scale" ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) : 1
Modifier les paramètres 'Java' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) : 1
Java home (default is: C:\Program Files\IBM\Java50): /opt/WAS61/java
Niveau de trace Java [ 1=Erreur, 2=Avertissement, 3=Information, 4=Débogage minimum, 5=Débogage moyen, 6=Débogage maximum,
    7=Tous ] (la sélection par défaut est 1) :
Arguments JVM (la sélection par défaut est ):
Modifier les paramètres 'Connexion' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :
Type de connexion de serveur MBean [ 1=Serveur conforme à JSR-160, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ]
(la sélection par défaut est 1) : 4
Modifier les paramètres 'WebSphere Application Server version 7.0' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :ID utilisateur WAS (la sélection par défaut est ) :
Entrez Mot de passe WAS (la sélection par défaut est ) :
Entrez de nouveau : Mot de passe WAS (la sélection par défaut est ) :
Nom d'hôte WAS (la sélection par défaut est localhost) : rhea00b02
```

```
Port WAS (la sélection par défaut est 2809) :
Protocole de connecteur WAS [ 1=rmi, 2=soap ] (la sélection par défaut est 1) :
Nom de profil WAS (la sélection par défaut est ) : valeur par défaut
---------------------------------------
Informations de chemin de classe WAS
Chemins de base WAS (la sélection par défaut est
C:\Program Files\IBM\WebSphere\AppServer;/opt/IBM/WebSphere/AppServer) : /opt/WAS61
Chemin de classes WAS (la sélection par défaut est
runtimes/com.ibm.ws.admin.client_6.1.0.jar;runtimes/com.ibm.ws.ejb.thinclient_7.0.0.jar) :
Répertoires JAR WAS (la sélection par défaut est lib;plugins) :
Modifier les paramètres 'Serveurs de grille WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) :
Aucun paramètre 'Serveurs de grille WebSphere eXtreme Scale' disponible
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier,
3=Supprimer, 4=Suivant, 5=Quitter] (la sélection par défaut est : 4) : 1
Serveurs de grille WebSphere eXtreme Scale (la sélection par défaut est ) : rhea00b02
Adresse URL de service JMX (la valeur par défaut est :
service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer) :

Paramètres 'Serveurs de grille WebSphere eXtreme Scale' settings:  WebSphere eXtreme Scale Grid Servers=rhea00b02
Modifiez les paramètres 'Serveurs de grille WebSphere eXtreme Scale', [1=Ajouter, 2=Modifier,
3=Supprimer, 4=Suivant, 5=Quitter] (la sélection par défaut est : 4) : 5
Modifier les paramètres 'Service de catalogue WebSphere eXtreme Scale' ? [ 1=Oui, 2=Non ]
(la sélection par défaut est 1) : 2
Cet agent se connectera-t-il à un TEMS ? [1=OUI, 2=NON]
(la sélection par défaut est 1) :
Nom d'hôte TEMS (la sélection par défaut est rhea00b02) :

Protocole de réseau [ip, sna, ip.pipe ou ip.spipe] (la sélection par défaut est ip.pipe) :

     Choisissez maintenant le prochain numéro de protocole parmi l'un des suivants :
     - ip
     - sna
     - ip.spipe
     - 0 pour aucun
Protocole de réseau 2 (la sélection par défaut est 0) :
Numéro de port IP.PIPE (la sélection par défaut est 1918) :
Entrez le nom de KDC_PARTITION (la sélection par défaut est null) :

Configurer la connexion TEMS secondaire ? [1=OUI, 2=NON] (la sélection par défaut est 2) :
Entrez Nom du réseau primaire optionnel ou 0 pour "aucun" (la sélection par défaut est 0) :
Configuration de l'agent terminée...
rhea00b02 #
```

Pour les déploiements WebSphere Application Server, vous n'avez pas besoin de créer plusieurs sous-noeuds. L'agent eXtreme Scale se connecte à l'agent de noeud pour collecter toutes les informations des serveurs d'applications dont il est responsable.

`SECTION=CAT` signifie une ligne de service de catalogue, tandis que `SECTION=OGS` signifie une ligne de configuration de serveur eXtreme Scale.

5. Configurez le port JMX pour tous les serveurs de conteneur eXtreme Scale.

   Si des serveurs de conteneur eXtreme Scale sont démarrés, sans l'argument **-JMXServicePort**, un serveur MBean reçoit un port dynamique. L'agent doit savoir à l'avance avec quel port JMX communiquer. L'agent ne fonctionne pas avec des ports dynamiques.

   Lorsque vous démarrez le serveur, vous devez spécifier l'argument `-JMXServicePort <numéro_port>` lorsque vous démarrez le serveur eXtreme Scale à l'aide de la commande de démarrage de serveur. L'exécution de cette commande garantit que le serveur JMX du processus écoute sur un port statique prédéfini.

   Pour les exemples précédents dans l'installation UNIX, deux serveurs eXtreme Scale doivent être démarrés avec des ports définis :

   a. "-JMXServicePort" "15000" (pour rhea00b02_c0)
   b. "-JMXServicePort" "15001" (pour rhea00b02_c1)
   a. Démarrez l'agent eXtreme Scale.

En supposant que l'instance `inst1` ait été créée, comme dans l'exemple précédent, exécutez les commandes ci-après.

1) `cd <install_ITM>/bin`
2) `itmcmd agent –o inst1 start xt`

b. Arrêtez l'agent eXtreme Scale.

En supposant que l'instance “inst1” correspond à l'instance créée, comme dans l'exemple précédent, exécutez les commandes ci-après.

1) `cd <install_ITM>/bin`
2) `itmcmd agent –o inst1 stop xt`

6. Activez les statistiques pour tous les serveurs de conteneur eXtreme Scale.

Pour enregistrer les statistiques, l'agent utilise les beans gérés de statistiques eXtreme Scale. La spécification des statistiques eXtreme Scale doit être activée à l'aide de l'une des méthodes suivantes :

- en configurant les propriétés des serveurs pour activer toutes les statistiques au démarrage de la totalité des serveurs : all=enabled
- à l'aide de l'utilitaire d'exemple xsadmin pour activer les statistiques pour tous les conteneurs actifs : paramètres -setstatsspec all=enabled

Résultats

Une fois que tous les serveurs sont configurés et démarrés, les données des beans gérés sont affichées sur la console d'IBM Tivoli Portal. Les espaces de travail prédéfinis montrent les graphiques et mesures de données au niveau de chaque noeud.

Les espaces de travail suivants sont définis : **noeud eXtreme Scale serveurs de grilles** pour tous les noeuds surveillés.

- vue Transactions eXtreme Scale
- vue Fragment primaire eXtreme Scale
- vue Mémoire eXtreme Scale
- vue ObjectMap eXtreme Scale

Vous pouvez également configurer votre propre espace de travail. Pour plus d'informations, reportez-vous aux informations sur la personnalisation des espaces de travail, dans le centre de documentation d'IBM Tivoli Monitoring.

Surveillance des applications eXtreme Scale à l'aide de CA Wily Introscope

CA Wily Introscope est un produit de gestion tiers qui permet de détecter et de diagnostiquer les problèmes de performances dans les environnements d'application d'entreprise. eXtreme Scale inclut des détails sur la configuration de CA Wily Introscope pour introspecter certaines portions de l'environnement d'exécution de eXtreme Scale afin d'afficher et de valider rapidement les applications eXtreme Scale. CA Wily Introscope fonctionne de manière efficace pour les déploiements autonomes et WebSphere Application Server.

Présentation

Pour surveiller les applications eXtreme Scale avec CA Wily Introscope, vous devez placer des paramètres dans les fichiers PBD (ProbeBuilderDirective) qui vous permettent d'accéder aux informations de surveillance de eXtreme Scale.

Avertissement : Les points d'instrumentation d'Introscope peuvent changer avec chaque corriger ou version. Lorsque vous installez un nouveau groupe de correctifs ou une nouvelle version, recherchez dans la documentation les modifications apportées aux points d'instrumentation.

Vous pouvez configurer des fichiers PBD (ProbeBuilderDirective) de CA Wily Introscope pour surveiller vos applications eXtreme Scale. CA Wily Introscope est un produit de gestion des applications à l'aide duquel vous pouvez détecter, prioriser et diagnostiquer de manière proactive les problèmes de performances dans vos environnements d'application Web, composite et complexe.

Paramètres des fichiers PBD pour la surveillance du service de catalogue

Vous pouvez utiliser un ou plusieurs des paramètres ci-après dans votre fichier PBD pour surveiller le service de catalogue.

```
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl changeDefinedCompleted
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewChangeCompleted
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewAboutToChange
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeat
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCluster
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCurrentLeader
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatDeadServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewLeader
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl
importRouteInfo BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl heartbeat
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl joinPlacementGroup
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}" TraceOneMethodOfClass:
com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl classifyServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.BalanceGridEventListener shardActivated
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.BalanceGridEventListener shardDeactivate
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
```

Classes de surveillance du service de catalogue

HAControllerImpl
La classe HAControllerImpl gère le cycle de vie du groupe central et les événements de retour d'informations. Vous pouvez surveiller cette classe pour déterminer les modifications et la structure du groupe central.

ServerAgent
La classe ServerAgent est chargée de communiquer les événements du groupe central avec le service de catalogue. Vous pouvez surveiller les divers appels de signal de présence pour identifier les événements principaux.

PlacementServiceImpl
La classe PlacementServiceImpl coordonne les conteneurs. Vous pouvez utiliser les méthodes de cette classe pour surveiller les événements de jointure et de positionnement.

BalanceGridEventListener
La classe BalanceGridEventListener contrôle la position de leader du catalogue. Vous pouvez surveiller cette classe pour déterminer quel service de catalogue sert actuellement de leader.

Paramètres des fichiers PBD pour la surveillance des conteneurs

Vous pouvez utiliser un ou plusieurs des paramètres ci-après dans votre fichier PBD pour surveiller les conteneurs.

```
TraceOneMethodOfClass: com.ibm.ws.objectgrid.ShardImpl processMessage
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.plugins.CommittedLogSequenceListenerProxy applyCommitted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.plugins.CommittedLogSequenceListenerProxy sendApplyCommitted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.map.BaseMap evictMapEntries
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.checkpoint.CheckpointMapImpl$CheckpointIterator activateListener
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl changeDefinedCompleted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewChangeCompleted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewAboutToChange
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent batchProcess
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeat
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCluster
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCurrentLeader
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatDeadServer
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewLeader
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewServer
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
```

Classes de surveillance des conteneurs

ShardImpl

La classe ShardImpl contient la méthode processMessage. La méthode processMessage est celle des demandes client. Avec cette méthode, vous pouvez obtenir les temps de réponse côté serveur et le nombre de demandes. En observant les résultats sur tous les serveurs et en surveillant l'utilisation des segments de mémoire, vous pouvez déterminer si la grille est équilibrée.

CheckpointIterator

La classe CheckpointIterator contient l'appel de méthode activateListener qui place les fragments primaires en mode homologue. Lorsque les fragments primaires sont placés en mode homologue, le fragment réplique est au même niveau que le fragment primaire une fois la méthode exécutée. Lorsqu'une réplique est régénérée à partir d'un fragment primaire complet, cette opération peut durer un certain temps. Le système n'ayant pas intégralement récupéré tant que cette opération n'est pas terminée, vous pouvez utiliser cette classe pour surveiller la progression de l'opération.

CommittedLogSequenceListenerProxy

La classe CommittedLogSequenceListenerProxy contient deux méthodes intéressantes. La méthode applyCommitted est exécutée pour chaque transaction et la méthode sendApplyCommitted est exécutée lorsque le fragment réplique extrait des informations. Le ratio de fréquence d'exécution de ces deux méthodes peut vous indiquer dans quelle mesure le fragment réplique est capable de suivre le fragment primaire.

Paramètres des fichiers PBD pour la surveillance des clients

Vous pouvez utiliser un ou plusieurs des paramètres ci-après dans votre fichier PBD pour surveiller les clients.

```
TraceOneMethodOfClass: com.ibm.ws.objectgrid.client.ORBClientCoreMessageHandler sendMessage
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.corba.cluster.ClusterStore bootstrap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.corba.cluster.ClusterStore epochChangeBootstrap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.map.BaseMap evictMapEntries
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.cluster.orb.routing.SelectionServiceImpl routeFailed
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.cluster.orb.routing.SelectionServiceImpl routeFailed
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.SessionImpl getMap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.ObjectGridImpl getSession
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TurnOn: ObjectMap
SetFlag: ObjectMap
IdentifyClassAs: com.ibm.ws.objectgrid.ObjectMapImpl ObjectMap
TraceComplexMethodsifFlagged: ObjectMap BlamePointTracerDifferentMethods
"OGclient|{classname}|{method}"
```

Classes de surveillance des clients

ORBClientCoreMessageHandler
La classe ORBClientCoreMessageHandler est chargée d'envoyer les demandes d'application aux conteneurs. Vous pouvez surveiller la méthode sendMessage pour le temps de réponse des clients et le nombre de demandes.

ClusterStore
La classe ClusterStore contient les informations de routage côté client.

BaseMap
La classe BaseMap contient la méthode evictMapEntries qui est appelée lorsque l'expulseur souhaite supprimer des entrées de la mappe.

SelectionServiceImpl
La classe SelectionServiceImpl effectue les décisions de routage. Si le client décide de procéder à un basculement, vous pouvez utiliser cette classe pour afficher les actions réalisées à partir de ces décisions.

ObjectGridImpl
La classe ObjectGridImpl contient la méthode getSession que vous pouvez surveiller pour afficher le nombre de demandes pour cette méthode.

Surveillance d'eXtreme Scale à l'aide de Hyperic HQ

Hyperic HQ est une solution de surveillance tiers disponible gratuitement comme solution à code source ouvert ou produit d'entreprise. WebSphere eXtreme Scale inclut un plug-in qui permet aux agents Hyperic HQ de reconnaître les serveurs de conteneur eXtreme Scale et de fournir et regrouper des statistiques à l'aide de beans de gestion eXtreme Scale. Vous pouvez utiliser Hyperic HQ pour surveiller les déploiements eXtreme Scale autonomes.

Avant de commencer

- Ce jeu d'instructions concerne Hyperic Version 4.0. Si vous possédez une version plus récente de Hyperic, reportez-vous à la documentation de Hyperic pour plus d'informations, telles que les noms de chemin et la méthode de démarrage des agents et des serveurs.
- Téléchargez les installations des agents et serveurs Hyperic. Une installation de serveur doit être en cours d'exécution. Pour détecter tous les serveurs eXtreme Scale, un agent Hyperic doit être en cours d'exécution sur chaque machine sur laquelle un serveur eXtreme Scale est en cours d'exécution. Pour les informations de téléchargement et le support de documentation, voir le site Web Hyperic.
- Vous devez avoir accès aux fichiers `objectgrid-plugin.xml` et `hqplugin.jar`. Ces fichiers se trouvent dans le répertoire *racine_install_wxs*`/hyperic/etc`.

Pourquoi et quand exécuter cette tâche

En intégrant eXtreme Scale au logiciel de surveillance Hyperic HQ, vous pouvez surveiller et afficher graphiquement les mesures sur les performances de votre environnement. Vous configurez cette intégration en utilisant une implémentation de plug-in sur chaque agent.

Procédure

1. Démarrez vos serveurs eXtreme Scale. Le plug-in Hyperic recherche les processus locaux à connecter aux machines virtuelles Java qui exécutent eXtreme Scale. Pour se connecter correctement aux machines virtuelles Java, chaque serveur doit être démarré avec l'option **-jmxServicePort**. Pour plus d'informations sur le démarrage des serveurs à l'aide de l'option **-jmxServicePort**, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.
2. Placez le fichier `extremescale-plugin.xml` et le fichier `wxshyperic.jar` dans les répertoires de plug-in appropriés du serveur et des agents, dans votre configuration Hyperic. Pour être intégrées à Hyperic, les installations des agents et du serveur doivent avoir accès au plug-in et aux fichiers JAR (archive Java). Le serveur peut permuter dynamiquement les configurations, mais vous devez effectuer l'intégration avant de démarrer l'un des agents.
   a. Placez le fichier `extremescale-plugin.xml` dans le répertoire `plugin` du serveur, qui se trouve à l'emplacement suivant :
      hyperic_home`/server_home/hq-engine/server/default/deploy/hq.ear/hq-plugins`
   b. Placez le fichier `extremescale-plugin.xml` dans le répertoire `plugin` de l'agent, qui se trouve à l'emplacement suivant :
      agent_home`/bundles/gent-4.0.2-939/pdk/plugins`
   c. Placez le fichier `wshyperic.jar` dans le répertoire `lib` de l'agent, qui se trouve à l'emplacement suivant :
      agent_home`/bundles/gent-4.0.2-939/pdk/lib`
3. Configurez l'agent. Le fichier `agent.properties` fait office de point de configuration pour l'environnement d'exécution de l'agent. Cette propriété se trouve dans le répertoire *rép_base_agent*`/conf`. Les clés suivantes sont facultatives, mais importantes pour le plug-in eXtreme Scale :
   - `autoinventory.defaultScan.interval.millis=<`*durée_en_millisecondes*`>`

     Définit l'intervalle en millisecondes entre les reconnaissances d'Agent.
   - `log4j.logger.org.hyperic.hq.plugin.extremescale.XSServerDetector=DEBUG`

     : Active les instructions de débogage prolixes à partir du plug-in eXtreme Scale.
   - `username=<`*username*`>` : Définit le nom d'utilisateur JMX (Java Management Extensions) si la sécurité est activée.
   - `password=<`*motdepasse*`>` : Définit le mot de passe JMX si la sécurité est activée.
   - `sslEnabled=<`*true*`|`*false*`>` : Indique au plug-in s'il doit utiliser SSL (Secure Sockets Layer). La valeur est `false` par défaut.
   - `trustPath=<`*chemin*`>` : Définit le chemin sécurisé de la connexion SSL.
   - `trustType=<`*type*`>` : Définit le type sécurisé de la connexion SSL.

- `trustPass=<motdepasse>` : Définit le mot de passe sécurisé de la connexion SSL.

4. Démarrez la reconnaissance des agents. Les agents Hyperic envoient des informations de reconnaissance et des mesures au serveur. Utilisez le serveur pour personnaliser les vues de données et regrouper les objets d'inventaire logiques afin de générer des informations utiles. Une fois que le serveur est disponible, vous devez exécuter le script de lancement ou démarrer le service Windows pour l'agent :
   - Linux *agent_home*`/bin/hq-agent.sh start`
   - Windows Démarrez l'agent avec le service Windows.

   Une fois que vous avez démarré les agents, les serveurs sont détectés et les groupes sont configurés. Vous pouvez vous connecter à la console du serveur et choisir les ressources à ajouter à la base de données d'inventaire du serveur. La console du serveur se trouve à l'URL suivante par défaut : `http://<nom_hôte_serveur>:7080/`
5. Les statistiques doivent être activées pour que Hyperic puisse collecter des données statistiques.

   Utilisez l'action de contrôle **SetStatsSpec** sur la console Hyperic pour eXtreme Scale. Allez à la ressource, puis utilisez la liste déroulante **Action de contrôle** sous l'onglet **Contrôle** afin de spécifier un paramètre `SetStatsSpec` avec `ALL=enabled` dans la zone de texte **Arguments du contrôle**.

   Les serveurs de catalogue ne sont pas détectés par le filtre défini sur la console Hyperic. Voir les informations concernant la propriété **statsSpec** dans Fichier de propriétés du serveur, qui activent les statistiques au démarrage des conteneurs. Diverses options d'activation des statistiques sont décrites dans «Surveillance des statistiques de serveur avec des beans gérés (MBeans)», à la page 585 et dans Exemple : utilitaire **xsadmin**.
6. Surveillez les serveurs à l'aide de la console Hyperic. Une fois que les serveurs ont été ajoutés au modèle d'inventaire, leurs services ne sont plus requis.
   - **Vue Tableau de bord** : Lorsque vous avez affiché les événements de détection des ressources, vous vous êtes connecté à la vue du tableau de bord principal. Il s'agit d'une vue générique qui sert de centre de messagerie que vous pouvez personnaliser. Vous pouvez exporter des graphiques ou des objets d'inventaire dans ce tableau de bord principal.
   - **Vue Ressources** : Vous pouvez interroger et afficher l'intégralité du modèle d'inventaire à partir de cette page. Une fois que les services ont été ajoutés, chaque serveur eXtreme Scale est correctement libellé et répertorié sous la section des serveurs. Vous pouvez cliquer sur chacun des serveurs pour consulter les mesures de base.
7. Affichez l'intégralité de l'inventaire du serveur dans la page d'affichage des ressources. Dans cette page, vous pouvez sélectionner plusieurs serveurs ObjectGrid et les regrouper. Une fois que vous avez regroupé un ensemble de ressources, leurs mesures communes peuvent être représentées graphiquement pour montrer les superpositions et les différences entre les membres du groupe. Pour afficher une superposition, sélectionnez les mesures dans l'écran de votre groupe de serveurs. La mesure est affichée dans la zone de représentation graphique. Pour afficher une superposition pour tous les membres du groupe, cliquez sur le nom de mesure souligné. Vous pouvez exporter les graphiques, vues de noeud et superpositions comparatives de votre choix dans le tableau de bord principal, à l'aide du menu **Outils**.

Surveillance des informations eXtreme Scale dans DB2

Lorsque le chargeur JPALoader ou JPAEntityLoader est utilisé avec DB2 comme base de données dorsale, des informations spécifiques de eXtreme Scale peuvent être transmises à DB2. Vous pouvez afficher ces informations à l'aide d'un outil de contrôle des performances tel que DB2 Performance Expert qui permet de surveiller les applications eXtreme Scale qui accèdent à la base de données.

Avant de commencer

Voir «Collecte de trace», à la page 668 pour plus d'informations sur les différentes méthodes de définition de trace que vous pouvez utiliser.

Pourquoi et quand exécuter cette tâche

Lorsque le chargeur est configuré pour utiliser DB2 comme base de données dorsale, les informations eXtreme Scale suivantes peuvent être transmises à DB2 à des fins de surveillance :

- **Utilisateur :** spécifie le nom de l'utilisateur qui s'authentifie auprès de eXtreme Scale. Si l'authentification standard n'est pas utilisée, ce sont les principaux de l'authentification qui sont utilisés.
- **Nom du poste de travail :** spécifie le nom d'hôte, l'adresse IP du serveur de conteneur eXtreme Scale.
- **Nom de l'application :** spécifie le nom de l'unité de persistance ObjectGrid (si spécifié).
- **Informations de comptabilité :** indique l'ID unité d'exécution, le type de transaction, l'ID de transaction et la chaîne de connexion.

Pour savoir comment surveiller l'accès à la base de données, informez-vous sur DB2 Performance Expert.

Procédure

- Pour activer toutes les informations du client eXtreme Scale, définissez les chaînes de trace suivantes :

  ```
  ObjectGridClientInfo*=event=enabled
  ```

- Pour tout activer sauf les informations utilisateur, utilisez l'un des paramètres suivants :
  -
    ```
    ObjectGridClientInfo*=event=enabled,ObjectGridClientInfoUser=event=disabled
    ```
    ou
  -
    ```
    ObjectGridClientInfo=event=enabled
    ```

Résultats

Une fois que vous avez activé la fonction de trace, les données s'affichent dans l'outil de surveillance des performances, tel que DB2 Performance Expert.

Exemple

Dans l'exemple suivant, l'utilisateur `bob` est authentifié en tant qu'utilisateur eXtreme Scale. L'application accède à la grille de données `mygrid` en utilisant l'unité de persistance `DB2Hibernate`. Le serveur de conteneur s'appelle `XS_Server1`. Informations résultantes :

- **Utilisateur**=`bob`

- **Nom de poste de travail**=XS_Server1,192.168.1.101
- **Nom d'application**=mygrid,DB2Hibernate
- **Informations comptabilité**=1, DEFAULT,FE7954BD-0126-4000-E000-2298094151DB,com.ibm.db2.jcc.t4.b@71787178

Dans l'exemple suivant, l'utilisateur bob est authentifié en utilisant un jeton WebSphere Application Server.. L'application accède à la grille de données mygrid en utilisant l'unité de persistance DB2OpenJPA. Le serveur de conteneur s'appelle XS_Server2. Informations résultantes :

- **Utilisateur**

  =acme.principal.UserPrincipal[Bob],acme.principal.GroupPrincipal[admin]
- **Nom de poste de travail**=XS_Server2,192.168.1.102
- **Nom d'application**=mygrid,DB2OpenJPA
- **Informations comptabilité**=188,DEFAULT,FE72BC63-0126-4000-E000-851C092A4E33,com.ibm.ws.rsadapter.jdbc.WSJccSQLJConnection@2b432b43

Chapitre 9. Optimisation des performances

Vous pouvez optimiser les paramètres dans l'environnement pour augmenter les performances générales de votre environnement WebSphere eXtreme Scale.

Optimisation des systèmes d'exploitation et des paramètres réseau

En modifiant les paramètres de connexion, il est possible de réduire les temps d'attente TCP (Transmission Control Protocol) et la modification des tampons TCP permet d'améliorer les débits de transmission.

Systèmes d'exploitation

De tous les systèmes d'exploitation, Windows est celui qui a le moins besoin d'être optimisé, au contraire de Solaris, qui nécessite un maximum d'optimisation. Les informations suivantes, qui concernent chacun des systèmes spécifiés, sont susceptibles d'améliorer les performances de WebSphere eXtreme Scale. Vous devez procéder à l'optimisation en fonction de la charge de votre réseau et de vos applications.

Windows

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
Tcpip\Parameters
MaxFreeTcbs = dword:00011940
MaxHashTableSize = dword:00010000
MaxUserPort = dword:0000fffe
TcpTimedWaitDelay = dword:0000001e
```

Solaris

```
ndd -set /dev/tcp tcp_time_wait_interval 60000
fndd -set /dev/tcp tcp_keepalive_interval 15000
ndd -set /dev/tcp tcp_fin_wait_2_flush_interval 67500
ndd -set /dev/tcp tcp_conn_req_max_q 16384
ndd -set /dev/tcp tcp_conn_req_max_q0 16384
ndd -set /dev/tcp tcp_xmit_hiwat 400000
ndd -set /dev/tcp tcp_recv_hiwat 400000
ndd -set /dev/tcp tcp_cwnd_max 2097152
ndd -set /dev/tcp tcp_ip_abort_interval 20000
ndd -set /dev/tcp tcp_rexmit_interval_initial 4000
ndd -set /dev/tcp tcp_rexmit_interval_max 10000
ndd -set /dev/tcp tcp_rexmit_interval_min 3000
ndd -set /dev/tcp tcp_max_buf 4194304
```

AIX

```
/usr/sbin/no -o tcp_sendspace=65536
/usr/sbin/no -o tcp_recvspace=65536
/usr/sbin/no -o udp_sendspace=65536
/usr/sbin/no -o udp_recvspace=65536
/usr/sbin/no -o somaxconn=10000
/usr/sbin/no -o tcp_nodelayack=1
/usr/sbin/no –o tcp_keepinit=40
/usr/sbin/no –o tcp_keepintvl=10
```

LINUX

```
sysctl -w net.ipv4.tcp_timestamps=0
sysctl -w net.ipv4.tcp_tw_reuse=1
sysctl -w net.ipv4.tcp_tw_recycle=1
sysctl -w net.ipv4.tcp_fin_timeout=30
sysctl -w net.ipv4.tcp_keepalive_time=1800
sysctl -w net.ipv4.tcp_rmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_wmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_max_syn_backlog=4096
```

HP-UX

```
ndd -set /dev/tcp tcp_ip_abort_cinterval 20000
```

Propriétés ORB

Java

(Obsolète) Les propriétés ORB (Object Request Broker) modifient le comportement du transport de la grille de données. Ces propriétés peuvent être définies avec un fichier `orb.properties`, comme paramètres dans la console d'administration de WebSphere Application Server ou en tant que propriétés personnalisées dans ORB dans la console d'administration de WebSphere Application Server.

Obsolète : 8.6+ Object Request Broker (ORB) est obsolète. Si vous n'utilisiez pas ORB dans une édition précédente, utilisez IBM eXtremeIO (XIO) pour le mécanisme de transport. Si vous utilisez ORB, migrez la configuration vers XIO.

orb.properties

Le fichier `orb.properties` se trouve dans le répertoire `java/jre/lib`. Lorsque vous modifiez le fichier `orb.properties` dans un répertoire WebSphere Application Server `java/jre/lib`, les propriétés ORB sont mises à jour sur l'agent de noeud et toutes les autres machines virtuelles Java (JVM) qui utilisent l'environnement d'exécution Java (JRE). Si vous ne souhaitez pas ce comportement, utilisez des propriétés personnalisées ou les paramètres ORB de la console d'administration WebSphere Application Server.

Paramètres WebSphere Application Server par défaut

Par défaut, WebSphere Application Server a certaines propriétés définies dans ORB. Ces paramètres se trouvent dans les services de conteneur de serveur d'applications et le gestionnaire de déploiement. Ces paramètres par défaut remplacent les paramètres que vous créez dans le fichier `orb.properties`. Pour chaque propriété décrite, reportez-vous à la section **Où définir** pour déterminer l'emplacement de définition de la valeur suggérée.

Paramètres de descripteur de fichier

Pour UNIX et Linux, il existe une limite pour le nombre de fichiers ouverts autorisés par processus. C'est le système d'exploitation qui spécifie le nombre permis de fichiers ouverts. Si cette valeur est trop basse, une erreur d'allocation de mémoire se produit sur AIX, et trop de fichiers ouverts sont consignés.

Dans la fenêtre de terminal UNIX, augmentez cette valeur au-dessus de la valeur par défaut du système. Dans le cas de grosses machines SMP avec des clones, fixez une valeur illimitée.

Pour les configurations AIX, définissez la valeur unlimited avec la commande : `ulimit -n unlimited`.

Pour les configurations Solaris définissez la valeur `16384` avec la commande `ulimit -n 16384`.

Pour afficher la valeur en cours, utilisez la commande `ulimit –a`.

Paramètres de référence

Les paramètres ci-après peuvent servir de référence, mais il ne s'agit pas nécessairement des meilleurs paramètres pour chaque environnement. Il convient de comprendre les paramètres pour définir des valeurs adaptées à votre environnement.

```
com.ibm.CORBA.RequestTimeout=30
com.ibm.CORBA.ConnectTimeout=10
com.ibm.CORBA.FragmentTimeout=30
com.ibm.CORBA.LocateRequestTimeout=10
com.ibm.CORBA.ThreadPool.MinimumSize=256
com.ibm.CORBA.ThreadPool.MaximumSize=256
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ConnectionMultiplicity=1
com.ibm.CORBA.MinOpenConnections=1024
com.ibm.CORBA.MaxOpenConnections=1024
com.ibm.CORBA.ServerSocketQueueDepth=1024
com.ibm.CORBA.FragmentSize=0
com.ibm.CORBA.iiop.NoLocalCopies=true
com.ibm.CORBA.NoLocalInterceptors=true
```

Descriptions des propriétés

Paramètres d'expiration

Les paramètres ci-après concernent le délai d'attente respecté par l'ORB avant d'abandonner des opérations de demande. Utilisez ces paramètres pour empêcher la création d'un trop grand nombre d'unités d'exécution dans une situation anormale.

Délai d'expiration de la demande

Nom de la propriété : com.ibm.CORBA.RequestTimeout

Valeur valide : entier indiquant le nombre de secondes.

Valeur suggérée : 30

Où définir : WebSphere Application Server administrative console

Description : indique le délai d'attente en secondes d'une réponse à une demande. Cette propriété influence la durée de la reprise en ligne du client en cas d'indisponibilité du réseau. Si vous spécifiez une valeur trop faible pour cette propriété, les demandes risquent d'arriver à expiration par inadvertance. Définissez soigneusement la valeur de cette propriété pour éviter les dépassements de délai d'attente.

Délai d'expiration de la connexion

Nom de la propriété : com.ibm.CORBA.ConnectTimeout

Valeur valide : entier indiquant le nombre de secondes.

Valeur suggérée : 10

Où définir : fichier `orb.properties`

Description : indique le délai d'attente en secondes d'une connexion de socket. Cette propriété, comme celle du délai d'expiration de la demande, peut influencer la durée de la reprise en ligne du client en cas d'indisponibilité du réseau. En règle générale, cette propriété doit avoir une valeur inférieure à la valeur de délai d'attente de la demande, car le délai d'établissement des connexions est relativement constant.

Délai d'expiration des fragments

Nom de la propriété : com.ibm.CORBA.FragmentTimeout

Valeur valide : entier indiquant le nombre de secondes.

Valeur suggérée : 30

Où définir : fichier `orb.properties`

Description : indique le délai d'attente en secondes d'une demande de fragment. Cette propriété est similaire à celle du délai d'expiration de la demande.

Paramètres du pool d'unités d'exécution

Ces propriétés restreignent la taille du pool d'unités d'exécution à un nombre spécifique d'unités d'exécution. Les unités d'exécution sont utilisées par l'ORB pour distribuer les demandes du serveur une fois qu'elles ont été reçues sur le socket. Si vous affectez des valeurs de propriété trop basses, vous allongez la file d'attente de sockets et éventuellement les délais.

Multiplicité des connexions

Nom de la propriété : com.ibm.CORBA.ConnectionMultiplicity

Valeur valide : entier indiquant le nombre de connexions entre le client et le serveur. La valeur par défaut est 1. Si vous spécifiez une valeur supérieure, le multiplexage est défini entre plusieurs connexions.

Valeur suggérée : 1

Où définir : fichier `orb.properties`**Description :** permet à ORB d'utiliser plusieurs connexions à un serveur. En théorie, la définition de cette valeur promeut le parallèlisme sur les connexions. En pratique, la multiplicité des connexions n'améliore pas les performances. Ne spécifiez pas ce paramètre.

Connexions ouvertes

Noms des propriétés : com.ibm.CORBA.MinOpenConnections, com.ibm.CORBA.MaxOpenConnections

Valeur valide : entier pour le nombre de connexions.

Valeur suggérée : 1024

Où définir : WebSphere Application Server administrative console**Description :** indique les nombres minimal et maximal de connexions ouvertes. L'ORB conserve un cache des connexions établies avec les clients. Ces connexions sont purgées lorsque cette valeur est transmise. La purge des connexions peut nuire au comportement de la grille de données.

Peut augmenter

Nom de la propriété : com.ibm.CORBA.ThreadPool.IsGrowable

Valeur valide : booléenne. Affectée de `true` ou `false`.

Valeur suggérée : `false`

Où définir : fichier `orb.properties`**Description :** si la valeur est `true`, le pool d'unités d'exécution qu'utilise ORB pour les demandes entrantes peuvent croître au-delà de ce que le pool prend en charge. Si la taille du pool est dépassée, des unités d'exécution sont créées pour traiter la

demande, mais elles ne sont pas placées dans un pool. Empêchez le pool d'unités d'exécution de croître en définissant la valeur `false`.

Longueur de la file d'attente des sockets de serveur

Nom de la propriété : com.ibm.CORBA.ServerSocketQueueDepth

Valeur valide : entier pour le nombre de connexions.

Valeur suggérée : `1024`

Où définir : fichier `orb.properties`**Description :** Indique la longueur de la file d'attente des connexions entrantes des clients. L'ORB place les connexions entrantes des clients en file d'attente. Si la file d'attente est saturée, les connexions sont refusées. Le refus de connexion peut affecter le comportement de la grille de données.

Taille du fragment

Nom de la propriété : com.ibm.CORBA.FragmentSize

Valeur valide : entier qui définit le nombre d'octets. La valeur par défaut est 1024.

Valeur suggérée : `0`

Où définir : fichier `orb.properties`**Description :** Indique la taille de paquet maximale utilisée par l'ORB lors de l'envoi d'une demande. Si la taille d'une demande est supérieure à la taille limite du fragment, la demande est divisée en fragments de demande qui sont envoyés séparément, puis réassemblés sur le serveur. La fragmentation des demandes est utile sur les réseaux non fiables où les packets doivent parfois être renvoyés. Toutefois, si le réseau est fiable, la division des demandes en fragments peut entraîner un traitement superflu.

Copies non locales

Nom de la propriété : com.ibm.CORBA.iiop.NoLocalCopies

Valeur valide : booléenne. Affectée de `true` ou `false`.

Valeur suggérée : `true`

Où définir : console d'administration WebSphere Application Server, paramètre de **transmission par référence**. **Description :** Indique si l'ORB utilise la transmission par référence. Par défaut, l'ORB utilise un appel de transmission par valeur. Un appel de transmission par valeur occupe plus de place et entraîne une sérialisation plus importante dans le chemin lorsqu'une interface est démarrée localement. Si vous spécifiez la valeur true, l'ORB utilise une méthode de transmission par référence, qui est plus efficace que l'appel de transmission par valeur.

Intercepteurs non locaux

Nom de la propriété : com.ibm.CORBA.NoLocalInterceptors

Valeur valide : booléenne. Affectée de `true` ou `false`.

Valeur suggérée : `true`

Où définir : fichier `orb.properties` **Description :** indique si l'ORB démarre les intercepteurs de demande, même lorsque des demandes locales (processus internes) sont faites. Les intercepteurs utilisés par WebSphere eXtreme Scale pour la gestion de la sécurité et des routes ne sont pas obligatoires si la demande est gérée au sein du processus. Les intercepteurs qui circulent entre les processus ne sont requis que pour les

opérations RPC (Remote Procedure Call). En indiquant la non-utilisation d'intercepteurs locaux, vous pouvez éviter le traitement supplémentaire que génère l'utilisation d'intercepteurs locaux.

Avertissement : Si vous utilisez la sécurité WebSphere eXtreme Scale, affectez à la propriété com.ibm.CORBA.NoLocalInterceptors la valeur `false`. L'infrastructure de sécurité utilise des intercepteurs pour l'authentification.

Optimisation d'IBM eXtremeIO (XIO)

Vous pouvez utiliser les propriétés du serveur XIO pour optimiser le comportement du transport XIO dans la grille de données.

Propriétés du serveur pour l'optimisation de XIO

Vous pouvez définir les propriétés suivantes dans le fichier des propriétés du serveur :

maxXIONetworkThreads
Définit le nombre maximum d'unités d'exécution à allouer dans le pool d'unités d'exécution du réseau de transport eXtremeIO.

Valeur par défaut :`50`

minXIONetworkThreads
Définit le nombre minimum d'unités d'exécution à allouer dans le pool d'unités d'exécution du réseau de transport eXtremeIO.

Valeur par défaut :`50`

maxXIOWorkerThreads
Définit le nombre maximum d'unités d'exécution à allouer dans le pool d'unités d'exécution de traitement des demandes de transport.

Valeur par défaut :`128`

minXIOWorkerThreads
Définit le nombre minimum d'unités d'exécution à allouer dans le pool d'unités d'exécution de traitement des demandes de transport.

Valeur par défaut :`128`

8.6+ transport

Indique le type de transport à utiliser pour tous les serveurs dans le domaine de service de catalogue Vous pouvez définir la valeur `XIO` ou `ORB`.

Lorsque vous utilisez **`startOgServer`** ou **`startXsServer`** , vous n'avez pas besoin de définir cette propriété. Le script remplace cette propriété. Toutefois, si vous démarrez les serveurs avec une autre méthode, la valeur de cette propriété est utilisée.

Cette valeur s'applique au service de catalogue uniquement.

Si le paramètre **`-transport`** figure dans le script de démarrage et que la propriété serveur **`transport`** est définie sur un serveur de catalogue, la valeur du paramètre **`-transport`** est utilisée.

8.6+ xioTimeout
Définit le délai d'expiration des demandes de serveur qui utilisent le transport IBM eXtremeIO (XIO) en secondes. La valeur doit être supérieure ou égale à une seconde.

Valeur par défaut : 30 secondes

Optimisation des machines virtuelles Java

Java

Vous devez prendre en compte plusieurs aspects spécifiques de l'optimisation des machines virtuelles Java (JVM) pour optimiser les meilleures performances WebSphere eXtreme Scale. Dans la plupart des cas, quelques paramètres JVM spéciaux sont nécessaires ou aucun. Si de nombreux objets sont stockés dans la grille de données, définissez une taille de pile appropriée pour éviter de manquer de mémoire.

IBM eXtremeMemory

En configurant eXtremeMemory, vous pouvez stocker des objets dans la mémoire native plutôt que dans le segment de mémoire Java. La configuration eXtremeMemory active eXtremeIO, un nouveau mécanisme de transport. En retirant des objets du segment de mémoire Java, vous pouvez éviter les pauses de récupération d'espace, ce qui permet de bénéficier de performances plus constantes et de temps de réponse plus prévisibles. Pour plus d'informations, voir «Configuration d'IBM eXtremeMemory», à la page 357.

Plateformes testées

Les tests de performance ont été réalisés principalement sur des ordinateurs AIX (32 voies), Linux (quatre voies) et Windows (huit voies). Avec des ordinateurs AIX haut de gamme, vous pouvez tester des scénarios qui font appel à de nombreuses unités d'exécution pour identifier et corriger les points de conflit.

Récupération de place

WebSphere eXtreme Scale crée des objets temporaires associés à chaque transaction tels que la demande et la réponse ainsi la séquence de journal. Ces objets affectant l'efficacité de la récupération de place, l'optimisation de la récupération de place est cruciale.

Toutes les machines virtuelles Java actuelles utilisent des algorithmes de récupération de place en parallèle, ce qui implique que l'utilisation d'un plus grand nombre de coeurs peut réduire les pauses dans la récupération de place. Un serveur physique avec huit coeurs a une récupération plus rapide qu'un serveur physique avec quatre coeurs.

Lorsque l'application doit gérer une large quantité de données pour chaque partition, la récupération de place peut être un facteur déterminant. Un scénario de lecture principalement est performant même avec de grands segments (20 Go ou plus) si un collecteur générationnel est utilisé. Toutefois, une fois que le segment de réservation est rempli, une pause proportionnelle à la taille de pile réelle et au nombre de processeurs sur l'ordinateur se produit. Cette pause peut être longue sur les petites ordinateurs avec de grands segments de mémoire.

Machine virtuelle IBM pour la récupération de place Java

Pour la machine virtuelle IBM pour Java, utilisez le collecteur **optavgpause** pour les scénarios impliquant des mises à jour fréquentes (100 % des transactions modifient les entrées). Le collecteur **gencon** fonctionne d'une manière similaire au collecteur

optavgpause pour les scénarios où les données sont mises à jour peu fréquemment (10 % du temps au plus). Expérimentez les deux collecteurs pour savoir lequel est le mieux adapté à vos besoins. Utilisez la récupération de place prolixe pour vérifier le pourcentage de temps passé à la récupération de place. Des cas ont été relevés où 80 % de l'exécution sont consacrés à la récupération de place jusqu'à ce que l'optimisation corrige le problème.

Utilisez le paramètre **-Xgcpolicy** pour changer le mécanisme de collecte de place. La valeur du paramètre **-Xgcpolicy** peut être **-Xgcpolicy:gencon** ou **-Xgcpolicy:optavgpause**, selon le récupérateur de place que vous voulez utiliser.

- Dans une configuration WebSphere Application Server, définissez le paramètre **-Xgcpolicy** dans la console d'administration. Cliquez sur **Serveurs** > **Serveurs d'applications** > **server_name** > **Définition de processus** > **Java Virtual Machine**. Ajoutez le paramètre dans la zone des **arguments JVM génériques**.
- Dans une configuration autonome, envoyez le paramètre **-jvmArgs** au script de démarrage du serveur pour définir la récupération de place. Le paramètre **-jvmArgs** doit être le dernier paramètre envoyé au script.

Autres options de récupération de place

Avertissement : Si vous utilisez une machine virtuelle Java Oracle, il peut être nécessaire d'ajuster la récupération de place et d'optimiser la stratégie.

WebSphere eXtreme Scale prend en charge WebSphere Real Time Java. Avec WebSphere Real Time Java, la réponse du traitement des transactions pour WebSphere eXtreme Scale est plus cohérente et prévisible. En conséquence, l'impact de la récupération de place et de la planification des unités d'exécution est considérablement réduit. L'impact est réduit au point où l'écart type du temps de réponse est inférieur à 10 % du langage Java classique.

Performance de la JVM

WebSphere eXtreme Scale peut être exécuté sur différentes versions de Java Platform, Standard Edition. WebSphere eXtreme Scale prend en charge Java SE Version 6. Pour optimiser la productivité et les performances des développeurs, utilisez Java SE Version 6 ou une versions suivante , ou bien Java SE Version 7 pour tirer parti des annotations et de la récupération de place améliorée . WebSphere eXtreme Scale tourne sur les machines virtuelles Java 32 bits ou 64 bits.

WebSphere eXtreme Scale est testé avec un sous-ensemble de machines virtuelles disponibles mais la liste de prise en charge n'est pas exhaustive. Vous pouvez exécuter WebSphere eXtreme Scale sur n'importe quelle machine JVM de fournisseur au niveau d'édition 5 ou suivant. Toutefois, en cas de problème avec la machine JVM d'un fournisseur, vous devez contacter le fournisseur de la machine JVM pour obtenir une assistance. Si possible, utilisez la machine JVM de l'environnement d'exécution WebSphere sur n'importe quelle plateforme qui prend en charge WebSphere Application Server.

En règle générale, utilisez la dernière version disponible de Java Platform, Standard Edition pour obtenir de meilleures performances.

Taille de pile

Il est recommandé d'utiliser des segments de mémoire de 1 à 2 Go avec une machine virtuelle Java pour quatre coeurs. La taille de segment de mémoire optimal dépend des facteurs suivants :

- le nombre d'objets actifs présents dans le segment
- le degré de complexité des objets actifs présents dans le segment
- le nombre de coeurs utilisables par la machine virtuelle Java

Par exemple, une application qui stocke des tableaux de 1 Ko peut utiliser une segment de mémoire beaucoup plus grand qu'une application qui utilise des graphiques complexes de POJO.

Nombre d'unités d'exécution

Le nombre d'unités d'exécution dépend de quelques facteurs. Une limite existe pour le nombre d'unités d'exécution pouvant être gérées par un seul fragment. Un fragment est une instance de partition et peut être un fragment primaire ou une réplique. Avec un nombre plus important de fragments pour chaque JVM, vous disposez de plusieurs unités d'exécution avec chaque fragment supplémentaire fournissant plus de chemins simultanés d'accès aux données. Chaque fragment est aussi concurrent que possible même si l'accès simultané est limité.

Exigences de la fonction ORB (Object Request Broker)

Le kit de développement de logiciels IBM comprend une implémentation ORB IBM qui a été testée avec WebSphere Application Server et WebSphere eXtreme Scale. Pour faciliter le processus de prise en charge, utilisez une machine virtuelle Java IBM. Les autres implémentations de machines virtuelles Java utilisent une autre fonction ORB. L'ORB IBM est fourni uniquement avec les machines virtuelles IBM Java. WebSphere eXtreme Scale requiert une fonction ORB opérationnelle. Vous pouvez utiliser WebSphere eXtreme Scale avec les ORB d'autres fournisseurs. Toutefois, si vous avez un problème avec un ORB de fournisseur, vous devez contacter le fournisseur ORB pour obtenir une assistance. L'implémentation ORB IBM est compatible avec des machines virtuelles Java tierces et peut être remplacée si nécessaire.

Optimisation d'orb.properties

En laboratoire, le fichier suivant a été utilisé sur les grilles de données de jusqu'à 1 500 machines virtuelles Java. Le fichier `orb.properties` se trouve dans le dossier `lib` de l'environnement d'exécution.

```
# IBM JDK properties for ORB
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton

# WS Interceptors
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer

# WS ORB & Plugins properties
com.ibm.CORBA.ForceTunnel=never
com.ibm.CORBA.RequestTimeout=10
com.ibm.CORBA.ConnectTimeout=10

# Needed when lots of JVMs connect to the catalog at the same time
com.ibm.CORBA.ServerSocketQueueDepth=2048

# Clients and the catalog server can have sockets open to all JVMs
com.ibm.CORBA.MaxOpenConnections=1016

# Thread Pool for handling incoming requests, 200 threads here
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ThreadPool.MaximumSize=200
```

```
com.ibm.CORBA.ThreadPool.MinimumSize=200
com.ibm.CORBA.ThreadPool.InactivityTimeout=180000

# No splitting up large requests/responses in to smaller chunks
com.ibm.CORBA.FragmentSize=0
```

Optimisation de la valeur de l'intervalle des pulsations pour la détection des basculements

Le paramètre d'intervalle du signal de présence permet de configurer le laps de temps séparant deux vérifications par le système des serveurs en panne. Ce paramètre s'applique uniquement aux serveurs de catalogue.

Pourquoi et quand exécuter cette tâche

La configuration des basculements varie en fonction du type d'environnement que vous utilisez. Si vous utilisez un environnement autonome, vous pouvez configurer les basculements à l'aide de la ligne de commande. Si vous utilisez un environnement WebSphere Application Server Network Deployment, vous devez les configurer à partir de la console d'administration de WebSphere Application Server Network Deployment.

Procédure

- Configurez les basculements pour les environnements autonomes.

  Vous pouvez configurer les intervalles des pulsations du serveur de catalogue à l'aide du paramètre **-heartbeat** dans le fichier script **startOgServer** or **startXsServer**. Affectez à ce paramètre l'une des valeurs suivantes :

  Tableau 38. Intervalles de signal de présence

Valeur	Action	Description
0	Standard (par défaut)	Les basculements sont généralement détectés dans les 30 secondes.
-1	Elevé	Les basculements sont généralement détectés dans les 5 secondes.
1	Souple	Les basculements sont généralement détectés dans les 180 secondes.

  Un intervalle élevé entre les signaux de présence peut être utile si les processus et le réseau sont stables. Si le réseau ou les processus ne sont pas configurés de manière optimale, il peut manquer des signaux de présence, ce qui peut fausser la détection des incidents.

- Configurez les basculements pour les environnements WebSphere Application Server.

  Vous pouvez configurer WebSphere Application Server Network Deployment Version 7.0 et les versions suivantes pour autoriser WebSphere eXtreme Scale à basculer très rapidement. La durée par défaut de pour les incidents matériels est d'environ 200 secondes. Un incident matériel est un ordinateur physique, une panne du serveur, déconnexion de câble réseau ou une erreur du système d'exploitation. Les incidents dus aux pannes de processus ou à des échecs logiciels sont généralement basculés en moins d'une seconde. La détection des incidents logiciels est effectuée lorsque les sockets réseau du processus inactif sont fermés automatiquement par le système d'exploitation du serveur qui héberge le processus.

  Configuration des signaux de présence du groupe central

  Si WebSphere eXtreme Scale est exécuté dans un processus WebSphere Application Server, il hérite des caractéristiques de reprise en ligne des

paramètres du groupe central du serveur d'applications. Les sections suivantes décrivent comment configurer les paramètres des signaux de présence du groupe central pour différentes versions de WebSphere Application Server Network Deployment :

– **Mise à jour des paramètres des groupes centraux de WebSphere Application Server Network Deployment Version 7.0**

  WebSphere Application Server Network Deployment Version 7.0 fournit deux paramètres de groupe central qui peuvent être ajustés pour augmenter ou réduire le délai de détection des incidents :

  - **Période de transmission du signal de présence.** La valeur par défaut est de 30000 millisecondes.
  - **Période d'expiration du signal de présence.** La valeur par défaut est de 180000 millisecondes.

  Pour plus de détails sur la manière de modifier ces paramètres, voir la rubrique relative à la WebSphere Application Server Network Deployment reconnaissance et de détection des incidents dans le centre de documentation.

  Utilisez les paramètres suivants pour spécifier un délai de détection des incidents de 1500 ms pour les serveurs WebSphere Application Server Network Deployment Version 7 :

  - Spécifiez une période de transmission du signal de présence de 750 millisecondes.
  - Spécifiez une période d'expiration du signal de présence de 1500 millisecondes.

Que faire ensuite

Lorsque vous modifiez ces paramètres pour réduire les délais de basculement, certains points d'optimisation du système sont à prendre en compte. Tout d'abord, Java n'est pas un environnement en temps réel. Des unités d'exécution peuvent être retardées si la JVM connaît des délais de récupération de place importants. Les unités d'exécution risquent également d'être retardées si la charge de la machine qui héberge la JVM est considérable (à cause de la JVM elle-même ou d'autres processus exécutés sur cette machine). Si les unités d'exécution sont retardées, les signaux de présence risquent de ne pas être envoyés à temps. Au pire, ils risquent d'être retardés du délai requis pour la reprise en ligne. Si des unités d'exécution sont retardées, des incidents sont détectés à tort. Le système doit être optimisé et dimensionné de sorte à éviter la détection de faux incidents en production. Il est recommandé pour cela de tester la charge de manière adéquate.

Remarque : La version actuelle d'eXtreme Scale prend en charge WebSphere Real Time.

Optimisation de la récupération de place avec WebSphere Real Time

Utiliser WebSphere eXtreme Scale avec WebSphere Real Time augmente la cohérence et la prévisibilité des performances de débit qu'offre la stratégie standard de récupération de place employée dans le JRE (IBM Java SE Runtime Environment). Le ratio coûts/avantages est variable. WebSphere eXtreme Scale crée un grand nombre d'objets temporaires associés à chaque transaction. Ces objets temporaires s'occupent des demandes, des réponses, des séquences de journaux et des sessions. Sans WebSphere Real Time, les temps de réponse des transactions peuvent grimper à des centaines de millisecondes. Mais WebSphere Real Time utilisé avec WebSphere eXtreme Scale peut augmenter l'efficacité de la récupération de place et faire tomber à 10% les temps de réponse de la configuration autonome.

WebSphere Real Time en environnement autonome

Il est possible d'utiliser WebSphere Real Time avec WebSphere eXtreme Scale. En activant WebSphere Real Time, l'on obtient une récupération de place plus prévisible grâce à des temps de réponses stables et cohérents et des débits de transactions dans un environnement eXtreme Scale autonome.

Avantages de WebSphere Real Time

WebSphere eXtreme Scale crée un grand nombre d'objets temporaires associés à chaque transaction. Ces objets temporaires s'occupent des demandes, des réponses, des séquences de journaux et des sessions. Sans WebSphere Real Time, les temps de réponse des transactions peuvent grimper à des centaines de millisecondes. Mais WebSphere Real Time utilisé avec WebSphere eXtreme Scale peut augmenter l'efficacité de la récupération de place et faire tomber à 10% les temps de réponse de la configuration autonome.

Activer WebSphere Real Time

Installez WebSphere Real Time et WebSphere eXtreme Scale autonome sur les ordinateurs sur lesquels vous prévoyez d'exécuter eXtreme Scale. Définissez la variable d'environnement JAVA_HOME pour qu'elle pointe sur un JRE standard Java SE standard.

Définissez la variable d'environnement JAVA_HOME pour qu'elle pointe sur le WebSphere Real Time installé. Puis activez WebSphere Real Time comme indiqué ci-après.

1. Dans le fichier `objectgridRoot/bin/setupCmdLine.sh | .bat`, supprimez le commentaire de la ligne suivante :

   ```
   WXS_REAL_TIME_JAVA="-Xrealtime -Xgcpolicy:metronome
   -Xgc:targetUtilization=80"
   ```

2. Enregistrez le fichier.

WebSphere Real Time est à présent activé. Pour le désactiver, il vous suffit de repasser la même ligne en commentaire.

Pratiques recommandées

WebSphere Real Time confère aux transactions eXtreme Scale des temps de réponse plus prévisibles. Les résultats montrent que les temps de réponse d'une transaction eXtreme Scale s'améliorent de manière significative avec WebSphere Real Time si on les compare à ceux obtenus par le récupérateur de place par défaut de Java. L'activation de WebSphere Real Time avec eXtreme Scale est un must si la stabilité et les temps de réponse de votre application sont essentiels.

Les pratiques recommandées développées ci-après expliquent comment rendre WebSphere eXtreme Scale encore plus efficace grâce à une optimisation et une programmation fonction de la charge attendue.

- Définissez le bon niveau d'utilisation du processeur et de récupération de place.

  WebSphere Real Time donne les moyens de contrôler l'utilisation du processeur de manière à contrôler et à réduire l'impact de la récupération de place sur votre application. Le paramètre `-Xgc:targetUtilization=NN` permet de spécifier `NN` comme pourcentage du processeur qui est utilisé par votre application toutes les 20 secondes. Par défaut, cette valeur est de 80 % pour WebSphere eXtreme Scale, mais vous pouvez modifier le script dans le fichier `objectgridRoot/bin/`

`setupCmdLine.sh` pour définir un autre chiffre, `70`, par exemple, qui libère davantage de capacité processeur pour le récupérateur de place. Déployez suffisamment de serveurs pour maintenir la charge processeur en dessous de 80 % pour vos applications.

- Augmentez la taille de la mémoire dynamique.

  WebSphere Real Time utilise davantage de mémoire que le Java standard, aussi, prévoyez plus de mémoire dynamique pour votre WebSphere eXtreme Scale et définissez la taille du segment mémoire lors du démarrage des serveurs de catalogue avec le paramètre `–jvmArgs –XmxNNNM` dans la commande **ogStartServer**. Vous pouvez, par exemple, utiliser le paramètre `–jvmArgs –Xmx500M` pour démarrer des serveurs de catalogue et utilisez une taille mémoire appropriée pour démarrer les conteneurs. Vous pouvez fixer la taille de la mémoire à 60-70 % de la taille prévue par machine virtuelle Java pour vos données . Si vous ne définissez pas cette valeur, une erreur OutOfMemoryError risque de se produire. Vous pouvez également, si vous le souhaitez, utiliser le paramètre `–jvmArgs –Xgc:noSynchronousGCOnOOM` pour empêcher le comportement non déterministe lorsque la machine virtuelle Java est à court de mémoire.

- Ajustez les unités d'exécution pour la récupération de place.

  WebSphere eXtreme Scale crée un grand nombre d'objets temporaires associés à chaque transaction et aux unités d'exécution RPC (Remote Procedure Call). La récupération de place présente des avantages pour les performances si votre ordinateur dispose de suffisamment de cycles processeur. Le nombre d'unités d'exécution est de `1` par défaut. L'argument `–Xgcthreads n` permet de modifier ce nombre. La valeur suggérée pour cet argument est le nombre de coeurs qui sont disponibles en prenant en considération le nombre de machines virtuelles Java par ordinateur.

- Ajustez les performances pour les applications à exécution courte avec WebSphere eXtreme Scale.

  WebSphere Real Time est optimisé pour les applications à exécution longue. D'ordinaire, vous avez besoin d'exécuter des transactions WebSphere eXtreme Scale pendant deux heures en continu pour obtenir des données de performances fiables. Le paramètre `–Xquickstart` donne de meilleures performances à vos applications à exécution courte. Ce paramètre indique au compilateur JIT (just-in-time) d'utiliser un bas niveau d'optimisation.

- Réduisez la file d'attente des clients WebSphere eXtreme Scale et les relais clients WebSphere eXtreme Scale.

  Le principal avantage d'utiliser WebSphere eXtreme Scale avec WebSphere Real Time est de bénéficier de temps de réponse des transactions extrêmement fiables, qui représentent usuellement une amélioration de l'ordre de plusieurs fois l'écart constaté dans les temps de réponse des transactions. Toutes les demandes clients mises en file d'attente et tous les relais de ces demandes effectuées via d'autres logiciels ont un impact sur les temps de réponse qui échappent au contrôle de WebSphere Real Time et de WebSphere eXtreme Scale. Vous devez modifier les paramètres de vos unités d'exécution et de vos sockets pour conserver une charge à la fois ferme et fluide sans retards significatifs et vous devez diminuer la profondeur des files d'attente.

- Ecrivez des applications WebSphere eXtreme Scale qui utilisent les unités d'exécution WebSphere Real Time.

  Sans modifier votre application, vous pouvez obtenir des temps de réponse WebSphere eXtreme Scale des transactions extrêmement fiable représentant une amélioration de l'ordre de plusieurs fois l'écart standard dans les temps de réponse des transactions. Vous pouvez exploiter davantage les unités d'exécution

de vos applications transactionnelles en passant du threading Java standard au RealtimeThread, qui fournit un meilleur contrôle de la priorité des unités d'exécution et de la planification.

Actuellement, votre application comporte le code suivant :

```
public class WXSCacheAppImpl extends Thread implements WXSCacheAppIF
```

Vous pouvez remplacer ce code par le code suivant :

```
public class WXSCacheAppImpl extends RealtimeThread implements
WXSCacheAppIF
```

WebSphere Real Time sur WebSphere Application Server

Vous pouvez utiliser WebSphere Real Time avec eXtreme Scale dans un environnement WebSphere Application Server Network Deployment de version 7.0. L'activation de WebSphere Real Time permet d'obtenir une récupération de place plus prévisible avec des temps de réponses et des débits de transactions stables et cohérents.

Avantages

Utiliser WebSphere eXtreme Scale avec WebSphere Real Time augmente la cohérence et la prévisibilité des performances de débit qu'offre la stratégie standard de récupération de place employée dans le JRE (IBM Java SE Runtime Environment). Le ratio coûts/avantages est variable en fonction de plusieurs critères. Voici quelques-uns des principaux critères :

- capacités en serveurs : mémoire disponible, vitesse et taille des processeurs, vitesse et utilisation du réseau
- charges des serveurs : charge processeur soutenue, charge processeur de pointe
- configuration Java : taille des segments, utilisation cible, unités d'exécution de récupération de place
- configuration du mode copie de WebSphere eXtreme Scale : tableau d'octets ou stockage POJO
- points propres aux applications : utilisation des unités d'exécution, conditions requises et tolérance des réponses, taille des objets, etc.

En plus de la stratégie métronome de récupération de place utilisable dans WebSphere Real Time, il existe des stratégies optionnelles proposées par le JRE IBM standard. Ces stratégies, optthruput (stratégie par défaut), gencon, optavgpause et subpool sont spécifiquement conçues pour résoudre les différents besoins et environnements des applications. Pour plus d'informations sur ces stratégies, voir «Optimisation des machines virtuelles Java», à la page 609. En fonction des besoins de l'application et de l'environnement, ainsi que des ressources et des restrictions, le prototypage d'une ou plusieurs de ces stratégies peut vous garantir la satisfaction de ces besoins et vous aider à déterminer à coup sûr une stratégie optimale.

Possibilités avec WebSphere Application Server Network Deployment

1. Voici quelques-unes des versions prises en charge :
   - WebSphere Application Server Network Deployment version 7.0.0.5 et au-dessus
   - WebSphere Real Time V2 SR2 for Linux et au-dessus. Pour plus d'informations, voir IBM WebSphere Real Time V2 for Linux
   - WebSphere eXtreme Scale version 7.0.0.0 et au-dessus

- Linux 32 et 64 bits

2. Les serveurs WebSphere eXtreme Scale ne peuvent cohabiter avec WebSphere Application Server DMgr.
3. Real Time ne prend pas en charge DMgr.
4. Real Time ne prend pas en charge les agents de noeuds WebSphere.

Activer WebSphere Real Time

Installez WebSphere Real Time et WebSphere eXtreme Scale sur les ordinateurs sur lesquels vous prévoyez d'exécuter eXtreme Scale. Mettez au niveau SR2 le Java de WebSphere Real Time.

Vous pouvez spécifier comme suit les paramètres des machines virtuelles Java pour chaque serveur via la console WebSphere Application Server version 7.0.

Sélectionnez **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > **<serveur installé requis>**.

Dans la page qui s'affiche, choisissez Définition des processus.

Dans la page qui s'affiche alors, cliquez sur Machine virtuelle Java en haut de la colonne de droite (c'est là que vous pouvez définir pour chaque serveur la taille des segments, la récupération de place et d'autres indicateurs).

Définissez les indicateurs suivants dans la zone Arguments JVM génériques :

```
-Xrealtime -Xgcpolicy:metronome  -Xnocompressedrefs -Xgc:targetUtilization=80
```

Appliquez les modifications et enregistrez-les.

Pour utiliser Real Time dans WebSphere Application Server 7.0 avec des serveurs eXtreme Scale incluant les indicateurs JVM ci-dessus, vous devez créer une variable d'environnement JAVA_HOME.

Définissez JAVA_HOME comme suit :

1. Développez Environnement.
2. Sélectionnez Variables WebSphere.
3. La case Toutes les portées en dessous de Afficher la portée doit être cochée.
4. Sélectionnez le serveur requis dans la liste déroulante (ne sélectionnez pas de serveurs DMgr ou d'agents de noeuds).
5. Si la variable d'environnement JAVA_HOME n'apparaît pas dans la liste, sélectionnez Nouveau et spécifiez JAVA_HOME comme nom de la variable. Dans la zone Valeur, entrez le nom complet du chemin d'accès à Real Time.
6. Appliquez les modifications et enregistrez-les.

Pratiques recommandées

Vous trouverez un ensemble de pratiques recommandées dans la section Pratiques recommandées du chapitre «Optimisation de la récupération de place avec WebSphere Real Time», à la page 613. Il y a dans cette liste de pratiques recommandées pour un environnement WebSphere eXtreme Scale autonome des points qui diffèrent pour un déploiement dans un environnement WebSphere Application Server Network Deployment.

Vous devez placer des paramètres supplémentaires de ligne de commande JVM au même endroit que les paramètres de stratégie de récupération de place évoqués à la précédente section.

Une cible initiale acceptable pour les charges processeur soutenues est de 50 % avec des charges de pointe de courte durée grimpant jusqu'à 75 %. Au-delà, vous devez ajouter des capacités supplémentaires avant de constater une dégradation mesure de la prévisibilité et de la cohérence. Vous pouvez augmenter légèrement les performances si vous pouvez tolérer des temps de réponse plus longs. Au-delà d'un seuil de 80 % conduit souvent à une dégradation significative de la cohérence et de la prévisibilité.

Chapitre 10. Sécurité

WebSphere eXtreme Scale permet de sécuriser l'accès aux données et l'intégration de fournisseurs de sécurité externes. Les éléments de sécurité comprennent l'authentification, l'autorisation, la sécurité du transport, la sécurité de la grille de données, la sécurité locale et la sécurité JMX (MBean).

Scénario : protection de la grille de données dans eXtreme Scale

Les grilles de données WebSphere eXtreme Scale contiennent des informations sensibles et elles doivent être protégées.

Avant de commencer

- Installez le produit. Vous devez installer l'environnement d'exécution serveur et les clients. Pour les clients, vous pouvez utiliser des clients Java et .NET. Pour plus d'informations, voir Chapitre 4, «Installation», à la page 183.
- Si vous effectuez une mise à niveau à partir d'une version précédente, tous les serveurs de conteneur et de catalogue doivent être au même niveau d'édition. Pour plus d'informations, voir Chapitre 5, «Mise à niveau et migration de WebSphere eXtreme Scale», à la page 255.

Pourquoi et quand exécuter cette tâche

Pour un déploiement sécurisé, utilisez plusieurs couches de protection pour optimiser la sécurité. Les pare-feu sont le premier élément de protection pour segmenter le réseau. Le modèle à niveaux standard pour les applications Web est constitué des clients, d'un niveau présentation des serveurs HTTP, d'un niveau application constitué des serveurs d'application, d'un niveau données et d'un niveau stockage.

Les serveurs de grille de données eXtreme Scale sont déployés avec le niveau données. En règle générale, il convient de placer le serveurs de la couche présentation dans une zone DMZ (demilitarized zone) protégée par un pare-feu, et de placer les niveaux application, données et stockage dans des segments du réseau protégés par d'autres pare-feu. Ne déployez pas des serveurs eXtreme Scale dans une zone DMZ. Les serveurs eXtreme Scale doivent être protégés comme tous les éléments du niveau données, en fonction de la pratique acceptée par le secteur.

Cependant, pour optimiser la protection contre les menaces de sécurité, utilisez un mécanisme de protection efficace comprenant des mesures qui protègent le fonctionnement de eXtreme Scale et les données stockées dans la grille de données. Ces mesures supplémentaires fournissent une protection contre les attaques externes, mais interdisent également les accès non autorisés aux données par les employés et les sous-traitants qui pourraient avoir accès aux segments du réseau contenant les serveurs eXtreme Scale.

Exécutez la procédure de bout en bout suivante pour configurer la sécurité dans WebSphere eXtreme Scale, que vous utilisiez des serveurs autonomes, le Profil Liberty, le canevas OSGi ou WebSphere Application Server dans l'environnement :

Authentification d'une grille de données

Java

Vous pouvez utiliser le plug-in du gestionnaire de jetons de sécurité pour activer l'authentification serveur à serveur, ce qui signifie d'implémenter l'interface SecureTokenManager.

La méthode generateToken(Object) prend un objet protect, puis génère un jeton pouvant être compris par les autres. La méthode verifyTokens(byte[]) procède en sens contraire : elle reconvertit le jeton en objet d'origine.

Une implémentation SecureTokenManager simple utilise un algorithme de codage de base, tel qu'un algorithme XOR, pour coder l'objet dans un formulaire sérialisé et utiliser l'algorithme de décodage correspondant pour décoder le jeton. Cette implémentation n'est pas sécurisée et peut être facilement interrompue.

Implémentation par défaut de WebSphere eXtreme Scale

WebSphere eXtreme Scale met immédiatement à disposition une implémentation de cette interface. Cette implémentation par défaut utilise une paire de clés pour signer et vérifier la signature et une clé confidentielle pour chiffrer le contenu. Chaque serveur comporte un fichier de clés de type JCKES contenant la paire de clés, une clé privée et une clé publique ainsi qu'une clé confidentielle. Pour stocker les clés confidentielles, le fichier de clés doit être de type JCKES. En effet, ces clés servent à chiffrer et signer ou vérifier la chaîne secrète côté expéditeur. De plus, le jeton est associé à un délai d'expiration. Côté récepteur, les données sont vérifiées, déchiffrées et comparées à la chaîne secrète du récepteur. Des protocoles de communication SSL (Secure Sockets Layer) ne sont pas requis entre une paire de serveurs pour l'authentification car les clés privées et les clés publiques ont la même finalité. Toutefois, si la communication du serveur n'est pas chiffrée, les données peuvent être dérobées à la seule vue de la communication. Le jeton venant à expiration, la menace d'attaque par relecture est minimisée. Ce risque est considérablement réduit si tous les serveurs sont déployés derrière un pare-feu.

L'inconvénient de cette approche : les administrateurs de WebSphere eXtreme Scale doivent générer des clés et les transporter vers tous les serveurs, ce qui peut provoquer des failles de sécurité lors du transport.

Sécurité de grille de données

La sécurité d'une grille de données permet de garantir qu'un serveur qui y est ajouté dispose des données d'identifications correctes afin qu'elles ne contiennent aucun serveur malveillant. La sécurité de la grille de données utilise un mécanisme de chaîne secrète partagée.

Tous les serveurs WebSphere eXtreme Scale, y compris les serveurs de catalogue, choisissent une chaîne de secret partagé commune. Lorsqu'un serveur rejoint la grille de données, il est invité à présenter la chaîne secrète. Si cette chaîne correspond à celle du serveur président ou du serveur de catalogues, le serveur est accepté. Dans le cas contraire, la demande de jointure est rejetée.

L'envoi d'un texte en clair n'est pas sécurisé. L'infrastructure de sécurité WebSphere eXtreme Scale propose un plug-in de gestionnaire de jetons sécurisé permettant au serveur de sécuriser la valeur confidentielle avant son envoi. Vous devez choisir le

mode d'implémentation de cette opération. WebSphere eXtreme Scale propose une implémentation prête à l'emploi : l'opération de sécurité est implémentée pour chiffrer et signer la valeur confidentielle.

La chaîne confidentielle est définie dans le fichier `server.properties`. Reportez-vous à la rubrique Fichier de propriétés du serveur pour plus d'informations sur la propriété authenticationSecret.

Plug-in SecureTokenManager

Le plug-in de gestionnaire de jetons sécurisé est représenté par l'interface com.ibm.websphere.objectgrid.security.plugins.SecureTokenManager.

Pour plus d'informations sur ce plug-in, consultez la documentation relative à l'API SecureTokenManager.

La méthode generateToken(Object) prend un objet, puis génère un jeton ne pouvant être compris par les autres. La méthode verifyTokens(byte[]) suit le processus inverse : la méthode reconvertit le jeton à son format d'origine.

Une implémentation SecureTokenManager simple utilise un algorithme de codage simple, par exemple un algorithme OU exclusif (XOR), pour coder l'objet au format sérialisé, puis utilise l'algorithme de décodage pour décoder le jeton. Cette implémentation n'est pas sécurisée.

WebSphere eXtreme Scale propose une implémentation disponible immédiatement pour cette interface.

L'implémentation par défaut utilise une paire de clés pour signer et vérifier la signature et utilise une clé confidentielle pour en chiffrer le contenu. Chaque serveur a un fichier de clés de type JCKES pour stocker cette paire (une clé privée et une clé publique), ainsi qu'une clé confidentielle. Le fichier de clés doit être de type JCKES pour pouvoir stocker les clés confidentielles.

Ces clés sont utilisées pour chiffrer et signer ou vérifier la chaîne secrète côté envoi. Le jeton est associé à un délai d'expiration. Côté réception, les données sont vérifiées, déchiffrées et comparées à la chaîne secrète du récepteur. Les protocoles de communication SSL (Secure Sockets Layer) ne sont pas obligatoires pour l'authentification entre une paire de serveurs car les clés privées et les clés publiques ont les mêmes fonctions. Toutefois, si la communication avec les serveurs n'est pas chiffrée, les données peuvent être volées en regardant la communication. Le jeton expirant bientôt, la menace pesant sur les attaques de type replay est minime. Cette possibilité décroît même de manière significative lorsque tous les serveurs sont déployés derrière un pare-feu.

L'inconvénient de cette approche est que les administrateurs WebSphere eXtreme Scale doivent générer des clés et en assurer le transport vers tous les serveurs, au cours duquel la sécurité risque d'être enfreinte.

Exemples de scripts permettant de créer les propriétés du gestionnaire de jetons par défaut

Comme indiqué dans la section précédente, vous pouvez créer un fichier de clés contenant une paire de clés pour signer et vérifier la signature, ainsi qu'une clé secrète pour chiffrer le contenu.

Vous pouvez par exemple utiliser la commande de l'outil de clé JDK 6 pour créer les clés, comme ci-dessous :

```
keytool -genkeypair -alias keypair1 -keystore key1.jck -storetype JCEKS -keyalg
rsa -dname "CN=sample.ibm.com, OU=WebSphere eXtreme Scale" -storepass key111 -keypass
keypair1 -validity 10000

keytool -genseckey -alias seckey1 -keystore key1.jck -storetype JCEKS -keyalg
DES  -storepass key111 -keypass seckey1 -validity 1000
```

Ces deux commandes créent une paire de clés "keypair1" et une clé confidentielle "seckey1". Vous pouvez alors configurer les éléments suivants dans le fichier de propriétés du serveur :

```
secureTokenKeyStore=key1.jck
secureTokenKeyStorePassword=key111
secureTokenKeyStoreType=JCEKS
secureTokenKeyPairAlias=keypair1
secureTokenKeyPairPassword=keypair1
secureTokenSecretKeyAlias=seckey1
secureTokenSecretKeyPassword=seckey1
secureTokenCipherAlgorithm=DES
secureTokenSignAlgorithm=RSA
```

Configuration

Consultez la rubrique Propriétés du serveur pour plus d'informations sur les propriétés utilisées pour configurer le gestionnaire de jetons sécurisé.

Authentification et autorisation des clients

Vous pouvez activer la sécurité et l'authentification des données d'identification pour authentifier les clients. En outre, vous pouvez autoriser les clients d'administration à accéder à la grille de données.

Authentification des clients d'application

L'authentification du client d'application consiste à activer la sécurité client-serveur et l'authentification des données d'identification et à configurer un authentificateur et un générateur de données d'identification.

Procédure

- Activez la sécurité client-serveur.

  Vous devez activer la sécurité sur le client et sur le serveur pour pouvoir vous authentifier auprès de la grille d'objets.

  1. Activez la sécurité client.

     WebSphere eXtreme Scale fournit un exemple de fichier de propriétés client, le fichier `sampleClient.properties`, dans le répertoire *racine_was*/`optionalLibraries/ObjectGrid/properties` pour une installation WebSphere Application Server ou le répertoire `/ObjectGrid/properties` dans une installation de serveurs mixtes. Vous pouvez modifier ce fichier en y entrant les valeurs de votre choix. Affectez à la propriété **`securityEnabled`** la valeur `true` dans le fichier `objectgridClient.properties`. La propriété **`securityEnabled`** indique si la sécurité est activée. Lorsqu'un client se connecte à un serveur, les valeurs du côté client et du côté serveur doivent toutes deux être égales à `true` ou à `false`. Par exemple, si la sécurité du serveur connecté est activée, la valeur de la propriété doit être associée à `true` du côté client pour que le client puisse se connecter au serveur.

     L'interface com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration

représente le fichier `security.ogclient.props`. Vous pouvez utiliser l'API publique com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory pour créer une instance de cette interface avec les valeurs par défaut ou vous pouvez créer une instance en transmettant le fichier des propriétés de sécurité du client ObjectGrid. Le fichier `security.ogclient.props` contient d'autres propriétés. Pour plus de détails, voir la documentation de l'API ClientSecurityConfiguration et celle de l'API ClientSecurityConfigurationFactory.

2. Activez la sécurité serveur.

   Pour activer la sécurité côté serveur, vous pouvez donner la valeur `true` à la propriété **`securityEnabled`** dans le fichier `security.xml`. Un fichier XML de descripteur de sécurité vous permettra de spécifier la configuration de la sécurité de la grille en isolant celle-ci de tous les éléments de configuration n'ayant pas trait à la sécurité.

- Activez l'authentification des données d'identification.

  Une fois que le client eXtreme Scale a extrait l'objet Credential en utilisant l'objet CredentialGenerator, l'objet Credential est envoyé avec la demande du client au serveur eXtreme Scale. Le serveur authentifie l'objet Credential avant de traiter la demande. Si l'authentification de l'objet Credential réussit, un objet Subject est renvoyé pour représenter cet objet Credential. Cet objet Subject est alors utilisé pour autoriser la demande.

  Définissez la propriété **`credentialAuthentication`** dans les fichiers de propriétés du client et du serveur afin d'activer l'authentification des données d'identification. Pour plus d'informations, voir Fichier de propriétés du client et Fichier de propriétés du serveur.

  Le tableau suivant présente les mécanismes d'authentification à utiliser selon les paramètres.

Tableau 39. Authentification des données d'identification dans les paramètres du client et du serveur

Authentification des données d'identification du client	Authentification des données d'identification du serveur	Résultat
Non	Jamais	Désactivé
Non	Pris en charge	Désactivé
Non	Requis	Cas d'erreur
Pris en charge	Jamais	Désactivé
Pris en charge	Pris en charge	Activé
Pris en charge	Requis	Activé
Requis	Jamais	Cas d'erreur
Requis	Pris en charge	Activé
Requis	Requis	Activé

- Configurez un authentificateur.

  Le serveur eXtreme Scale utilise le plug-in Authenticator pour authentifier l'objet Credential. Une implémentation de l'interface Authenticator obtient l'objet Credential qu'elle authentifie ensuite auprès d'un registre d'utilisateurs, un serveur LDAP, par exemple, et ainsi de suite. eXtreme Scale ne fournit aucune configuration de registre. La connexion à un registre d'utilisateurs et l'authentification auprès de celui-ci doivent être implémentées dans ce plug-in.

  Par exemple, une implémentation d'Authenticator extrait l'ID utilisateur et le mot de passe des données d'identification, les utilise pour la connexion et la validation auprès d'un serveur LDAP et crée un objet Subject résultant de

l'authentification. L'implémentation peut utiliser les modules de connexion JAAS (Java Authentication and Authorization Service). Un objet Subject est retourné comme résultat de l'authentification.

Vous pouvez configurer l'authentificateur dans le fichier XML descripteur de sécurité, comme dans l'exemple qui suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config/security">

 <security securityEnabled="true"
        loginSessionExpirationTime="300">

  <authenticator
className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
        </authenticator>

 </security>

</securityConfig>
```

L'option **-clusterSecurityFile** lors du démarrage d'un serveur sécurisé permet de définir le fichier XML de sécurité. Voir le tutoriel de sécurité Java SE dans *Présentation du produit* pour plus d'informations.

- Configurez un générateur de données d'identification système.

  Le générateur de données d'identification système permet de représenter la fabrique des données d'identification système. Les données d'identification système sont identiques aux données d'identification de l'administrateur. Vous pouvez configurer l'élément SystemCredentialGenerator dans le fichier XML de sécurité du catalogue, comme indiqué dans cet exemple :

  ```
  <systemCredentialGenerator className ="com.ibm.websphere.objectgrid.security.plugins.
   builtins.UserPasswordCredentialGenerator">
            <property name="properties" type="java.lang.String" value="manager manager1"
        description="username password" />
   </systemCredentialGenerator>
  ```

  Pour que cet exemple soit parlant, le nom d'utilisateur et le mot de passe sont indiqués en clair. Dans un environnement de production, ne stockez pas ces informations en clair.

  WebSphere eXtreme Scale fournit un générateur de données d'identification système par défaut qui utilise les données d'identification du serveur. Si vous n'indiquez pas ce générateur de façon explicite, le générateur par défaut est utilisé.

Autorisation des clients d'application

L'autorisation du client d'application consiste en des classes d'autorisation ObjectGrid, des mécanismes d'autorisation, une période de vérification des droits et une autorisation "accès réservé au créateur".

Pourquoi et quand exécuter cette tâche

Pour eXtreme Scale, l'autorisation est accordée en fonction de l'objet et des droits du sujet. Le produit prend en charge deux sortes de mécanismes d'autorisation : le service JAAS (Java Authentication and Authorization Service) et l'autorisation personnalisée.

Il existe quatre types de classes d'autorisation :

- La classe MapPermission représente les autorisations d'accès aux données dans les mappes ObjectGrid.
- La classe ObjectGridPermission représente les autorisations d'accès à ObjectGrid.

- La classe ServerMapPermission représente les autorisations d'accès aux mappes ObjectGrid sur le serveur à partir d'un client.
- La classe AgentPermission représente les autorisations de démarrage d'un agent sur le serveur.

Pour plus d'informations sur les API et les autorisations associées, voir la rubrique sur la programmation de l'autorisation client dans le *Guide de programmation*.

Procédure

1. Définissez la période de vérification des autorisations.

   eXtreme Scale prend en charge la mise en cache des résultats de la vérification des droits d'accès aux mappes pour des raisons de performances. Sans ce mécanisme, lorsqu'une méthode, qui est répertoriée dans la liste des méthodes de votre classe d'autorisation, est appelée, l'environnement d'exécution appelle le mécanisme d'autorisation configuré pour autoriser l'accès. Lorsque cette période est définie, le mécanisme d'autorisation est appelé périodiquement. Pour la liste des méthodes de chaque classe d'autorisation, voir la rubrique sur la programmation d'autorisation client dans *Guide de programmation*.

   Les informations relatives à l'autorisation des droits se fondent sur l'objet Subject. Lorsqu'un client essaie d'accéder aux méthodes, l'environnement d'exécution eXtreme Scale recherche l'objet Subject dans le cache. Si cet objet est introuvable, l'environnement d'exécution vérifie les droits qui lui sont accordés, puis stocke les droits dans un cache.

   La période de vérification des droits doit être définie avant l'initialisation de la grille d'objets. Vous pouvez la configurer de deux manières différentes :

   Vous pouvez utiliser le fichier XML ObjectGrid pour définir une grille d'objets ainsi que la période de vérification des droits. Dans l'exemple suivant, cette période est définie pour une durée de 45 secondes :

   ```
   <objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
    authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS"
    permissionCheckPeriod="45">
     <bean id="bean id="TransactionCallback"
   className="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
   ...
   </objectGrids>
   ```

   Si vous souhaitez créer une grille d'objets à l'aide des API, appelez la méthode suivante pour définir la période de vérification des droits. Cette méthode peut uniquement être appelée avant l'initialisation de l'instance ObjectGrid. Elle s'applique uniquement au modèle de programmation eXtreme Scale local lorsque vous instanciez directement une instance ObjectGrid.

   ```
   /**
    * This method takes a single parameter indicating how often you
    * want to check the permission used to allow a client access. If the
    * parameter is 0 then every single get/put/update/remove/evict call
    * asks the authorization mechanism, either JAAS authorization or custom
    * authorization, to check if the current subject has permission. This might be
    * prohibitively expensive from a performance point of view depending on
    * the authorization implementation, but if you need to have ever call check the
    * authorization mechanism, then set the parameter to 0.
    * Alternatively, if the parameter is > 0 then it indicates the number
    * of seconds to cache a set of permissions before returning to
    * the authorization mechanism to refresh them. This value provides much
    * better performance, but if the back-end
    * permissions are changed during this time then the ObjectGrid can
    * allow or prevent access even though the back-end security
    * provider was modified.
    *
    * @param period the permission check period in seconds.
    */
   void setPermissionCheckPeriod(int period);
   ```

2. Configurez l'autorisation d'accès réservé au créateur uniquement.

Avec l'autorisation "accès réservé au créateur", seul l'utilisateur (représenté par les objets Principal qui lui sont associés) ayant inséré une entrée dans une mappe ObjectGrid peut accéder (lecture, mise à jour, invalidation et suppression) à cette entrée.

Le modèle d'autorisation d'accès aux mappes ObjectGrid existant se fonde sur le type d'accès et non sur les entrées de données. En d'autres termes, un utilisateur dispose de droits d'accès d'un certain type (par exemple lecture, écriture, insertion, suppression ou invalidation) à toutes les données de la mappe ou ne détient aucun droit d'accès à aucune donnée. En revanche, eXtreme Scale n'autorise pas l'accès à certaines données seulement. Cette fonction constitue une nouvelle manière d'octroyer aux utilisateurs des droits d'accès aux entrées de données.

Dans un scénario où différents utilisateurs accèdent à différents jeux de données, ce modèle peut être utile. Lorsqu'un utilisateur charge des données à partir du stockage de persistance dans les mappes ObjectGrid, l'accès peut être autorisé par le stockage de persistance. Dans ce cas, il est inutile d'accorder une autre autorisation dans la couche de mappes ObjectGrid. Vous devez seulement faire en sorte que la personne qui charge les données dans la mappe peut y accéder en activant la fonction "réservé au créateur".

Valeurs des attributs en mode Réservé au créateur :

`disabled`
La fonction "accès réservé au créateur" est désactivée.

`complement`
La fonction "accès réservé au créateur" est activée et vient s'ajouter à l'autorisation d'accès aux mappes. En d'autres termes, les deux fonctions (autorisation d'accès aux mappes et fonction "accès réservé au créateur") sont opérationnelles. Vous pouvez donc limiter les opérations aux données. Le créateur ne peut par exemple pas invalider les données.

`supersede`
La fonction "accès réservé au créateur" est activée et remplace l'autorisation d'accès aux mappes. En d'autres termes, elle se substitue à cette autorisation, qui n'est plus opérationnelle.

a. Configurez le mode d'accès réservé au créateur avec un fichier XML.

   Vous pouvez utiliser le fichier XML ObjectGrid pour définir une grille d'objets et choisir le mode `disabled`, `complement` ou `supersede`, comme dans l'exemple suivant :

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
         accessByCreatorOnlyMode="supersede"
        <bean id="TransactionCallback"
              classname="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
    ...
</objectGrids>
```

b. Configurez le mode d'accès réservé au créateur à l'aide d'un programme.

   Si vous souhaitez créer une grille d'objets à l'aide d'un programme, vous pouvez appeler la méthode suivante pour définir le mode "accès réservé au créateur". L'appel de cette méthode s'applique uniquement au modèle de programmation eXtreme Scale local lorsque vous instanciez directement l'instance ObjectGrid :

```
/**
 * Set the "access by creator only" mode.
 * Enabling "access by creator only" mode ensures that only the user (represented
 * by the Principals associated with it), who inserts the record into the map,
 * can access (read, update, invalidate, and remove) the record.
 * The "access by creator only" mode can be disabled, or can complement the
 * ObjectGrid authorization model, or it can supersede the ObjectGrid
 * authorization model. The default value is disabled:
```

```
 * {@link SecurityConstants#ACCESS_BY_CREATOR_ONLY_DISABLED}.
 * @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_DISABLED
 * @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_COMPLEMENT
 * @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_SUPERSEDE
 *
 * @param accessByCreatorOnlyMode the access by creator mode.
 *
 * @since WAS XD 6.1 FIX3
*/
void setAccessByCreatorOnlyMode(int accessByCreatorOnlyMode);
```

Autre exemple : imaginez un scénario selon lequel une grille de données bancaires contient un compte de mappe ObjectGrid dont les deux utilisateurs sont Manager1 et Employee1. Les règles d'autorisation d'eXtreme Scale accordent tous les droits d'accès à Manager1, mais uniquement les droits d'accès en lecture à Employee1. Les règles JAAS d'autorisation d'accès aux mappes ObjectGrid sont représentées ci-dessous :

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    Principal com.acme.PrincipalImpl "Manager1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission
        "banking.account", "all"
};
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    Principal com.acme.PrincipalImpl "Employee1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission
        "banking.account", "read, insert"
};
```

A faire : Observez quelle incidence la fonction "accès réservé au créateur" a sur l'autorisation :

- **disabled** Si la fonction "accès réservé au créateur" est désactivée, l'autorisation d'accès aux mappes reste inchangée. L'utilisateur "Manager1" peut accéder à toutes les données de la mappe "account". L'utilisateur "Employee1" peut lire toutes les données de la mappe et en insérer, mais il ne peut ni mettre à jour, ni invalider, ni supprimer ces données.
- **complement** Si la fonction "accès réservé au créateur" est activée avec l'option "complement", les deux fonctions sont opérationnelles. L'utilisateur "Manager1" peut accéder aux données de la mappe "account", mais uniquement s'il les a chargées dans celle-ci. L'utilisateur "Employee1" peut lire les données de cette mappe, mais uniquement s'il les a chargées. (Il ne peut cependant ni mettre à jour, ni invalider, ni supprimer les données de cette mappe.)
- **supersede** Si la fonction "accès réservé au créateur" est activée avec l'option "supersede", l'autorisation d'accès aux mappes n'est pas activée. L'autorisation "accès réservé au créateur" est alors la seule règle en vigueur. L'utilisateur "Manager1" détient les mêmes privilèges que ceux liés au mode "complement" : il peut accéder aux données de la mappe "account" uniquement s'il les a chargées dans la mappe. Toutefois, l'utilisateur "Employee1" détient maintenant les droits d'accès complet aux données de la mappe "account" s'il les a chargés dans la mappe. En d'autres termes, les règles d'autorisation définies dans les règles Java Authentication and Authorization Service (JAAS) ne sont pas appliquées.

Autoriser les clients d'administration

Dans la sécurité d'administration, vous pouvez autoriser les utilisateurs à accéder à la grille de données. Certaines conditions sont requises, en fonction de votre environnement d'installation WebSphere eXtreme Scale et des utilisateurs à autoriser à accéder à la grille de données.

Pourquoi et quand exécuter cette tâche

Lorsque des utilisateurs sont autorisés à accéder à une grille de données WebSphere eXtreme Scale, ils peuvent également être autorisés à effectuer des opérations de gestion à l'aide de la commande **xscmd** ou **stopOgServer**. La plupart des déployeurs de grilles de données restreignent l'accès administrateur à un sous-ensemble des utilisateurs qui peuvent accéder aux données de la grille.

Procédure

1. Configurez l'autorisations des opérations **xscmd** et la commande **stopOgServer**.

   Si vous utilisez la commande suivante pour accéder à la grille de données, vous pouvez être également autorisé à exécuter des actions d'administration, telles qu'exécuter la commande **listAllJMXAddresses** :

   ```
   ./xscmd.sh -user <user> -password <password> <other_parameters>
   ```

   Si l'utilisateur peut exécuter la commande précédente, il peut exécuter n'importe quelle opération **xscmd** ou commande **stopOgServer**.

   Lorsque les composants eXtreme Scale sont exécutés avec WebSphere Application Server, utilisez la console d'administration WebSphere Application Server pour activer le gestionnaire de sécurité. Pour limiter l'accès des applications aux ressources locales, cliquez sur **Sécurité** > **Sécurité globale** et cochez les cases **Activer la sécurité d'administration** et **Utilisez la sécurité Java 2** pour limiter l'accès de l'application aux ressources locales.

   L'accès aux opérations de gestion est contrôlé par le gestionnaire de sécurité WebSphere Application Server et il est accordé uniquement aux utilisateurs qui appartiennent au rôle WebSphere Administrator. Vous devez exécuter les commandes **xscmd** et **stopOgServer** depuis le répertoire WebSphere Application Server.
2. Configurez les autorisations d'administration dans les installations autonomes.

   Lorsque les composants eXtreme Scale s'exécutent dans un environnement autonome, des étapes supplémentaires sont requises pour implémenter la sécurité d'administration. Vous devez exécuter les serveurs de catalogue et les serveurs de conteneur en utilisant le gestionnaire de sécurité Java, ce qui nécessite un fichier de règles.

   Le fichier de règles se présente comme suit :

   A faire : Il contient généralement des entrées MapPermission, comme indiqué dans «Tutoriel sur la sécurité Java SE - Etape 5», à la page 106.

```
grant codeBase "file:${objectgrid.home}/lib/*" {
permission java.security.AllPermission;
};

grant principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=OGSample" {
permission javax.management.MBeanPermission "*", "getAttribute,setAttribute,invoke,queryNames";
};
```

   Si le client est une application Java Spring, l'entrée `AgentPermission` est nécessaire dans le fichier de règles pour permettre au compte `CN=manager` d'accéder à la grille de données depuis le client Spring.

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,
OU=OGSample" {
permission com.ibm.websphere.objectgrid.security.AgentPermission "*",
"com.ibm.ws.objectgrid.spring.PutAgent";
};
```

Dans cet exemple, seul le gestionnaire principal est autorisé à exécuter des opérations d'administration à l'aide de la commande **xscmd** ou **stopOgServer**. Vous pouvez ajouter d'autres lignes en fonction des besoins pour fournir des autorisations supplémentaires de bean géré de principal. Un autre type de principal est nécessaire si vous utilisez l'authentification LDAP.

Entrez la commande suivante : UNIX Linux

```
startOgServer.sh <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
-Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=$OBJECTGRID_HOME
```

UNIX Linux 8.6+

```
startXsServer.sh <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
-Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=$OBJECTGRID_HOME
```

Windows

```
startOgServer.bat <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
-Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=%OBJCTGRID_HOME%
```

Windows 8.6+

```
startXsServer.bat <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
 -Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=%OBJCTGRID_HOME%
```

Activation de l'authentification LDAP dans les serveurs de catalogue et de conteneur eXtreme scale

Activez les serveurs WebSphere eXtreme Scale et les serveurs de catalogue pour l'authentification LDAP (Lightweight Directory Access Protocol) avec un fichier de règles Java Authentication and Authorization Service (JAAS) utilisé pour l'autorisation.

Pourquoi et quand exécuter cette tâche

Dans cette tâche, vous utilisez LDAP comme mécanisme d'authentification qui permet d'accéder à la grille de données, en fonction des autorisations que vous définissez dans le fichier de configuration de la politique d'autorisation JAAS.

Procédure

1. Créez un fichier `wxs_ldap.config`, par exemple :

   ```
   LDAPLogin {
    com.ibm.websphere.objectgrid.security.plugins.builtins.SimpleLDAPLoginModule
         required
    providerURL="ldap://yourldapserver.yourcompany.com:389/"
    factoryClass="com.sun.jndi.ldap.LdapCtxFactory"
   };
   ```

2. Créez un fichier `wxs_ldap.auth.config`. Remplacez le principal par l'utilisateur qui se connecte à la grille de données. Remplacez également `YourGridName` par le nom de la grille de données. Répétez cette étape aussi souvent que nécessaire pour les utilisateurs et les grilles de données supplémentaires. Examinez l'exemple suivant :

   ```
   grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
       principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=sample" {
       permission com.ibm.websphere.objectgrid.security.MapPermission "*.*", "all";

       permission com.ibm.websphere.objectgrid.security.ObjectGridPermission "*.*", "all";
   };
   ```

   Vous pouvez également accorder des droits d'accès à l'ensemble des grilles de données, par exemple :

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=sample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "*", "all";

    permission com.ibm.websphere.objectgrid.security.ObjectGridPermission "*", "all";
};
```

3. Créez un fichier `security_ldap.xml`, par exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security
../objectGridSecurity.xsd"
xmlns="http://ibm.com/ws/objectgrid/config/security">
<security securityEnabled="true" loginSessionExpirationTime="300" >
        <authenticator className ="com.ibm.websphere.objectgrid.
security.plugins.builtins.LDAPAuthenticator">
        </authenticator>
    </security>
</securityConfig>
```

4. Modifiez le fichier `objectGridServer.properties` avec les propriétés suivantes. Si vous ne disposez pas d'un fichier `objectGridServer.properties`, vous pouvez utiliser le fichier `sampleServer.properties` qui se trouve dans le répertoire *rép_base_wxs*`/properties` pour créer le fichier de propriétés.

```
securityEnabled=true

credentialAuthentication=Required
```

5. Démarrez les serveurs de catalogue.

   Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

```
-Dobjectgrid.cluster.security.url=file:///security/security.xml
-Dobjectgrid.server.props="/security/objectGridServer.properties"
-Djava.security.policy="/security/wxs_ldap_auth.config"
```

   Pour démarrer les serveurs de catalogue dans WebSphere Application Server, dans «Activation de l'authentification LDAP dans les serveurs de catalogue et de conteneur eXtreme scale», à la page 629.

6. Démarrez les serveurs de conteneur.

```
Dobjectgrid.server.props="/security/objectGridServer.properties"
-Djava.security.policy="/security/wxs_ldap_auth.config"
```

   Pour démarrer les serveurs de conteneur dans WebSphere Application Server, voir «Activation de l'authentification LDAP dans les serveurs de catalogue et de conteneur eXtreme scale», à la page 629.

7. Modifiez le fichier `objectGridClient.properties` sur le client. Si WebSphere Application Server est le client, le fichier que vous mettez à jour est *was_profile_dir*`/properties`.

```
securityEnabled=true

credentialAuthentication=Supported
```

8. Configurez le client pour qu'il envoie les données d'identification de connexion LDAP nécessaires. Chargez un fichier de propriétés de client. Ce fichier peut contenir l'ID et le mot de passe de l'utilisateur. Si le fichier de propriétés ne contient pas l'ID et le mot de passe de l'utilisateur, ajoutez-les à la configuration dans le programme client. Dans l'exemple suivant, un fichier de propriétés de

client est chargé en utilisant un paramètre de programme. Ensuite, l'ID et le mot de passe utilisateur sont ajoutés à la configuration.

```
String userid = "CN=manager,O=acme,OU=sample";

String pw="password";

// crée un objet ClientSecurityConfiguration à l'aide du fichier spécifié
ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
.getClientSecurityConfiguration(args[0]);

//crée un générateur CredentialGenerator en utilisant l'utilisateur
et le mot de passe.
CredentialGenerator credGen = new UserPasswordCredentialGenerator
(userid,password);
clientSC.setCredentialGenerator(credGen);

// crée un ObjectGrid en se connectant au serveur de catalogue
ClientClusterContext ccContext = ogManager.connect
("cataloghostname:2809", clientSC, null);
ObjectGrid og = ogManager.getObjectGrid(ccContext,
"YourGridName");'
```

Activation de l'authentification par fichier de clés dans les serveurs de conteneur et de catalogue eXtreme Scale

Activez les serveurs WebSphere eXtreme Scale et les serveurs de catalogue pour l'authentification par fichier de clés avec un fichier de clés JAAS (Java Authentication and Authorization Service) utilisé pour l'autorisation.

Pourquoi et quand exécuter cette tâche

Dans cette tâche, vous utilisez un fichier de clés comme mécanisme d'authentification qui permet d'accéder à la grille de données en fonction des autorisations que vous définissez dans le fichier de configuration de la règle d'autorisation JAAS.

Procédure

1. Créez un fichier de clés avec des alias de connexion, comme indiqué dans «Tutoriel sur la sécurité Java SE - Etape 4», à la page 102.
2. Créez le fichier `wxs_keystore.config`. Remplacez le principal par l'utilisateur qui se connecte à la grille de données. Remplacez également `YourGridName` par le nom de la grille de données. Répétez cette étape pour les utilisateurs et les grilles de données supplémentaires nécessaires. Examinez l'exemple suivant :

   ```
   KeyStoreLogin {
   com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule
   required
   keyStoreFile="/security/sampleKS.jks";

   }
   ```

3. Créez un fichier `security_ldap.xml`, par exemple :

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security
   ../objectGridSecurity.xsd"
   xmlns="http://ibm.com/ws/objectgrid/config/security">
   <security securityEnabled="true" loginSessionExpirationTime="300" >
           <authenticator className="com.ibm.websphere.objectgrid.security
   ```

```
.plugins.builtins.KeyStoreLoginAuthenticator>
        </authenticator>
    </security>
</securityConfig>
```

4. Modifiez le fichier `objectGridServer.properties` avec les propriétés suivantes. Si vous ne disposez pas d'un fichier `objectGridServer.properties`, vous pouvez utiliser le fichier `sampleServer.properties` qui se trouve dans le répertoire *rép_base_wxs*`/properties` pour créer le fichier de propriétés. Pour plus d'informations, voir «Configuration du mécanisme de quorum», à la page 327.

```
securityEnabled=true

credentialAuthentication=Required
```

5. Démarrez les serveurs de catalogue.

Obsolète :  **8.6+** Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

```
startOgServer.sh catalogServer -clusterSecurityFile /security/security.xml
-serverProps /security/objectGridServer.properties -jvmArgs
-Djava.security.auth.login.config="/security/wxs_keystore.config"

-Djava.security.policy="/security/wxs_ldap_auth.config"
```

8.6+

```
startXsServer.sh catalogServer -clusterSecurityFile /security/security.xml
-serverProps /security/objectGridServer.properties -jvmArgs
-Djava.security.auth.login.config="/security/wxs_keystore.config"

-Djava.security.policy="/security/wxs_ldap_auth.config"
```

6. Démarrez les serveurs de conteneur.

```
startOgServer.sh c0 -objectgridFile /xml/objectgrid.xml
-deploymentPolicyFile /xml/deployment.xml
-catalogServiceEndPoints cataloghostname:2809
-serverProps /security/objectGridServer.properties
-jvmArgs -Djava.security.auth.login.config="/security/wxs_keystore.config"

-Djava.security.policy="/security/wxs_ldap_auth.config"
```

8.6+

```
startXsServer.sh c0 -objectgridFile /xml/objectgrid.xml
-deploymentPolicyFile /xml/deployment.xml
-catalogServiceEndPoints cataloghostname:2809
-serverProps /security/objectGridServer.properties
-jvmArgs -Djava.security.auth.login.config="/security/wxs_keystore.config"

-Djava.security.policy="/security/wxs_ldap_auth.config"
```

7. Modifiez le fichier `objectGridClient.properties` sur le client. Si WebSphere Application Server est le client, le fichier que vous mettez à jour est *was_profile_dir*`/properties`.

```
securityEnabled=true

credentialAuthentication=Supported
```

```
transportType=TCP/IP

singleSignOnEnabled=false
```

8. Modifiez l'application client pour qu'elle envoie les données d'identification de connexion du fichier de clés.

```
String userid = "CN=manager,O=acme,OU=sample";

String pw="password";
// crée un objet ClientSecurityConfiguration à l'aide du fichier spécifié
ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
.getClientSecurityConfiguration(args[0]);

// crée un CredentialGenerator en utilisant le nom de
l'utilisateur et le mot de passe fournis.
CredentialGenerator credGen =
new UserPasswordCredentialGenerator(userid,password);
clientSC.setCredentialGenerator(credGen);

// crée un ObjectGrid en se connectant au serveur de catalogue
ClientClusterContext ccContext = ogManager.connect
("cataloghostname:2809", clientSC, null);
ObjectGrid og = ogManager.getObjectGrid(ccContext,
"YourGridName");'
```

Configuration des types de transports sécurisés

TLS (Transport layer security) fournit une communication sécurisée entre le client et le serveur. Le mécanisme de communication utilisé dépend de la valeur du paramètre **transportType** spécifié dans les fichiers de configuration du client et du serveur.

Pourquoi et quand exécuter cette tâche

Lorsque vous utilisez SSL (Secure Sockets Layer), les paramètres de configuration SSL doivent être définis dans le client et le serveur. Dans un environnement Java SE, la configuration du protocole SSL s'effectue dans les fichiers de propriétés du client ou du serveur. Si le client ou le serveur se trouve dans WebSphere Application Server, vous pouvez utiliser les paramètres de transport CSIV2 existants WebSphere Application Server de vos serveurs de conteneur et de vos clients. Pour plus d'informations, voir «Intégration de la sécurité dans WebSphere Application Server», à la page 643.

Tableau 40. Protocole de transport à utiliser avec les paramètres de transport client et serveur.

Si les paramètres transportType sont différents entre le client et le serveur, le protocole résultant peut varier ou entraîner une erreur.

Propriété de client transportType	Propriété de serveur transportType	Résultat du protocole
TCP/IP	TCP/IP	TCP/IP
TCP/IP	SSL pris en charge	TCP/IP
TCP/IP	SSL requis	Erreur
SSL pris en charge	TCP/IP	TCP/IP
SSL pris en charge	SSL pris en charge	SSL (en cas d'échec du protocole SSL, TCP/IP)
SSL pris en charge	SSL requis	SSL
SSL requis	TCP/IP	Erreur
SSL requis	SSL pris en charge	SSL
SSL requis	SSL requis	SSL

Procédure

1. Pour définir la propriété **transportType** dans la configuration de sécurité du client, voir Fichier de propriétés du client.
2. Pour définir la propriété **transportType** dans la configuration de sécurité du conteneur et du serveur de catalogue, voir Fichier de propriétés du serveur.

Protocole TLS et couche de connexion sécurisée

WebSphere eXtreme Scale prend en charge les protocoles TCP/IP et TLS/SSL pour assurer une communication sécurisée entre les clients et les serveurs.

Activation de TLS/SSL dans les deux sens

TLS/SSL est parfois activé dans un seul sens. Par exemple, le certificat public du serveur est importé dans le fichier de clés certifiées, mais pas le certificat public du client n'est pas importé vers le fichier de clés certifiées du serveur. Toutefois, WebSphere eXtreme Scale utilise largement des agents de grille de données. Un agent de grille de données se caractérise par le fait qu'il crée un connexion lorsque le serveur répond au client. Le serveur eXtreme Scale fait alors office de client. Par conséquent, vous devez importer le certificat public du client dans le fichier de clés certifiées du serveur.

Activation de la sécurité de transport pour le kit JDK Oracle

WebSphere eXtreme Scale nécessite IBM Java Secure Sockets Extension (IBMJSSE) ou IBM Java Secure Sockets Extension 2 (IBMJSSE2). Les fournisseurs IBMJSSE et IBMJSSE2 contiennent une implémentation de référence prenant en charge les protocoles SSL et TLS, ainsi qu'un framework d'API.

Le kit JDK Oracle ne fournit pas les fournisseurs IBM JSSE et IBM JSSE2 et la sécurité de transport ne peut donc pas être activée avec un kit JDK Oracle. Pour l'activer, un kit JDK Oracle fourni avec WebSphere Application Server est requis. Le kit JDK Oracle fourni avec WebSphere Application Server contient les fournisseurs IBM JSSE et IBM JSSE2.

Voir «Configuration d'un ORB personnalisé», à la page 354 pour plus d'informations sur l'utilisation d'un JDK non-IBM pour WebSphere eXtreme Scale. Si `-Djava.endorsed.dirs` est configuré, il pointe vers les répertoires `objectgridRoot/lib/endorsed` et `JRE/lib/endorsed`. Le répertoire `objectgridRoot/lib/endorsed` est requis de sorte que la fonction ORB IBM est utilisée et le répertoire `JRE/lib/endorsed` est requis pour le chargement des fournisseurs JSSE IBM et JSSE IBM.

Etudiez l'étape 4 du tutoriel de sécurité dans *Présentation du produit* pour plus d'informations sur la définition des propriétés SSL requises, créer des fichiers de clés et de clés certifiées et pour démarrer des serveurs sécurisés dans WebSphere eXtreme Scale.

Définition des paramètres SSL (Secure Sockets Layer) des clients ou des serveurs

La manière de définir les paramètres SSL est différente pour les clients et les serveurs.

Pourquoi et quand exécuter cette tâche

TLS/SSL est parfois activé dans un seul sens. Par exemple, le certificat public du serveur est importé dans le fichier de clés certifiées, mais pas le certificat public du client n'est pas importé vers le fichier de clés certifiées du serveur. Toutefois, WebSphere eXtreme Scale utilise largement des agents de grille de données. Un agent de grille de données se caractérise par le fait qu'il crée une connexion lorsque le serveur répond au client. Le serveur eXtreme Scale fait alors office de client. Par conséquent, vous devez importer le certificat public du client dans le fichier de clés certifiées du serveur.

Procédure

- Définissez les paramètres SSL du client.

  Utilisez l'une des options suivantes pour définir les paramètres SSL sur le client :

  – Créez un objet com.ibm.websphere.objectgrid.security.config.SSLConfiguration à l'aide de la classe com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory.

  – Configurez les paramètres dans le fichier `client.properties`. Vous pouvez ensuite définir le fichier de propriétés comme propriété de client JVM ou utiliser les API WebSphere eXtreme Scale. Transmettez le fichier de propriétés dans la méthode ClientSecurityConfigurationFactory.getClientSecurityConfiguration(String) du client et utilisez l'objet retourné comme paramètre dans la méthode ObjectGridManager.connect(String, ClientSecurityConfiguration, URL).

- Configurez les paramètres SSL du serveur.

  Les paramètres SSL sont configurés pour les serveurs qui utilisent le fichier `server.properties`. Pour démarrer un serveur de conteneur ou de catalogue avec un fichier de propriétés spécifique, utilisez le paramètre **-serverProps** dans le script **startOgServer** ou **startXsServer**. Pour plus d'informations sur les paramètres SSL que vous pouvez définir pour les serveurs eXtreme Scale, voir Propriétés du serveur de sécurité.

Sécurité JMX (Java Management Extensions)

Vous pouvez sécuriser les invocations de beans gérés (MBean) dans un environnement réparti.

Pour plus d'informations sur les beans gérés disponibles, voir «Administration avec les beans gérés (MBeans)», à la page 537.

Dans une topologie de déploiement réparti, les beans gérés sont directement hébergés sur les serveurs de catalogue et les serveurs de conteneur. En général, la sécurité JMX dans une topologie répartie suit la spécification de sécurité JMX telle que spécifiée dans la spécification JMX (Java Management Extensions). Elle est composée des trois parties suivantes :

1. Authentification : le client distant doit être authentifié dans le serveur de connecteur.
2. Contrôle d'accès : le contrôle de l'accès des beans gérés définit les privilèges d'accès aux informations de beans gérés et les droits d'exécution des opérations de beans gérés.
3. Transfert sécurisé : le transfert entre le client et le serveur JMX peut être sécurisé à l'aide du protocole TLS/SSL.

Authentification

JMX offre des méthodes aux serveurs de connecteur pour authentifier les clients distants. Pour le connecteur RMI, l'authentification est effectuée en fournissant un objet qui implémente l'interface JMXAuthenticator lors de la création du serveur de connecteur. Par conséquent, eXtreme Scale implémente cette interface JMXAuthenticator à utiliser le plug-in ObjectGrid Authenticator pour authentifier les clients distants. Voir «Tutoriel sur la sécurité Java SE - Etape 2», à la page 98 pour plus de détails sur l'authentification d'un client par eXtreme Scale.

Le client JMX suit les API JMX pour offrir des données d'identification permettant la connexion au serveur de connecteur. L'infrastructure JMX transmet les données d'identification au serveur de connecteur et appelle l'implémentation JMXAuthenticator pour l'authentification. Comme décrit précédemment, l'implémentation JMXAuthenticator délègue ensuite l'authentification à l'implémentation de l'authentificateur ObjectGrid.

Passez en revue l'exemple présenté ci-dessous, qui décrit comment établir la connexion à un serveur de connecteur à l'aide de données d'identification :

```
javax.management.remote.JMXServiceURL jmxUrl = new JMXServiceURL(
        "service:jmx:rmi:///jndi/rmi://localhost:1099/objectgrid/MBeanServer");

    environment.put(JMXConnector.CREDENTIALS, new UserPasswordCredential("admin", "xxxxxx"));

    // Créez le JMXCconnectorServer
    JMXConnector cntor = JMXConnectorFactory.newJMXConnector(jmxUrl, null);

    // Connectez et appelez une opération sur le MBeanServer distant
    cntor.connect(environment);
```

Dans l'exemple précédent, un objet UserPasswordCredential est fourni avec l'ID utilisateur `admin` et le mot de passe `xxxxx`. Cet objet UserPasswordCredential est défini dans la mappe d'environnement qui est utilisée dans la méthode JMXConnector.connect(Map). Cet objet UserPasswordCredential est ensuite transmis au serveur par l'infrastructure JMX, puis à l'infrastructure d'authentification ObjectGrid pour authentification.

Le modèle de programmation client respecte strictement les spécifications JMX.

Contrôle d'accès

Un serveur de beans gérés JMX peut avoir accès aux informations sensibles et peut être en mesure d'effectuer des opérations sensibles. JMX offre le contrôle d'accès requis permettant d'identifier les clients pouvant accéder à telles ou telles informations et qui peut effectuer ces opérations. Le contrôle d'accès repose sur le modèle de sécurité Java standard en définissant des autorisations de contrôle d'accès au serveur de beans gérés et aux opérations correspondantes.

Pour le contrôle d'accès ou l'autorisation des opérations JMX, eXtreme Scale repose sur le support JAAS fourni par l'implémentation JMX. A n'importe quel stade de l'exécution d'un programme, il existe un ensemble d'autorisations maintenu dans une unité d'exécution. Lorsqu'une unité d'exécution appelle une opération de spécification JMX, ces autorisations sont connues sous le terme d'autorisations de maintien. Lorsqu'une opération JMX est effectuée, une vérification de sécurité est réalisée pour vérifier si l'autorisation requise est concernée par l'autorisation de maintien.

La définition des règles d'administration de beans gérés respecte le format de la stratégie Java. Par exemple, la stratégie suivante octroie à tous les signataires et

bases de code le droit d'extraire l'adresse JMX du serveur pour PlacementServiceMBean avec une restriction pour le domaine com.ibm.websphere.objectgrid domain.

```
grant {
    permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
     [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

Vous pouvez utiliser l'exemple de stratégie suivant pour compléter l'autorisation en fonction de l'identité du client distant. La stratégie octroie la même autorisation de bean géré que celle présentée dans l'exemple précédent, sauf pour les utilisateurs dont le nom X500Principal est : `CN=Administrator,OU=software,O=IBM,L=Rochester,ST=MN,C=US`.

```
grant principal javax.security.auth.x500.X500Principal "CN=Administrator,OU=software,O=IBM,
    L=Rochester,ST=MN,C=US" {permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
    [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

Les stratégies Java sont uniquement vérifiées si le gestionnaire de sécurité est activé. Démarrez les serveurs de catalogue et les serveurs de conteneur à l'aide de l'argument JVM -Djava.security.manager pour forcer le contrôle d'accès des opérations de beans gérés.

Transfert sécurisé

Le transfert entre le client et le serveur JMX peut être sécurisé à l'aide du protocole TLS/SSL. Si le type de transfert du serveur de catalogues ou du serveur conteneur est défini sur SSL_Required ou SSL_Supported, vous devez utiliser le protocole SSL pour établir la connexion au serveur JMX.

Pour utiliser le protocole SSL, vous devez configurer le fichier de clés certifiées, le type de fichier de clés certifiées et le mot de passe du fichier de clés certifiées sur le client MBean en utilisant les propriétés système -D :

1. `-Djavax.net.ssl.trustStore=TRUST_STORE_LOCATION`
2. `-Djavax.net.ssl.trustStorePassword=TRUST_STORE_PASSWORD`
3. `-Djavax.net.ssl.trustStoreType=TRUST_STORE_TYPE`

Si vous utilisez com.ibm.websphere.ssl.protocol.SSLSocketFactory comme fabrique de sockets SSL dans le fichier*rép_base_java*`/jre/lib/security/java.security`, utilisez les propriétés suivantes :

1. `-Dcom.ibm.ssl.trustStore=TRUST_STORE_LOCATION`
2. `-Dcom.ibm.ssl.trustStorePassword=TRUST_STORE_PASSWORD`
3. `-Dcom.ibm.ssl.trustStoreType=TRUST_STORE_TYPE`

Pour obtenir ces informations lorsque le protocole TLS/SSL (Transport Layer Security/Secure Sockets Layer) est activé dans des configurations autonomes, vous devez démarrer les serveurs de catalogue et de conteneur avec l'ensemble de ports de service JMX. Utilisez l'une des méthodes suivantes pour configurer le port de service JMX :

- Utilisez l'option **-JMXServicePort** dans le script **startOgServer** ou **startXsServer**.
- Si vous utilisez un serveur embarqué, appelez la méthode setJMXServicePort dans l'interface ServerProperties pour définir le port de service JMX.

La valeur par défaut du port de service JMX sur les serveurs de catalogue est 1099. Vous devez utiliser un numéro de port différent pour chaque machine virtuelle Java dans votre configuration. Si vous souhaitez utiliser JMX/RMI, vous devez spécifier explicitement l'option **-JMXServicePort** et le numéro de port, même si vous souhaitez utiliser la valeur de port par défaut.

Vous devez définir le port de service JMX si vous souhaitez afficher les informations du serveur de conteneur à partir du serveur de catalogue. Par exemple, le port est requis lorsque vous utilisez la commande **xscmd -c showMapSizes**.

Définissez le port de connecteur JMX afin d'éviter la création d'un port éphémère. Utilisez l'une des méthodes suivantes pour configurer le port de connecteur JMX.

- Utilisez l'option **-JMXConnectorPort** dans le script **startOgServer** ou **startXsServer**.
- Si vous utilisez un serveur embarqué, appelez la méthode setJMVConnectorPort dans l'interface ServerProperties.

Intégration de la sécurité à des fournisseurs externes

Pour protéger vos données, le produit peut intégrer plusieurs fournisseurs de sécurité.

WebSphere eXtreme Scale peut intégrer un programme de sécurité externe. Ce programme externe doit fournir des services d'authentification et d'autorisation pour WebSphere eXtreme Scale. WebSphere eXtreme Scale dispose de points de plug-in pour intégrer une implémentation de sécurité.WebSphere eXtreme Scale a été intégré avec succès aux composants suivants :

- Protocole LDAP (Lightweight Directory Access Protocol)
- Kerberos
- Sécurité ObjectGrid
- Tivoli Access Manager
- Service JAAS (Java Authentication and Authorization Service)

eXtreme Scale utilise le fournisseur de sécurité pour les tâches suivantes :

- Authentifier les clients sur les serveurs.
- Autoriser les clients à accéder à certains artefacts eXtreme Scale ou à préciser ce qui peut être fait avec les artefacts eXtreme Scale.

eXtreme Scale propose les types d'autorisations suivants :

Autorisation de mappes
Les clients et les groupes peuvent être autorisés à insérer, lire, mettre à jour, expulser ou supprimer des opérations sur les mappes.

Autorisation ObjectGrid
Les clients ou les groupes peuvent être autorisés à effectuer des requêtes de type objet ou entité sur objectGrids.

Autorisation de l'agent DataGrid
Les clients ou les groupes peuvent être autorisés à permettre aux agents DataGrid d'être déployés en une base de données ObjectGrid.

Autorisation de mappes côté serveur
Les clients ou les groupes peuvent être autorisés à répliquer une mappe de serveur côté client ou à créer un index dynamique pour la mappe de serveur.

Autorisation d'administration
Les clients ou les groupes peuvent être autorisés à effectuer des tâches administratives.

Remarque : Si la sécurité est activée pour le dorsal, rappelez-vous que ces paramètres ne sont plus suffisants pour protéger vos données. Les paramètres de sécurité de votre base de données ou autre magasin de données ne sont transférés en aucune façon vers votre cache. Vous devez protéger séparément les données à présent mises en cache en utilisant le mécanisme de sécurité eXtreme Scale, qui inclut l'authentification, l'autorisation et la sécurité du niveau de transport.

Important : Utilisez un kit de développement ou d'environnement d'exécution Version 1.6 ou version suivante pour prendre en charge la sécurité de transport SSL avec WebSphere eXtreme Scale Version 7.1.1 et les versions ultérieures.

Sécurisation du service de données REST

Vous pouvez sécuriser un bon nombre d'aspects du service de données REST. L'accès au service de données REST d'eXtreme Scale peut être sécurisé via l'authentification et l'autorisation. Il peut également être contrôlé par des règles de configuration à portée de service, appelées règles d'accès. La sécurité des transports est le troisième élément concerné en matière de sécurisation.

Pourquoi et quand exécuter cette tâche

L'accès au service de données REST d'eXtreme Scale peut être sécurisé via l'authentification et l'autorisation. L'authentification et l'autorisation s'effectuent grâce à l'intégration à la sécurité d'eXtreme Scale.

L'accès peut également être contrôlé par des règles de configuration à portée service, appelées règles d'accès. Il existe deux types de règles d'accès : droits d'opérations du service qui contrôlent les opérations CRUD autorisées par le service et les droits d'accès aux entités qui contrôlent les opérations CRUD autorisées pour un type donné d'entité.

La sécurité des transports est fournie par la configuration du conteneur d'hébergement pour les connexions entre le client Web et le service REST. Et la sécurité du transport est fournie par la configuration client eXtreme Scale (pour le service REST aux connexions de grille eXtreme Scale).

Procédure

- Contrôlez l'authentification et l'autorisation.

  L'accès au service de données REST d'eXtreme Scale peut être sécurisé via l'authentification et l'autorisation. L'authentification et l'autorisation s'effectuent par l'intégration à la sécurité eXtreme Scale.

  Le service de données REST eXtreme Scale utilise la sécurité eXtreme Scale, pour l'authentification et l'autorisation, pour déterminer les utilisateurs qui peuvent accéder au service et les opérations qu'ils sont autorisés à effectuer via le service. Le service de données REST eXtreme Scale utilise soit des données d'identification globales configurées, avec utilisateur et mot de passe, soit des données d'identification dérivées d'une d'authentification HTTP BASIC qui est

envoyée avec chaque transaction à la grille de données eXtreme Scale où s'effectuent l'authentification et l'autorisation.

1. Configurez dans la grille l'authentification et l'autorisation des clients eXtreme Scale. Voir «Intégration de la sécurité à des fournisseurs externes», à la page 638 pour des explications détaillées sur la manière de configurer l'authentification et l'autorisation des clients eXtreme Scale.
2. Configurez le client eXtreme Scale, qui est utilisé par le service REST, pour la sécurité.

   Le service de données REST d'eXtreme Scale fait appel à la bibliothèque des clients eXtreme Scale lorsqu'il communique avec la grille eXtreme Scale. Il en résulte que le client eXtreme Scale doit être configuré pour la sécurité d'eXtreme Scale.

   L'authentification du client eXtreme Scale est activée via des propriétés dans le fichier des propriétés du client objectgrid. Au minimum, les attributs suivants doivent être activés lorsqu'on utilise la sécurité du client avec le service REST :

   ```
   securityEnabled=true
   credentialAuthentication=Supported [-ou-] Required
   credentialGeneratorProps=utilisateur:motdepasse [-ou-]
   {xor encoded utilisateur:motdepasse}
   ```

   A faire : L'utilisateur et le mot de passe spécifiés dans la propriété credentialGeneratorProps doivent correspondre à un ID du registre d'authentification et disposer de droits de règles ObjectGrid suffisants pour se connecter à des ObjectGrids et en créer.

   Un exemple de fichier de règles de client objectgrid se trouve dans *rép_base_servicerest*`/security/security.ogclient.properties`. Voir également Fichier de propriétés du client.
3. Configurez la sécurité du service de données REST d'eXtreme Scale.

   Le fichier des propriétés de configuration du service de données REST eXtreme Scale doit contenir les entrées suivantes pour pouvoir être intégré à la sécurité eXtreme Scale :

   ```
   ogClientPropertyFile=nom_fichier
   ```

   ogClientPropertyFile est l'adresse du fichier de propriétés qui contient les propriétés du client ObjectGrid mentionnées au point précédent. Lorsque la sécurité est activée, le service REST utilise ce fichier pour initialiser le client eXtreme Scale afin de communiquer avec la grille.

   ```
   loginType=basic [-ou-] none
   ```

   La propriété loginType configure le service REST pour le type d'ouverture de session. Si la valeur `none` est définie, l'ID utilisateur “global” et le mot de passe définis par credentialGeneratorProps seront envoyés à la grille pour chaque transaction. Si la valeur `basic` est spécifiée, le service REST présente au client une authentification HTTP BASIC en demandant des données d'identification qu'il envoie dans chaque transaction lorsqu'il communique avec la grille.

   Pour plus d'informations sur les propriétés ogClientPropertyFile et loginType, voir Fichier de propriétés du service de données REST.

- Appliquez des règles d'accès.

  L'accès peut également être contrôlé par des règles de configuration de portée service, désignées sous le nom de règles d'accès. Il existe deux types de règles d'accès : les droits d'opérations du service qui contrôlent les opérations CRUD autorisées par le service et les droits d'accès aux entités qui contrôlent les opérations CRUD autorisées pour un type donné d'entité.

Le service de données REST d'eXtreme Scale autorise, si on le souhaite, des règles d'accès configurables à accéder de manière restreinte au service et aux entités contenues dans ce dernier. Ces règles d'accès sont spécifiées dans le fichier des propriétés des droits d'accès du service REST. Le nom de ce fichier est spécifié dans le fichier des propriétés du service de données REST par la propriété wxsRestAccessRightsFile. Pour plus d'informations sur cette propriété, voir Fichier de propriétés du service de données REST. Ce fichier est un fichier de propriétés Java classique avec des paires clé/valeur. Il existe deux types de règles d'accès : les droits d'opérations du service qui contrôlent les opérations CRUD autorisées par le service et les droits d'accès aux entités qui contrôlent les opérations CRUD autorisées pour un type donné d'entité.

1. Configurez les droits d'opérations du service.

   Les droits d'opérations du service spécifient les droits d'accès qui s'appliquent à tous les ObjectGrids exposés via le service REST ou à toutes les entités de l'ObjectGrid individuel qui est spécifié.

   Utilisez la syntaxe suivante.

   ```
   serviceOperationRights=droit_opérations_service
   serviceOperationRights.nom_grille -ou- *=droit_opérations_service
   ```

   où

   – serviceOperationRights peut être l'un des suivants : [NONE, READSINGLE, READMULTIPLE, ALLREAD, ALL]

   – serviceOperationRights.*nom_grille -ou-* * implique que le droit d'accès s'applique à tous les ObjectGrids, autrement le nom d'un ObjectGrid spécifique peut être fourni.

   Par exemple :

   ```
   serviceOperationsRights=ALL
   serviceOperationsRights.*=NONE
   serviceOperationsRights.EMPLOYEEGRID=READSINGLE
   ```

   Le premier exemple spécifie que toutes les opérations du service sont autorisées pour tous les ObjectGrids exposés par ce service REST. Le deuxième exemple est semblable au premier car il s'applique également à tous les ObjectGrids exposés par le service REST, mais il spécifie des droits d'accès NONE, ce qui signifie qu'aucune opération du service n'est autorisée sur les ObjectGrids. Le dernier exemple spécifie comment contrôler les opérations du service pour une grille spécifique ; ici seules les opérations de lecture qui donnent un seul enregistrement sont autorisées pour toutes les entités de la grille EMPLOYEEGRID.

   La valeur par défaut utilisée par le service REST est `serviceOperationsRights=ALL`, ce qui signifie que toutes les opérations sont autorisées pour tous les ObjectGrids exposés par ce service. Cela diffère de l'implémentation Microsoft pour laquelle la valeur par défaut est `NONE`, si aucune opération n'est autorisée sur le service REST.

   Important : Les droits d'opérations du service sont évalués dans l'ordre dans lequel ils sont spécifiés dans ce fichier, ce qui fait que le dernier droit à être spécifié prendra le pas sur les droits qui viennent avant lui.

2. Configurez les droits d'accès aux entités.

   Les droits d'ensembles d'entités spécifient les droits d'accès qui s'appliquent aux entités de l'ObjectGrid spécifique qui est exposé via le service REST. Ces droits permettent d'imposer un contrôle bien plus étroit et bien plus granulaire de l'accès à des entités d'un ObjectGrid individuel que ne le permettent les droits d'opérations du service.

Utilisez la syntaxe suivante.

```
entitySetRights.nom_grille.nom_entité=droit_ensemble_entités
```

où

– *droit_ensemble_entités* peut être l'un des droits suivants

Tableau 41. Droits d'accès à des entités. Valeurs prises en charge.

Droit d'accès	Description
NONE	Refuse tout droit d'accès aux données
READSINGLE	Autorise la lecture d'un seul élément de données
READMULTIPLE	Autorise la lecture d'ensembles de données
ALLREAD	Autorise toutes les opérations de lecture (élément simple ou ensembles de données)
WRITEAPPEND	Autorise la création de nouveaux éléments de données dans les ensembles de données
WRITEREPLACE	Autorise le remplacement de données
WRITEDELETE	Autorise la suppression d'éléments de données dans les ensembles de données
WRITEMERGE	Autorise la fusion de données
ALLWRITE	Autorise toutes les opérations d'écriture (création, remplacement, fusion ou suppression) de données
ALL	Autorise la création, la lecture, la modification et la suppression de données

– *nom_entité* est le nom d'un ObjectGrid spécifique au sein du service REST

– *nom_grille* est le nom d'une entité spécifique au sein de l'ObjectGrid spécifié

Remarque : Si les droits d'opérations du service et les droits d'ensembles d'entités sont spécifiés en même temps pour un ObjectGrid et ses entités, le droit appliqué sera le plus restrictif des deux, comme le montrent les exemples qui suivent. Rappelez-vous également que les droits d'ensembles d'entités sont évalués dans l'ordre où ils sont spécifiés dans le fichier. Le dernier droit à être spécifié prendra le pas sur ceux qui viennent avant lui.

Exemple 1 : Si serviceOperationsRights.NorthwindGrid=READSINGLE et entitySetRights.NorthwindGrid.Customer=ALL sont spécifiés. READSINGLE sera appliqué pour l'entité Customer.

Exemple 2 : Si serviceOperationsRights.NorthwindGrid=ALLREAD est spécifié et qu'entitySetRights.NorthwindGrid.Customer=ALLWRITE l'est aussi, seules des opérations de lecture seront autorisées pour toutes les entités de NorthwindGrid. Mais, en ce qui concerne Customer, ses droits d'ensembles d'entités empêcheront toute lecture (puisque c'est ALLWRITE qui est spécifié) et de ce fait l'entité Customer aura NONE comme droit d'accès.

- Sécurisez les transports.

  La sécurité du transport est fournie par la configuration du conteneur d'hébergement pour les connexions entre le client Web et le service REST. La sécurité du transport est fournie par la configuration du client eXtreme Scale pour les connexions entre le service Web et la grille eXtreme Scale.

1. Sécurisez la connexion entre le client et le service REST. La sécurité des transports pour cette connexion est fournie par l'environnement du conteneur hébergeant et non dans eXtreme Scale.
2. Sécurisez la connexion entre le service REST et la grille eXtreme Scale. La sécurité des transports pour cette connexion est configurée dans eXtreme Scale. Voir «Protocole TLS et couche de connexion sécurisée», à la page 634.

Intégration de la sécurité dans WebSphere Application Server

Lorsque vous déployez WebSphere eXtreme Scale dans un environnement WebSphere Application Server, vous vous pouvez simplifier le flux d'authentification et la configuration de la sécurité de la couche de transport à partir de WebSphere Application Server.

Flux d'authentification simplifié

Lorsque les clients et serveurs eXtreme Scale sont exécutés dans WebSphere Application Server et dans le même domaine de sécurité, vous pouvez utiliser l'infrastructure de sécurité de WebSphere Application Server pour propager les données d'accès pour l'authentification du client sur le serveur eXtreme Scale. Par exemple, si un servlet agit en tant que client eXtreme Scale pour se connecter à un serveur eXtreme Scale du même domaine de sécurité et si le servlet est déjà authentifié, il est possible de propager le jeton d'authentification du client (servlet) vers le serveur, puis utiliser l'infrastructure de sécurité de WebSphere Application Server pour reconvertir ce jeton en données d'accès du client.

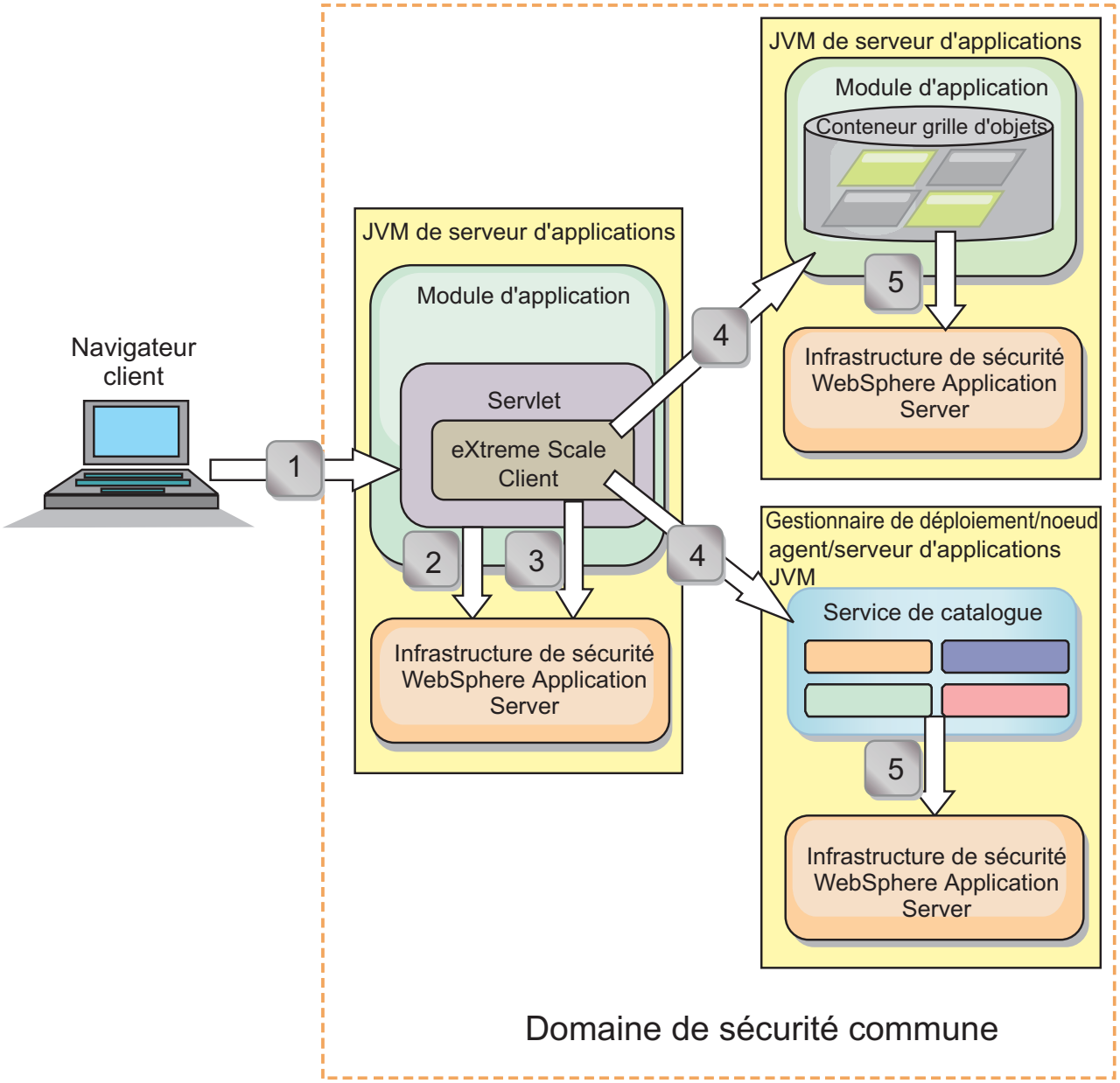


Figure 71. Flux d'authentification pour les serveurs dans le même domaine de sécurité

Dans le diagramme précédent, les serveurs d'applications se trouvent dans le même domaine de sécurité. Un serveur d'applications héberge l'application Web qui est également un client eXtreme Scale. L'autre serveur d'applications héberge le serveur de conteneur. Le gestionnaire de déploiement ou la machine virtuelle Java héberge le service de catalogue.

Remarque : Utilisez ce type de configuration dans les environnements de déploiement. Cependant, pour les environnements de production, exécutez les serveurs de catalogue dans des processus distincts et, dans la mesure du possible, sur un système différent de celui des serveurs de conteneur.
Les flèches dans le diagramme indiquent le flux du processus d'authentification :

1. Un utilisateur d'application d'entreprise utilise un navigateur Web pour se connecter au premier serveur d'applications avec un nom d'utilisateur et un mot de passe.

2. Le premier serveur d'applications envoie le nom d'utilisateur et le mot de passe du client à l'infrastructure de sécurité WebSphere Application Server pour s'authentifier auprès du registre des utilisateurs. Par exemple, ce registre d'utilisateurs peut être un serveur LDAP. Par conséquent, les informations de sécurité sont stockées dans l'unité d'exécution du serveur d'applications.
3. Le fichier JSP (JavaServer Pages) fait office de client eXtreme Scale pour extraire les informations de sécurité à partir de l'unité d'exécution du serveur. Le fichier JSP appelle l'infrastructure de sécurité WebSphere Application Server pour obtenir les jetons de sécurité qui représentent l'utilisateur d'application d'entreprise.
4. Le client eXtreme Scale, ou un fichier JSP, envoie des jetons de sécurité avec la demande au serveur de conteneur et au service de catalogue qui est hébergé sur les autres machines virtuelles Java. Le serveur de catalogue et le serveur de conteneur utilisent les jetons de sécurité WebSphere Application Server comme données d'identification du client eXtreme Scale.
5. Les serveurs de catalogue et de conteneur envoient les jetons de sécurité à l'infrastructure de sécurité WebSphere Application Server pour les convertir en informations de sécurité utilisateur. Ces informations sont représentées par un objet Subject qui contient les principaux, les données d'identification publiques et données d'identification privées. Cette conversion peut se produire, car les serveurs d'applications qui hébergent le client eXtreme Scale, le serveur de catalogue et le serveur de conteneur partagent les mêmes jetons LTPA (Lightweight Third-Party Authentication (LTPA) WebSphere Application Server.

Intégration de l'authentification

Intégration de la sécurité répartie à WebSphere Application Server :

Pour le modèle réparti, utilisez les classes suivantes :

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenAuthenticator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredential

Pour des exemples d'utilisation de ces classes, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 115.

Côté serveur, utilisez l'authentificateur WSTokenAuthentication pour authentifier l'objet WSTokenCredential.

Intégration de la sécurité locale à WebSphere Application Server:

Pour le modèle ObjectGrid local, utilisez les classes suivantes :

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectSourceImpl
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectValidationImpl

Pour plus d'informations sur ces classes, voir Programmation de la sécurité locale. Vous pouvez configurer la classe WSSubjectSourceImpl en tant que plug-in SubjectSource et la classe WSSubjectValidationImpl en tant que plug-in SubjectValidation.

Support de sécurité de la couche de transport dans WebSphere Application Server

Lorsqu'un client, un serveur de conteneur ou un serveur de catalogue eXtreme Scale s'exécute dans un processus WebSphere Application Server, la sécurité du transport eXtreme Scale est gérée par les paramètres de transport CSIV2 WebSphere Application Server. Pour le client ou le serveur eXtreme Scale, n'utilisez pas les propriétés du client ou du serveur eXtreme Scale pour définir les paramètres SSL. Tous les paramètres SSL doivent être définis dans la configuration WebSphere Application Server.

Cependant, le serveur de catalogues est légèrement différent. Le serveur de catalogue dispose de ses propres chemins de transport propriétaires qui ne peuvent pas être gérés par les paramètres de transport CSIV2 WebSphere Application Server. Par conséquent, il reste nécessaire de configurer les propriétés dans le fichier de propriétés du serveur pour le serveur de catalogue. Pour plus d'informations, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 115.

Configuration de la sécurité client dans un domaine de service de catalogue

La configuration de sécurité du client sur un domaine de service de catalogue permet de définir les propriétés de configuration de l'authentification client par défaut. Ces propriétés sont utilisées lorsque aucun fichier de propriétés du client n'est situé dans la machine virtuelle Java (JVM) qui héberge le client ou que le client ne dispose pas d'un programme pour spécifier les propriétés de sécurité. Si un fichier de propriétés client existe, les propriétés que vous spécifiez dans la console remplacent les valeurs dans le fichier. Vous pouvez remplacer ces propriétés en spécifiant un fichier `splicer.properties` avec la propriété personnalisée com.ibm.websphere.xs.sessionFilterProps ou en raccordant le fichier EAR d'application.

Avant de commencer

- Vous devez connaître l'implémentation CredentialGenerator que vous utilisez pour authentifier les clients avec la grille de données distante. Vous pouvez utiliser l'une des implémentations qui sont fournies par WebSphere eXtreme Scale : UserPasswordCredentialGenerator ou WSTokenCredentialGenerator.

  Vous pouvez également utiliser une implémentation personnalisée de l'interface CredentialGenerator. L'implémentation personnalisée doit se trouver dans le chemin d'accès aux classes du client d'exécution et le serveur. Si vous configurez un scénario de sessions HTTP avec WebSphere Application Server, vous devez placer l'implémentation dans le chemin d'accès aux classes du gestionnaire de déploiement et le chemin d'accès aux classes du serveur d'applications dans lequel le client est en cours d'exécution.
- Vous devez disposer d'un domaine de service de catalogue défini. Pour plus d'informations, voir «Création de domaines de service de catalogue dans WebSphere Application Server», à la page 307.

Pourquoi et quand exécuter cette tâche

Vous devez configurer la sécurité du client sur le domaine de service de catalogue lorsque vous avez activé l'authentification des données d'identification côté serveur, en configurant l'un des scénarios suivants :

- La stratégie de sécurité côté serveur a la propriété **credentialAuthentication** affectée de la valeur `Required`.
- La stratégie de sécurité côté serveur a la propriété **credentialAuthentication** affectée de la valeur `Supported` ET un mécanisme **authorizationMechanism** a été spécifié dans le fichier XML ObjectGrid.

Dans ces scénarios, des données d'identification doivent être transmises à partir du client. Ces informations sont extraites de la méthode getCredential dans une classe qui implémente l'interface CredentialGenerator. Dans un scénario de configuration de session HTTP, l'environnement d'exécution doit connaître l'implémentation CredentialGenerator à utiliser pour générer une identification qui est envoyée à une grille de données distante. Si vous ne spécifiez pas la classe d'implémentation CredentialGenerator à utiliser, la grille de données distante refuserait les demandes du client, car le client ne peut pas être authentifié.

Procédure

Définissez les propriétés de sécurité du client. Dans la console d'administration WebSphere Application Server, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaine de service de catalogue** > *catalog_service_domain_name* > **Propriétés de sécurité du client**. Spécifiez les propriétés de sécurité du client sur la page et enregistrez vos modifications. Voir «Propriétés de sécurité du client», à la page 325 pour la liste des propriétés que vous pouvez définir.

Résultats

Les propriétés de sécurité du client que vous avez configurées dans le domaine de service de catalogue sont utilisées comme valeurs par défaut. Les valeurs que vous indiquez remplacent les propriétés définies dans les fichiers `client.properties`.

Que faire ensuite

Configurez vos applications qu'elles utilisent WebSphere eXtreme Scale pour la gestion des sessions. Pour plus d'informations, voir «Fractionnement automatique des applications pour la gestion de session HTTP dans WebSphere Application Server», à la page 377.

Configuration de la sécurité de la grille de données et de SSL pour .NET

.NET

Vous pouvez configurer .NET et Java pour communiquer sur SSL (Secure Sockets Layer) et utiliser la logique d'authentification UserPassword.

Avant de commencer

Vous devez disposer des fichiers `key.jks` et `trust.jks` correspondant à votre environnement. Pour plus d'informations sur la création des fichiers de clés et des fichiers de clés certifiées, voir «Tutoriel sur la sécurité Java SE - Etape 6», à la page 111.

Procédure

1. Activez et configurez la sécurité dans les serveurs. Si la sécurité n'est pas encore configurée sur les serveurs, vous pouvez procédez comme suit pour la configurer avec l'exemple d'authentificateur externe.
   a. Obtenez les exemples de fichiers de sécurité. Téléchargez les exemples de fichiers dans le fichier `security_extauth.zip` depuis le wikiWebSphere eXtreme Scale.
      - `xsjaas3.config` : définit la configuration JAAS (Java Authentication and Authorization Service).
      - `sampleKS3.jks` : contient le fichier de clés des valeurs utilisateurs et mot de passe JAAS.
      - `security3.xml` : définit l'authentificateur à utiliser pour la sécurité.
   b. Modifiez le fichier `xsjaas3.config` et définissez le chemin d'accès au fichier `sampleKS3.jks`.
   c. Si vous voulez générer votre propre fichier de clés privées au lieu d'utiliser le modèle de fichier `sampleKS3.jks`, utilisez l'utilitaire **keytool** pour générer la clé privée.

      ```
      keytool -genkey -alias myalias -keysize 2048 -keystore key.jks
      -keyalg rsa -dname "CN=www.mydomain.com" -storepass password
      -keypass password -validity 3650
      ```

   d. Modifiez `sampleServer.properties` pour activer la sécurité. Le fichier `sampleServer.properties` se trouve dans le répertoire *racine_install_wxs*\`properties`. Supprimez la mise en commentaire et modifiez les valeurs de propriété suivantes :

      ```
      securityEnabled=true
      secureTokenManagerType=none
      alias=ogsample
      contextProvider=IBMJSSE2
      protocol=SSL
      keyStoreType=JKS
      keyStore=../../../../xio.test/etc/test/security/key.jks
      keyStorePassword=ogpass
      trustStoreType=JKS
      trustStore=../../../../xio.test/etc/test/security/trust.jks
      trustStorePassword=ogpass
      ```

   e. Démarrez les serveurs de catalogue et de conteneur.

      ```
      startXsServer.bat cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809 -objectgridFile gettingstarted\xml\objectgrid.xml
      -deploymentPolicyFile gettingstarted\xml\deployment.xml -serverProps ..\properties\sampleServer.properties
      -clusterSecurityFile security3.xml -jvmArgs -Djava.security.auth.login.config="xsjaas3.config"

      startXsServer.bat c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml
      -deploymentPolicyFile gettingstarted\xml\deployment.xml -serverProps ..\properties\sampleServer.properties
      -clusterSecurityFile security3.xml -jvmArgs -Djava.security.auth.login.config="xsjaas3.config"
      ```

2. Configurez la sécurité du client .NET.
   a. Facultatif : A l'aide de l'utilitaire keytool, extrayez le certificat public du fichier `key.jks` que vous avez configuré pour le serveur.

      ```
      keytool -export -alias myalias -keystore key.jks -file public.cer
      -storepass password
      ```

      Importez cette clé publique vers le magasin de certificats Windows avec l'outil de gestion des certificats, `certmgr.msc`, pour importer la clé vers le dossier des certificats 'Trusted Root Certification Authority' ou 'Trusted People'. (La propriété **keyStore** dans le fichier `client.properties` pointe vers ce fichier)
   b. Modifiez le fichier `Client.Net.properties` pour y placer les valeurs de propriétés suivantes :

```
securityEnabled=true
credentialAuthentication=supported
authenticationRetryCount=3
credentialGeneratorAssembly=IBM.WebSphere.Caching.CredentialGenerator,
Version=8.6.0.0,
Culture=neutral,PublicKeyToken=b439a24ee43b0816
credentialGeneratorProps=manager manager1
transportType=ssl-supported
publicKeyFile=<name>.cer
```

La valeur de la propriété credentialGeneratorProps, `manager manager1` est utilisée comme nom d'utilisateur et mot de passe envoyés au serveur dans l'objet Credential.

La propriété **publicKeyFile** est affectée d'un chemin relatif d'accès à l'environnement d'exécution .NET. Si la propriété **publicKeyFile** n'est pas définie, le fichier `public.cer` est recherché dans le magasin de certificats Windows 'a'. Si la propriété **publicKeyFile** est définie, le fichier spécifié est utilisé comme fichier de certificat public SSL. Si le fichier spécifié est introuvable, le client .NET tente de trouver un fichier `public.cer` correspondant dans le magasin de certificats.

c. Copiez le fichier *net_client_home*\`IBM.WebSphere.Caching.CredentialGenerator.dll` vers le répertoire *net_client_home*\sample\SimpleClient\bin\<*ConfigurationName*>.

d. Générez l'exemple avec le contexte de projet *ConfigurationName*. Exécutez l'exemple sur le serveur.

Activation de l'autorisation de grille de données

WebSphere eXtreme Scale fournit plusieurs points de contact de sécurité permettant d'intégrer les mécanismes personnalisés. Dans le modèle de programmation local, la principale fonction de sécurité est l'autorisation, qui n'est associée à aucune prise en charge de l'authentification. Vous devez vous authentifier indépendamment de l'authentification WebSphere Application Server existante. Toutefois, vous pouvez utiliser les plug-in fournis permettant d'obtenir et de valider des objets Subject.

Pourquoi et quand exécuter cette tâche

Vous pouvez activer la sécurité locale avec le fichier descripteur XML ObjectGrid ou à l'aide d'un programme.

Procédure

Activez la sécurité locale avec le fichier descripteur XML ObjectGrid.
Le fichier `secure-objectgrid-definition.xml` utilisé dans l'exemple d'application d'entreprise ObjectGridSample est présenté dans l'exemple suivant. Pour activer la sécurité, associez l'attribut securityEnabled à la valeur `true`.

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
        authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS">
    ...
</objectGrids>
```

Que faire ensuite

Démarrez le serveur de conteneur et le serveur de catalogue avec la sécurité activée.

Démarrage et arrêt des serveurs sécurisés

La sécurité est activée en spécifiant les configurations de sécurité lorsque vous démarrez et arrêtez des serveurs.

Démarrage des serveurs sécurisés dans un environnement autonome

Pour démarrer les serveurs autonomes sécurisés, vous envoyez les fichiers de configuration appropriés en définissant des paramètres dans la commande **startOgServer** ou**startXsServer**.

8.6+ Pourquoi et quand exécuter cette tâche

Obsolète : 8.6+ Les commandes **startOgServer** et **stopOgServer** démarrent les serveurs qui utilisent le mécanisme de transport ORB (Object Request Broker). ORB est obsolète, mais vous pouvez continuer à utiliser ces scripts si vous utilisiez ORB dans une édition précédente. Le mécanisme de transport IBM eXtremeIO (XIO) remplace ORB. Utilisez les scripts **startXsServer** et **stopXsServer** pour démarrer et arrêter les serveurs qui utilisent le transport XIO.

Procédure

- Démarrez les serveurs de conteneur.

  Le démarrage d'un serveur de conteneur sécurisé requiert le fichier de configuration de sécurité suivant :

  – **Fichier de propriétés du serveur :** ce fichier permet de configurer les propriétés de sécurité spécifiques au serveur. Pour plus d'informations, voir Fichier de propriétés du serveur.

  Indiquez l'emplacement de ce fichier de configuration en fournissant l'argument suivant dans le script **startOgServer** ou **startXsServer** :

  -serverProps
  Spécifie l'emplacement du fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques du serveur. Le nom de fichier spécifié pour cette propriété correspond à un format de chemin de fichier classique, tel que `../security/server.properties`.

  Entrez les lignes suivantes lorsque vous exécutez la commande **startOgServer** ou **startXsServer** : UNIX Linux

```
startOgServer.sh <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
-Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=$OBJECTGRID_HOME
```

UNIX Linux 8.6+

```
startXsServer.sh <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
-Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=$OBJECTGRID_HOME
```

Windows

```
startOgServer.bat <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
-Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=%OBJECTGRID_HOME%
```

Windows 8.6+

```
startXsServer.bat <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config
 -Djava.security.manager -Djava.security.policy="auth.policy" -Dobjectgrid.home=%OBJECTGRID_HOME%
```

- Démarrez les serveurs de catalogue.

Pour démarrer un service de catalogue sécurisé, vous devez disposer des fichiers de configuration suivants :

– **Fichier descripteur XML de la sécurité :** décrit les propriétés de sécurité communes à tous les serveurs, y compris les serveurs de catalogue et de conteneur. Un exemple de propriété est la configuration de l'authentificateur qui représente le registre utilisateur et le mécanisme d'authentification.
– **Fichiers des propriétés du serveur :** configure les propriétés de sécurité spécifiques du serveur.

Indiquez l'emplacement de ces fichiers de configuration en fournissant l'argument suivant au script **startOgServer** or **startXsServer** :

-clusterSecurityFile et -clusterSecurityUrl
Ces arguments indiquent l'emplacement du fichier XML du descripteur de sécurité. Utilisez le paramètre **-clusterSecurityFile** pour spécifier un fichier local ou le paramètre **-clusterSecurityUrl** pour indiquer l'adresse URL du fichier `objectGridSecurity.xml`.

-serverProps
Spécifie l'emplacement du fichier de propriétés du serveur qui contient les propriétés de sécurité spécifiques du serveur. Le nom de fichier spécifié pour cette propriété correspond à un chemin classique, tel que `c:/tmp/og/catalogserver.props`.

Démarrage des serveurs sécurisés dans WebSphere Application Server

Pour démarrer des serveurs sécurisés dans WebSphere Application Server, vous devez spécifier les fichiers de configuration de la sécurité dans les arguments génériques Java virtual machine (JVM).

Procédure

- Associez les serveurs de catalogue WebSphere eXtreme Scale aux serveurs d'applications WebSphere en utilisant la console d'administration. Dans la console d'administration, cliquez sur **Administration du système** > **WebSphere eXtreme Scale** > **Domaines de service de catalogue**.
- Associez les serveurs de conteneur WebSphere eXtreme Scale à des serveurs d'applications WebSphere spécifiques en déployant un fichier EAR (enterprise archive) qui contient les descripteurs XML nécessaires à la grille. Pour plus d'informations sur cette procédure, voir «Tutoriel : Intégration de la sécurité WebSphere eXtreme Scale à WebSphere Application Server», à la page 115.
- Définissez les arguments JVM (Java virtual machine) qui pointent vers les fichiers de configuration pour sécuriser les serveurs de catalogue et de conteneur. Pour plus d'informations sur cette procédure, voir Authentification des demandes des clients dans WebSphere Application Server et Autorisation d'accès à la grille de données dans WebSphere Application Server. En outre, définissez `securityEnabled="true"` dans le fichier `objectgrid xml` de chaque grille de données. Après avoir défini les arguments JVM et activé la sécurité dans les grilles de données, vous pouvez démarrer les serveurs ou les clusters qui font office de serveurs de catalogue ou de conteneur eXtreme Scale.
- Démarrez les serveurs de catalogue et de conteneurs avec la console d'administration WebSphere Application Server ou utilisez la ligne de commande WebSphere Application Server.

Que faire ensuite

«Arrêt des serveurs sécurisés»

Arrêt des serveurs sécurisés

L'arrêt des serveurs de catalogue sécurisés ou des serveurs de conteneur requiert un fichier de configuration de sécurité.

Procédure

- Arrêtez un serveur de catalogue sécurisé ou un serveur de conteneur dans les environnements autonomes. Dans les environnements autonomes, arrêtez les serveurs de catalogue et de conteneur WebSphere eXtreme Scale en utilisant la fonction de désassemblage de la commande **xscmd** ou **stopXsServer** ou **stopOgServer**.

  Limitez l'accès à ces opérations aux administrateurs autorisés uniquement, comme indiqué dans la section Autorisation d'accès pour les opérations d'administration dans lesenvironnements autonomes. Lorsque l'authentification ou SSL sont utilisés, les commandes **stopXsServer** et **stopOgServer** nécessitent d'envoyer un fichier de propriétés de client sous la forme d'un paramètre. Le contenu de ce fichier est décrit dans Authentification des demandes des clients dans les environnements autonomes et Protection des données qui transitent entre eXtreme Scale et les environnements autonomes avec le chiffrement SSL.
- Utilisez la console d'administration WebSphere Application Server pour arrêter eXtreme Scale le serveur qui s'exécute avec WebSphere Application Server. La sécurité administrative WebSphere Application Server doit être configurée pour limiter l'accès au démarrage et à l'arrêt des serveurs aux administrateurs autorisés, comme indiqué dans Autorisation d'accès pour les opérations administratives dans WebSphere Application Server.

Configuration de WebSphere eXtreme Scale pour utiliser FIPS 140-2

Federal Information Processing Standard (FIPS) 140-2 spécifie les niveaux requis de chiffrement du protocole Transport Layer Security/Secure Sockets Layer (TLS/SSL). Ce standard garantit une protection élevée des données envoyées via la connexion.

Avant de commencer

- Vous devez utiliser un environnement IBM Runtime Environment. Pour plus d'informations, voir «Java SE : points à prendre en considération», à la page 68.
- Configurez la sécurité de la couche de transport et la couche des sockets sécurisés dans les deux sens. Le fichier de clés certifiées du serveur de catalogue doit contenir les certificats autosignés des serveurs de conteneur. Les serveurs de conteneur doivent contenir les certificats autosignés du serveur de catalogue. Pour plus d'informations, voir «Protocole TLS et couche de connexion sécurisée», à la page 634.

Pourquoi et quand exécuter cette tâche

Procédez comme suit pour configurer les serveurs de catalogue et de conteneur dans l'installation autonome WebSphere eXtreme Scale pour utiliser FIPS.

Si vous utilisez WebSphere eXtreme Scale intégré dans WebSphere Application Server, les serveurs de catalogue et les serveurs de conteneur héritent des propriétés de sécurité du serveur d'applications. Pour plus d'informations sur la

configuration de FIPS avec WebSphere Application Server, voir Configuration des fichiers Federal Information Processing Standard Java Secure Socket Extension. Lorsqu'un serveur de catalogue s'exécute dans WebSphere Application Server, une partie des communications est contrôlée par le fichier `server.properties`. Mettez à jour le fichier `server.properties` pour qu'il contienne les mêmes propriétés que celles des serveurs de catalogue autonomes.

Procédure

1. Modifiez le fichier `java.security`. L'emplacement de `java.security` dépend de la configuration de la machine virtuelle Java (JVM) :
   - Si vous utilisez la machine JVM par défaut avec le produit, le fichier se trouve dans le répertoire *racine_install_wxs*`/java/jre/lib/security`.
   - Si vous utilisez une machine virtuelle Java différente, éditez le fichier dans le répertoire *rép_base_java*`/jre/lib/security`.

   Le fichier doit contenir le texte suivant :

   ```
   security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
   security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
   security.provider.3=com.ibm.crypto.provider.IBMJCE
   security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
   security.provider.5=com.ibm.security.cert.IBMCertPath
   security.provider.6=com.ibm.security.sasl.IBMSASL
   security.provider.7=com.ibm.xml.crypto.IBMXMLCryptoProvider
   security.provider.8=com.ibm.xml.enc.IBMXMLEncProvider
   security.provider.9=org.apache.harmony.security.provider.PolicyProvider
   security.provider.10=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
   ```

2. Editez les fichiers de propriétés du serveur de catalogue et des serveurs de conteneur.

   Ces fichiers doivent contenir les propriétés et valeurs suivantes :

   ```
   contextProvider=IBMJSSE2
   transportType=SSL-Required
   ```

   Pour plus d'informations sur les propriétés de serveur, voir Fichier de propriétés du serveur.

3. Configurez des paires de clés qui utilisent l'algorithme de génération de clé RSA dans le fichier de clés du serveur de catalogue et des serveurs de conteneur. La longueur de clé minimale est de 1 024 bits.
4. Redémarrez les serveurs de conteneur et de catalogue.

   Lorsque vous démarrez les serveurs de catalogue, vous devez définir les arguments JVM (Java virtual machine). Les arguments que vous utilisez dépendent de la version de Java SE que vous utilisez.
   - Pour Java 5 et Java 6 jusqu'à SR 9, spécifiez l'argument **`-Dcom.ibm.jsse2.JSSEFIPS=true`** lorsque vous démarrez le serveur.
   - Pour Java 6 SR 10 et les versions suivantes ou Java 7, spécifiez l'argument **`-Dcom.ibm.jsse2.usefipsprovider=true`** lorsque vous démarrez le serveur.

   Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 650.

Configuration des profils de sécurité pour l'utilitaire `xscmd`

En créant un profil de sécurité, vous pouvez utiliser les paramètres de sécurité enregistrés pour utiliser l'utilitaire **`xscmd`** avec des environnements sécurisés.

Avant de commencer

Pour plus d'informations sur la configuration de l'utilitaire **xscmd**, voir «Administration avec l'utilitaire **xscmd**», à la page 515.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser le paramètre **-ssp** *profile_name* ou **--saveSecProfile** *profile_name* avec le reste de la commande **xscmd** pour enregistrer un profil de sécurité. Le profil peut contenir des paramètres pour les noms d'utilisateur et les mots des générateurs de données d'identification, des fichiers de clés, des fichiers de clés certifiées et des types de transport.

Le groupe de commandes **ProfileManagement** dans l'utilitaire **xscmd** contient des commandes de gestion de vos profils de sécurité.

Procédure

- Enregistrez un profil de sécurité.

  Pour enregistrer un profil de sécurité, utilisez le paramètre **-ssp** *profile_name* ou **--saveSecProfile** *profile_name* avec le reste de la commande. L'ajout de ce paramètre à la commande enregistre les paramètres suivants :

  ```
  -al,--alias <alias>
  -arc,--authRetryCount <integer>
  -ca,--credAuth <support>
  -cgc,--credGenClass <className>
  -cgp,--credGenProps <property>
  -cxpv,--contextProvider <provider>
  -ks,--keyStore <filePath>
  -ksp,--keyStorePassword <password>
  -kst,--keyStoreType <type>
  -prot,--protocol <protocol>
  -pwd,--password <password>
  -ts,--trustStore <filePath>
  -tsp,--trustStorePassword <password>
  -tst,--trustStoreType <type>
  -tt,--transportType <type>
  -user,--username <username>
  ```

  Les profils de sécurité sont enregistrés dans le répertoire *user_home*`\.scmd\profiles\security\<`*nom_profil*`>.properties`.

  Important : N'incluez pas l'extension de nom de fichier `.properties` dans le paramètre *nom_profil*. Cette extension est automatiquement ajoutée au nom de fichier.
- Utilisez un profil de sécurité enregistré.

  Pour utiliser un profil de sécurité enregistré, ajoutez le paramètre **-sp** *profile_name* ou **--securityProfile** *profile_name* à la commande que vous exécutez. Exemple de commande : `xscmd -c listHosts -cep myhost.mycompany.com -sp myprofile`
- Listez les commandes dans le groupe de commandes **ProfileManagement**.

  Exécutez la commande **xscmd -lc ProfileManagement**.
- Listez les profils de sécurité existants.

  Exécutez la commande **xscmd -c listProfiles -v**.
- Affichez les paramètres enregistrés dans un profil de sécurité.

  Exécutez la commande **xscmd -c showProfile -pn** *profile_name*.
- Supprimez un profil de sécurité existant.

Exécutez la commande **xscmd -c RemoveProfile -pn** *profile_name*.

Sécurisation des connexions client J2C

Utilisez l'architecture Java 2 Connector (J2C) pour sécuriser les connexions entre les clients WebSphere eXtreme Scale et vos applications.

Pourquoi et quand exécuter cette tâche

Les applications font référence à la fabrique de connexions, qui établit la connexion à la grille de données distante. Chaque fabrique de connexions héberge une connexion client eXtreme Scale unique qui est réutilisée pour tous les composants d'application.

Important : Etant donné que la connexion client eXtreme Scale peut inclure un cache local, il est important que les applications ne partagent pas de connexion. Il doit exister une fabrique de connexions pour chaque instance d'application pour éviter tout problème de partage d'objets entre les applications.

Vous pouvez définir le générateur de données d'identification à l'aide de l'API ou dans le fichier de propriétés client. Dans ce dernier, vous utilisez les propriétés securityEnabled et credentialGenerator. L'exemple de code suivant est présenté sur plusieurs lignes en raison des contraintes liées à la publication :

```
securityEnabled=true
credentialGeneratorClass=com.ibm.websphere.objectgrid.security.plugins.builtins.
     UserPasswordCredentialGenerator
credentialGeneratorProps=operator XXXXXX
```

Le générateur de données d'identification et les données d'identification du fichier de propriétés client sont utilisés pour l'opération de connexion de eXtreme Scale et pour les données d'identification J2C par défaut. Les données d'identification qui sont spécifiées avec l'API sont donc utilisées lors de la connexion J2C pour la connexion J2C. Toutefois, si aucune donnée d'identification n'est spécifiée au moment de la connexion J2C, c'est le générateur de données d'identification du fichier de propriétés client qui est utilisé.

Procédure

1. Configurez un accès sécurisé dans lequel la connexion J2C représente le client eXtreme Scale. Utilisez la propriété de fabrique de connexions ClientPropertiesResource ou ClientPropertiesURL pour configurer l'authentification client.

   Si vous utilisez WebSphere eXtreme Scale avec WebSphere Application Server, spécifiez les propriétés du client dans la configuration du domaine de services de catalogue. Lorsque la fabrique de connexions fait référence au domaine, elle utilise automatiquement cette configuration.
2. Configurez les propriétés de sécurité client afin qu'elles utilisent la fabrique de connexions qui référence l'objet générateur de données d'identification approprié pour eXtreme Scale. Ces propriétés sont également compatibles avec la sécurité serveur eXtreme Scale. Par exemple, utilisez le générateur de données d'identification WSTokenCredentialGenerator pour les données d'identification WebSphere lorsque eXtreme Scale est installé avec WebSphere Application Server. Vous pouvez aussi utiliser le générateur de données d'identification UserPasswordCredentialGenerator lorsque vous exécutez eXtreme Scale dans un environnement autonome. Dans l'exemple suivant, les données d'identification sont transmises par voie de programme à l'aide de l'appel d'API au lieu d'utiliser la configuration dans les propriétés client :

```
XSConnectionSpec spec = new XSConnectionSpec();
spec.setCredentialGenerator(new UserPasswordCredentialGenerator
("operator", "xxxxxx"));
Connection conn = connectionFactory.getConnection(spec);
```

3. (Facultatif) Désactivez le cache local, si nécessaire.

   Toutes les connexions J2C provenant d'une fabrique de connexions unique partagent un cache local unique. Les autorisations d'entrée de grille et de mappe sont validées sur le serveur, mais pas dans le cache local. Lorsqu'une application utilise plusieurs données d'identification pour créer des données J2C et que la configuration utilise des autorisations spécifiques pour les entrées de grille et les mappes pour ces données d'identification, vous devez désactiver le cache local. Désactivez le cache local en utilisant la propriété de fabrique de connexions ObjectGridResource ou ObjectGridURL. Pour plus d'informations sur la désactivation du cache local, voir «Configuration du cache local», à la page 362.

4. (Facultatif) Définissez les paramètres de règle de sécurité, si nécessaire.

   Si l'application J2EE contient le configuration de fichier RAR (resource adapter archive) de l'adaptateur de ressources eXtreme Scale imbriqué, il peut s'avérer nécessaire que vous définissiez des paramètres de règle de sécurité supplémentaires dans le fichier de règles de sécurité de l'application. Par exemple, ces règles sont requises :

   ```
   permission com.ibm.websphere.security.WebSphereRuntimePermission
   "accessRuntimeClasses";
   permission java.lang.RuntimePermission "accessDeclaredMembers";
   permission javax.management.MBeanTrustPermission "register";
   permission java.lang.RuntimePermission "getClassLoader";
   ```

   De plus, tout fichier de propriétés ou de ressources utilisé par les fabriques de connexions requiert des autorisations d'accès aux fichiers ou autres, tels que `permission java.io.FilePermission "filePath";`. Pour WebSphere Application Server, le fichier de règles est `META-INF/was.policy` et il est situé dans le fichier J2EE EAR.

Résultats

Les propriétés de sécurité du client que vous avez configurées dans le domaine de service de catalogue sont utilisées comme valeurs par défaut. Les valeurs que vous indiquez remplacent les propriétés définies dans les fichiers `client.properties`.

Que faire ensuite

Utilisez les API d'accès aux données eXtreme Scale pour développer les composants client qui doivent utiliser des transactions.

Chapitre 11. Résolution des incidents

Outre les journaux et de trace, les messages et les notes sur l'édition mentionnés dans la présente section, vous pouvez utiliser des outils de surveillance pour identifier et résoudre les incidents tels que l'emplacement des données dans l'environnement, la disponibilité des serveurs dans la grille de données, etc. Si vous utilisez un environnement WebSphere Application Server, vous pouvez utiliser PMI (Performance Monitoring Infrastructure). Si vous utilisez un environnement autonome, vous pouvez utiliser l'outil de surveillance d'un fournisseur, tel que CA Wily Introscope ou Hyperic HQ. Vous pouvez également utiliser et personnaliser l'utilitaire **xscmd** pour afficher des informations textuelles sur votre environnement.

Identification et résolution des incidents pour WebSphere eXtreme Scale

Pour isoler et résoudre les problèmes rencontrés avec vos produits IBM, vous pouvez utiliser les informations relatives à l'identification et la résolution des incidents. Ces informations contiennent des consignes concernant l'utilisation des ressources de détermination des problèmes fournies avec vos produits IBM, y compris WebSphere eXtreme Scale .

Techniques d'identification et de résolution d'incidents

L'*identification et la résolution des incidents* est une approche systématique de la phase de résolution d'un problème. L'objectif est de déterminer les raisons pour lesquelles une opération ne fonctionne pas comme prévu et d'expliquer la procédure à suivre pour résoudre le problème. Certaines techniques courantes peuvent être utiles pour la tâche d'identification d'incident.

La première étape du processus consiste à décrire complètement le problème. La description de l'incident vous permet, et permet également au technicien de maintenance IBM, de savoir où commencer à chercher la cause de l'incident. A cette étape, vous devez vous poser les questions de base suivantes :

- Quels sont les symptômes du problème ?
- Où survient le problème ?
- Quand survient le problème ?
- Dans quelles conditions le problème survient-il ?
- Le problème peut-il être reproduit ?

Les réponses à ces questions permettent généralement de décrire avec précision le problème, ce qui vous permet de le résoudre.

Quels sont les symptômes du problème ?

Lorsque vous commencez à décrire un problème, la question la plus évidente est «Quel est le problème ?» Cette question peut sembler simple ; toutefois, vous pouvez la scinder en plusieurs questions plus concentrées qui permettent d'avoir une vue plus descriptive du problème. Ces questions sont notamment les suivantes :

- Qui ou qu'est-ce qui a généré un rapport de problème ?
- Quels sont les codes et les messages d'erreur ?

- Comment le système a échoué ? Par exemple, boucle, arrêt, plantage, dégradation des performances, résultat incorrect.

Où survient le problème ?

Il n'est pas toujours aisé de déterminer où se trouve le problème, mais il s'agit pourtant de l'une des étapes les plus importantes. De nombreuses couches technologiques peuvent exister entre le composant qui signale l'incident et le composant défaillant. Les réseaux, la grille de données et les serveurs ne sont que quelques exemples de composants à prendre en compte lorsque vous cherchez à en savoir plus sur les problèmes rencontrés.

Les questions suivantes vous aident à vous concentrer sur l'endroit où survient le problème pour isoler la couche de problème :

- Le problème est-il spécifique à une plateforme ou à un système d'exploitation, ou concerne-t-il plusieurs plateformes ou systèmes d'exploitation ?
- L'environnement et la configuration actuels sont-ils pris en charge ?
- Tous les utilisateurs rencontrent-ils ce problème ?
- (Pour les installations multisite.) Tous les sites rencontrent-ils ce problème ?

Ce n'est pas parce qu'une couche signale le problème que celui-ci provient obligatoirement de cette couche. Pour identifier l'endroit d'où provient un problème, vous devez comprendre l'environnement dans lequel ce problème se produit. Prenez quelques instants pour décrire complètement l'environnement du problème, notamment le système d'exploitation et sa version, tous les logiciels correspondants et leur version, ainsi que les informations sur le matériel. Vérifiez que la configuration de votre environnement est prise en charge ; de nombreux problèmes peuvent être provoqués par l'utilisation de logiciels incompatibles qui ne sont pas destinés à être exécutés ensemble ou dont l'exécution simultanée n'a pas été testée.

Quand survient le problème ?

Dressez la liste chronologique des événements qui ont conduit à l'apparition de l'incident, en particulier si l'incident ne s'est produit qu'une seule fois. Vous pouvez très aisément dresser une liste chronologique en procédant à l'envers : partez du moment où une erreur a été signalée (aussi précisément que possible, même à la milliseconde près), et remontez en arrière à l'aide des journaux et des informations disponibles. Il vous suffit généralement de remonter jusqu'au premier événement suspicieux signalé dans un journal de diagnostic.

Pour dresser une liste chronologique détaillée des événements, répondez aux questions suivantes :

- Le problème se produit-il uniquement à certains moments du jour ou de la nuit ?
- Selon quelle fréquence le problème se produit-il ?
- Quelle séquence d'événements a provoqué la survenue du problème ?
- Le problème s'est-il produit après un changement apporté à l'environnement, tel qu'une mise à niveau ou l'installation d'un logiciel ou d'un matériel ?

En répondant à ces questions, vous définissez un cadre de référence dans lequel mener vos recherches.

Dans quelles conditions le problème survient-il ?

Il est très important de savoir quelles applications et quels systèmes étaient en cours d'exécution lorsque l'incident s'est produit. Les questions suivantes concernant l'environnement vous aideront à identifier la cause de l'incident :

- Le problème se produit-il toujours lorsque vous effectuez la même tâche ?
- Est-ce qu'une séquence d'événements doit se produire pour provoquer le problème ?
- Est-ce que l'exécution d'autres applications échoue également ?

En répondant à ces questions, vous pouvez décrire l'environnement dans lequel l'incident se produit, et identifier les éventuelles dépendances. Notez cependant que si plusieurs incidents se produisent de manière quasi-simultanée, cela ne signifie pas nécessairement que ces incidents sont liés.

Le problème peut-il être reproduit ?

Du point de vue de l'identification et de la résolution, le problème idéal est celui qui peut être reproduit. En général, lorsqu'un problème peut être reproduit, vous disposez d'un plus grand nombre d'outils ou de procédures pour en savoir plus sur ces problèmes. Par conséquent, les problèmes que vous pouvez reproduire sont souvent plus faciles à déboguer et résoudre.

Toutefois, ces problèmes présentent un inconvénient : si le problème en question a un impact commercial considérable, vous ne voulez pas qu'il se reproduise. Si possible, recréez l'incident dans un environnement de test ou de développement, garantissant davantage de souplesse et de contrôle lors de l'analyse.

- Le problème peut-il être recréé sur un système test ?
- Est-ce que plusieurs utilisateurs ou applications rencontrent le même type de problème ?
- Le problème peut-il être recréé en exécutant une seule commande, un jeu de commandes ou une application particulière ?

Recherche des bases de connaissances

Vous pouvez souvent trouver des solutions à vos problèmes en effectuant des recherches dans les bases de connaissances IBM. Vous pouvez optimiser vos résultats en utilisant les ressources disponibles, les outils de support et les méthodes de recherche.

Pourquoi et quand exécuter cette tâche

Vous pouvez trouver des informations utiles en effectuant des recherches dans le centre de documentation WebSphere eXtreme Scale . Cependant, il est parfois nécessaire de pousser les recherches plus loin pour trouver des réponses à vos questions ou résoudre certains problèmes.

Procédure

Pour rechercher les informations dont vous avez besoin dans les bases de connaissances, utilisez une ou plusieurs des approches suivantes :

- Recherchez un contenu à l'aide d'IBM Support Assistant (ISA).

ISA est un plan de travail de serviçabilité gratuit qui vous permet de répondre à certaines questions et de résoudre les problèmes relatifs aux logiciels IBM. Vous trouverez les instructions de téléchargement et d'installation d'ISA sur le site Web d'ISA.

- Recherchez le contenu dont vous avez besoin à l'aide du portail de support IBM.

  Le portail de support IBM est une vue centralisée et unifiée de tous les outils de support technique et de toutes les informations relatives à l'ensemble des systèmes, logiciels et services IBM. Le portail de support IBM permet d'accéder au portefeuille d'assistance électronique d'IBM à partir d'un emplacement unique. >Vous pouvez en personnaliser les pages pour cibler les informations et les ressources dont vous avez besoin pour empêcher un problème ou en résoudre plus rapidement un. Familiarisez-vous avec le portail de support IBM en visionnant les vidéos de démonstration (https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos) sur cet outil. Ces vidéos vous présentent le portail de support IBM, explorent la fonction d'identification et de résolution des problèmes et d'autres ressources et expliquent comment personnaliser la page en déplaçant, ajoutant et supprimant des portlets.
- Recherchez du contenu relatif à WebSphere eXtreme Scale en utilisant l'une des ressources techniques supplémentaires suivantes :
  - WebSphere eXtreme Scale - Notes sur l'édition
  - Site Web d'assistance WebSphere eXtreme Scale
  - Forum WebSphere eXtreme Scale
- Recherchez un contenu à l'aide de la fonction de recherche générique IBM. Vous pouvez utiliser la fonction de recherche générique IBM en saisissant votre chaîne de recherche dans la zone de recherche, dans la partie supérieure de toute page ibm.com.
- Recherchez un contenu à l'aide de tout moteur de recherche externe, tel que Google, Yahoo ou Bing. Si vous utilisez un moteur de recherche externe, vos résultats risquent d'inclure davantage d'informations en dehors du domaine ibm.com. Cependant, en consultant certains forums et blogues en dehors du domaine ibm.com, vous pouvez parfois y trouver des informations utiles pour résoudre des problèmes sur les produits IBM.

  Conseil : Incluez «IBM» et le nom du produit dans votre recherche si vous recherchez des informations sur un produit IBM.

Obtention de correctifs

Il existe peut-être un correctif produit permettant de résoudre votre problème.

Procédure

Pour rechercher et installer des correctifs, procédez comme suit :

1. Procurez-vous les outils requis pour obtenir le correctif. Utilisez IBM Update Installer pour installer et appliquer plusieurs types de modules de maintenance pour WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client. Ce programme d'installation de mises à jour étant soumis à des opérations de maintenance régulières, veuillez utiliser sa dernière version.
2. Déterminez le correctif dont vous avez besoin. Pour sélectionner le correctif le plus récent, voir Recommended fixes for WebSphere eXtreme Scale . Lors de la sélection d'un correctif, le document de téléchargement pour ce correctif s'ouvre.
3. Téléchargez le correctif. Dans le document de téléchargement, cliquez sur le lien relatif au correctif le plus récent dans la section «Download package».

4. Appliquez le correctif. Suivez les instructions de la section «Installation Instructions» du document de téléchargement.
5. Abonnez-vous pour recevoir des notifications hebdomadaires par courrier électronique sur les correctifs et d'autres informations du service de support IBM.

Obtention de correctifs à partir de Fix Central

Vous pouvez utiliser Fix Central pour rechercher les correctifs recommandés par le service de support IBM pour divers produits, y compris WebSphere eXtreme Scale . Avec Fix Central, vous pouvez rechercher, sélectionner, commander et télécharger des correctifs pour votre système et avez le choix entre plusieurs options de distribution. Un correctif de produit WebSphere eXtreme Scale peut être disponible pour résoudre votre incident.

Procédure

Pour rechercher et installer des correctifs, procédez comme suit :

1. Procurez-vous les outils requis pour obtenir le correctif. Si votre programme d'installation de mises à jour n'est pas installé, procurez-le vous. Vous pouvez télécharger le programme d'installation à partir de Fix Central. Ce site fournit des instructions de téléchargement, d'installation et de configuration pour le programme d'installation de mises à jour.
2. Sélectionnez le produit, puis cochez une ou plusieurs cases correspondant à l'incident que vous souhaitez résoudre.
3. Identifiez et sélectionnez le correctif requis.
4. Téléchargez le correctif.
   a. Ouvrez le document de téléchargement, puis suivez le lien indiqué dans la section «Download Package».
   b. Lorsque vous téléchargez le fichier, vérifiez que le nom du fichier de maintenance n'est pas modifié. Cette modification peut être intentionnelle ou découler de certains navigateurs Web ou utilitaires de téléchargement.
5. Appliquez le correctif.
   a. Suivez les instructions de la section «Installation Instructions» du document de téléchargement.
   b. Pour plus d'informations, reportez-vous à la rubrique «Installing fixes with the Update Installer» dans la documentation du produit.
6. Facultatif : Abonnez-vous pour recevoir des notifications hebdomadaires par courrier électronique sur les correctifs et d'autres mises à jour du service de support IBM.

Comment prendre contact avec le service de support IBM

Le service de support IBM assure l'assistance relative aux problèmes rencontrés avec les produits, répond aux questions (FAQ) et aide les utilisateurs à résoudre les incidents liés au produit.

Avant de commencer

Une fois que vous avez essayé de rechercher votre réponse ou votre solution à l'aide d'autres options d'auto-assistance, telles que les notes sur l'édition, vous pouvez contacter le service de support IBM. Pour contacter le service de support IBM, votre société ou organisation doit avoir souscrit en contrat de maintenance IBM en cours de validité, et vous devez être autorisé à soumettre des problèmes à IBM. Pour obtenir des informations sur les types de support disponibles,

reportez-vous à la rubrique Support portfolio du «*Software Support Handbook*».

Procédure

Pour signaler un problème au service de support IBM, procédez comme suit :

1. Définissez la nature du problème, collectez des informations générales, puis identifiez le niveau de gravité du problème. Pour plus d'informations, reportez-vous à la rubrique Getting IBM support du *Software Support Handbook*.
2. Collectez des informations de diagnostic.
3. Soumettez le problème au service de support IBM de l'une des manières suivantes :
   - Avec IBM Support Assistant (ISA). Pour plus d'informations, voir «IBM Support Assistant for WebSphere eXtreme Scale», à la page 698 ou «Collecte des données avec IBM Support Assistant Data Collector», à la page 697.
   - En ligne via le Portail de support IBM : Vous pouvez ouvrir, mettre à jour et afficher toutes vos demandes de service à partir du portlet Demande de service de la page Demande de service.
   - Par téléphone : Pour connaître le numéro à appeler dans votre pays ou région, reportez-vous à la page Web Directory of worldwide contacts.

Résultats

Si le problème signalé concerne un incident logiciel, ou de la documentation manquante ou incorrecte, le service de support IBM crée un rapport officiel d'analyse de programme (APAR). Ce rapport expose le problème en détail. Dans la mesure du possible, le service de support IBM propose une solution palliative que vous pouvez mettre en oeuvre en attendant la finalisation de l'APAR et la mise à disposition d'un correctif . IBM publie chaque jour sur le site Web du service de support IBM, les APAR traités afin que les utilisateurs susceptibles de rencontrer le même problème puissent prendre connaissance de la solution à appliquer.

Echange d'informations avec IBM

Pour diagnostiquer ou identifier un incident, vous avez peut-être besoin de fournir au service de support IBM des données et des informations issues de votre système. Dans d'autres cas, le service de support IBM peut vous fournir des outils ou utilitaires à utiliser pour l'identification de l'incident.

Envoi d'informations au service de support IBM

Pour réduire le temps requis pour la résolution de votre incident, vous pouvez envoyer des informations de trace et de diagnostic au service de support IBM.

Procédure

Pour soumettre des informations de diagnostic au service de support IBM, procédez comme suit :

1. Ouvrez un enregistrement PMR.
2. Collectez les données de diagnostic dont vous avez besoin. Les données de diagnostic vous aident à réduire le temps de résolution de votre enregistrement PMR. Vous pouvez collecter les données de diagnostic manuellement ou automatiquement :
   - Collecte manuelle des données.
   - Collecte automatique des données.
3. Compressez les fichiers à l'aide du format de fichier `.zip` ou `.tar`.

4. Transférez les fichiers à IBM. Vous pouvez utiliser l'une des méthodes suivantes pour transférer les fichiers à IBM :
   - IBM Support Assistant
   - L'outil de demande de service
   - Méthodes de téléchargement de données standard : FTP, HTTP
   - Méthodes de téléchargement de données sécurisées : FTPS, SFTP, HTTPS
   - Courrier électronique

   Si vous utilisez un produit z/OS et que vous faites appel à ServiceLink / IBMLink pour soumettre des enregistrements PMR, vous pouvez envoyer les données de diagnostic au service de support IBM dans un courrier électronique ou via FTP.

   Toutes ces méthodes d'échange de données sont expliquées sur le site Web du service de support IBM.

Réception d'informations du service de support IBM

Il arrive parfois qu'un technicien de maintenance IBM vous demande de télécharger des outils de diagnostic ou d'autres fichiers. Vous pouvez utiliser FTP pour télécharger ces fichiers.

Avant de commencer

Assurez-vous que votre technicien de maintenance IBM vous a fourni le serveur préféré à utiliser pour le téléchargement des fichiers, ainsi que les noms de répertoire et de fichier exacts auxquels accéder.

Procédure

Pour télécharger des fichiers à partir du service de support IBM, procédez comme suit :

1. Utilisez FTP pour vous connecter au site indiqué par votre technicien de maintenance IBM, puis connectez-vous en tant qu'utilisateur `anonymous`. Utilisez votre adresse électronique comme mot de passe :
2. Accédez au répertoire approprié :
   a. Accédez au répertoire `/fromibm`.
      ```
      cd fromibm
      ```
   b. Accédez au répertoire fourni par votre technicien de maintenance IBM.
      ```
      cd nomdurépertoire
      ```
3. Activez le mode binaire pour votre session.
   ```
   binary
   ```
4. Utilisez la commande **get** pour télécharger le fichier indiqué par votre technicien de maintenance IBM.
   ```
   get nomfichier.extension
   ```
5. Mettez fin à votre session FTP.
   ```
   quit
   ```

Abonnement aux mises à jour de support

Pour rester informé des informations importantes sur les produits IBM que vous utilisez, vous pouvez vous abonner aux mises à jour.

Pourquoi et quand exécuter cette tâche

En vous abonnant pour recevoir les mises à jour sur le produit, vous pouvez recevoir les mises à jour et informations techniques importantes sur des ressources et outils de support IBM spécifiques. Vous pouvez vous abonner aux mises à jour à l'aide de l'une des deux approches suivantes :

Abonnements aux média sociaux

Le flux RSS suivant est disponible pour le produit :

- Flux RSS pour le forum WebSphere eXtreme Scale

Pour des informations générales sur RSS, dont la procédure d'initiation et la liste des pages Web IBM compatibles RSS, visitez le site Flux RSS du service de support logicielIBM.

Mes notifications

Avec Mes notifications, vous pouvez vous abonner aux mises à jour de support de tout produit IBM. Mes notifications remplace Mon Support, qui est un outil similaire que vous avez peut-être utilisé dans le passé. Avec Mes Notifications, vous pouvez indiquer que vous souhaitez recevoir des annonces quotidiennes ou hebdomadaires par courrier électronique. Vous pouvez spécifier le type d'informations à recevoir (par exemple, des publications, des astuces et des conseils, des notifications flash sur les produits (ou alertes), des téléchargements et des pilotes). Mes notifications vous permet de personnaliser et classer les produits sur lesquels vous souhaitez être informé et les méthodes de distribution qui répondent le mieux à vos besoins.

Procédure

Pour vous abonner aux mises à jour de support :

1. Abonnez-vous au flux RSS pour le forum WebSphere eXtreme Scale .
   a. Dans la page d'abonnement, cliquez sur l'icône du flux RSS.
   b. Sélection l'option que vous souhaitez utiliser pour vous abonner au flux.
   c. Cliquez sur **M'abonner**.
2. Abonnez-vous à Mes notifications en accédant au Portail de support IBM et cliquez sur **Mes notifications** dans le portlet **Notifications**.
3. Ouvrez une session à l'aide de votre ID IBM et de votre mot de passe, puis cliquez sur **Envoyez**.
4. Indiquez les types de mise à jour à recevoir et la manière de les recevoir.
   a. Cliquez sur l'onglet **M'abonner**.
   b. Sélectionnez la marque de logiciel ou le type de matériel approprié.
   c. Sélectionnez un ou plusieurs produits par nom, puis cliquez sur **Continuer**.
   d. Sélectionnez vos préférences de mode de réception des mises à jour (e-mail, en ligne dans un dossier désigné ou comme flux RSS ou Atom).
   e. Sélectionnez les types de mise à jour de documentation à recevoir, par exemple, les nouvelles informations sur les téléchargements de produit et les commentaires des groupes de discussion.
   f. Cliquez sur **Submit**.

Résultats

Tant que vous ne modifiez pas vos flux RSS et vos préférences Mes notifications, vous recevez les notifications des mises à jour que vous avez demandées. Vous

pouvez modifier vos préférences si nécessaire (par exemple, si vous arrêtez d'utiliser un produit et commencez à en utiliser un autre).

Activation de la consignation

Vous pouvez utiliser des journaux pour surveiller et traiter les problèmes liés à votre environnement.

Pourquoi et quand exécuter cette tâche

Les journaux sont enregistrés dans des emplacements différents et les formats dépendant de votre configuration.

Procédure

- **Activez des journaux dans un environnement autonome.**

  Avec les serveurs de catalogue autonomes, les journaux se trouvent dans le répertoire dans lequel vous exécutez la commande start. Pour les serveurs de conteneur, vous pouvez utiliser l'emplacement par défaut ou définir un emplacement de journal personnalisé :

  – **Emplacement de journal par défaut :** les journaux se trouvent dans le répertoire où la commande serveur a été exécutée. Si vous démarrez les serveurs dans le répertoire *rép_base_wxs*`/bin`, les journaux et les fichiers de trace se trouvent dans les sous-répertoires `logs/<`*`nom_serveur`*`>` du répertoire `bin`.

  – **Emplacement de journal personnalisé :** pour définir un autre emplacement pour les journaux des serveurs de conteneur, créez un fichier de propriétés, tel que `server.properties` contenant :

  ```
  workingDirectory=<directory>
  traceSpec=
  systemStreamToFileEnabled=true
  ```

  La propriété **`workingDirectory`** est le répertoire racine des journaux et du fichier de trace facultatif. WebSphere eXtreme Scale crée un répertoire avec le nom du serveur de conteneur avec un fichier `SystemOut.log`, un fichier `SystemErr.log` et un fichier de trace. Pour utiliser un fichier de propriétés au démarrage des conteneurs, utilisez l'option **`-serverProps`** et spécifiez l'emplacement du fichier de propriétés du serveur.

- **Activez les journaux dans WebSphere Application Server.**

  Voir WebSphere Application Server: Activation et désactivation de la consignation pour plus d'informations.

- **Extrayez les fichiers FFDC.**

  Les fichiers FFDC sont destinés au support technique d'IBM, pour le débogage. Ces fichiers peuvent être demandés par le support technique d'IBM en cas de problème. Ces fichiers se trouvent dans un répertoire libellé `ffdc` et contiennent des fichiers similaires au suivant :

  ```
  server2_exception.log
  server2_20802080_07.03.05_10.52.18_0.txt
  ```

- .NET 8.6+ **Activez les journaux dans un client .NET.** Les journaux dans un client .NET sont configurés par défaut et écrits dans le répertoire `logs` sur le client. Pour plus d'informations sur les journaux d'un client .NET, voir «Journaux du client .NET», à la page 667.

Que faire ensuite

Affichage des fichiers journaux dans leur emplacement spécifié. Les messages courants à rechercher dans le fichier `SystemOut.log` sont les messages de confirmation du démarrage, comme dans l'exemple suivant :

```
CWOBJ1001I: ObjectGrid Server catalogServer01 is ready to process requests.
```

Pour plus d'informations sur un message spécifique dans les fichiers journaux, voir Messages.

Configuration de la journalisation distante

Vous pouvez activer la journalisation distante pour enregistrer les entrées de journal sur un serveur distant. La journalisation distante peut être utile lorsque vous devez définir un niveau de journal de débogage détaillé pour isoler un problème ou surveiller un comportement sur une longue période.

Avant de commencer

- Vous devez disposer d'un serveur syslog qui écoute les événements et les capture.
- Les noms des serveurs de catalogue, de conteneur et d'applications (si vous utilisez WebSphere Application Server) ne doivent contenir que des caractères alphanumériques. Syslog RFC 1364 n'autorise pas les caractères non alphanumériques pour la zone TAG. La zone TAG contient le nom du serveur dans les messages syslog.

Pourquoi et quand exécuter cette tâche

Utilisez la journalisation distante pour analyser les données d'historique. Les serveurs dans l'environnement conservent un nombre limité de journaux dans le système. Configurez la journalisation distante si vous voulez enregistrer un plus grand nombre de fichiers journaux pour l'analyse. Le serveur de journalisation distant agrège les données de plusieurs serveurs. Vous pouvez configurer l'ensemble de la topologie des serveurs de catalogue et de conteneur pour envoyer les fichiers au même serveur de journalisation distant.

Procédure

1. Configurez la journalisation distante sur chaque serveur de catalogue ou de conteneur. Activez la journalisation distante en modifiant les propriétés suivantes dans le fichier des propriétés du serveur :

   8.6+ syslogEnabled
   Active la journalisation distante pour analyser les données d'historique. Vous devez disposer d'un serveur syslog qui écoute les événements et les capture.

   Valeur par défaut : `false`

   8.6+ syslogHostName
   Indique le nom d'hôte ou l'adresse IP du serveur distant sur lequel vous voulez consigner les données d'historique.

   8.6+ syslogHostPort
   Indique le numéro de port du serveur distant sur lequel vous voulez consigner les données d'historique.

   Valeurs valides : `0-65535`

Default: 512

8.6+ syslogFacility
Indique le type de fonction de journalisation à distance utilisé.

Valeurs valides : `kern, user, mail, daemon, auth, syslog, lpr, news, uucp, cron, authpriv, ftp, sys0, sys1, sys2, sys3,local0, local1, local2, local3, local4, local5, local6, local7`

Valeur par défaut : `user`

8.6+ syslogThreshold
Indique le seuil de gravité des messages à envoyer au serveur de journalisation distant. Pour envoyer des messages d'avertissement et graves, entrez la valeur `WARNING`. Pour envoyer des messages graves, sélectionnez `SEVERE`.

Valeurs valides : `SEVERE, WARNING`

Valeur par défaut : `WARNING`

2. Redémarrez les serveurs de catalogue et de conteneur sur lesquels vous avez modifié les propriétés. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479.

Résultats

Les messages sont envoyés au serveur de journalisation distant configuré pour l'archivage et l'analyse.

Journaux du client .NET

.NET

Des journaux dans un client .NET sont configurés par défaut et écrits dans des fichiers dans le répertoire `logs` et le journal des événements Windows.

Fichiers journaux par défaut

Les fichiers journaux suivants sont générés par défaut.

- **`SystemOut.log`** : contient des messages d'information, d'erreur, d'avertissement et d'échec. Ce fichier se trouve dans le répertoire `logs/` du client.
- **`SystemErr.log`** : contient des messages d'erreur et d'échec. Ce fichier se trouve dans le répertoire `logs/` du client.
- **Journal des événements Windows :** les erreurs fatales sont placées dans le journal des événements Windows. Des erreurs fatales se produisent lorsque le client ne peut plus accepter des transactions. Les messages WebSphere eXtreme Scale sont consignés dans le journal des événements Windows comme messages `WXSEventLog`.

Journaux de trace et de l'outil de diagnostic de premier niveau

Les journaux de trace et l'outil de diagnostic de premier niveau ne sont pas activés sur les clients .NET. Si vous devez collecter les journaux de trace ou de l'outil de diagnostic de premier niveau d'un client .NET, contactez l'équipe du support pour obtenir une assistance supplémentaire. Pour plus d'informations, voir «Comment prendre contact avec le service de support IBM», à la page 661.

Collecte de trace

Vous pouvez utiliser une trace pour surveiller et traiter les problèmes liés à votre environnement. Vous devez fournir une trace pour un serveur lorsque vous contactez le support IBM.

Pourquoi et quand exécuter cette tâche

La collecte d'une trace peut vous aider à surveiller et corriger les problèmes dans votre déploiement de WebSphere eXtreme Scale. La manière dont vous collectez la trace dépend de votre configuration. Voir «Options de trace du serveur», à la page 669 pour la liste des spécifications de trace que vous pouvez collecter.

Procédure

- **Collectez la trace dans un environnement WebSphere Application Server.**

  Si votre catalogue et les serveurs de conteneur se trouvent dans un environnement WebSphere Application Server, voir WebSphere Application Server : Utilisation avec la fonction de trace pour plus d'informations.

- **Collectez la trace avec le catalogue autonome ou la commande de démarrage de serveur.**

  Vous pouvez définir la fonction de trace sur un service de catalogue ou serveur de conteneur en utilisant les paramètres **-traceSpec** et **-traceFile** avec la commande de démarrage de serveur. Par exemple :

  ```
  startOgServer.sh catalogServer -traceSpec ObjectGridPlacement=all=enabled -traceFile  /home/user1/logs/trace.log
  ```

  8.6+

  ```
  startXsServer.sh catalogServer -traceSpec ObjectGridPlacement=all=enabled -traceFile  /home/user1/logs/trace.log
  ```

  Le paramètre **-traceFile** est facultatif. Si vous ne définissez pas un emplacement **-traceFile**, le fichier de trace est placé dans le même endroit que les fichiers journaux du système.Pour des informations sur ces paramètres, voir «Script **startOgServer** (ORB)», à la page 499 and «Script **startXsServer** (XIO)», à la page 485.

- **Collectez la trace sur le catalogue autonome ou serveur de conteneur avec un fichier de propriétés.**

  Pour collecter une trace à partir d'un fichier de propriétés, créez un fichier, tel que `server.properties`, avec le contenu suivant :

  ```
  workingDirectory=<directory>
  traceSpec=<trace_specification>
  systemStreamToFileEnabled=true
  ```

  La propriété **workingDirectory** est le répertoire racine des journaux et du fichier de trace facultatif. Si la valeur **workingDirectory** n'est pas définie, le répertoire de travail par défaut est l'emplacement utilisé pour démarrer les serveurs (par exemple, *rép_base_wxs*`/bin`). Pour utiliser un fichier de propriétés au cours du démarrage du serveur, utilisez le paramètre **-serverProps** avec la commande **startOgServer** et fournissez l'emplacement du fichier de propriétés du serveur.Pour plus d'informations sur le fichier de propriétés du serveur et l'utilisation du fichier, voir Fichier de propriétés du serveur.

- Java **Collectez la trace sur un client autonome Java.**

  Vous pouvez démarrer la collecte de trace sur un client autonome en ajoutant des propriétés système au script de démarrage pour l'application client. Dans l'exemple suivant, les paramètres de trace sont spécifiés pour l'application `com.ibm.samples.MyClientProgram` :

```
java -DtraceSettingsFile=MyTraceSettings.properties
-Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager
-Djava.util.logging.configureByServer=true com.ibm.samples.MyClientProgram
```

Pour plus d'informations, voir WebSphere Application Server: Activation de la fonction de trace sur les applications client et autonomes.

- .NET 8.6+ **Collectez la trace sur un client .NET.**

  La fonction de trace n'est pas activée par défaut pour les clients .NET. Si vous devez collecter la trace pour un client .NET, contactez le support pour obtenir une assistance supplémentaire. Pour plus d'informations, voir «Comment prendre contact avec le service de support IBM», à la page 661.

- Java **Collectez la trace avec l'interface ObjectGridManager.**

  Vous pouvez également définir la trace lors de la phase d'exécution sur une interface ObjectGridManager. La définition de la sur une interface ObjectGridManager permet d'extraire la trace sur un client eXtreme Scale lorsqu'il se connecte à eXtreme Scale et valide des transactions. Pour définir la trace sur une interface ObjectGridManager, fournissez une spécification de trace et un journal de trace.

  ```
  ObjectGridManager manager = ObjectGridManagerFactory.getObjectGridManager();
  ...
  manager.setTraceEnabled(true);
  manager.setTraceFileName("logs/myClient.log");
  manager.setTraceSpecification("ObjectGridReplication=all=enabled");
  ```

  Pour plus d'informations sur l'interface ObjectGridManager, voir Interaction avec un objet ObjectGrid en utilisant l'interface ObjectGridManager.

- **Collectez la trace sur les serveurs de conteneur avec l'utilitaire `xscmd`.**

  Pour collecter une trace à l'aide de l'utilitaire **`xscmd`**, utilisez la commande **`setTraceSpec`**. Utilisez l'utilitaire **`xscmd`** pour collecter la trace sur un environnement autonome lors de la phase d'exécution et non pas au démarrage. Vous pouvez collecter la trace sur tous les serveurs et services de catalogue ou filtrer les serveurs en fonction du nom ObjectGrid, et d'autres propriétés. Par exemple, pour collecter la trace ObjectGridReplication avec un accès au serveur de service de catalogue, exécutez :

  ```
  xscmd -c setTraceSpec -spec "ObjectGridReplication=all=enabled"
  ```

  Vous pouvez également désactiver la trace en affectant à la spécification de trace la valeur `*=all=disabled`.

Résultats

Les fichiers de trace sont écrits dans l'emplacement défini.

Options de trace du serveur

Vous pouvez activer la trace pour fournir des informations sur votre environnement au service d'assistance IBM.

A propos de la trace

La trace de WebSphere eXtreme Scale est divisée en plusieurs composants. Vous pouvez définir le niveau de trace à utiliser pour un serveur de catalogue ou un serveur de conteneur. Les niveaux de trace courants sont les suivants : all, debug, entryExit et event.

Voici un exemple de chaîne de trace :

```
ObjectGridComponent=level=enabled
```

Vous pouvez concaténer les chaînes de trace. Utilisez le symbole * (astérisque) pour spécifier une valeur générique, telle que `ObjectGrid*=all=enabled`. Si vous devez fournir une trace au service d'assistance IBM, une chaîne de trace spécifique est demandée. Par exemple, en cas de problème de réplication, la trace `ObjectGridReplication=debug=enabled` peut être demandée.

Spécification de la trace

ObjectGrid
Moteur général du cache central.

ObjectGridCatalogServer
Service de catalogue général.

ObjectGridChannel
Communications statiques de la topologie de déploiement.

ObjectGridClientInfo
Informations sur le client DB2.

ObjectGridClientInfoUser
Informations sur l'utilisateur DB2.

ObjectgridCORBA
Communications dynamiques de la topologie de déploiement.

ObjectGridDataGrid
API AgentManager.

ObjectGridDynaCache
Fournisseur de cache dynamique de WebSphere eXtreme Scale.

ObjectGridEntityManager
API EntityManager. A utiliser avec l'option Projector.

ObjectGridEvictors
Expulseurs pré-intégrés d'ObjectGrid.

ObjectGridJPA
Chargeurs JPA (Java Persistence API).

ObjectGridJPACache
Plug-in de cache JPA.

ObjectGridLocking
Gestionnaire de verrouillage des entrées de cache d'ObjectGrid.

8.6+ ObjectGridLogHandler
Informations de journalisation distante.

ObjectGridMBean
Beans de gestion.

ObjectGridMonitor
Infrastructure de la surveillance de l'historique.

ObjectGridNative
Trace de code natif WebSphere eXtreme Scale, y compris le code natif eXtremeMemory.

ObjectGridOSGi
Les composants d'intégration OSGi WebSphere eXtreme Scale.

ObjectGridPlacement
Service de positionnement des fragments de serveur de catalogues.

ObjectGridQuery
Requête ObjectGrid.

ObjectGridReplication
Service de réplication.

ObjectGridRouting
Détails du routage client/serveur.

ObjectGridSecurity
Trace de la sécurité.

ObjectGridSerializer
Infrastructure de plug-in DataSerializer.

ObjectGridStats
Statistiques d'ObjectGrid.

ObjectGridTransactionManager
Gestionnaire de transaction WebSphere eXtreme Scale.

ObjectGridWriteBehind
Ecriture différée d'ObjectGrid.

ObjectGridXA
Trace du transaction multipartition.

ObjectGridXM
Trace IBM eXtremeMemory générale.

ObjectGridXMEviction
Trace d'expulsion eXtremeMemory.

ObjectGridXMTransport
Trace de transport générale eXtremeMemory.

ObjectGridXMTransportInbound
Trace de transport entrant eXtremeMemory.

ObjectGridXMTransportOutbound
Trace de transport sortant eXtremeMemory.

Projector
Moteur dans l'API EntityManager.

QueryEngine
Moteur de requête des API Object Query et EntityManager Query.

QueryEnginePlan
Trace du plan de requête.

TCPChannel
Canal TCP/IP IBM eXtremeIO.

XsByteBuffer
Trace de mémoire tampon d'octets WebSphere eXtreme Scale.

Traitement des problèmes HPEL (High Performance Extensible Logging)

HPEL est une fonction de journalisation et de trace que vous pouvez utiliser dans les environnements autonomes et WebSphere Application Server. Vous pouvez utiliser HPEL pour stocker et accéder aux informations de journal, de trace, `System.err` et `System.out` générées par le serveur d'applications ou les applications. HPEL est une alternative à la fonction de trace et de journal de base qui fournit les journaux JVM (Java virtual machine), les fichiers de trace et les fichiers journaux de service. Ces fichiers s'appellent généralement `SystemOut.log/SystemErr.log`, `trace.log` et `activity.log`. HPEL fournit un référentiel de données de journalisation, un référentiel de données de trace et un fichier journal de texte.

Pourquoi et quand exécuter cette tâche

Au lieu de la fonction de consignation existante, vous pouvez utiliser HPEL, qui est désactivé par défaut. En mode HPEL, le contenu des journaux et de trace est écrit dans un fichier journal ou un référentiel de données de trace dans un format binaire propriétaire. Par conséquent, la désactivation de HPEL peut améliorer les performance des serveurs en accélérant la journalisation et la fonction de trace. Activez HPEL avec les fichiers de propriétés des serveurs de conteneur et de catalogue. Après avoir activé HPEL, tous les fichiers de journalisation et les fichiers journaux résultants WebSphere eXtreme Scale sont placés dans le référentiel HPEL défini.

Procédure

1. Définissez des propriétés pour activer la journalisation HPEL. Modifiez le Fichier de propriétés du serveur de chaque serveur de conteneur et de chaque serveur de catalogue avec les propriétés à utiliser.

   8.6+ hpelEnable
   Indique si HPEL (High Performance Extensible Logging) est activé. La journalisation HPEL est activée lorsque la propriété a la valeur `true`.

   Valeur par défaut : `false`

   8.6+ hpelRepositoryLocation
   Indique l'emplacement du référentiel de journalisation HPEL.

   Valeur par défaut : `"."` (l'emplacement d'exécution)

   8.6+ hpelEnablePurgeBySize
   Indique si HPEL purge les fichiers journaux en fonction de la taille. Vous pouvez définir la taille des fichiers avec la propriété hpelMaxRepositorySize.

   Valeur par défaut : `true` (activé)

   8.6+ hpelEnablePurgeByTime
   Indique si HPEL purge les fichiers journaux en fonction de l'heure. Définissez le délai avec la propriété hpelMaxRetentionTime.

   Valeur par défaut : `true` (enabled)

   8.6+ hpelEnableFileSwitch
   Indique si le fichier HPEL est activé pour créer un fichier à une heure spécifiée. Utilisez la propriété hpelFileSwitchHour pour indiquer l'heure de création d'un fichier.

Valeur par défaut : `false` (désactivé)

8.6+ hpelEnableBuffering
Indique si la mise en mémoire tampon HPEL est activée.

Valeur par défaut : `false` (disabled)

8.6+ hpelIncludeTrace
Indique si les fichiers texte HPEL incluent le traçage.

Valeur par défaut : `false` (disabled)

8.6+ hpelOutOfSpaceAction
Indique l'action à effectuer lorsque l'espace disque a été dépassé.

Valeur par défaut : `PurgeOld`

Valeurs possibles : `PurgeOld, StopServer, StopLogging`

8.6+ hpelOutputFormat
Indique le format des fichiers journaux à générer.

Valeur par défaut : `Basic`

Valeurs possibles : `Basic, Advanced, CBE-1.0.1`

8.6+ hpelMaxRepositorySize
Indique la taille maximale des fichiers en mégaoctets. Cette valeur est utilisée lorsque vous activez la propriété hpelEnablePurgeBySize.

Valeur par défaut :`50`

8.6+ hpelMaxRetentionTime
Indique la durée de conservation maximale des fichiers, en heures.

Valeur par défaut : `48`

8.6+ hpelFileSwitchHour
Indique l'heure à laquelle un fichier doit être créé. Cette valeur est utilisée lorsque la propriété hpelEnableFileSwitch est activée.

Valeur par défaut : `0`

2. Redémarrez les serveurs sur lesquels vous avez modifié le fichier de propriétés de serveur pour définir les propriétés HPEL. Après avoir activé HPEL et redémarré le serveur, les informations de journalisation WebSphere eXtreme Scale précédentes ne sont plus disponibles. Les informations de journalisation précédentes sont remplacées par les informations HPEL équivalentes. Pour plus d'informations, voir «Démarrage et arrêt des serveurs sécurisés», à la page 479 et «Démarrage et arrêt des serveurs dans un environnement WebSphere Application Server», à la page 508.
3. Utilisez l'afficheur de journaux HPEL de ligne de commande pour afficher les fichiers journaux. L'afficheur de journal de ligne de commande est une solution puissante, mais simple pour afficher les informations de journalisation. Pour une référence détaillée des options d'affichage de ligne de commande, voir le centre de documentation WebSphere Application Server : LogViewer command-line tool.
   a. Depuis une invite, accédez au répertoire `bin`. Windows

      ```
      C:\Program Files\IBM\WebSphere\eXtremeScale\ObjectGrid\bin
      ```

      Linux UNIX

      ```
      /opt/IBM/WebSphere/eXtremeScale/ObjectGrid/bin
      ```

b. Exécutez la commande suivante pour obtenir de l'aide sur l'afficheur de journal : Windows

```
logViewer -help
```

Linux UNIX

```
./logViewer.sh -help
```

4. Voici quelques-unes des commandes que vous pouvez utiliser avec l'afficheur de journal :
   - Exécutez la commande suivante pour créer un fichier journal de format legacy `legacyFormat.log` qui contient uniquement les enregistrements INFO, WARNING et SEVERE: Windows

     ```
     logViewer -outLog ..\logs\legacyFormat.log -minLevel INFO
     -maxLevel SEVERE
     ```

     Linux UNIX

     ```
     ./logViewer.sh -outLog ../logs/legacyFormat.log -minLevel INFO -maxLevel SEVERE
     ```

     Utilisez un éditeur de texte pour afficher le fichier journal de format legacy que vous avez créé.
   - Exécutez la commande suivante pour afficher uniquement les enregistrements de journal pour l'unité d'exécution 0 : Windows

     ```
     logViewer -thread 0
     ```

     Linux UNIX

     ```
     ./logViewer.sh -thread 0
     ```
   - Exécutez la commande suivante pour afficher uniquement les messages WARNING : Windows

     ```
     logViewer -level WARNING
     ```

     Linux UNIX

     ```
     ./logViewer.sh -level WARNING
     ```
   - Exécutez la commande suivante pour récupérer tous les enregistrements de journal QUI NE PROVIENNENT PAS des consignateurs qui commencent par `com.ibm` : Windows

     ```
     logViewer -excludeLoggers com.ibm.*
     ```

     Linux UNIX

     ```
     ./logViewer.sh -excludeLoggers com.ibm.*
     ```
   - Exécutez la commande suivante pour extraire le référentiel des messages WARNING et SEVERE uniquement et enregistrer le fichiers résultants dans un nouveau répertoire : Windows

```
logViewer -minLevel WARNING -maxLevel SEVERE -extractToNewRepository ..\logs\newHPELRepository
```

Linux UNIX

```
./logViewer.sh -minLevel WARNING -maxLevel SEVERE -extractToNewRepository ../logs/newHPELRepository
```

   - Exécutez la commande suivante pour exporter le contenu du référentiel résultant dans un fichier journal de format texte : Windows

```
logViewer -repositoryDir ..\logs\newHPELRepository -outLog ..\logs\newFormat.log
```

Linux UNIX

```
./logViewer.sh -repositoryDir ../logs/newHPELRepository -outLog
../logs/newFormat.log
```

Utilisez un éditeur de texte pour afficher le fichier journal résultant.

Analyse des journaux et des données de trace

Vous pouvez utiliser les outils d'analyse de journal pour analyser le fonctionnement de l'environnement d'exécution et résoudre les problèmes qui y apparaissent.

Pourquoi et quand exécuter cette tâche

Vous pouvez générer des rapports à partir des fichiers journaux et de trace existants dans l'environnement. Ces rapports graphiques peuvent être utilisés pour les objectifs suivants :

- **Analyse de l'état et des performances de l'environnement d'exécution :**
  - Cohérence de l'environnement de déploiement
  - Fréquence de consignation
  - Exécution de la topologie et topologie configurée
  - Modifications non planifiées de la topologie
  - Etat de quorum
  - Etat de la réplication des partitions
  - Statistiques de mémoire, rendement, utilisation du processeur, etc.
- **Pour traiter les problèmes dans l'environnement :**
  - Vues topologiques à des points spécifiques dans le temps
  - Statistiques de mémoire, rendement, utilisation du processeur au cours des problèmes
  - Niveaux de groupe de correctifs en cours, paramètres d'optimisation
  - Etat de quorum

Présentation de l'analyse du journal

Vous pouvez utiliser l'outil **xsLogAnalyzer** pour vous aider traiter les problèmes dans l'environnement.

Tous les messages de reprise en ligne

Affiche le nombre total de messages de reprise en ligne sous la forme d'un graphique dans le temps. Affiche également une liste des messages de reprise en ligne, y compris les serveurs qui ont été affectés.

Tous les messages critiques eXtreme Scale

Affiche les ID de message et les explications associées et les actions utilisateur qui peuvent vous faire gagner du temps dans la recherche des messages.

Toutes les exceptions

Affiche les cinq premières exceptions, y compris les messages et leur nombre et les serveurs affectés par l'exception.

Résumé de la topologie

Affiche le diagramme de la configuration de la topologie en fonction des fichiers journaux. Vous pouvez utiliser ce récapitulatif pour comparer avec votre configuration réelle, en identifiant les erreurs de configuration.

Cohérence de topologie : tableau de comparaison ORB (Object Request Broker)

Affiche les paramètres ORB de l'environnement. Vous pouvez utiliser ce tableau pour déterminer si les paramètres de l'environnement sont cohérents.

Vue Tableau chronologique d'événements

Affiche un diagramme chronologique des différentes actions qui ont eu lieu sur la grille de données, y compris les événements de cycle de vie, les exceptions, les messages critiques et les événements de diagnostic de premier niveau (FFDC).

Exécution de l'analyse du journal

Vous pouvez exécuter l'outil **xsLogAnalyzer** sur un ensemble de fichiers journaux et de trace à partir de n'importe quel ordinateur.

Avant de commencer

- Activez les journaux et la trace. Pour plus d'informations, reportez-vous aux rubriques «Activation de la consignation», à la page 665 et «Collecte de trace», à la page 668.
- Collectez vos fichiers journaux. Les fichiers journaux peuvent se trouver dans des emplacements différents selon la façon dont vous les avez configurés. Si vous utilisez les paramètres de journal par défaut, vous pouvez obtenir les fichiers journaux dans les emplacements suivants :
  - Dans une installation autonome : *racine_install_wxs*/bin/logs/<*server_name*>
  - Dans une installation intégrée à WebSphere Application Server : *racine_was*/logs/<*server_name*>
- Collectez vos fichiers de trace. Ils peuvent se trouver dans des emplacements différents selon la façon dont vous les avez configurés. Si vous utilisez les paramètres de trace par défaut, vous pouvez obtenir les fichiers de trace dans les emplacements suivants :
  - Dans une installation autonome : si aucune valeur de trace n'est définie, les fichiers de trace sont écrits dans le même emplacement que les fichiers journaux système.
  - Dans une installation intégrée à WebSphere Application Server : *racine_was*/profiles/*server_name*/logs.

  Copiez les fichiers journaux et de trace vers l'ordinateur à partir duquel vous avez l'intention d'utiliser l'outil d'analyse de journal.
- Si vous voulez créez des scanners personnalisés, créez un fichier de propriétés de spécifications de scanner et un fichier de configuration avant d'exécuter l'outil. Pour plus d'informations, voir «Création de scanners personnalisés pour l'analyse de journal», à la page 678.

Procédure

1. Exécutez l'outil **xsLogAnalyzer**.

   Le script se trouve dans les emplacements suivants :

- Dans une installation autonome : *racine_install_wxs*/ObjectGrid/bin
- Dans une installation intégrée à WebSphere Application Server : *racine_was*/bin

Conseil : Si les fichiers journaux sont volumineux, utilisez les paramètres **-startTime**, **-endTime**, et **-maxRecords** lorsque vous exécutez le rapport pour limiter le nombre d'entrées de journal analysées. L'utilisation de ces paramètres lorsque vous exécutez le rapport améliore la clarté et l'exécution du rapport. Vous pouvez exécuter plusieurs rapports sur un même groupe de fichiers journaux.

```
xsLogAnalyzer.sh|bat -logsRoot c:\myxslogs -outDir c:\myxslogs\out
-startTime 11.09.27_15.10.56.089 -endTime 11.09.27_16.10.56.089 -maxRecords 100
```

-logsRoot
Spécifie le chemin absolu du répertoire des journaux à évaluer (requis).

-outDir
Spécifie un répertoire pour y placer la sortie du rapport. Si vous ne définissez pas une valeur, le rapport est écrit dans l'emplacement racine de l'outil **xsLogAnalyzer**.

-startTime
Spécifie l'heure de début de l'évaluation dans les journaux. La date est au format suivant :
année.mois.jour_heure.minute.seconde.milliseconde

-endTime
Spécifie l'heure de fin de l'évaluation dans les journaux. La date est au format suivant : *année.mois.jour_heure.minute.seconde.milliseconde*

-trace Spécifie une chaîne de trace, telle que ObjectGrid*=all=enabled.

-maxRecords
Spécifie le nombre maximal d'enregistrements pour générer le rapport. La valeur par défaut est 100. Si vous définissez la valeur 50, les 50 premiers enregistrements sont générés pour la période définie.

2. Ouvrez les fichiers générés. Si vous n'avez pas défini de répertoire de sortie, les rapports sont générés dans le dossier report_*date_time*. Pour ouvrir la page principale des rapports, ouvrez le fichier index.html.
3. Utilisez les rapports pour analyser les données des journaux. Suivez les conseils ci-dessous pour optimiser les performances du rapport affiché :
   - Pour optimiser les performances des requêtes sur les données des journaux, utilisez des informations aussi spécifiques que possibles. Par exemple, la recherche de server dure plus longtemps et retourne plus de résultats que server_host_name.
   - Certaines vues ont un nombre limité de points de données affichés simultanément. Vous pouvez ajuster le segment de temps affiché en changeant les données en cours, telles que les heures de début et de fin, dans la vue.

Que faire ensuite

Pour plus d'informations sur le traitement de l'outil **xsLogAnalyzer** et les rapports générés, voir «Traitement des problèmes d'analyse de journal», à la page 679.

Création de scanners personnalisés pour l'analyse de journal

Vous pouvez créer des scanners personnalisés pour l'analyse de journal. Après avoir configuré le scanner, les résultats sont générés dans les rapports lorsque vous exécutez l'outil **xsLogAnalyzer**. Le scanner personnalisé recherche les enregistrements d'événement dans les journaux en fonction des expressions régulières que vous avez définies.

Procédure

1. Créez un fichier de propriétés de spécification de scanner qui définit l'expression générale à exécuter pour le scanner personnalisé.
   a. Créez et enregistrez un fichier de propriétés. Le fichier doit se trouver dans le répertoire *loganalyzer_root*/config/custom. Vous pouvez attribuer le nom de choix. Le fichier est utilisé par le nouveau scanner ; il est donc utile de nommer le scanner dans le fichier des propriétés. Par exemple, *my_new_server_scanner_spec*.properties.
   b. Incluez les propriétés suivantes dans le fichier *my_new_server_scanner_spec*.properties :

      ```
      include.regular_expression = REGULAR_EXPRESSION_TO_SCAN
      ```

      La variable *REGULAR_EXPRESSION_TO_SCAN* est une expression régulière en fonction de laquelle vous filtrez les fichiers journaux.

      Exemple : pour analyser les instances des lignes qui contiennent les chaînes "xception" et "rror", quel que soit l'ordre, affectez la valeur suivante à la propriété **include.regular_expression** :

      ```
      include.regular_expression = (xception.+rror)|(rror.+xception)
      ```

      Cette expression régulière permet d'enregistrer les événements si la chaîne "rror" se trouve avant ou après la chaîne "xception".

      Exemple :Pour analyser chaque ligne des journaux pour rechercher les lignes qui contiennent la chaîne "xception" ou "rror" quel que soit l'ordre, affectez la valeur suivante à la propriété **include.regular_expression** :

      ```
      include.regular_expression = (xception)|(rror)
      ```

      Cette expression régulière permet d'enregistrer les événements si la chaîne "rror" ou "xception" existe.
2. Créez un fichier de configuration que l'outil **xsLogAnalyer** utilise pour créer le scanner.
   a. Créez et enregistrez un fichier de configuration. Le fichier doit se trouver dans le répertoire *loganalyzer_root*/config/custom. Vous pouvez nommer le fichier *scanner_name*Scanner.config, où *scanner_name* est le nom unique du nouveau scanner. Par exemple, vous pouvez nommer le fichier serverScanner.config
   b. Incluez les propriétés suivantes dans le fichier *scanner_name*Scanner.config :

      ```
      scannerSpecificationFiles = LOCATION_OF_SCANNER_SPECIFICATION_FILE
      ```

      La variable *LOCATION_OF_SCANNER_SPECIFICATION_FILE* est le chemin et l'emplacement du fichier de spécification que vous avez créé au cours de l'étape précédente. Par exemple : *loganalyzer_root*/config/custom/my_new_scanner_spec.properties. Vous pouvez aussi définir plusieurs fichiers de spécification de scanner en utilisant une liste d'éléments séparés par un point-virgule :

```
scannerSpecificationFiles = LOCATION_OF_SCANNER_SPECIFICATION_FILE1;LOCATION_OF_SCANNER_SPECIFICATION_FILE2
```

3. Exécutez l'outil **xsLogAnalyzer**. Pour plus d'informations, voir «Exécution de l'analyse du journal», à la page 676.

Résultats

Après avoir exécuté l'outil **xsLogAnalyzer**, le rapport contient de nouveaux onglets pour les scanners personnalisés que vous avez configurés. Chaque onglet contient les vues suivantes :

Graphiques
Graphique qui illustre les événements enregistrés. Les événements sont affichés dans leurs ordre de découverte.

Tableaux
Représentation tabulaire des événements enregistrés.

Etats récapitulatifs

Traitement des problèmes d'analyse de journal

Utilisez les informations de dépannage pour identifier et éliminer les problèmes avec l'outil **xsLogAnalyzer** et ses rapports générés.

Procédure

- **Problème :** manque de mémoire lors de l'utilisation de l'outil **xsLogAnalyzer** pour générer des rapports. Exemple d'erreur possible : `java.lang.OutOfMemoryError: GC overhead limit exceeded`.

  Solution : l'outil **xsLogAnalyzer** s'exécute dans une machine JVM (Java virtual machine). Vous pouvez configurer la machine JVM pour augmenter la taille de segment avant d'exécuter l'outil **xsLogAnalyzer** en définissant certains paramètres lorsque vous exécutez l'outil. L'augmentation de la taille du segment permet de stocker plus d'enregistrements dans la mémoire JVM. Commencez avec 2 048 M en supposant que le système d'exploitation dispose d'une mémoire principale suffisante. Dans la même instance de ligne de commande dans laquelle vous voulez exécuter l'outil **xsLogAnalyzer**, définissez la taille de segment de mémoire JVM maximale :

  ```
  java -Xmx HEAP_SIZE m
  ```

  La valeur *HEAP_SIZE* peut être un entier et représente le nombre de mégaoctets alloués au segment de mémoire JVM. Par exemple, vous pouvez exécuter `java -Xmx2048m`. Si vous continuez de recevoir des messages indiquant un manque de mémoire ou que vous ne disposez pas des ressources pour allouer 2 048 Mo ou plus, limitez le nombre d'événements stockés dans le segment de mémoire. Vous pouvez limiter le nombre d'événements dans le segment de mémoire en envoyant le paramètre **-maxRecords** dans la commande **xsLogAnalyzer**.

- **Problème :** lorsque vous ouvrez un rapport généré depuis l'outil **xsLogAnalyzer**, le navigateur se bloque et ne charge pas la page.

  Cause : les fichiers HTML générés sont trop volumineux et le navigateur ne peut pas les charger. Ces fichiers sont volumineux, car la portée des fichiers journaux que vous analysez est trop grande.

  Solution : utilisez les paramètres **-startTime**, **-endTime**, et **-maxRecords** lorsque vous exécutez l'outil **xsLogAnalyzer** pour limiter le nombre d'entrées de journal analysées. L'utilisation de ces paramètres lorsque vous exécutez le rapport améliore la clarté et l'exécution du rapport. Vous pouvez exécuter plusieurs rapports sur un même groupe de fichiers journaux.

Identification et résolution des incidents liés à l'installation du produit

IBM Installation Manager est un programme d'installation commun à de nombreux logiciels IBM qui vous permet d'installer cette version de WebSphere eXtreme Scale.

Résultats

Remarques sur la journalisation et le traçage :

- Pour consulter aisément les journaux, ouvrez le gestionnaire d'installation et cliquez sur **File > View Log**. Pour ouvrir un fichier journal individuel, il vous suffira alors de le sélectionner dans le tableau et de cliquer sur l'icône **Open log file**.
- Les journaux sont situés dans le répertoire `logs`, à l'emplacement des données applicatives d'Installation Manager. Exemple :
  - Windows **Installation par un administrateur :**

```
C:\Documents and Settings\All Users\Application Data\IBM\Installation Manager
```

  - Windows **Installation par un non administrateur :**

```
C:\Documents and Settings\nom_utilisateur\Application Data\IBM\Installation Manager
```

  - UNIX Linux **Installation par un administrateur :**

```
/var/IBM/InstallationManager
```

  - UNIX Linux **Installation par un non administrateur :**

```
rép_utilisateur/var/ibm/InstallationManager
```

- Les principaux fichiers journaux sont horodatés et conservés dans le répertoire `logs` au format XML. Ils peuvent donc être consultés à l'aide d'un navigateur Web standard.
- Le fichier `log.properties` du répertoire `logs` indique le niveau de journalisation ou de traçage appliqué par Installation Manager. Pour activer la fonction de traçage des plug-ins de WebSphere eXtreme Scale, par exemple, créez un fichier `log.properties` contenant les informations suivantes :

```
com.ibm.ws=DEBUG
com.ibm.cic.agent.core.Engine=DEBUG
global=DEBUG
```

  Redémarrez Installation Manager au besoin. Installation Manager fournira ainsi les traces des plug-ins de WebSphere eXtreme Scale.

Remarques sur la résolution des problèmes :

- UNIX Linux Par défaut, certains systèmes HP-UX sont configurés pour ne pas utiliser de système DNS pour résoudre les noms d'hôte. Il est par conséquent possible qu'Installation Manager ne puisse pas se connecter à un référentiel externe.

  Vous pouvez vérifier la connexion au référentiel à l'aide de l'utilitaire Ping mais nslookup ne renverra aucun résultat.

  Demandez à votre administrateur système de configurer votre machine afin qu'elle utilise un système DNS ou utilisez l'adresse IP du référentiel.
- Dans certains cas, il peut être nécessaire d'ignorer les mécanismes de vérification existants dans Installation Manager.
  - Sur certains systèmes de fichiers réseau, l'espace disque peut ne pas être signalé correctement et il peut être nécessaire d'ignorer la vérification d'espace disque et de poursuivre l'installation.

Pour désactiver la vérification d'espace disque, spécifiez la propriété système suivante dans le fichier `config.ini` du répertoire *racine_install_IM*`/eclipse/configuration` et redémarrez Installation Manager :

```
cic.override.disk.space=tailleunité
```

où *taille* est un chiffre entier et *unité* est laissé vide pour octets, a la valeur k pour kilo, m pour mégaoctets ou g pour gigaoctets. Par exemple :

```
cic.override.disk.space=120 (120 octets)
cic.override.disk.space=130k (130 kilooctets)
cic.override.disk.space=140m (140 mégaoctets)
cic.override.disk.space=150g (150 gigaoctets)
cic.override.disk.space=true
```

Installation Manager indique que la taille de l'espace disque est Long.MAX_VALUE. Au lieu d'afficher une grande quantité d'espace disque disponible, N/A s'affiche.

– Pour ignorer la vérification des prérequis pour le système d'exploitation, ajoutez `disableOSPrereqChecking=true` au fichier `config.ini` dans *racine_install_IM*`/eclipse/configuration` et redémarrez Installation Manager.

Si vous devez utiliser l'une de ces méthodes, contactez le support IBM pour obtenir de l'aide et développer une solution n'impliquant pas d'ignorer les mécanismes de vérification d'Installation Manager.

- Pour plus d'informations sur l'utilisation d'Installation Manager, accédez au centre de documentation d'IBM Installation Manager Version 1.5.

  Pour en savoir plus sur la dernière version d'Installation Manager, lisez les notes sur l'édition. Pour accéder aux notes sur l'édition, procédez comme suit :

  – Windows Cliquez sur **Démarrer > Programmes > IBM Installation Manager > Release Notes**.

  – UNIX Linux Accédez au sous-répertoire de la documentation dans le répertoire dans lequel Installation Manager est installé et ouvrez le fichier `readme.html`.

- Si une erreur fatale se produit lors de l'installation du produit, effectuez les étapes suivantes :
  – Faites une copie de sauvegarde de votre répertoire d'installation de produit actuel au cas où le service de support IBM aurait besoin de l'examiner ultérieurement.
  – Utilisez Installation Manager pour désinstaller tout ce que vous avez installé dans l'emplacement d'installation du produit (groupe de packages). Vous risquez de rencontrer des erreurs, mais vous pouvez les ignorer en toute sécurité.
  – Supprimez tout ce qui reste dans le répertoire d'installation du produit.
  – Utilisez Installation Manager pour réinstaller le produit dans le même emplacement ou dans un nouvel emplacement.

Remarque sur les informations de version et d'historique : Les commandes **`versionInfo`** et **`historyInfo`** retournent les informations de version et d'historique basées sur toutes les activités d'installation, de désinstallation, de mise à jour et de rétrogradation effectuées sur le système.

Traitement des problèmes d'intégration du cache

Utilisez ces informations pour traiter les problèmes de la configuration de l'intégration du cache, y compris ceux associés aux configurations de session HTTP et de cache dynamique.

Procédure

- **Problème :** les ID de session HTTP ne sont pas réutilisés.

  Cause : vous pouvez réutiliser les ID de session. Si vous créez une grille de données pour la persistance des sessions dans la version 7.1.1 ou une version ultérieure, la réutilisation des ID de session est automatiquement activée. Toutefois, si vous avez créé des configurations dans des versions antérieures, ce paramètre est peut être déjà défini avec une valeur incorrecte.

  Solution : vérifiez les paramètres suivants pour déterminer si vous avez activé la réutilisation des ID de session HTTP :

  – La propriété `reuseSessionId` dans le fichier `splicer.properties` doit avoir la valeur `true`.

  – La propriété personnalisée `HttpSessionIdReuse` doit avoir la valeur `true`. Cette propriété personnalisée peut être définie dans l'un des chemins suivants dans la console d'administration WebSphere Application Server :

  - **Serveurs** > *server_name* > **Getion de session** > **Propriétés personnalisées**
  - **Clusters dynamiques** > *dynamic_cluster_name* > **Modèle de serveur** > **Gestin de session** > **Propriétés personnalisés**
  - **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name*, puis sous Infrastructure du serveur, cliquez sur **Java et gestion des processus** > **Définition de processus** > **Java virtual machine** > **Propriétés personnalisées**
  - **Serveurs** > **Types de serveur** > **Serveurs d'applications WebSphere** > *server_name* > **Paramètres de conteneur Web** > **Conteneur Web**

  Si vous mettez à jour les valeurs des propriétés personnalisées, reconfigurez la gestion des sessions eXtreme Scale afin que le fichier `splicer.properties` détecte la modification

- **Problème :** lorsque vous utilisez un grille de données pour stocker les sessions HTTP et que la charge des transactions est élevée, le message CWOBJ0006W figure dans le fichier `SystemOut.log`.

  ```
  CWOBJ0006W: An exception occurred:
  com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
  java.util.ConcurrentModificationException
  ```

  Ce message apparaît lorsque le paramètre **`replicationInterval`** dans le fichier `splicer.properties` a une valeur supérieure à zéro et que l'application Web modifie l'objet List défini comme attribut dans la session HTTPSession.

  Solution : clonez l'attribut qui contient l'objet List modifié et placez l'attribut cloné dans l'objet session.

- 8.6+ **Problème :** lors de l'exécution des applications Web avec la spécification Servlet 3,0, les filtres d'application Web et les programmes d'écoute ne sont pas appelés par la gestion de session WebSphere eXtreme Scale. Par exemple, les programmes d'écoute ne sont pas rappelés lorsque les sessions sont invalidées à l'aide d'explusion conteneur distant avec WebSphere eXtreme Scale.

  Cause : WebSphere eXtreme Scale n'identifie pas les filtres ni les programmes d'écoute définis en utilisant des annotations ou un programme.

  Solution : les filtres et programmes d'écoute doivent être explicitement déclarés dans le fichier web.xml de l'application Web.

Traitement des problèmes du plug-in de mémoire cache JPA

Java

Utilisez ces informations pour traiter les problèmes de configuration du plug-in de mémoire cache JPA. Ces problèmes peuvent se produire dans les deux configurations Hibernate et OpenJPA.

Procédure

- **Problème :** l'exception suivante s'affiche : `CacheException: Failed to get ObjectGrid server`.

  Avec la valeur d'attribut `EMBEDDED` ou `EMBEDDED_PARTITION` **`ObjectGridType`**, la mémoire cache eXtreme Scale tente d'obtenir une instance de serveur de l'environnement d'exécution. Dans un environnement Java Platform, Standard Edition, un serveur eXtreme Scale avec un service de catalogue intégré est démarré. Le service de catalogue intégré essaie d'écouter sur le port 2809. Si ce port est utilisé par un autre processus, l'erreur se produit.

  Solution : si des noeuds finals de service de catalogue externes sont spécifiés, par exemple, avec le fichier `objectGridServer.properties`, cette erreur se produit si le nom d'hôte ou le port ne sont spécifiés correctement. Corrigez le conflit de port.

- **Problème :** l'exception suivante s'affiche : `CacheException: Failed to get REMOTE ObjectGrid for configured REMOTE ObjectGrid. objectGridName = [ObjectGridName], PU name = [persistenceUnitName]`

  Cette erreur se produit, car le cache ne peut pas obtenir l'instance ObjectGrid à partir des noeuds finals de service de catalogue fournis.

  Solution : ce problème se produit généralement lorsque le nom ou le port hôte est incorrect.

- **Problème :** l'exception suivante s'affiche : `CacheException: Cannot have two PUs [persistenceUnitName_1, persistenceUnitName_2] configured with same ObjectGridName [ObjectGridName] of EMBEDDED ObjectGridType`

  Cette exception se produit si un grand nombre d'unités de persistance sont configurées et que les mémoires caches eXtreme Scale de ces unités sont configurées avec le même nom ObjectGrid et la même valeur d'attribut `EMBEDDED` **`ObjectGridType`**. Ces configurations d'unités de persistance peuvent être dans les mêmes fichiers `persistence.xml` ou dans des fichiers différents.

  Solution : vous devez vérifier que le nom d'ObjectGrid est unique pour chaque unité de persistance lorsque la valeur de l'attribut **`ObjectGridType`** est `EMBEDDED`.

- **Problème :** l'exception suivante s'affiche : `CacheException: REMOTE ObjectGrid [ObjectGridName] does not include required BackingMaps [mapName_1, mapName_2,...]`

  Avec un type ObjectGrid REMOTE, si l'ObjectGrid obtenu côté client ne dispose pas des mappes de sauvegarde complètes d'entités pour prendre en charge le cache de l'unité de persistance, cette exception se produit. Supposons par exemple que cinq classes d'entités sont répertoriées dans la configuration des unités de persistance mais que l'ObjectGrid obtenu ne dispose que de deux mappes de sauvegarde. Même si l'ObjectGrid obtenu peut avoir 10 BackingMaps, si l'une des cinq BackingMaps d'entité requises est introuvable dans le 10 mappes de sauvegarde, cette exception se produit toujours.

  Solution : vérifiez que votre configuration de mappes de sauvegarde prend en charge la mémoire cache de l'unité de persistance.

Traitement des incidents IBM eXtremeMemory

Suivez la procédure ci-dessous pour résoudre les problèmes eXtremeMemory.

Procédure

Problème : Si la ressource partagée, libstdc++.so.5, n'est pas installée, lorsque vous démarrez le serveur de conteneur, les bibliothèques natives IBM eXtremeMemory ne se chargent pas.

Linux **Symptôme :** Sur les systèmes d'exploitation Linux 64 bits, si vous tentez de démarrer un serveur de conteneur avec la propriété serveur enableXM définie sur `true` et si la ressource partagée `libstdc++.so.5` n'est pas installée, vous recevez un message d'erreur similaire au suivant :

```
00000000 Initialization W CWOBJ0006W: An exception occurred: java.lang.reflect.InvocationTargetException
 at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
 at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:56)
 at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:39)
 at java.lang.reflect.Constructor.newInstance(Constructor.java:527)
 at com.ibm.websphere.objectgrid.server.ServerFactory.initialize(ServerFactory.java:350)
 at com.ibm.websphere.objectgrid.server.ServerFactory$2.run(ServerFactory.java:303)
 at java.security.AccessController.doPrivileged(AccessController.java:202)
 at com.ibm.websphere.objectgrid.server.ServerFactory.getInstance(ServerFactory.java:301)
 at com.ibm.ws.objectgrid.InitializationService.main(InitializationService.java:302)

Caused by: com.ibm.websphere.objectgrid.ObjectGridRuntimeException: java.lang.UnsatisfiedLinkError:
 OffheapMapdbg (Not found in java.library.path)
 at com.ibm.ws.objectgrid.ServerImpl.<init&gt;(ServerImpl.java:1033)
... 9 more Caused by: java.lang.UnsatisfiedLinkError: OffheapMapdbg (Not found in java.library.path)
 at java.lang.ClassLoader.loadLibraryWithPath(ClassLoader.java:1011)
 at java.lang.ClassLoader.loadLibraryWithClassLoader(ClassLoader.java:975)
 at java.lang.System.loadLibrary(System.java:469)
 at com.ibm.ws.objectgrid.io.offheap.ObjectGridHashTableOH.initializeNative(ObjectGridHashTableOH.java:112)
 at com.ibm.ws.objectgrid.io.offheap.ObjectGridHashTableOH.<clinit&gt;(ObjectGridHashTableOH.java:87)
 at java.lang.J9VMInternals.initializeImpl(Native Method)
 at java.lang.J9VMInternals.initialize(J9VMInternals.java:200)
 at com.ibm.ws.objectgrid.ServerImpl.<init&gt;(ServerImpl.java:1028)
... 9 more
```

Cause : La ressource partagée `libstdc++.so.5` n'a pas été installée.
Diagnostic du problème : Pour vérifier si la ressource `libstdc++.so.5` est installée, lancez la commane suivante à partir du répertoire ObjectGrid/native de votre installation :

```
ldd libOffheapMap.so
```

Si la bibliothèque partagée n'est pas installée, vous recevez l'erreur suivante :

```
ldd libOffheapMap.so
libstdc++.so.5 => not found
```

Résolution du problème : Utilise le programme d'installation de package de votre distribution Linux 64 bits pour installer le fichier de ressources requis. Ce package peut se présenter sous la forme `compat-libstdc++-33.x86_64` ou `libstdc++5`. Une fois la ressource installée, vérifiez que le package `libstdc++5` est installé en lançant la commande suivante à partir du répertoire ObjectGrid de votre installation :

```
ldd libOffheapMap.so
```

Traitement des problèmes d'administration

Utilisez les informations suivantes pour traiter les problèmes d'administration, notamment le démarrage et l'arrêt des serveurs, en utilisant l'utilitaire **xscmd**, etc.

Procédure

- **Problème :** les scripts d'administration manquent dans le répertoire *profile_root*`/bin` d'une installation WebSphere Application Server.

Cause : lorsque vous mettez à jour l'installation, les nouveaux fichiers scripts ne sont pas installés automatiquement dans les profils.

Solution : si vous voulez exécuter un script depuis le répertoire *profile_root*/bin, annulez l'extension et étendez de nouveau le profils avec la dernière version. Pour plus d'informations, voir Annulation de l'extension d'un profil en utilisant l'invite de commande et «Création et augmentation de profils pour WebSphere eXtreme Scale», à la page 243.

- **Problème :** lorsque vous exécutez une commande **xscmd**, le message suivant s'affiche :

```
java.lang.IllegalStateException: Placement service MBean not available.
[]
        at
com.ibm.websphere.samples.objectgrid.admin.OGAdmin.main(OGAdmin.java:1449)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:60)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:37)
        at java.lang.reflect.Method.invoke(Method.java:611)
        at com.ibm.ws.bootstrap.WSLauncher.main(WSLauncher.java:267)
Ending at: 2011-11-10 18:13:00.000000484
```

  Cause :problème de connexion avec le serveur de catalogue.

  Solution : vérifiez que les serveurs de catalogue sont actifs et disponibles via le réseau. Ce message peut aussi être généré lorsque vous disposez d'un domaine de services de catalogue défini et que moins de deux serveurs de catalogue sont actifs. L'environnement n'est pas disponible tant que deux serveurs de catalogue ne sont pas démarrés.

- **Problème :** lorsque vous exécutez une commande **xscmd**, le message suivant s'affiche :

```
CWXSI0066E: Unmatched argument argument_name was detected.
```

  Cause : Vous avez entré un format de commande qui n'est pas reconnu par l'utilitaire **xscmd**.

  Solution : Vérifiez le format de la commande. Vous pouvez rencontrer ce problème lorsque vous exécutez des expressions régulières avec la commande **-c findbyKey**. Pour plus d'informations, voir «Demande, affichage et invalidation des données», à la page 524.

- 8.6+ **Problème :** toutes les commandes start, stop et **xscmd** échouent avec l'erreur java.lang.UnsupportedClassVersionError.

  Par exemple, vous pouvez recevoir l'une des erreurs suivantes lorsque vous utilisez start, stop oul'utilitaire**xscmd**:

```
The java class could not be loaded. java.lang.UnsupportedClassVersionError:
(com/ibm/ws/xs/admin/wxscli/WXSAdminCLI) bad major version at offset=6
```

```
The java class could not be loaded. java.lang.UnsupportedClassVersionError:
(com/ibm/ws/objectgrid/server/impl/ProcessLauncher) bad major version at offset=6
```

  Cause : les commandes sont en cours d'exécution avec une version non prise en charge Java pour WebSphere eXtreme Scale.

  Solution : mettez à jour la variable d'environnement *JAVA_HOME* pour qu'elle pointe vers une installation Java Development Kit (JDK) prise en charge. Pour les versions prises en charge et les instructions de mise à jour du kit JDK, voir «Java SE : points à prendre en considération», à la page 68.

Traitement des incidents de surveillance des données

Utilisez ces informations pour traiter les problèmes d'activités de surveillance que vous effectuez à l'aide de la console d'administration Web WebSphere eXtreme Scale ou d'autres utilitaires pour surveiller les performances de votre environnement d'application.

Procédure

Problème : vous ne pouvez pas basculer entre les domaines avec des paramètres de sécurité différents dans la console Web WebSphere eXtreme Scale.
Vous pouvez basculer entre deux domaines non sécurisés. Vous également basculer entre deux domaines sécurisés avec la même configuration de sécurité. Cependant, vous ne pouvez pas basculer entre un domaine non sécurisé et un domaine sécurisé ou entre deux domaines sécurisés avec des paramètres de sécurité différents.
Diagnotics : utilisez la commande **startOgServer** pour démarrer deux serveurs de catalogue différents dans des domaines distincts. Chaque serveur de catalogue n'a pas connaissance de l'existence des autres serveurs. Cependant, les deux serveurs de catalogue sont démarrés avec le même nom de domaine. Lorsque vous ne définissez pas le nom de domaine, les deux serveurs de catalogue démarrent dans des domaines différents avec le nom par défaut DefaultDomain. En outre, la console de surveillance affiche des données pour un seul des domaines de serveur de catalogue.
Cause : lorsque vous changez de domaines dans la console de surveillance, vous vous connectez au second domaine. Cependant, aucune grille de données du domaine s'affiche et les données de la première grille de domaine figurent toujours dans la vue. Par conséquent, au cours de l'exécution, les deux serveurs de catalogue sont exécutés dans des domaines distincts de même nom, DefaultDomain.
Solution : détermines les noms de domaine utilisés lorsque les serveurs de catalogue démarrent dans les deux domaines. Pour identifier les noms de domaine, analysez la syntaxe de commande **startOgServer** et identifiez le domaine défini. Comme ce scénario de problème n'est pas pris en charge, effectuez les opérations suivantes pour afficher les statistiques du domaine de service de catalogue correct :

1. Arrêtez les serveurs de catalogue et vérifiez qu'ils sont configurés pour démarrer avec des noms de domaine unique.
2. Redémarrez la console de surveillance.
3. Facultatif : Si une indisponibilité n'est pas possible, exécutez une deuxième console de surveillance pour surveiller le second domaine.

Traitement des problèmes de plusieurs configurations de centre de données

Utilisez ces informations pour traiter les problèmes de plusieurs configurations, y compris de la liaison entre les domaines de service de catalogue.

Avant de commencer

Vous devez utiliser l'utilitaire **xscmd** pour résoudre les problèmes liés aux configurations de centre de données. Pour plus d'informations, voir «Administration avec l'utilitaire **xscmd**», à la page 515.

Procédure

- 8.6+ **Problème :** vous devez déterminer si la réplication des données est synchronisée sur les serveurs de conteneur et les domaines de service de catalogue.

  Solution : exécutez la commande **xscmd -c showReplicationState** ou **xscmd.sh -c showDomainReplicationState**. Ces commandes affichent des informations sur

l'état de la réplication dans l'environnement. Pour plus d'informations, voir «Surveillance avec l'utilitaire **xscmd**», à la page 571.

- 8.6+ **Problème :** vous devez identifier les domaines de service de catalogue liés au domaine de service de catalogue local.

  Solution : exécutez la commande **xscmd -c showLinkedDomains**. Cette commande répertorie les domaines de service de catalogue externe liés au domaine de service de catalogue local.

- 8.6+ **Problème :** vous souhaitez détecter les problèmes de configuration associés aux liens de fragment primaire aux domaines de service de catalogue, sans avoir à vérifier l'ensemble des résultats de la commande **xscmd -c showLinkedPrimaries**.

  Solution : utilisez l'option **xscmd -hc** ou **xscmd --linkHealthCheck**. La commande vérifie que les fragments primaires disposent du nombre approprié de liens de domaine de service de catalogue. La commande répertorie les fragments primaires dont le nombre de liens est erroné. S'ils sont tous liés correctement (par exemple, le domaine est lié à un autre domaine, tous les fragments primaires doivent avoir un lien), vous recevez un message indiquant qu'ils sont liés :

```
CWXSI0092I: Tous les fragments primaires pour la grille de données {0} et le groupe de mappes
{1} disposent du nombre de liens correct aux
fragments primaires externes.
```

  En cas de problèmes, essayez les solutions suivantes :

  – Vérifiez vos paramètres réseau et de pare-feu pour vous assurer que les serveurs qui hébergent les serveurs de conteneur dans les domaines peuvent communiquer entre eux.
  – Recherchez dans les fichiers journaux SystemOut et de l'outil de diagnostic de premier niveau des messages d'erreur plus spécifiques pour les fragments primaires avec des liens incorrects.
  – Supprimez et rétablissez le lien entre les domaines.

- **Problème :** données manquantes dans un ou plusieurs domaine de services de catalogue. Par exemple, vous pouvez exécuter la commande **xscmd -c establishLink**. Lorsque vous examinez les données de chaque domaine de service de catalogue lié, les données sont différentes par rapport à la commande **xscmd -c showMapSizes**.

  Solution : vous pouvez traiter ce problème avec la commande **xscmd -c showLinkedPrimaries**. Cette commande consigne chaque fragment primaire, y compris les fragments primaires externes qui sont liés.

  Dans le scénario décrit, vous pouvez constater en exécutant la commande **xscmd -c showLinkedPrimaries** que les fragments primaires du premier domaine de service de catalogue sont liés à ceux du second domaine de service de catalogue et que ce dernier n'a pas de liens au premier domaine de service de catalogue. Vous pouvez réexécuter la commande **xscmd -c establishLink** depuis le second domaine de service de catalogue vers le premier domaine de service de catalogue.

Traitement des problèmes des chargeurs

Java

Utilisez ces informations pour traiter les problèmes liés aux chargeurs de base de données.

Procédure

- **Problème :** Le chargeur ne parvient pas à communiquer avec la base de données. Une exception LoaderNotAvailableException se produit.

  Explication : Un échec du plug-in du chargeur est possible lorsque celui-ci ne parvient pas à communiquer avec le système dorsal de la base de données. Cette situation peut se produire si la connexion réseau ou le serveur de base de données est inactif. Le chargeur à écriture différée met les mises à jour en file d'attente et tente périodiquement d'insérer les modifications apportées aux données dans le chargeur. Ce dernier doit signaler le problème de connectivité à l'environnement d'exécution ObjectGrid en générant une exception LoaderNotAvailableException.

  Solution : L'implémentation du chargeur doit pouvoir distinguer entre une défaillance de base de données et une défaillance de chargeur physique. En cas d'échec lié aux données, une exception LoaderException ou OptimisticCollisionException doit être générée, alors qu'en cas de défaillance physique du chargeur, une exception LoaderNotAvailableException doit être générée. ObjectGrid gère ces deux exceptions de manières différentes :

  – Si une exception LoaderException est détectée par le chargeur à écriture différée, celui-ci considère l'exception comme un échec, par exemple si une clé en double a été identifiée. Le chargeur à écriture différée détaille la mise à jour et examine chaque enregistrement séparément pour isoler la raison de l'échec. Si une exception {{LoaderException}} est à nouveau détectée lors de la mise à jour de l'enregistrement concerné, un enregistrement d'échec de la mise à jour est créé et consigné dans la mappe des mises à jour ayant échoué.

  – Si une exception LoaderNotAvailableException est interceptée par le chargeur à écriture différée, celui-ci considère que l'échec est dû à l'impossibilité de se connecter à la base de données, par exemple, lorsque le système dorsal de la base de données est inactif, lorsque la connexion à une base de données est indisponible ou lorsque le réseau est inactif. Le chargeur à écriture différée attend 15 secondes, puis tente à nouveau la mise à jour par lots de la base de données.

  Une erreur fréquente consiste à générer une exception LoaderException alors qu'il doit s'agir d'une exception LoaderNotAvailableException. Dans ce cas, tous les enregistrements mis en file d'attente dans le chargeur à écriture différée deviennent des enregistrements de mise à jour ayant échoué, ce qui rend l'isolation des défaillances dans le système dorsal inutile.

- **Problème :** Lorsque vous utilisez un chargeur OpenJPA avec DB2 dans WebSphere Application Server, une exception de curseur fermé se produit.

  L'exception suivante provient de DB2 et figure dans le fichier journal org.apache.openjpa.persistence.PersistenceException :

  ```
  [jcc][t4][10120][10898][3.57.82] Invalid operation: result set is closed.
  ```

  Solution : par défaut, le serveur d'applications attribue à la propriété personnalisée resultSetHoldability la valeur 2 (CLOSE_CURSORS_AT_COMMIT). Cette propriété amène DB2 à fermer son resultSet/curseur au niveau des limites de la transaction. Pour supprimer l'exception, affectez à la propriété personnalisée, la valeur 1 (HOLD_CURSORS_OVER_COMMIT). Définissez la propriété personnalisée resultSetHoldability dans le chemin suivant dans la cellule WebSphere Application Server : **Ressources** > **Fournisseur JDBC** > **DB2 Universal JDBC Driver Provider** > **DataSources** > *data_source_name* > **Propriétés personnalisées** > **Nouveau**.

- **Problème** DB2 affiche une exception : `The current transaction has been rolled back because of a deadlock or timeout. Reason code "2".. SQLCODE=-911, SQLSTATE=40001, DRIVER=3.50.152`

  Cette exception se produit en raison d'un problème de conflit de verrouillage lorsque vous exécutez OpenJPA avec DB2 dans WebSphere Application Server. Le niveau d'isolement par défaut pour WebSphere Application Server est Lecture reproductible (RR), qui obtient des verrous de longue durée avec DB2.**Solution :**

  Définissez le niveau d'isolement Read Committed pour réduire les conflits de verrouillage. Définissez la propriété personnalisée de source de données webSphereDefaultIsolationLevel pour spécifier le niveau d'isolement 2(TRANSACTION_READ_COMMITTED) dans le chemin suivant dans la cellule WebSphere Application Server : **Ressources** > **Fournisseur JDBC** > ***JDBC_provider*** > **Sources de données** > ***data_source_name*** > **Propriétés personnalisées** > **Nouveau**. Pour plus d'informations sur la propriété personnalisée webSphereDefaultIsolationLevel et les niveaux d'isolement de transaction, voir Conditions de définition des niveaux d'isolement de l'accès aux données.
- **Problème :** Lorsque vous utilisez la fonction de préchargement de JPALoader ou JPAEntityLoader, le message CWOBJ1511 suivant ne s'affiche pas pour la partition dans un serveur de conteneur : `CWOBJ1511I: GRID_NAME:MAPSET_NAME:PARTITION_ID (primary) is open for business.`

  A la place, une exception TargetNotAvailableException est généré dans le serveur de conteneur qui active la partition définie par la propriété preloadPartition.

  Solution : Affectez à l'attribut preloadMode la valeur `true` si vous utilisez un chargeur JPALoader ou JPAEntityLoader pour précharger les données dans la mappe. Si la propriété preloadPartition du chargeur JPALoader et JPAEntityLoader a une valeur comprise entre `0` et `total_number_of_partitions - 1`, il tente de précharger les données à partir de la base de données dorsale dans la mappe. Le fragment de code ci-dessous illustre comment l'attribut preloadMode est défini pour activer le préchargement asynchrone :

  ```
  BackingMap bm = og.defineMap( "map1" );
  bm.setPreloadMode( true );
  ```

  Vous pouvez également définir l'attribut preloadMode à l'aide d'un fichier XML, comme le montre l'exemple suivant :

  ```
  <backingMap name="map1" preloadMode="true" pluginCollectionRef="map1"
   lockStrategy="OPTIMISTIC" />
  ```

Traitement des problèmes de configuration XML

Lorsque vous configurez eXtreme Scale, vous pouvez rencontrer un comportement inattendu avec vos fichiers XML. Les sections ci-après décrivent les problèmes qui peuvent se produire et les solutions.

Procédure

- **Problème :** votre stratégie de déploiement et les fichiers XML ObjectGrid doivent concorder.

  La règle de déploiement et les fichiers XML ObjectGrid doivent concorder. Si les noms ObjectGrid et les noms de la mappe ne concordent pas, des erreurs se produisent.

  Si la liste de la backingMap dans un fichier XML ObjectGrid ne correspond pas à la liste des références de la mappe d'un fichier XML de règle de déploiement, une erreur se produit sur le serveur de catalogue.

Par exemple, le fichier XML ObjectGrid et le fichier XML de la règle de déploiement ci-dessous permettent de démarrer un processus de conteneur. Le fichier de la règle de déploiement contient davantage de références de mappe que celles listées dans le fichier XML ObjectGrid.

ObjectGrid.xml - exemple incorrect

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="accounting">
            <backingMap name="payroll" readOnly="false" />
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

deploymentPolicy.xml - exemple incorrect

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
 xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
    <objectgridDeployment objectgridName="accounting">
        <mapSet name="mapSet1" numberOfPartitions="4" minSyncReplicas="1"
     maxSyncReplicas="2" maxAsyncReplicas="1">
            <map ref="payroll"/>
            <map ref="ledger"/>
        </mapSet>
    </objectgridDeployment>
</deploymentPolicy>
```

Messages : un message d'erreur apparaît dans le fichier `SystemOut.log` lorsque la stratégie de déploiement est incompatible avec le fichier XML ObjectGrid. Pour l'exemple précédent, ce message est le suivant :

```
CWOBJ3179E: La référence à la mappe Ledger dans le groupe de mappes mapSet1 du fichier
de descripteur de déploiement Accouting ne fait pas référence à une
mappe de sauvegarde valide dans le XML
de l'ObjectGrid.
```

S'il manque dans la règle de déploiement des références aux backingMaps répertoriées dans le fichier XML ObjectGrid, un message d'erreur est écrit dans le fichier `SystemOut.log`. Par exemple :

```
CWOBJ3178E: La mappe Ledger dans Accounting de l'ObjectGrid référencée dans le XML
de l'ObjectGrid est introuvable dans le fichier de descripteur du déploiement.
```

Solution : identifiez le fichier contient la liste correcte et modifiez le code approprié en conséquence.

- **Problème :** des noms ObjectGrid incorrects entre les fichiers XML génèrent également une erreur.

  Le nom de l'ObjectGrid est référencé à la fois dans le fichier XML ObjectGrid et le fichier XML de la règle de déploiement.

  Message : une exception ObjectGridException se produit causée par l'exception IncompatibleDeploymentPolicyException. Exemple :

  Causée par :
  com.ibm.websphere.objectgrid.IncompatibleDeploymentPolicyException :
  L'objectgridDeployment avec objectGridName "accountin" n'a pas d'objectGrid correspondant dans le XML ObjectGrid.

  Le fichier XML ObjectGrid est la liste principale des noms ObjectGrid. Si une règle de déploiement a un nom ObjectGrid qui n'est pas contenu dans le fichier XML ObjectGrid, une erreur se produit.

  Solution : vérifiez les détails tels que l'orthographe du nom ObjectGrid. Supprimez les noms redondants ou ajoutez les noms ObjectGrid manquants dans le fichier XML ObjectGrid ou le fichier XML de la règle de déploiement. Dans l'exemple de message, la valeur de l'objectGridName est mal orthographiée ("accountin" au lieu de "accounting").

- **Problème :** des attributs dans le fichier XML ne peuvent pas être affectés de certaines valeurs. Les valeurs acceptées par ces attributs sont énumérées par le schéma. La liste suivante indique certains de ces attributs :
  - Attribut authorizationMechanism sur l'élément objectGrid
  - Attribut copyMode sur l'élément backingMap
  - Attribut lockStrategy sur l'élément backingMap
  - Attribut ttlEvictorType sur l'élément backingMap
  - Attribut type sur l'élément property
  - initialState sur l'élément objectGrid
  - evictionTriggers sur l'élément backingMap

  Si une valeur non valide est attribuée à l'un de ces attributs, la validation XML échoue. Dans l'exemple suivant de fichier XML, une valeur incorrecte INVALID_COPY_MODE est utilisée :

  Exemple INVALID_COPY_MODE

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
   xmlns="http://ibm.com/ws/objectgrid/config">
      <objectGrids>
          <objectGrid name="accounting">
              <backingMap name="payroll" copyMode="INVALID_COPY_MODE"/>
          <objectGrid/>
     </objectGrids>
  </objectGridConfig>
  ```

  Le message suivant s'affiche dans le fichier journal.

  ```
  CWOBJ2403E : le fichier XML n'est pas valide. Un problème a été détecté
  avec < null > à la ligne 5. Le message d'erreur est cvc-enumeration-valid :
  Value 'INVALID_COPY_MODE' is not facet-valid with respect to enumeration
  '[COPY_ON_READ_AND_COMMIT, COPY_ON_READ, COPY_ON_WRITE, NO_COPY, COPY_TO_BYTES]'.
  La valeur doit être l'une des valeurs énumérées.
  ```

- **Problème :** des attributs ou des balises manquants ou incorrects dans le fichier XML provoquent des erreurs, telles que dans l'exemple suivant dans lequel le fichier XML ObjectGrid ne contient pas la balise de fin < /objectGrid > :

  attributs manquants- exemple XML

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
   xmlns="http://ibm.com/ws/objectgrid/config">
      <objectGrids>
          <objectGrid name="accounting">
              <backingMap name="payroll" />

     </objectGrids>
  </objectGridConfig>
  ```

  Message :

  ```
  CWOBJ2403E : le fichier XML n'est pas valide. Un problème a été détecté
  avec < null > à la ligne 7. Le message d'erreur est le suivant :
  la balise de fin du type d'élément "objectGrid" doit se terminer avec le
  délimiteur '>'.
  ```

  Une exception ObjectGridException concernant le fichier XML non valide se produit avec le nom du fichier XML.

  Solution : vérifiez que les balises et attributs nécessaires figurent dans les fichiers XML avec le format correct.

- **Problème :** si un fichier XML est formaté avec une syntaxe incorrecte ou manquante, le message CWOBJ2403E apparaît dans le journal. Par exemple, le message suivant s'affiche lorsqu'un guillemet manque pour l'un des attributs XML.

  ```
  CWOBJ2403E : le fichier XML n'est pas valide.
  Un problème a été détecté avec < null > à la ligne 7.
  Le message d'erreur est le suivant : un guillemet ouvrant est attendu pour
  l'attribut "maxSyncReplicas" associé à un type d'élément "mapSet".
  ```

Une exception ObjectGridException concernant le fichier XML non valide se produit également.

Solution : il existe différentes solutions pour une erreur de syntaxe XML. Consultez la documentation appropriée sur l'écriture de script XML.

- **Problème :** le référencement d'une collection de plug-in inexistante rend un fichier XML non valide. Par exemple, lorsque vous utilisez XML pour définir des plug-in BackingMap, l'attribut pluginCollectionRef de l'élément backingMap doit faire référence à une collection backingMapPluginCollection. L'attribut pluginCollectionRef doit correspondre aux éléments backingMapPluginCollection.

  Message :

  Si l'attribut pluginCollectionRef ne correspond à aucun attribut d'ID de l'un des éléments backingMapPluginConfiguration, le message suivant ou un message similaire s'affiche dans le fichier journal.

  ```
  [7/14/05 14:02:01:971 CDT] 686c060e XmlErrorHandl E CWOBJ9002E :
  Message informatif en anglais uniquement : fichier XML non valide.
  Ligne : 14 ; URI :
  null ; Message : la clé 'pluginCollectionRef' avec
  la valeur 'bookPlugins' est introuvable pour la contrainte d'identité de
  l'élément 'objectGridConfig'.
  ```

  Le fichier XML suivant est utilisé pour produire l'erreur. Notez que l'attribut pluginCollectionRef de la BackingMap book est défini sur bookPlugins et que l'ID de la backingMapPluginCollection est collection1.

  Référencement d'un attribut XML inexistant - Exemple

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ibm.com/ws/objectgrid/config../objectGrid.xsd"
   xmlns="http://ibm.com/ws/objectgrid/config">
     <objectGrids>
        <objectGrid name="bookstore">
           <backingMap name="book" pluginCollectionRef="bookPlugin" />
        </objectGrid>
     </objectGrids>
     <backingMapPluginCollections>
      <backingMapPluginCollection id="collection1">
       <bean id="Evictor"
         className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" />
      </backingMapPluginCollection>
     </backingMapPluginCollections>
  </objectGridConfig>
  ```

  Solution :

  Pour résoudre le problème, assurez-vous que la valeur de chaque pluginCollectionRef correspond à l'ID de l'un des éléments de la backingMapPluginCollection. Modifiez simplement le nom de pluginCollectionRef en collection1 pour ne pas recevoir cette erreur. Une autre solution consiste à modifier l'ID de la backingMapPluginCollection existante pour le faire correspondre à pluginCollectionRef, ou d'ajouter une backingMapPluginCollection ayant un ID correspond à pluginCollectionRef.

- **Problème :** IBM Software Development Kit (SDK) Version 5 contient une implémentation de la fonction Java API for XML Processing (JAXP) à utiliser pour la validation XML par rapport à un schéma. Lorsque vous utilisez un kit de développement de logiciels ne contenant pas cette implémentation, les tentatives de validation risquent d'échouer.

  Lorsque vous tentez de valider XML avec un kit de développement de logiciels n'ayant pas l'implémentation nécessaire, le fichier journal contient l'erreur suivante :

  ```
  La validation XML XmlConfigBuild est activée
  SystemErr R com.ibm.websphere.objectgrid
  SystemErr R at com.ibm.ws.objectgrid.ObjectGridManagerImpl.getObjectGridConfigurations
   (ObjectGridManagerImpl.java:182)
  SystemErr R at com.ibm.ws.objectgrid.ObjectGridManagerImpl.createObjectGrid(ObjectGridManagerImpl.java:309)
  SystemErr R at com.ibm.ws.objectgrid.test.config.DocTest.main(DocTest.java:128)
  SystemErr R Caused by: java.lang.IllegalArgumentException : aucun attribut implémenté
  ```

```
SystemErr R at org.apache.crimson.jaxp.DocumentBuilderFactoryImpl.setAttribute(DocumentBuilderFactoryImpl.java:93)
SystemErr R at com.ibm.ws.objectgrid.config.XmlConfigBuilder.<init>XmlConfigBuilder.java:133)
SystemErr R at com.ibm.websphere.objectgrid.ProcessConfigXML$2.runProcessConfigXML.java:99)...
```

Le kit de développement de logiciels utilisé ne contient pas l'implémentation de la fonction JAXP nécessaire pour valider les fichiers XML en fonction d'un schéma.

Solution : si vous souhaitez valider XML en utilisant un kit SDK qui ne contient pas l'implémentation JAXP, téléchargez Apache Xerces, et incluez ses fichiers JAR (Java archive) dans le chemin d'accès aux classes. Pour éviter ce problème, après avoir téléchargé Xerces et inclus les fichiers JAR dans le chemin d'accès aux classes, vous pouvez valider le fichier XML.

Traitement des incidents liés aux exceptions de dépassement de délai d'attente pour une transaction multipartition

Java

Le scénario décrit est un exemple de transaction multipartition qui est à l'origine d'une exception de dépassement de délai de verrouillage. Selon l'état de la transaction, les solutions montrent la façon dont vous pouvez manuellement résoudre ce problème.

Avant de commencer

Implémentez la gestion des exceptions dans l'application. Pour plus d'informations, voir Implémentation de gestion des exceptions dans les scénarios de verrouillage.

L'exception suivante s'affiche :

```
Caused by: com.ibm.websphere.objectgrid.LockTimeoutException:
Local-40000139-DEF8-05EA-E000-64A856931719 timed out waiting for
lock mode S to be granted for map name: TS2_MapP, key: key12
granted = X
lock request queue
->[WXS-40000139-DEF6-FA84-E000-1CB456931719, state = Granted,
requested 73423 milli-seconds ago, marked to keep current mode false,
snapshot mode 0, mode = X, thread name = xIOReplicationWorkerThreadPool : 29]
->[Local-40000139-DEF8-05EA-E000-64A856931719, state = Waiting for
5000 milli-seconds, marked to keep current mode false, snapshot mode 0,
mode = S, thread name = xIOWorkerThreadPool : 28]
dump of all locks for WXS-40000139-DEF6-FA84-E000-1CB456931719
Key: key12, map: TS2_MapP
strongest currently granted mode for key is X
->[WXS-40000139-DEF6-FA84-E000-1CB456931719, state = Granted,
requested 73423 milli-seconds ago, marked to keep current mode false,
snapshot mode 0, mode = X, thread name = xIOReplicationWorkerThreadPool : 29]
dump of all locks for Local-40000139-DEF8-05EA-E000-64A856931719
```

Ce message représente la chaîne transmise en tant que paramètre lorsque l'exception est créée et émise.

Procédure

Problème : vous obtenez une exception de dépassement de délai de verrouillage et le détenteur du verrou est une transaction multipartition, ou le dossier du journal augmente avec les messages de journal.
Diagnostic :
un message apparaît de manière répétée jusqu'à remplir le dossier des journaux, par exemple :

```
00000099 TransactionLog I  CWOBJ8705I: Automatic resolution
of transaction WXS-40000139-DF01-216D-E002-1CB456931719 at RM:TestGrid:TestSet2:20
is still waiting for a decision.
Another attempt to resolve the transaction will occur in 30 seconds.
```

Identifiez le type de transaction à l'origine du verrou. Si le préfixe dans l'identificateur de transaction est WXS-, cela indique qu'il s'agit d'une transaction multipartition. Si le préfixe dans l'identificateur de transaction est Local-, cela indique qu'il s'agit d'une transaction à une partition.
Cause : il est fort probable que l'application détienne le verrou, car aucune validation ou annulation n'a eu lieu.
Solution : déterminez l'état de la transaction et la durée de l'état. Utilisez l'utilitaire de commande **xscmd -c listindoubts** avec l'option -d (pour une sortie détaillée) ou le bean géré de la transaction.

Résolution des exceptions de délai d'attente de verrouillage

Java

En utilisant la commande **xscmd -c listindoubt**, vous pouvez afficher l'état d'une transaction et déterminer l'action à exécuter.

Résolution des exceptions de délai d'attente de verrouillage à l'aide de la commande xscmd -c listindoubts

Procédure

- Affichez la liste détaillée des transactions dans votre environnement : xscmd -c listindoubt -d La commande peut retourner l'un des états suivants :
  - Toutes les transactions validées
  - Préparé
  - Un gestionnaire de transactions (TM) manquant
- Exécutez les actions appropriées pour résoudre la transaction. **Problème :** toutes les transactions validées

```
[1] WXS-40000139-DEF8-EF60-E002-1CB456931719
Timestamp               Partition    Role  State     Container   Resync  Attempts
------------------------------------------------- --------------------------------
2012-09-19 10:40:19.824  TestSet1:11  TM  COMMITT  MPTBasic2_C-0  Primary  0
2012-09-19 10:40:19.824  TestSet1:7  RM  PREPARED  MPTBasic0_C-1  Primary  0
2012-09-19 10:40:19.839  TestSet2:20  RM  PREPARED  MPTBasic2_C-0  Primary  0
2012-09-19 10:40:19.824  TestSet2:6    RM   PREPARED  MPTBasic0_C-1  Primary  0
```

Solution : valider les partitions de gestionnaire de ressources (RM) et ignorez la transaction.

1. Exécutez la commande suivante pour valider la partition RM dans la transaction WXS-40000139-DEF8-EF60-E002-1CB456931719: xscmd -c listIndoubts -xid WXS-40000139-DEF8-EF60-E002-1CB456931719 -cm -rm
2. Exécutez la commande suivante pour ignorer cette transaction : xscmd -c listIndoubts -xid WXS-40000139-DEF8-EF60-E002-1CB456931719 -f

Problème : transactions préparées

```
[1] WXS-40000139-DEF6-FA84-E000-1CB456931719

Timestamp               Partition    Role   State   Container     Resync Attempts
------------------------------------------------------------------ --------------
2012-09-19 10:38:11.603  TestSet1:10  RM  PREPARED  MPTBasic2_C-0  Primary  0
2012-09-19 10:38:11.588  TestSet1:5  TM  PREPARED  MPTBasic2_C-0  Primary  0
2012-09-19 10:38:11.603  TestSet2:11  RM  PREPARED  MPTBasic2_C-0  Primary  0
2012-09-19 10:38:11.619  TestSet2:13  RM  PREPARED  MPTBasic2_C-0  Primary  0
```

Solution : annulez la partition TM et les partitions RM suivantes. Ensuite ignorez la transaction.

1. Exécutez la commande suivante pour annuler la partition TM dans la transaction WXS-40000139-DEF6-FA84-E000-1CB456931719: `xscmd -c listIndoubts -xid WXS-40000139-DEF6-FA84-E000-1CB456931719 -r -tm`
2. Exécutez la commande suivante pour annuler les partitions RM dans cette transaction : `xscmd -c listIndoubts -xid WXS-40000139-DEF6-FA84-E000-1CB456931719 -r -rm`
3. Exécutez la commande suivante pour ignorer cette transaction : `xscmd -c listIndoubts -xid WXS-40000139-DEF6-FA84-E000-1CB456931719 -f`

Problème : gestionnaire de transactions manquant

```
[1] WXS-40000139-DEF8-EF31-E000-1CB456931719

Timestamp               Partition    Role   State   Container     Resync Attempts
------------------------------------------------------------------ --------------
2012-09-19 10:40:19.777  TestSet1:11  RM  PREPARED  MPTBasic2_C-0  Primary  0
2012-09-19 10:40:19.792  TestSet2:5  RM  PREPARED  MPTBasic2_C-0  Primary  0
2012-09-19 10:40:19.777  TestSet2:6    RM  PREPARED  MPTBasic2_C-1  Primary  0
```

Solution : annulez les partitions RM.

– Exécutez la commande suivante pour annuler les partitions RM dans la transaction WXS-40000139-DEF8-EF31-E000-1CB456931719 : `xscmd -c listIndoubts -xid WXS-40000139-DEF8-EF31-E000-1CB456931719 -r`

Traitement des problèmes de sécurité

Utilisez ces informations pour traiter les problèmes de configuration de sécurité.

Procédure

- **Problème :** l'extrémité client de la connexion nécessite SSL (Secure Sockets Layer), avec le paramètre transportType affecté de la valeur `SSL-Required`. Toutefois, l'extrémité serveur de la connexion ne prend pas en charge SSL et le paramètre transportType a la valeur `TCP/IP`. En conséquence, l'exception suivante est chaînée à une autre exception dans les fichiers journaux :

```
java.net.ConnectException: connect: Address is invalid on local machine, or
port is not valid on remote machine
    at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:389)
    at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:250)
    at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:237)
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:385)
    at java.net.Socket.connect(Socket.java:540)
    at
com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:155)
    at
com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:167)
```

  L'adresse dans cette exception peut être un serveur de catalogue, un serveur de conteneur ou un client.

  Solution: voir «Configuration des types de transports sécurisés», à la page 633 pour le tableau des configurations de sécurité valides entre les clients et les serveurs.

- Lorsqu'un agent est utilisé, le client envoie l'appel de l'agent au serveur et le serveur envoie la réponse au client pour accuser réception de l'appel de l'agent. Lorsque l'agent a terminé le traitement, le serveur établit une connexion pour envoyer les résultats de l'agent. Ainsi, le serveur de conteneur devient un client du point de vue de la connexion. Par conséquent, TLS ou SSL est configuré, veillez à importer le certificat public client dans le fichier de clés certifiées du serveur.
- **Problème :** lorsque les utilisateurs sont autorisés à accéder à une grille de données WebSphere eXtreme Scale, ils peuvent être également autorisés à

exécuter des opérations de gestion en utilisant la commande **xscmd** ou **stopOgServer**. La plupart des déployeurs de grilles de données restreignent l'accès administrateur à un sous-ensemble des utilisateurs qui peuvent accéder aux données de la grille.

Si vous utilisez la commande suivante pour accéder à la grille de données, vous pouvez être également autorisé à exécuter des actions d'administration, telles qu'exécuter la commande listAllJMXAddresses :

```
./xscmd.sh -user <user> -password <password> <other_parameters>
```

Si cette opération fonctionne pour cet utilisateur, une opération **xscmd** peut également être exécutée par le même utilisateur.

Solution : lorsque les composants eXtreme Scale sont exécutés avec WebSphere Application Server, utilisez la console d'administration WebSphere Application Server pour activer le gestionnaire de sécurité. Cliquez sur **Sécurité** > **Sécurité globale** et cochez les cases **Activer la sécurité administrative** et **Utiliser la sécurité Java 2** pour limiter l'accès des applications aux ressources locales.

L'accès aux opérations de gestion est contrôlé par le gestionnaire de sécurité WebSphere Application Server et il est accordé uniquement aux utilisateurs qui appartiennent au rôle WebSphere Administrateur. Vous devez exécuter la commande **xscmd** depuis le répertoire WebSphere Application Server.

Lorsque les composants eXtreme Scale s'exécutent dans un environnement autonome, des étapes supplémentaires sont requises pour implémenter la sécurité d'administration. Vous devez exécuter les serveurs de catalogue et les serveurs de conteneur en utilisant le gestionnaire de sécurité Java, ce qui nécessite un fichier de règles.

Le fichier de règles se présente comme suit :

A faire : Il existe généralement des entrées MapPermission, comme décrit dans «Tutoriel sur la sécurité Java SE - Etape 5», à la page 106.

```
grant codeBase "file:${objectgrid.home}/lib/*" {
permission java.security.AllPermission;
};

grant principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=OGSample" {
permission javax.management.MBeanPermission "*", "getAttribute,setAttribute,invoke,queryNames";
};
```

Dans cet exemple, seul le gestionnaire principal est autorisé à exécuter des opérations d'administration à l'aide de la commande **xscmd**. D'autres lignes peuvent être ajoutées en fonction des besoins pour accorder des autorisations supplémentaires de bean géré aux principaux. Un autre type de principal est nécessaire si vous utilisez l'authentification LDAP.

Entrez la commande suivante : UNIX Linux

```
startOgServer.sh <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config -Djava.security.manager
-Djava.security.policy="auth.policy" -Dobjectgrid.home=$OBJECTGRID_HOME
```

UNIX Linux 8.6+

```
startXsServer.sh <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config -Djava.security.manager
-Djava.security.policy="auth.policy" -Dobjectgrid.home=$OBJECTGRID_HOME
```

Windows

```
startOGServer.bat <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config -Djava.security.manager
-Djava.security.policy="auth.policy" -Dobjectgrid.home=%OBJCTGRID_HOME%
```

Windows 8.6+

```
startXsServer.bat <arguments> -jvmargs -Djava.security.auth.login.config=jaas.config -Djava.security.manager
-Djava.security.policy="auth.policy" -Dobjectgrid.home=%OBJCTGRID_HOME%
```

You specify -Djava.security.policy in this case, instead of -Djava.security.auth.policy.

Collecte des données avec IBM Support Assistant Data Collector

Exécutez IBM Support Assistant Data Collector pour collecter les données de traitement des problèmes dans l'environnement WebSphere eXtreme Scale. A l'aide de cet outil, vous pouvez accélérer la reproduction d'un problème avec les niveaux de trace RAS appropriés définis et réduire les tâches d'envoi d'informations de journal appropriées à IBM Support.

Avant de commencer

Avant d'exécuter l'outil, vous devez disposer des informations de configuration système suivantes prêtes pour les fournir à l'outil :

- Nom de fichier pour enregistrer les données collectées
- Répertoire *rép_base_java*
- Répertoire *rép_base_wxs*
- Répertoire de travail utilisé par WebSphere eXtreme Scale
- Emplacement des fichiers script supplémentaires utilisés pour démarrer les serveurs

Pourquoi et quand exécuter cette tâche

Dans les versions précédentes de WebSphere eXtreme Scale, l'outil IBM Support Assistant Lite était utilisé pour collecter les journaux pour le traitement des problèmes. L'outil IBM Support Assistant Lite est toujours fourni avec le produit dans le répertoire *rép_base_wxs*/isalite_wxs. IBM Support Assistant Data Collector est un outil plus interactif qui s'installe avec la version 8.6 et les versions suivantes. IBM Support Assistant Data Collector facilite la collecte des données en enregistrant les entrées, ce qui réduit la saisie répétitive lors de l'entrées dans la console. Pour plus d'informations, voir IBM Support Assistant Data Collector.

Procédure

1. Lancez l'outil. L'outil s'exécute en mode console via le démarrage du script de lancement à partir de la ligne de commande. Le script de l'outil est installé dans le répertoire *rép_base_wxs*/isalite_dc.
   - Windows **isadc.bat**
   - Linux UNIX **isadc.sh**
2. Fournissez les informations système à l'outil. A chaque étape, les choix sont présentés sous forme de listes numérotées ; vous entrez le numéro de votre sélection et vous appuyez sur la touche Entrée. Lorsque des données sont requises en entrée, vous êtes invité à entrer votre réponse avant d'appuyer sur la touche Entrée. Vous pouvez trouver les informations de collecte pour chaque type d'incident dans les documents MustGather correspondants. Vous pouvez également indiquer le nom du fichier compressé et l'emplacement du répertoire dans lequel vous souhaitez sauvegarder les informations.
3. Arrêtez l'outil de collecte en tapant **quit** en mode Console.

Résultats

Les informations environnementales ci-après sont regroupées dans un fichier compressé que vous avez nommé pour sauvegarder les données :

- Fichiers journaux
- Informations de version eXtreme Scale
- Informations de version Java
- Informations sur la structure de répertoires *rép_base_wxs*, ainsi que les fichiers stockés dans divers répertoires. Les fichiers en tant que tels ne sont pas enregistrés dans le fichier compressé.
- Scripts du répertoire `bin`.

Que faire ensuite

Contactez IBM et fournissez le fichier compressé généré avec IBM Support Assistant Data Collector. Pour plus d'informations, voir «Comment prendre contact avec le service de support IBM», à la page 661.

IBM Support Assistant for WebSphere eXtreme Scale

IBM Support Assistant permet de collecter des données, d'analyser des symptômes et d'accéder à des informations sur les produits.

IBM Support Assistant Lite

IBM Support Assistant Lite for WebSphere eXtreme Scale assure une collecte automatique des données et l'analyse des symptômes pour l'identification des problèmes et de leurs causes.

IBM Support Assistant Lite réduit le temps consacré à la reproduction des problèmes en adaptant son ensemble de niveaux de traçabilité (niveaux de fiabilité, de disponibilité et de facilité de maintenance, qui sont définis automatiquement par l'outil) afin de simplifier l'identification des problèmes. Mais si cette assistance ne suffit pas et que vous ayez besoin de l'aide d'un technicien, IBM Support Assistant Lite réduit également le temps consacré à l'envoi des informations appropriées au support technique d'IBM.

IBM Support Assistant Lite est inclus dans chaque installation de WebSphere eXtreme Scale version 7.1.0

IBM Support Assistant

IBM Support Assistant (ISA) permet d'accéder rapidement à des ressources de produits, de formation et de support qui pourront vous aider à répondre de vous-mêmes à vos questions et à résoudre les problèmes rencontrés avec des logiciels IBM sans avoir besoin de contacter le support IBM. Différents plug-in spécifiques à différents produits vous permettent de personnaliser IBM Support Assistant en fonction des produits particuliers que vous avez installés. IBM Support Assistant peut également collecter des données système, des fichiers journaux et d'autres informations qui aideront le support technique d'IBM à déterminer la cause des problèmes.

IBM Support Assistant est un utilitaire qui s'installe sur le poste de travail et non sur le serveur WebSphere eXtreme Scale lui-même. En effet, sa mémoire et ses besoins en ressources risqueraient d'affecter de manière négative les performances du serveur WebSphere eXtreme Scale. Les composants portables de diagnostics qui sont inclus dans l'Assistant sont conçus pour avoir un impact minimal sur le fonctionnement normal d'un serveur.

IBM Support Assistant est utilisable des manières suivantes :

- pour effectuer des recherches dans des sources IBM et non IBM de connaissances et d'information sur plusieurs produits IBM afin de répondre à une question ou de résoudre un problème
- pour trouver des informations complémentaires dans des ressources Web dédiées à un produit donné (pages d'accueil du produit et de son support, groupes de discussions et forums d'utilisateurs, ressources d'acquisition de compétences et de formations, informations de résolution des problèmes et FAQ)
- pour renforcer vos capacités à diagnostiquer les problèmes d'un produit donné grâce aux outils de diagnostics ciblés proposés par l'Assistant
- pour simplifier la collecte des données de diagnostic afin de vous aider, IBM et vous, à résoudre vos problèmes (collecte de données générales ou liées à un symptôme particulier)
- pour vous aider à signaler des problèmes au support IBM via une interface personnalisée en ligne avec possibilité d'attacher aux incidents signalés les données de diagnostic mentionnées plus haut ou toute autre information

Enfin, la fonctionnalité Updater intégrée permet de mettre à jour le support pour d'autres produits logiciels et d'autres fonctionnalités au fur et à mesure de leur disponibilité. Pour configurer IBM Support Assistant afin de l'utiliser avec WebSphere eXtreme Scale, commencez par l'installer à l'aide des fichiers fournis dans l'image téléchargée à partir de la page Web IBM Support Overview (http://www-947.ibm.com/support/entry/portal/Overview/Software/Other_Software/IBM_Support_Assistant. Ensuite, utilisez IBM Support Assistant pour repérer et installer les mises à jour de produits qui vous intéressent. Vous pouvez également choisir d'installer des plug-in pour d'autres logiciels IBM de votre environnement. Vous trouverez des informations complémentaires et la dernière version d'IBM Support Assistant à la page Web IBM Support Assistant (http://www.ibm.com/software/support/isa/).

Remarques

Les références aux produits, logiciels et services d'IBM n'impliquent pas qu'ils soient distribués dans tous les pays dans lesquels IBM exerce son activité. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. L'évaluation et la vérification de son fonctionnement en conjonction avec d'autres produits, hormis ceux expressément désignés par IBM, relèvent de la responsabilité de l'utilisateur.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594 USA

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Requests

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Marques

IBM, le logo IBM et ibm.com sont marques d'International Business Machines Corp., dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web “Copyright and trademark information” à l'adresse www.ibm.com/legal/copytrade.shtml.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Oracle Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Index

Caractères spéciaux

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

U

V

W

X

Z