IBM WebSphere eXtreme Scale Version 7.1.1
Version 7 Release 1

Verwaltung

Diese Ausgabe bezieht sich auf Version 7, Release 1, Modifikation 1 von WebSphere eXtreme Scale und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellen

Informationen zur Veröffentlichung *Verwaltung*

Der Dokumentationssatz zu WebSphere eXtreme Scale umfasst drei Handbücher, die die erforderlichen Informationen zur Verwendung des Produkts WebSphere eXtreme Scale, zur Programmierung für das Produkt und zur Verwaltung des Produkts enthalten.

Bibliothek von WebSphere eXtreme Scale

Die Bibliothek von WebSphere eXtreme Scale enthält die folgenden Bücher:

- Die Veröffentlichung *Produktübersicht* enthält eine Übersicht über die Konzepte von WebSphere eXtreme Scale, einschließlich Anwendungsfallszenarien und Lernprogrammen.
- Im *Installationshandbuch* wird beschrieben, wie Sie allgemeine Topologien von WebSphere eXtreme Scale installieren.
- Die Veröffentlichung *Verwaltung* enthält die für Systemadministratoren erforderlichen Informationen, z. B. Planung von Anwendungsimplementierungen, Kapazitätsplanung, Installation und Konfiguration des Produkts, Starten und Stoppen von Servern, Überwachung der Umgebung und Sicherung der Umgebung.
- Die Veröffentlichung *Programmierung* enthält Informationen für Anwendungsentwickler zur Entwicklung von Anwendungen für WebSphere eXtreme Scale unter Verwendung der bereitgestellten API-Informationen.

Zum Herunterladen der Handbücher rufen Sie die Bibliotheksseite von WebSphere eXtreme Scale auf.

Sie finden die in dieser Bibliothek enthaltenen Informationen auch im Information Center von WebSphere eXtreme Scale Version 7.1.1.

Veröffentlichungen offline verwenden

Alle Veröffentlichungen in der Bibliothek von WebSphere eXtreme Scale enthalten Links zum Information Center mit dem folgenden Stamm-URL: `http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r1m1`. Diese Links führen Sie direkt zu den zugehörigen Informationen. Wenn Sie jedoch offline arbeiten und auf einen dieser Links klicken, können Sie den Titel des Links in den anderen Veröffentlichungen in der Bibliothek suchen. Die API-Dokumentation, das Glossar und die Nachrichtenreferenzen sind in den PDF-Veröffentlichungen nicht verfügbar.

Zielgruppe

Dieses Handbuch ist hauptsächlich für Systemadministratoren, Sicherheitsadministratoren und Systembediener bestimmt.

Aktualisierungen für dieses Handbuch

Sie erhalten Aktualisierungen zu diesem Handbuch, indem Sie die jeweils aktuelle Version des Handbuchs von der Bibliotheksseite von WebSphere eXtreme Scale herunterladen.

Hinweise zu Rückmeldungen

Wenden Sie sich an das Dokumentationsteam. Haben Sie die benötigten Informationen gefunden? Sind die Informationen präzise und vollständig? Senden Sie Ihre Kommentare zu dieser Dokumentation per E-Mail an wasdoc@us.ibm.com.

Kapitel 1. Einführung

Nach der Installation des Produkts können Sie das Einführungsbeispiel verwenden, um die Installation zu testen und das Produkt zum ersten Mal zu verwenden.

Lernprogramm: Einführung in WebSphere eXtreme Scale

Nach der Installation von WebSphere eXtreme Scale in einer eigenständigen Umgebung können Sie die Einführungsbeispielanwendung als einfache Einführung in die Funktionalität des Produkts als speicherinternes Datengrid verwenden.

Lernziele

- Vertrautmachen mit der ObjectGrid-XML-Deskriptordatei und den XML-Deskriptordateien für die Implementierungsrichtlinie, die Sie zum Konfigurieren Ihrer Umgebung verwenden
- Katalog- und Container-Server mit den Konfigurationsdateien starten
- Vertrautmachen mit der Entwicklung einer Clientanwendung
- Ausführung der Clientanwendung zum Einfügen von Daten in das Datengrid
- Überwachung der Datengrids mit der Webkonsole

Erforderliche Zeit

60 Minuten

Lerneinheit 1 des Lernprogramms "Einführung": Datengrids mit Konfigurationsdateien definieren

Zum Konfigurieren einfacher Datengrids verwenden Sie die Dateien `objectgrid.xml` und `deployment.xml`, die in der einführenden Beispielanwendung (gettingstarted) bereitgestellt werden.

Das Beispiel verwendet die Dateien `objectgrid.xml` und `deployment.xml`, die im Verzeichnis *WXS-Installationsstammverzeichnis*`/ObjectGrid/gettingstarted/xml` enthalten sind. Diese Dateien werden an die Startbefehle übergeben, mit denen Container-Server und ein Katalogserver gestartet werden. Die Datei `objectgrid.xml` ist die ObjectGrid-XML-Deskriptordatei. Die Datei `deployment.xml` ist die XML-Deskriptordatei für die ObjectGrid-Implementierungsrichtlinie. Diese Dateien definieren eine verteilte Topologie.

ObjectGrid-XML-Deskriptordatei

Eine ObjectGrid-XML-Deskriptordatei wird verwendet, um die Struktur des ObjectGrids definieren, das von der Anwendung verwendet wird. Sie enthält eine Liste mit BackingMap-Konfigurationen. Diese BackingMaps speichern die Cachedaten. Im Folgenden sehen Sie eine Beispieldatei `objectgrid.xml`. Die ersten Zeilen der Datei enthalten den erforderlichen Header für jede ObjectGrid-XML-Datei. Diese Beispieldatei definiert das ObjectGrid `Grid` mit den BackingMaps `Map1` und `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">
            <backingMap name="Map1" />
            <backingMap name="Map2" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

XML-Deskriptordatei für Implementierungsrichtlinie

Während des Starts wird eine XML-Deskriptordatei für die Implementierungsrichtlinie an den Container-Server übergeben. Eine Implementierungsrichtlinie muss zusammen mit einer ObjectGrid-XML-Datei verwendet werden und mit der ObjectGrid-XML kompatibel sein, mit der sie verwendet wird. Für jedes objectgridDeployment-Element in der Implementierungsrichtlinie muss ein entsprechendes ObjectGrid-Element in der ObjectGrid-XML-Datei vorhanden sein. Die backingMap-Elemente, die im objectgridDeployment-Element definiert werden, müssen mit den backingMap-Elementen in der ObjectGrid-XML konsistent sein. Jedes backingMap-Element darf nur in einem einzigen mapSet-Element referenziert werden.

Die XML-Deskriptordatei für die Implementierungsrichtlinie ist für die Verwendung mit der entsprechenden ObjectGrid-XML-Datei `objectgrid.xml` bestimmt. Im folgenden Beispiel enthalten die ersten Zeilen der Datei `deployment.xml` den erforderlichen Header für jede XML-Deskriptordatei für die Implementierungsrichtlinie. Die Datei definiert das Element "objectgridDeployment" für das Grid "ObjectGrid", das in der Datei `objectgrid.xml` definiert ist. Beide im Grid "ObjectGrid" definierten BackingMaps, Map1 und Map2, sind im MapSet "mapSet" enthalten, in dem die Attribute "numberOfPartitions", "minSyncReplicas" und "maxSyncReplicas" konfiguriert sind.

```
<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>
```

Mit dem Attribut "numberOfPartitions" des Elements "mapSet" wird die Anzahl der Partitionen für das MapSet angegeben. Es ist ein optionales Attribut und hat standardmäßig den Wert 1. Die Anzahl der Partitionen muss der geplanten Datengridkapazität angemessen sein.

Mit dem Attribut "minSyncReplicas" des MapSets wird die Mindestanzahl synchroner Replikate für jede Partition im MapSet angegeben. Es ist ein optionales Attribut und hat standardmäßig den Wert 0. Es werden erst dann primäre Shards und Replikat-Shards verteilt, wenn die Domäne in der Lage ist, die Mindestanzahl synchroner Replikate zu unterstützen. Für die Unterstützung des minSyncReplicas-Werts benötigen Sie einen Container mehr, als der minSyncReplicas-Wert vorgibt. Wenn die Anzahl synchroner Replikate unter den Wert von minSyncReplicas fällt, werden keine Schreiboperationen für diese Partition mehr zugelassen.

Mit dem Attribut "maxSyncReplicas" des MapSets wird die maximale Anzahl synchroner Replikate für jede Partition im MapSet angegeben. Es ist ein optionales Attribut und hat standardmäßig den Wert 0. Es werden keine weiteren synchronen Replikate für eine Partition verteilt, wenn eine Domäne diese Anzahl synchroner Replikate für diese bestimmte Partition erreicht. Das Hinzufügen von Containern, die dieses ObjectGrid unterstützen, kann zu einer höheren Anzahl synchroner Replikate führen, wenn der maxSyncReplicas-Wert noch nicht erreicht ist. Das Beispiel setzt maxSyncReplicas auf 1, d. h., die Domäne verteilt maximal ein synchrones Replikat. Wenn Sie mehrere Container-Server-Instanzen starten, wird nur ein einziges synchrones Replikat in einer der Container-Server-Instanzen verwendet.

Prüfpunkt der Lerneinheit

In dieser Lerneinheit haben Sie Folgendes gelernt:

- Maps definieren, die die Daten in der ObjectGrid-XML-Deskriptordatei speichern
- XML-Implementierungsdeskriptordatei für die Definition der Partitions- und Replikatanzahl für das Datengrid verwenden

Lerneinheit 2 des Lernprogramms "Einführung": Clientanwendung erstellen

Wenn Sie Daten in Ihrem Datengrid einfügen, löschen, aktualisieren und abrufen möchten, müssen Sie eine Clientanwendung schreiben. Das Einführungsbeispiel (gettingstarted) enthält eine Clientanwendung, die Sie verwenden können, um sich mit der Erstellung einer eigenen Clientanwendung vertraut zu machen.

Die Datei `Client.java` im Verzeichnis *WXS-Installationsstammverzeichnis*`/ObjectGrid/gettingstarted/client/src/` ist das Clientprogramm, das veranschaulicht, wie die Verbindung zu einem Katalogserver hergestellt, die ObjectGrid-Instanz abgerufen und die API ObjectMap verwendet wird. Die API ObjectMap speichert Daten in Form von Schlüssel/Wert-Paaren und eignet sich ideal für das Caching von Objekten ohne Beziehungen.

Wenn Sie Objekte zwischenspeichern müssen, die Beziehungen haben, verwenden Sie die API EntityManager.

1. Verbindung zum Katalogservice durch Anfordern einer ClientClusterContext-Instanz herstellen.

   Verwenden Sie zum Herstellen einer Verbindung zum Katalogserver die Methode "connect" der API "ObjectGridManager". Mit der Methode connect, die verwendet wird, muss nur der Katalogserverendpunkt im Format *`Hostname:Port`* angegeben werden. Sie können mehrere Katalogserverendpunkte angeben, indem Sie die Liste der *`Hostname:Port`*-Werte durch Kommas voneinander trennen. Das folgende Code-Snippet veranschaulicht, wie die Verbindung zu einem Katalogserver hergestellt und eine ClientClusterContext-Instanz angefordert wird:

   ```
   ClientClusterContext ccc = ObjectGridManagerFactory.getObjectGridManager().connect("localhost:2809", null, null);
   ```

   Wenn die Verbindungen zu den Katalogserver erfolgreich hergestellt werden, gibt die Methode connect eine ClientClusterContext-Instanz zurück. Die ClientClusterContext-Instanz ist erforderlich, um das ObjectGrid von der API "ObjectGridManager" abzurufen.

2. ObjectGrid-Instanz anfordern.

   Zum Anfordern einer ObjectGrid-Instanz verwenden Sie die Methode "getObjectGrid" der API "ObjectGridManager". Die Methode "getObjectGrid" erfordert die ClientClusterContext-Instanz und den Namen der Datengridinstanz. Die

ClientClusterContext-Instanz wird während der Verbindung zum Katalogserver angefordert. Der Name der ObjectGrid-Instanz ist `Grid` und in der Datei `objectgrid.xml` angegeben. Das folgende Code-Snippet veranschaulicht, wie das Datengrid durch Aufruf der Methode "getObjectGrid" der API "ObjectGridManager" angefordert wird.

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

3. Session-Instanz abrufen.

   Sie können eine Session-Instanz von der angeforderten ObjectGrid-Instanz abrufen. Eine Session-Instanz ist erforderlich, um die ObjectMap-Instanz abzurufen und die Transaktionsdemarkation durchzuführen. Das folgende Code-Snippet veranschaulicht, wie eine Session-Instanz durch Aufruf der Methode "getSession" der API "ObjectGrid" abgerufen wird.

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

4. ObjectMap-Instanz abrufen.

   Nach dem Abrufen einer Session-Instanz können Sie durch Aufruf der Methode "getMap" der API "Session" eine ObjectMap-Instanz von der Session-Instanz abrufen. Sie müssen den Namen der Map als Parameter an die Methode getMap übergeben, um die ObjectMap-Instanz abzurufen. Das folgende Code-Snippet veranschaulicht, wie eine ObjectMap-Instanz durch Aufruf der Methode "getMap" der API "Session" angefordert wird.

   ```
   ObjectMap map1 = sess.getMap("Map1");
   ```

5. ObjectMap-Methoden verwenden.

   Nach dem Anfordern einer ObjectMap-Instanz können Sie die API ObjectMap verwenden. Beachten Sie, dass die Schnittstelle "ObjectMap" eine transaktionsorientierte Map ist und eine Transaktionsdemarkation durch die Verwendung der Methoden "begin" und "commit" der API "Session" erfordert. Wenn keine explizite Transaktionsdemarkation in der Anwendung stattfindet, werden die ObjectMap-Operationen über Transaktionen mit automatischer Festschreibung ausgeführt.

   Das folgende Code-Snippet veranschaulicht, wie die API "ObjectMap" mit einer Transaktion mit automatischer Festschreibung verwendet wird.

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

   Das folgende Code-Snippet veranschaulicht, wie die API "ObjectMap" mit expliziter Transaktionsdemarkation verwendet wird.

   ```
   sess.begin();
   map1.insert(key1, value1);
   sess.commit();
   ```

Prüfpunkt der Lerneinheit

In dieser Lerneinheit haben Sie gelernt, wie eine einfache Clientanwendung für die Ausführung von Datengridoperationen erstellt wird.

Lerneinheit 3 des Lernprogramms "Einführung": Beispielclientanwendung "gettingstarted" ausführen.

Verwenden Sie die folgenden Schritte, um das erste Datengrid zu starten und einen Client für die Interaktion mit dem Datengrid auszuführen.

Das Script `env.sh|bat` wird von den anderen Scripts aufgerufen, um die erforderlichen Umgebungsvariablen zu setzen. Normalerweise müssen Sie dieses Script nicht ändern.

- UNIX Linux `./env.sh`

- Windows `env.bat`

Zum Ausführen der Anwendung müssen Sie zuerst den Katalogserviceprozess starten. Der Katalogservice ist die Steuerzentrale des Datengrids. Er überwacht die Positionen der Container-Server und steuert die Verteilung der Daten auf die Host-Container-Server. Nach dem Starten des Katalogservice können Sie die Container-Server starten, in denen die Anwendungsdaten für das Datengrid gespeichert werden. Wenn Sie mehrere Kopien der Daten speichern möchten, können Sie mehrere Container-Server starten. Nach dem Start aller Server können Sie die Clientanwendung ausführen, um Daten aus dem Datengrid einzufügen, zu aktualisieren, zu entfernen und abzurufen.

1. Öffnen Sie eine Terminalsitzung oder ein Befehlszeilenfenster.
2. Verwenden Sie den folgenden Befehl, um zum Verzeichnis `gettingstarted` zu navigieren:

   `cd` *`WXS-Installationsstammverzeichnis`*`/ObjectGrid/gettingstarted`

   Ersetzen Sie *WXS-Installationsstammverzeichnis* durch den Pfad des eXtreme-Scale-Installationsstammverzeichnisses bzw. den Stammdateipfad (*WXS-Installationsstammverzeichnis*) der entpackten Testversion von eXtreme Scale.
3. Führen Sie das folgende Script aus, um einen Katalogserviceprozess auf dem lokalen Host (localhost) zu starten:

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

   Der Katalogserviceprozess wird im aktuellen Terminalfenster ausgeführt.

   Sie können den Katalogservice auch mit dem Befehl **`startOgServer`** starten. Führen Sie **`startOgServer`** im Verzeichnis *WXS-Installationsstammverzeichnis*`/ObjectGrid/bin` aus:

   - UNIX Linux `startOgServer.sh cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`
   - Windows `startOgServer.bat cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`
4. Öffnen Sie eine weitere Terminalsitzung bzw. ein weiteres Befehlszeilenfenster, und führen Sie den folgenden Befehl aus, um eine Container-Server-Instanz zu starten:

   - UNIX Linux `./runcontainer.sh server0`
   - Windows `runcontainer.bat server0`

   Der Container-Server wird im aktuellen Terminalfenster ausgeführt. Sie können diesen Schritt mit einem anderen Servernamen wiederholen, wenn Sie mehrere Container-Server-Instanzen für die Replikationsunterstützung starten möchten.

   Sie können die Container-Server auch mit dem Befehl **`startOgServer`** starten. Führen Sie **`startOgServer`** im Verzeichnis *WXS-Installationsstammverzeichnis*`/ObjectGrid/bin` aus:

   - UNIX Linux `startOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml -deploymentPolicyFile gettingstarted\xml\deployment.xml`
   - Windows `startOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml -deploymentPolicyFile gettingstarted\xml\deployment.xml`
5. Öffnen Sie eine weitere Terminalsitzung bzw. ein weiteres Befehlszeilenfenster, um Clientbefehle auszuführen.

Das Script `runclient.sh|bat` führt den einfachen CRUD-Client aus und startet die angegebene Operation. Das Script `runclient.sh|bat` wird mit den folgenden Parametern ausgeführt:

- UNIX Linux `./runclient.sh Befehl Wert1 Wert2`
- Windows `runclient.bat Befehl Wert1 Wert2`

Für *Befehl* können Sie eine der folgenden Optionen einsetzen:

- Geben Sie `i` ein, um *Wert2* in das Datengrid mit dem Schlüssel *Wert1* einzufügen.
- Geben Sie `u` ein, um das Objekt mit dem Schlüssel *Wert1* in *Wert2* zu aktualisieren.
- Geben Sie `d` ein, um das Objekt mit dem Schlüssel *Wert1* zu löschen.
- Geben Sie `g` ein, um das Objekt mit dem Schlüssel *Wert1* abzurufen und anzuzeigen.

a. Fügen Sie dem Datengrid Daten hinzu:
   - UNIX Linux `./runclient.sh i key1 helloWorld`
   - Windows `runclient.bat i key1 helloWorld`

b. Suchen und zeigen Sie den Wert an:
   - UNIX Linux `./runclient.sh g key1`
   - Windows `runclient.bat g key1`

c. Aktualisieren Sie den Wert:
   - UNIX Linux `./runclient.sh u key1 goodbyeWorld`
   - Windows `runclient.bat u key1 goodbyeWorld`

d. Löschen Sie den Wert:
   - UNIX Linux `./runclient.sh d key1`
   - Windows `runclient.bat d key1`

Prüfpunkt der Lerneinheit

In dieser Lerneinheit haben Sie Folgendes gelernt:

- Katalogserver und Container-Server starten
- Beispielclientanwendung ausführen

Lerneinheit 4 des Lernprogramms "Einführung": Umgebung überwachen

Sie können das Dienstprogramm **xscmd** und Webkonsoltools verwenden, um Ihre Datengridumgebung zu überwachen.

Überwachung mit der Webkonsole

Mit der Webkonsole können Sie aktuelle Statistiken und Protokollstatistiken in einem Diagramm darstellen. Diese Konsole enthält einige vorkonfigurierte Diagramme für allgemeine Übersichten und eine angepasste Berichtseite, die Sie verwenden können, um aus den verfügbaren Statistiken Diagramme zu erstellen. Sie können die Diagrammfunktionen in der Überwachungskonsole von WebSphere eXtreme Scale verwenden, um die allgemeine Leistung der Datengrids in Ihrer Umgebung anzuzeigen.

Installieren Sie die Webkonsole als optionales Feature, wenn Sie den Installationsassistenten ausführen:

1. Starten Sie den Konsolserver. Das Script **startConsoleServer.bat|sh** zum Starten des Konsolservers befindet sich im Verzeichnis *WXS-Installationsstammverzeichnis*/ObjectGrid/bin Ihrer Installation.
2. Melden Sie sich an der Konsole an.
   a. Rufen Sie im Webbrowser den URL `https://Ihr.Konsol.Host:7443` auf. Ersetzen Sie `Ihr.Konsol.Host` durch den Hostnamen des Servers, auf dem Sie die Konsole installiert haben.
   b. Melden Sie sich an der Konsole an.
      - **Benutzer-ID:** `admin`
      - **Kennwort:** `admin`

      Die Begrüßungsseite der Konsole erscheint.
3. Bearbeiten Sie die Konsolkonfiguration. Klicken Sie auf **Einstellungen** > **Konfiguration**, um die Konsolkonfiguration zu überprüfen. Die Konsolkonfiguration enthält Informationen wie die folgenden:
   - Tracezeichenfolge für den eXtreme-Scale-Client, z. B. `*=all=disabled`
   - Administratorname und -kennwort
   - E-Mail-Adresse des Administrators
4. Stellen Sie Verbindungen zu Katalogservern her, die Sie überwachen möchten, und verwalten Sie diese. Wiederholen Sie die folgenden Schritte, um die einzelnen Katalogserver zur Konfiguration hinzuzufügen.
   a. Klicken Sie auf **Einstellungen** > **eXtreme-Scale-Katalogserver**.
   b. Fügen Sie einen neuen Katalogserver hinzu.

      1) Klicken Sie auf das Symbol "Hinzufügen" (), um einen vorhandenen Katalogserver zu registrieren.
      2) Geben Sie Informationen wie den Hostnamen und den Listener-Port an. Weitere Informationen zur Portkonfiguration und zu den Portstandardwerten finden Sie unter „Netzports planen” auf Seite 65.
      3) Klicken Sie auf **OK**.
      4) Vergewissern Sie sich, dass der Katalogserver der Navigationsstruktur hinzugefügt wurde.
5. Zeigen Sie den Verbindungsstatus an. Im Feld **Aktuelle Domäne** wird der Name der Katalogservicedomäne angezeigt, die momentan verwendet wird, um Informationen in der Webkonsole anzuzeigen. Der Verbindungsstatus wird neben dem Namen der Katalogservicedomäne angezeigt.
6. Statistiken für die Datengrids und Server anzeigen oder einen angepassten Bericht erstellen.

Überwachung mit dem Dienstprogramm "xscmd"

1. Öffnen Sie ein Befehlszeilenfenster. Setzen Sie in der Befehlszeile die entsprechenden Umgebungsvariablen.
   a. Setzen Sie die Umgebungsvariable *CLIENT_AUTH_LIB*:
      - Windows `set CLIENT_AUTH_LIB=<`*Pfad_zu_Sicherheits-JARs_oder_Klassen*`>`
      - UNIX `set CLIENT_AUTH_LIB=<`*Pfad_zu_Sicherheits-JARs_oder_Klassen*`> export CLIENT_AUTH_LIB`

2. Wechseln Sie in das Verzeichnis *WXS-Ausgangsverzeichnis*/bin.
   ```
   cd WXS-Ausgangsverzeichnis/bin
   ```
3. Führen Sie verschiedene Befehle aus, um Informationen zu Ihrer Umgebung anzuzeigen.
   - Alle Online-Container-Server für das Datengrid "Grid" und das MapSet "mapSet" anzeigen:
     ```
     xscmd -c showPlacement -g Grid -ms mapSet
     ```
   - Routing-Informationen für das Datengrid anzeigen.
     ```
     xscmd -c routetable -g Grid
     ```
   - Anzahl der Map-Einträge im Datengrid anzeigen.
     ```
     xscmd -c showMapSizes -g Grid -ms mapSet
     ```

Server stoppen

Wenn Sie mit Ihren Arbeiten in der Clientanwendung und der Überwachung der Beispielumgebung des Lernprogramms "Einführung" fertig sind, können Sie die Server stoppen.

- Wenn Sie die Scriptdateien zum Starten der Server verwendet haben, verwenden Sie die Tastenkombination <Strg+c>, um den Katalogserverprozess und die Container-Server in den entsprechenden Fenstern zu stoppen.
- Wenn Sie den Befehl **startOgServer** zum Starten der Server verwendet haben, verwenden Sie den Befehl **stopOgServer** zum Stoppen der Server.

  Container-Server stoppen:

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

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

  Katalogserver stoppen:

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

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

Prüfpunkt der Lerneinheit

In dieser Lerneinheit haben Sie Folgendes gelernt:

- Webkonsole starten und Konsole mit dem Katalogserver verbinden
- Datengrid- und Serverstatistiken überwachen
- Server stoppen

Kapitel 2. Planung

Bevor Sie WebSphere eXtreme Scale installieren und Ihre Datengridanwendungen implementieren, müssen Sie Ihr Cachingtopologie festlegen, die Kapazitätsplanung durchführen, die Hardware- und Softwarevoraussetzungen prüfen, die Einstellungen für den Netzbetrieb und die Optimierung prüfen usw. Sie können auch die Prüfliste für Betriebsbereitschaft verwenden, um sicherzustellen, dass Ihre Umgebung für die Implementierung von Anwendungen bereit ist.

Eine Beschreibung der bewährten Verfahren für das Entwerfen von eXtreme-Scale-Anwendungen finden Sie im folgenden Artikel auf developerWorks: Principles and best practices for building high performing and highly resilient WebSphere eXtreme Scale applications.

Übersicht über die Planung

Vor der Verwendung von WebSphere eXtreme Scale in einer Produktionsumgebung sollten Sie sich die folgenden Punkte zur Optimierung der Implementierung ansehen.

Installationshinweise

Sie können WebSphere eXtreme Scale in einer eigenständigen Umgebung installieren, oder Sie können die Installation mit WebSphere Application Server integrieren. Um sicherzustellen, dass Sie Ihre Server künftig nahtlos aktualisieren können, müssen Sie Ihre Umgebung entsprechend planen. Für eine optimale Leistung müssen Katalogserver auf anderen Maschinen als die Container-Server ausgeführt werden. Wenn Sie Ihre Katalogserver und Container-Server auf derselben Maschine ausführen müssen, verwenden Sie separate Installationen von WebSphere eXtreme Scale für die Katalog- und Container-Server. Wenn Sie zwei Installationen verwenden, können Sie die Installation, in der der Katalogserver ausgeführt wird, zuerst aktualisieren.

Hinweise zur Cachingtopologie

Ihre Architektur kann lokales speicherinternes Datencaching oder verteiltes Client/Server-Caching verwenden. Jeder Typ von Cachetopologie hat Vor- und Nachteile. Die Cachingtopologie, die Sie implementieren, richtet sich nach den Anforderungen Ihrer Umgebung und Ihrer Anwendung. Weitere Informationen zu den verschiedenen Cachingtopologien finden Sie unter „Topologie planen” auf Seite 10.

Hinweise zur Datenkapazität

Die folgende Liste enthält die zu beachtenden Punkte:

- **Anzahl der Systeme und Prozessoren:** Wie viele physische Maschinen und Prozessoren sind in der Umgebung erforderlich?
- **Anzahl der Server:** Wie viele eXtreme-Scale-Server sind für die Speicherung der eXtreme-Scale-Maps erforderlich?
- **Anzahl der Partitionen:** Das in den Maps gespeicherte Datenvolumen ist ein Faktor für die Bestimmung der Anzahl erforderlicher Partitionen.
- **Anzahl der Replikate:** Wie viele Replikate sind für jedes primäre Shard in der Domäne erforderlich?

- **Synchrone oder asynchrone Replikation:** Sind die Daten elementar, so dass eine synchrone Replikation erforderlich ist? Oder hat die Leistung eine höhere Priorität, so dass die asynchrone Replikation die richtige Wahl ist?
- **Größe der Heapspeicher:** Welches Datenvolumen wird auf jedem Server gespeichert?

Ausführliche Erläuterungen zu jedem dieser Hinweise finden Sie in „Umgebungskapazität planen" auf Seite 55.

Topologie planen

Mit WebSphere eXtreme Scale kann Ihre Architektur speicherinternes Daten-Caching oder verteiltes Client/Server-Daten-Caching verwenden. Die Architektur kann verschiedene Beziehungen zu Ihren Datenbanken haben. Sie können die Topologie auch so konfigurieren, dass sie mehrere Rechenzentren umspannt.

WebSphere eXtreme Scale erfordert für den Betrieb eine minimale zusätzliche Infrastruktur. Die Infrastruktur setzt sich aus Scripts für die Installation, das Starten und Stoppen von Java-EE-Anwendungen in einem Server zusammen. Die zwischengespeicherten Daten werden in den Container-Servern gespeichert, und Clients stellen über Fernzugriff eine Verbindung zum Server her.

Speicherinterne Umgebungen

Wenn Sie die Implementierung in einer lokalen, speicherinternen Umgebung durchführen, wird WebSphere eXtreme Scale in einer einzigen Java Virtual Machine ausgeführt und nicht repliziert. Zum Konfigurieren einer lokalen Umgebung können Sie eine ObjectGrid-XML-Datei oder die ObjectGrid-APIs verwenden.

Verteilte Umgebungen

Wenn Sie die Implementierung in einer verteilten Umgebung durchführen, wird WebSphere eXtreme Scale in einer Reihe von Java Virtual Machines ausgeführt, was die Leistung, die Verfügbarkeit und die Skalierbarkeit erhöht. Mit dieser Konfiguration können Sie Datenreplikation und Partitionierung verwenden. Zusätzliche Server können hinzugefügt werden, ohne die vorhandenen eXtreme Scale-Server erneut starten zu müssen. Wie bei einer lokalen Umgebung ist in einer verteilten Umgebung eine ObjectGrid-XML-Datei oder eine entsprechende programmgesteuerte Konfiguration erforderlich. Außerdem müssen Sie eine XML-Implementierungsrichtliniendatei mit Konfigurationsdetails bereitstellen.

Sie können einfache Implementierungen erstellen oder große Implementierungen in Terabytegröße, in denen Tausende von Servern erforderlich sind.

Lokaler Speichercache

Im einfachsten Fall kann WebSphere eXtreme Scale als lokaler (nicht verteilter) speicherinterner Datengrid-Cache verwendet werden. Dies kann insbesondere für Anwendungen mit sehr vielen gemeinsamen Zugriffen von Vorteil sein, in denen mehrere Threads auf transiente Daten zugreifen und diese ändern müssen. Die in einem lokalen Datengrid gespeicherten Daten können indexiert und mit Abfragen abgerufen werden. Abfragen helfen Ihnen bei der Arbeit mit großen speicherinternen Datasets. Die mit Java Virtual Machine (JVM) bereitgestellte Unterstützung ist zwar einsatzfähig, besitzt aber eine eingeschränkte Datenstruktur.

Die lokale speicherinterne Cachetopologie für WebSphere eXtreme Scale wird verwendet, um einen konsistenten, transaktionsorientierten Zugriff auf temporäre Daten in einer einzelnen Java Virtual Machine zu unterstützen.

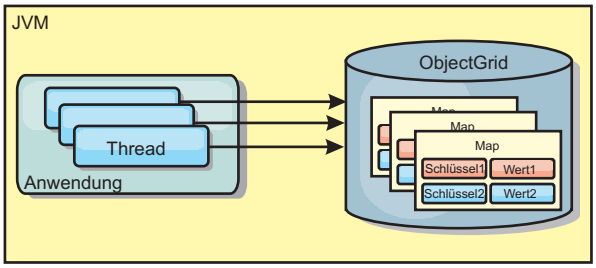


Abbildung 1. Szenario mit einem lokalen speicherinternen Speichercache

Vorteile

- Einfaches Setup: Ein ObjectGrid kann programmgesteuert oder deklarativ über die ObjectGrid-XML-Implementierungsdeskriptordatei oder mit anderen Frameworks wie Spring erstellt werden.
- Schnell: Jede BackingMap kann gesondert für eine optimale Speicherauslastung und gemeinsamen Zugriff optimiert werden.
- Ideal für Topologien mit einer einzigen JVM und eines kleinen Datasets oder für das Caching von Daten, auf die häufig zugegriffen wird.
- Transaktionsorientiert: BackingMap-Aktualisierungen können zu einer einzigen Arbeitseinheit gruppiert und als letzter Teilnehmer in zweiphasige Transaktionen wie JTA-Transaktionen (Java Transaction Architecture) integriert werden.

Nachteile

- Keine Fehlertoleranz.
- Die Daten werden nicht repliziert. Speichercaches eignen sich am besten für schreibgeschützte Referenzdaten.
- Keine Skalierbarkeit. Die für die Datenbank erforderliche Speicherkapazität kann die JVM möglicherweise nicht bereitstellen.
- Es treten Probleme auf, wenn JVMs hinzugefügt werden.
  - Die Daten sind nicht so einfach partitionierbar.
  - Der Status muss in den JVMs manuell repliziert werden, da die einzelnen Cacheinstanzen ansonsten verschiedene Versionen derselben Daten enthalten könnten.
  - Das Ungültigmachen von Einträgen ist kostenintensiv.
  - Jeder Cache muss einzeln vorbereitet werden. Die Vorbereitungs- oder Aufwärmphase ist der Zeitraum, in dem eine Gruppe von Daten geladen wird, damit der Cache mit gültigen Daten gefüllt wird.

Einsatz

Die Implementierungstopologie mit dem lokalen Speichercache sollte nur verwendet werden, wenn die Menge der zwischenzuspeichernden Daten klein ist (in eine einzige JVM passt) und relativ stabil ist. Bei diesem Ansatz müssen veraltete Daten toleriert werden. Die Verwendung von Evictor (Bereinigungsprogramm), um nur die am häufigsten verwendeten oder die zuletzt verwendeten Daten im Cache zu verwalten, kann dabei helfen, den Cache klein zu halten und die Relevanz der Daten zu erhöhen.

Auf Peers replizierter lokaler Cache

Sie müssen sicherstellen, dass der Cache synchronisiert wird, wenn mehrere Prozesse mit unabhängigen Cacheinstanzen vorhanden sind. Um sicherzustellen, dass die Cacheinstanzen synchronisiert werden, richten Sie einen auf Peers replizierten Cache mithilfe von Java Message Service (JMS) ein.

WebSphere eXtreme Scale enthält zwei Plug-ins, die Transaktionsänderungen automatisch zwischen Peer-ObjectGrid-Instanzen weitergeben. Das Plug-in "JMSObjectGridEventListener" gibt eXtreme-Scale-Änderungen automatisch über JMS weiter.

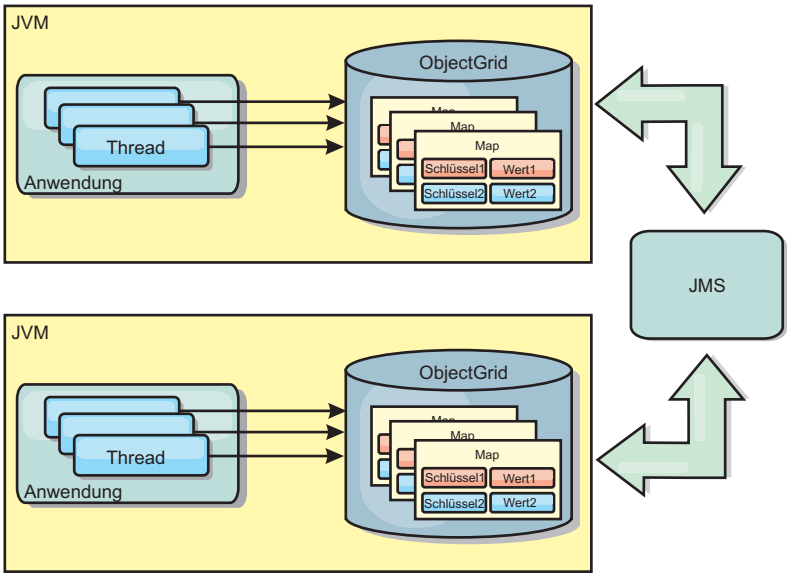


Abbildung 2. Auf Peers replizierter Cache mit Änderungen, die über JMS weitergegeben werden

Wenn Sie in einer Umgebung mit WebSphere Application Server arbeiten, ist auch das TranPropListener-Plug-in verfügbar. Das TranPropListener-Plug-in verwendet den High Availability Manager (kurz HA-Manager), um die Änderungen an jede Peer-Cacheinstanz weiterzugeben.

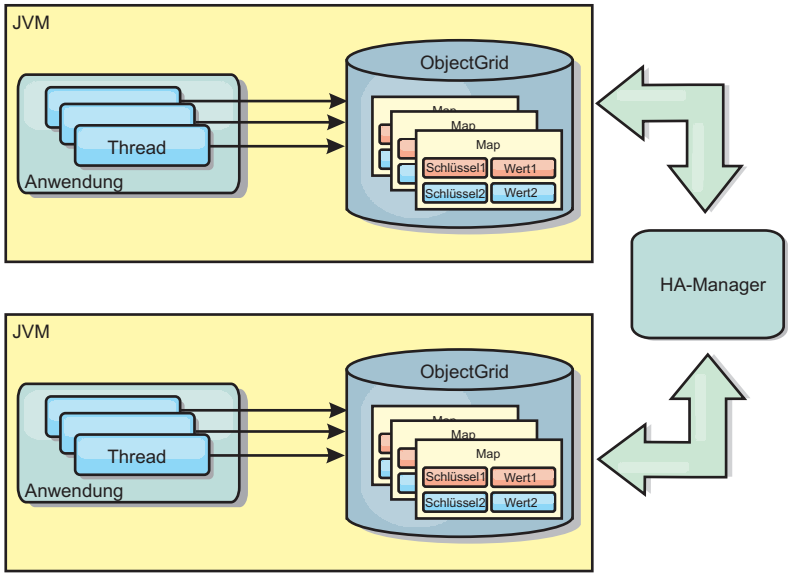


Abbildung 3. Auf Peers replizierter Cache mit Änderungen, die über den High Availability Manager weitergegeben werden

Vorteile

- Die Gültigkeit der Daten ist höher, weil sie häufiger aktualisiert werden.
- Mit dem TranPropListener-Plug-in kann eXtreme Scale wie die lokale Umgebung über das Programm oder deklarativ über die XML-Implementierungsdeskriptordatei von eXtreme Scale oder mit anderen Frameworks wie Spring erstellt werden. Die Integration mit dem High Availability Manager erfolgt automatisch.
- Jede BackingMap kann gesondert für eine optimale Speicherauslastung und gemeinsamen Zugriff optimiert werden.
- BackingMap-Aktualisierungen können zu einer einzigen Arbeitseinheit gruppiert und als letzter Teilnehmer in zweiphasige Transaktionen wie JTA-Transaktionen (Java Transaction Architecture) integriert werden.
- Ideal für Topologien mit wenigen JVMs und einem angemessen kleinen Dataset oder für das Caching von Daten, auf die häufig zugegriffen wird.
- Änderungen an eXtreme Scale werden in allen Peerinstanzen von eXtreme Scale repliziert. Die Änderungen sind so lange konsistent, wie eine permanente Subskription verwendet wird.

Nachteile

- Die Konfiguration und Verwaltung für JMSObjectGridEventListener kann komplex sein. eXtreme Scale kann über das Programm oder deklarativ über die XML-Implementierungsdeskriptordatei von eXtreme Scale oder mit anderen Frameworks wie Spring erstellt werden.
- Nicht skalierbar: Die für die Datenbank erforderliche Speicherkapazität kann die JVM möglicherweise nicht bereitstellen.
- Funktioniert nicht ordnungsgemäß, wenn Java Virtual Machines hinzugefügt werden:
  - Die Daten sind nicht so einfach partitionierbar.
  - Das Ungültigmachen von Einträgen ist kostenintensiv.
  - Jeder Cache muss einzeln vorbereitet werden.

Einsatz

Verwenden Sie die Implementierungstopologie nur, wenn das zwischenzuspeichernde Datenvolumen klein ist, in eine einzige JVM passt und relativ stabil ist.

Integrierter Cache

WebSphere-eXtreme-Scale-Grids können in vorhandenen Prozessen als integrierte eXtreme-Scale-Server ausgeführt oder als externe Prozesse verwaltet werden.

Integrierte Grids sind hilfreich, wenn Sie mit einem Anwendungsserver wie WebSphere Application Server arbeiten. Sie können eXtreme-Scale-Server, die nicht integriert sind, über Befehlszeilenscripts starten und in einem Java-Prozess ausführen.

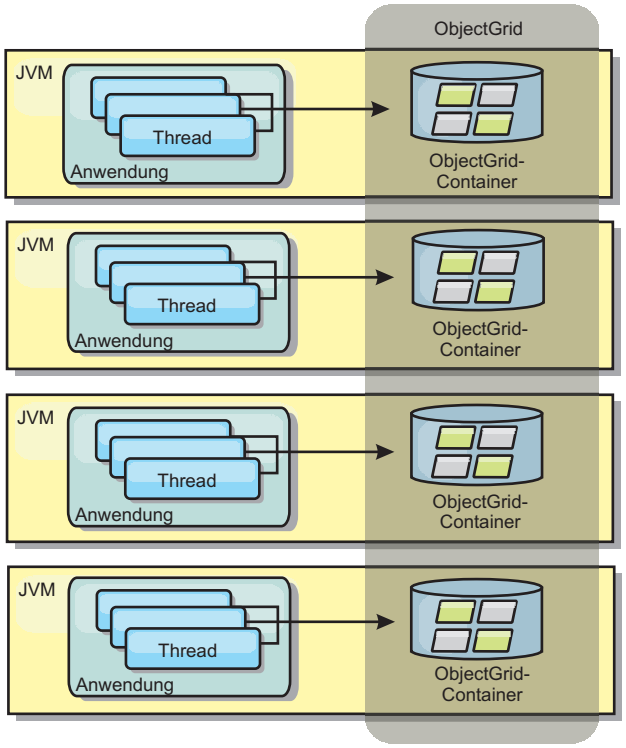


Abbildung 4. Integrierter Cache

Vorteile

- Vereinfachte Verwaltung, da weniger Prozesse zu verwalten sind
- Vereinfachte Anwendungsimplementierung, weil das Grid den Klassen-Loader der Clientanwendung verwendet
- Unterstützung von Partitionierung und hoher Verfügbarkeit

Nachteile

- Erhöhter Speicherbedarf im Clientprozess, weil alle Daten im Prozess erfasst werden
- Erhöhte CPU-Auslastung für die Bearbeitung von Clientanforderungen
- Erschwerte Verarbeitung von Anwendungsupgrades, da Clients dieselben Java-Anwendungsarchivdateien wie die Server verwenden

- Weniger Flexibilität. Die Skalierung von Clients und Grid-Servern ist nicht linear. Wenn Server extern definiert werden, haben Sie mehr Flexibilität bei der Verwaltung der Prozessanzahl.

Einsatz

Verwenden Sie integrierte Grids, wenn im Clientprozess reichlich Speicher für die Griddaten und potenzielle Failover-Daten frei ist.

Weitere Informationen finden Sie im Abschnitt zum Aktivieren des Clientinvalidierungsmechanismus in der Veröffentlichung *Verwaltung*.

Verteilter Cache

In den meisten Fällen wird WebSphere eXtreme Scale als gemeinsam genutzter Cache verwendet, um einen transaktionsgesteuerten Zugriff auf Dateien durch mehrere Komponenten zu ermöglichen, wo ansonsten eine traditionelle Datenbank verwendet werden würde. Bei der Verwendung eines gemeinsam genutzten Caches muss keine Datenbank konfiguriert werden.

Kohärenz des Caches

Der Cache ist kohärent, weil alle Clients dieselben Daten im Cache sehen. Jede einzelne Information wird im Cache eines einzigen Servers gespeichert. Auf diese Weise werden unnötige Datensatzkopien verhindert, die potenziell verschiedene Versionen der Daten enthalten könnten. Ein kohärenter Cache kann außerdem mehr Daten aufnehmen, wenn dem Datengrid weitere Server hinzugefügt werden. Die Skalierung des Caches erfolgt linear zum Wachstum des Grids. Da Clients über Prozedurfernaufrufe auf Daten in diesem Datengrid zugreifen, wird der Cache auch als ferner Cache bezeichnet. Durch die Datenpartitionierung enthält jeder Prozess einen eindeutigen Teil des Gesamtdatasets. Größere Datengrids können mehr Daten aufnehmen und und mehr Anforderungen für diese Daten bearbeiten. Aufgrund der Kohärenz müssen auch keine Daten zum Ungültigmachen im Datengrid verteilt werden, weil es keine veralteten Daten gibt. Der kohärente Cache enthält jeweils nur die aktuelle Kopie jeder Information.

Wenn Sie in einer Umgebung mit WebSphere Application Server arbeiten, ist auch das TranPropListener-Plug-in verfügbar. Das TranPropListener-Plug-in verwendet die Komponente "High Availability Manager" (kurz HA-Manager) von WebSphere Application Server, um die Änderungen an die einzelnen Peer-ObjectGrid-Cacheinstanzen weiterzugeben.

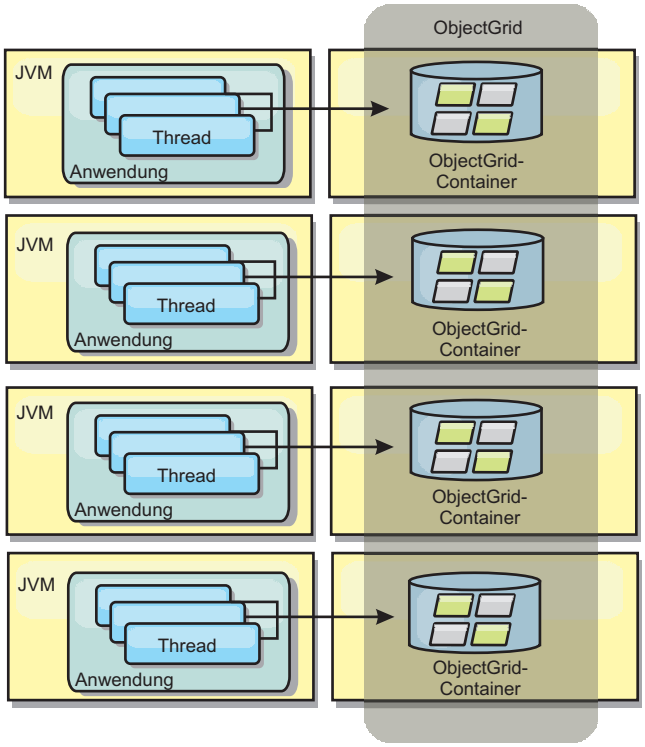


Abbildung 5. Verteilter Cache

Naher Cache

Clients können optional einen lokalen integrierten Cache haben, wenn eXtreme Scale in einer verteilten Topologie verwendet wird. Dieser optionale Cache wird als naher Cache bezeichnet. Er ist ein unabhängiges ObjectGrid in jedem Client, das als Cache für den fernen serverseitigen Cache dient. Der nahe Cache wird standardmäßig aktiviert, wenn eine optimistische Sperrstrategie oder keine Sperrstrategie konfiguriert ist, und kann nicht verwendet werden, wenn eine pessimistische Sperrstrategie konfiguriert ist.

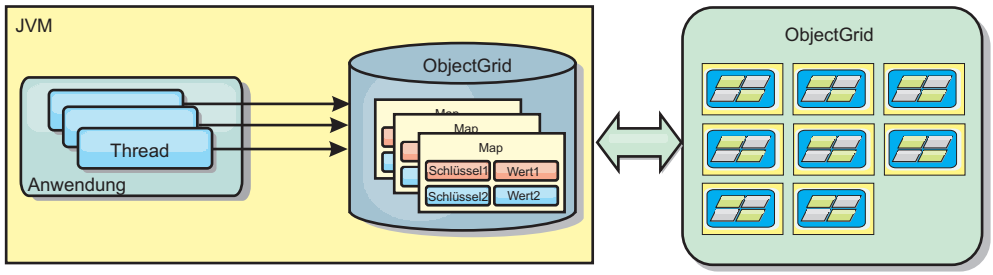


Abbildung 6. Naher Cache

Ein naher Cache ist sehr schnell, weil er den speicherinternen Zugriff auf einen Teil der gesamten zwischengespeicherten Daten ermöglicht, die fern in den Servern von eXtreme Scale gespeichert sind. Der nahe Cache ist nicht partitioniert und enthält Daten aus allen fernen eXtreme-Scale-Partitionen. WebSphere eXtreme Scale kann bis zu drei Cacheschichten haben. Diese sind im Folgenden erläutert:

1. Der Cache auf der Transaktionsschicht enthält alle Änderungen für eine einzelne Transaktion. Der Transaktionscache enthält eine Arbeitskopie der Daten, bis

die Transaktion festgeschrieben wird. Wenn eine Clienttransaktion Daten aus einer ObjectMap anfordert, wird zuerst die Transaktion geprüft.

2. Der nahe Cache auf der Clientschicht enthält einen Teil der Daten aus der Serverschicht. Wenn die Daten nicht auf Transaktionsschicht zu finden sind, werden die Daten (sofern verfügbar) von der Cacheschicht abgerufen und in den Transaktionscache eingefügt.
3. Das Datengrid auf der Serverschicht enthält den Hauptteil der Daten und wird von allen Clients gemeinsam genutzt. Die Serverschicht kann partitioniert werden, was die Zwischenspeicherung großer Datenvolumen ermöglicht. Wenn der nahe Cache des Clients die Daten nicht enthält, werden die Daten von der Serverschicht abgerufen und in den Clientcache eingefügt. Die Serverschicht kann auch ein Loader-Plug-in (Ladeprogramm) haben. Wenn das Grid die angeforderten Daten nicht enthält, wird der Loader aufgerufen, der die Daten aus dem Back-End-Datenspeicher abruft und in das Grid einfügt.

Zum Inaktivieren des nahen Caches setzen Sie das Attribut "numberOfBuckets" in der eXtreme-Scale-Deskriptorkonfiguration für das Überschreiben des Clients auf "0". Einzelheiten zu den Sperrstrategien von eXtreme Scale finden Sie im Abschnitt zum Sperren von Map-Einträgen. Der nahe Cache kann auch mit einer gesonderten Bereinigungsrichtlinie und anderen Plug-ins konfiguriert werden, die die eXtreme-Scale-Deskriptorkonfiguration für das Überschreiben des Clients verwenden.

Vorteil

- Schnelle Antwortzeiten, weil alle Zugriffe auf die Daten lokal erfolgen. Indem die Daten zuerst im nahen Cache gesucht werden, wird eine Konsultation des Servergrid gespart, wodurch selbst die fernen Daten lokal zugänglich werden.

Nachteile

- Die Verweildauer veralteter Daten erhöht sich, weil der nahe Cache auf jeder Schicht unter Umständen nicht mit den aktuellen Daten im Datengrid synchronisiert sind.
- Es muss ein Bereinigungsprogramm zum Ungültigmachen von Daten verwendet werden, um Speicherengpässe zu verhindern.

Einsatz

Verwenden Sie diese Technik, wenn die Antwortzeiten wichtig und veraltete Daten tolerierbar sind.

Datenbankintegration: Write-behind, Inline- und Neben-Caching

WebSphere eXtreme Scale wird als Front-End für eine traditionelle Datenbank verwendet und macht Leseaktivitäten überflüssig, die normalerweise an die Datenbank übertragen werden. Ein kohärenter Cache kann direkt oder indirekt über einen ORM (Object Relational Mapper) mit einer Anwendung verwendet werden. Der kohärente Cache kann dann die Datenbank bzw. das Back-End von Leseaktivitäten entlasten. In einem geringfügig komplexeren Szenario, wie z. B. beim transaktionsorientierten Zugriff auf einen Datenbestand, in dem nur einige der Daten traditionelle Persistenzgarantien erfordern, können Sie Filter verwenden, um selbst die Schreibtransaktionen auszulagern.

Sie können WebSphere eXtreme Scale als hoch flexiblen speicherinternen Datenbankverarbeitungsbereich konfigurieren. WebSphere eXtreme Scale ist jedoch kein ORM. Das Produkt weiß nicht, woher die Daten im Datengrid stammen. Eine An-

wendung oder ein ORM kann Daten in einem eXtreme-Scale-Server ablegen. Die Datenquelle ist dafür verantwortlich sicherzustellen, dass sie mit der Datenbank, aus der die Daten stammen, konsistent bleibt. Das bedeutet, dass eXtreme Scale Daten, die automatisch aus einer Datenbank extrahiert werden, nicht ungültig machen kann. Die Anwendung bzw. der Mapper muss diese Funktion bereitstellen und die Daten verwalten, die in eXtreme Scale gespeichert werden.

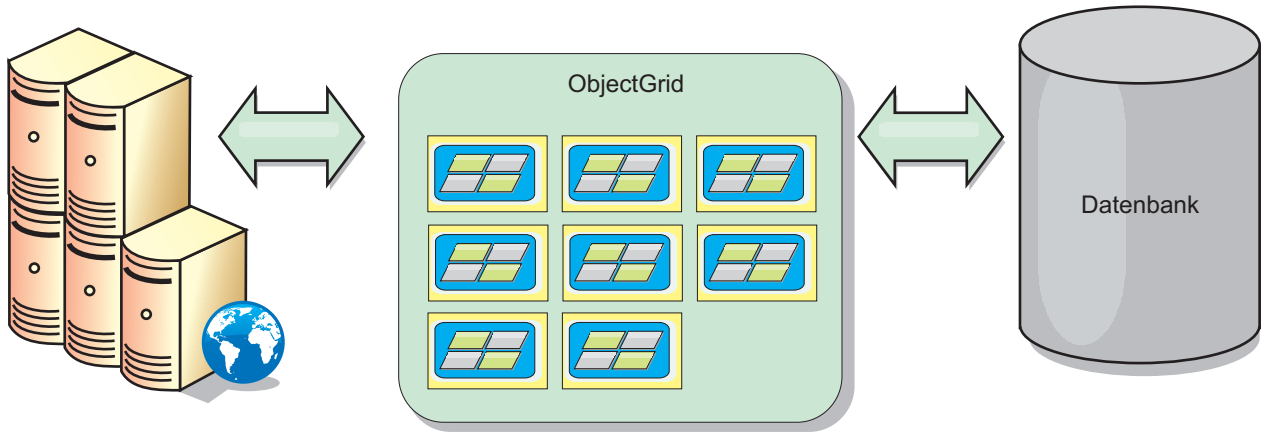


Abbildung 7. ObjectGrid als Datenbankpuffer

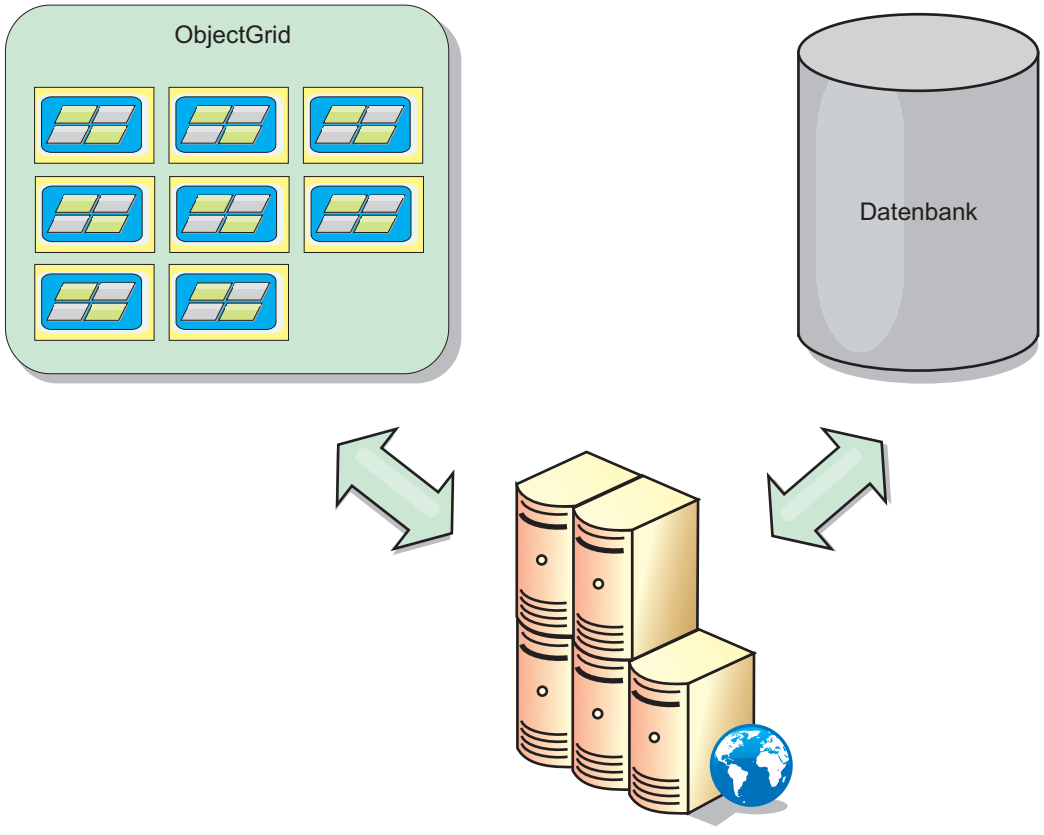


Abbildung 8. ObjectGrid als Nebencache

Teilcache und vollständiger Cache

WebSphere eXtreme Scale kann als Teilcache oder als vollständiger Cache eingesetzt werden. In einem Teilcache wird nur ein Teil der gesamten Daten gespeichert,

wohingegen in einem vollständigen Cache alle Daten gespeichert werden. Ein Teilcache kann nach und nach bedarfsgesteuert gefüllt werden. Der Zugriff auf Teilcaches erfolgt gewöhnlich über Schlüssel (und nicht über Indizes oder Abfragen), da die Daten nur teilweise verfügbar sind.

Teilcache

Wenn ein Schlüssel nicht im Teilcache vorhanden ist oder wenn die Daten nicht verfügbar sind und ein Cachefehler auftritt, wird die nächste Schicht aufgerufen. Die Daten werden beispielsweise aus der Datenbank abgerufen und in die Cacheschicht des Datengrids eingefügt. Bei der Verwendung einer Abfrage oder eines Index wird nur auf die derzeit geladenen Werte zugegriffen, und die Anforderungen werden nicht an die anderen Schichten weitergeleitet.

Vollständiger Cache

Ein vollständiger Cache enthält alle erforderlichen Daten, und der Zugriff kann über Attribute ohne Schlüsselfunktion mit Indizes oder Abfragen erfolgen. Ein vollständiger Cache wird vorher mit Daten aus der Datenbank geladen, bevor die Anwendung versucht, auf die Daten zuzugreifen. Ein vollständiger Cache kann als Datenbankersatz dienen, nachdem die Daten geladen wurden. Da alle Daten verfügbar sind, können Abfragen und Indizes verwendet werden, um Daten zu suchen und zusammenzufassen.

Nebencache

Wenn WebSphere eXtreme Scale als Nebencache verwendet wird, wird das Back-End für das Datengrid verwendet.

Nebencache

Sie können das Produkt als Nebencache für die Datenzugriffsschicht einer Anwendung konfigurieren. In diesem Szenario wird WebSphere eXtreme Scale verwendet, um Objekte temporär zu speichern, die normalerweise aus einer Back-End-Datenbank abgerufen werden. Anwendungen prüfen, ob das Datengrid die Daten enthält. Wenn die Daten im Datengrid enthalten sind, werden die Daten an den Aufrufenden zurückgegeben. Wenn die Daten nicht vorhanden sind, werden die Daten aus der Back-End-Datenbank abgerufen. Anschließend werden die Daten in das Datengrid eingefügt, damit die nächste Anforderung die zwischengespeicherte Kopie verwenden kann. Die folgende Abbildung veranschaulicht, wie WebSphere eXtreme Scale als Nebencache mit einer beliebigen Datenzugriffsschicht wie OpenJPA oder Hibernate verwendet werden kann.

Nebencache-Plug-ins für Hibernate und OpenJPA

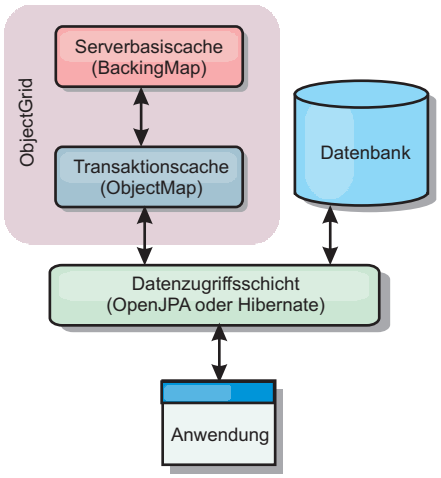


Abbildung 9. Nebencache

Cache-Plug-ins für OpenJPA und Hibernate sind in WebSphere eXtreme Scale enthalten. Damit können Sie das Produkt als automatischen Nebencache verwenden. Durch die Verwendung von WebSphere eXtreme Scale als Cache-Provider kann die Leistung beim Lesen und Abfragen von Daten verbessert und die Belastung der Datenbank verringert werden. WebSphere eXtreme Scale bietet im Vergleich mit integrierten Cacheimplementierungen verschiedene Vorteile, weil der Cache automatisch in allen Prozessen repliziert wird. Wenn ein Client einen Wert zwischenspeichert, können alle andere Clients den zwischengespeicherten Wert verwenden.

Inline-Cache

Sie können das Inline-Caching für ein Datenbank-Back-End oder als Nebencache für eine Datenbank konfigurieren. Beim Inline-Caching wird eXtreme Scale als primäres Mittel für die Interaktion mit den Daten verwendet. Bei der Verwendung von eXtreme Scale als Inline-Cache interagiert die Anwendung über ein Ladeprogramm-Plug-in mit dem Back-End.

Inline-Cache

Bei Verwendung als Inline-Cache interagiert WebSphere eXtreme Scale über ein Loader-Plug-in mit dem Back-End. Dieses Szenario kann den Datenzugriff vereinfachen, weil Anwendungen direkt auf die APIs von eXtreme Scale zugreifen können. Es werden verschiedene Caching-Szenarien in eXtreme Scale unterstützt, um sicherzustellen, dass die Daten im Cache und die Daten im Back-End synchronisiert sind. Die folgende Abbildung veranschaulicht, wie ein Inline-Cache mit der Anwendung und dem Back-End interagiert.

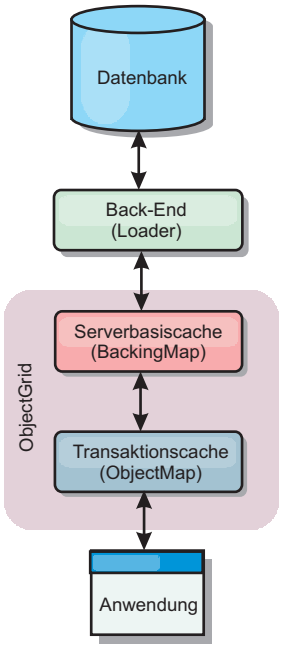


Abbildung 10. Inline-Cache

Die Option für Inline-Caching vereinfacht den Datenzugriff, weil sie Anwendungen den direkten Zugriff auf die eXtreme-Scale-APIs ermöglicht. WebSphere eXtreme Scale unterstützt mehrere Szenarien mit Inline-Caching:

- Read-through
- Write-Through
- Write-behind

Szenario mit Read-through-Caching

Ein Read-through-Cache ist ein Teilcache, in den nach und nach Dateneinträge nach Schlüssel geladen werden, wenn diese angefordert werden. Dies geschieht, ohne dass der Aufrufende wissen muss, wie die Einträge geladen werden. Wenn die Daten nicht im eXtreme-Scale-Cache gefunden werden, ruft eXtreme Scale die fehlenden Daten vom Loader-Plug-in ab, das die Daten aus der Back-End-Datenbank lädt und in den Cache einfügt. Nachfolgende Anforderungen für denselben Datenschlüssel werden im Cache gefunden, bis der Eintrag gelöscht, ungültig gemacht oder durch Bereinigung entfernt wird.

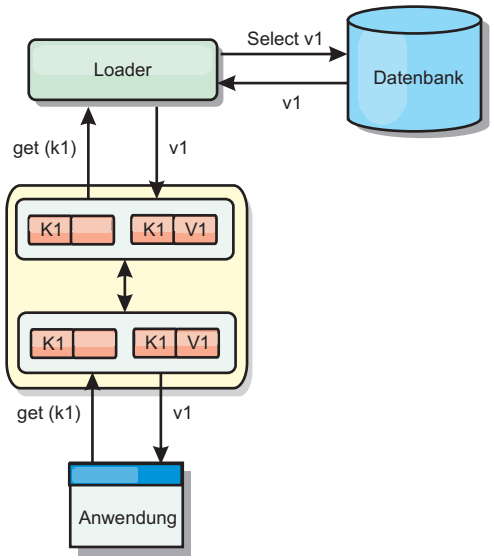


Abbildung 11. Read-through-Caching

Szenario mit Write-Through-Caching

In einem Write-Through-Cache (Durchschreibcache) erfolgt bei jedem Schreibvorgang in den Cache ein synchroner Schreibvorgang über den Loader in die Datenbank. Diese Methode gewährleistet die Konsistenz mit dem Back-End, verringert aber die Schreibleistung, weil die Datenbankoperation synchron erfolgt. Da der Cache und die Datenbank beide aktualisiert werden, werden bei nachfolgenden Leseoperationen dieselben Daten im Cache gefunden und Datenbankaufrufe vermieden. Ein Write-Through-Cache wird häufig in Kombination mit einem Read-through-Cache verwendet.

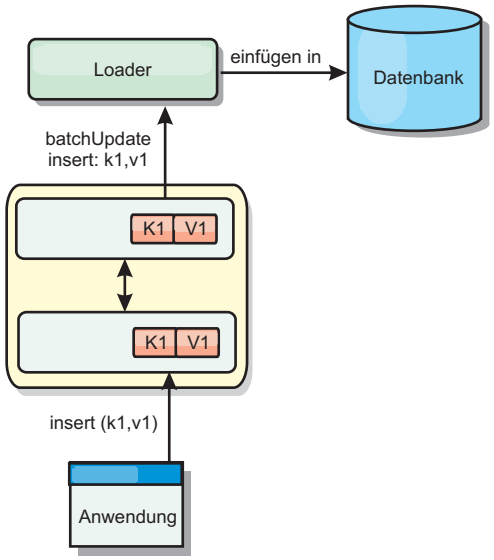


Abbildung 12. Write-Through-Caching

Szenario mit Write-behind-Caching

Die Datenbanksynchronisation kann verbessert werden, indem Änderungen asynchron geschrieben werden. Dies wird als Write-behind- oder Write-back-Cache

(Rückschreibcache) bezeichnet. Änderungen, die normalerweise synchron in den Loader geschrieben werden, werden stattdessen in eXtreme Scale gepuffert und über einen Hintergrund-Thread in die Datenbank geschrieben. Die Schreibleistung wird erheblich verbessert, weil die Datenbankoperation aus der Clienttransaktion entfernt wird und die Schreibvorgänge in die Datenbank komprimiert werden können.

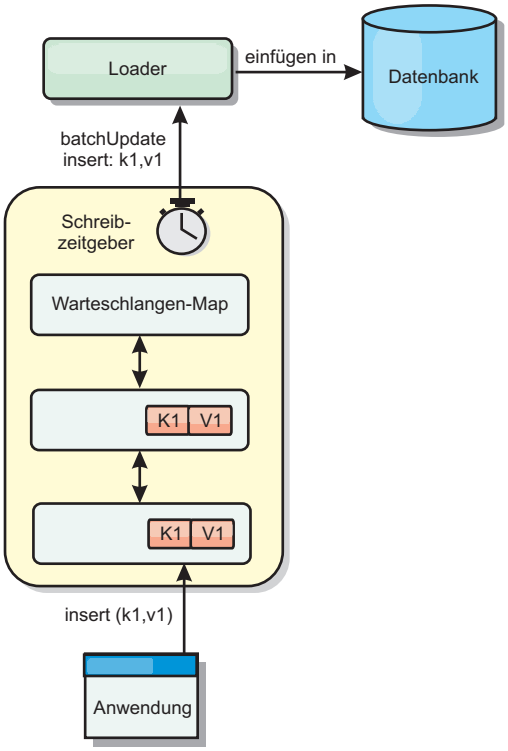


Abbildung 13. Write-behind-Caching

Write-behind-Caching

Sie können Write-behind-Caching verwenden, um die Kosten für die Aktualisierung einer Datenbank, die Sie als Back-End verwenden, zu reduzieren.

Übersicht über das Write-behind-Caching

Beim Write-behind-Caching werden Aktualisierungen für das Loader-Plug-in asynchron in die Warteschlange eingereiht. Sie können die Leistung von Aktualisierungs-, Einfüge- und Entfernungsoperationen für die Map verbessern, indem Sie die eXtreme-Scale-Transaktion von der Datenbanktransaktion entkoppeln. Die asynchrone Aktualisierung wird nach einer zeitbasierten Verzögerung (z. B. fünf Minuten) oder einer eintragsbasierten Verzögerung (z. B. 1000 Einträge) durchgeführt.

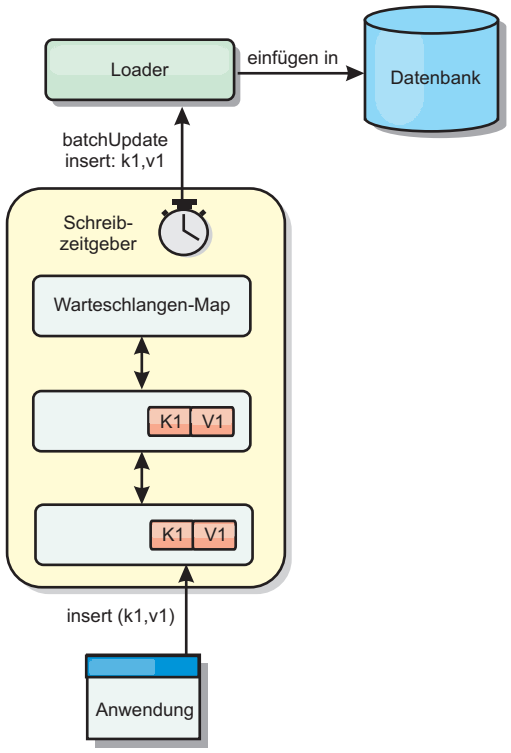


Abbildung 14. Write-behind-Caching

Bei der Write-behind-Konfiguration in einer BackingMap wird ein Thread zwischen dem Loader (Ladeprogramm) und der Map erstellt. Anschließend delegiert der Loader Datenanforderungen über den Thread gemäß den Konfigurationseinstellungen in der Methode "BackingMap.setWriteBehind". Wenn eine eXtreme-Scale-Transaktion einen Eintrag in einer Map einfügt, aktualisiert oder entfernt, wird ein LogElement-Objekt für jeden dieser Datensätze erstellt. Diese Elemente werden an den Write-behind-Loader gesendet und in eine spezielle ObjectMap, eine so genannte Warteschlangen-Map, eingereiht. Jede BackingMap mit aktivierter Write-behind-Einstellung hat ihre eigenen Warteschlangen-Maps. Ein Write-behind-Thread entfernt die in die Warteschlange eingereihten Daten aus den Warteschlangen-Maps und überträgt Sie mit Push in den echten Back-End-Loader.

Der Write-behind-Loader sendet nur LogElement-Objekte der Typen "insert" (Einfügen), "update" (Aktualisieren) und "delete" (Löschen) an den echten Loader. Alle anderen Typen von LogElement-Objekten, wie z. B. EVICT, werden ignoriert.

Die Write-behind-Unterstützung ist eine Erweiterung des Loader-Plug-ins, das Sie verwenden, um eXtreme Scale mit der Datenbank zu integrieren. Sehen Sie sich beispielsweise die Informationen zur Konfiguration eines JPA-Loaders im Abschnitt „JPA-Loader konfigurieren” auf Seite 361 an.

Vorteile

Das Aktivieren der Write-behind-Unterstützung hat die folgenden Vorteile:

- **Isolation von Back-End-Fehlern:** Durch das Write-behind-Caching können Back-End-Fehler isoliert werden. Wenn die Back-End-Datenbank ausfällt, werden Aktualisierungen in die Warteschlangen-Map eingereiht. Die Anwendungen können

weiterhin Transaktionen an eXtreme Scale senden. Nach der Wiederherstellung des Back-Ends werden die Daten in der Warteschlangen-Map mit Push an das Back-End übertragen.

- **Geringere Back-End-Last:** Der Write-behind-Loader fasst die Aktualisierungen auf Schlüsselbasis so zusammen, dass nur eine einzige zusammenfasste Aktualisierung pro Schlüssel in der Warteschlangen-Map vorhanden ist. Bei dieser Zusammenfassung verringert sich die Anzahl der Aktualisierungen für die Back-End-Datenbank.
- **Verbesserte Transaktionsleistung:** Die Zeiten einzelner eXtreme-Scale-Transaktionen verringern sich, weil sie nicht auf die Synchronisation der Daten mit dem Back-End warten müssen.

Hinweise zum Anwendungsdesign

Das Aktivieren der Write-behind-Unterstützung ist zwar einfach, aber eine Anwendung mit Write-behind-Unterstützung zu entwerfen, bedarf sorgfältiger Überlegungen. Ohne Write-behind-Unterstützung ist die Back-End-Transaktion in die ObjectGrid-Transaktion eingeschlossen. Die ObjectGrid-Transaktion wird vor der Back-End-Transaktion gestartet und endet erst nach Abschluss der Back-End-Transaktion.

Wenn die Write-behind-Unterstützung aktiviert ist, endet die ObjectGrid-Transaktion vor dem Start der Back-End-Transaktion. Die ObjectGrid-Transaktion und die Back-End-Transaktion sind entkoppelt.

Referenzielle Integritätsbedingungen

Jede BackingMap, die mit Write-behind-Unterstützung konfiguriert ist, hat einen eigenen Write-behind-Thread, der die Daten mit Push an das Back-End überträgt. Deshalb werden die Daten, die in einer einzigen ObjectGrid-Transaktion in verschiedenen Maps aktualisiert wurden, im Back-End in verschiedenen Back-End-Transaktionen aktualisiert. Beispiel: Transaktion T1 aktualisiert den Schlüssel "key1" in der Map "Map1" und den Schlüssel "key2" in der Map "Map2". Die Aktualisierungen von Schlüssel key1 in der Map Map1 und von Schlüssel "key2" in der Map "Map2" werden in einer jeweils anderen Back-End-Transaktion von einem jeweils anderen Write-behind-Thread durchgeführt. Wenn es Beziehungen zwischen den in Map1 und Map2 gespeicherten Daten, wie z. B. Integritätsbedingungen über Fremdschlüssel, im Back-End gibt, können die Aktualisierungen fehlschlagen.

Beim Design der referenziellen Integritätsbedingungen in Ihrer Back-End-Datenbank müssen Sie sicherstellen, dass solche nicht ausführbaren Aktualisierungen zugelassen werden.

Sperrverhalten von Warteschlangen-Maps

Ein weiterer wichtiger Unterschied im Transaktionsverhalten ist das Sperrverhalten. ObjectGrid unterstützt drei verschiedene Sperrstrategien: PESSIMISTIC (Pessimistisch), OPTIMISITIC (Optimistisch) und NONE (Keine). Die Write-behind-Warteschlangen-Map verwendet die pessimistische Sperrstrategie, unabhängig davon, welche Sperrstrategie für die zugehörige BackingMap konfiguriert ist. Es gibt zwei verschiedene Typen von Operationen, die eine Sperre für die Warteschlangen-Map anfordern:

- Wenn eine ObjectGrid-Transaktion festgeschrieben wird oder eine Flush-Operation (Map-Flush oder Sitzungs-Flush) stattfindet, liest die Transaktion den Schlüssel in der Warteschlangen-Map und setzt eine S-Sperre für den Schlüssel.

- Wenn eine ObjectGrid-Transaktion festgeschrieben wird, versucht die Transaktion die S-Sperre für den Schlüssel in eine X-Sperre zu aktualisieren.

Anhand dieses zusätzlichen Verhaltens für die Warteschlangen-Map sind einige Unterschiede im Sperrverhalten erkennbar.

- Wenn die Benutzer-Map mit einer pessimistischen Sperrstrategie konfiguriert ist, sind die Unterschiede im Sperrverhalten nicht gravierend. Bei jedem Aufruf einer Flush- oder Festschreiboperation (Commit) wird eine S-Sperre für denselben Schlüssel in der Warteschlangen-Map gesetzt. Während der Festschreibung wird nicht nur eine X-Sperre für den Schlüssel in der Benutzer-Map, sondern auch für den Schlüssel in der Warteschlangen-Map angefordert.
- Wenn die Benutzer-Map mit einer optimistischen Sperrstrategie oder ohne Sperrstrategie konfiguriert ist, folgt die Benutzertransaktion dem Muster der pessimistischen Sperrstrategie. Bei jedem Aufruf einer Flush- oder Festschreiboperation (Commit) wird eine S-Sperre für denselben Schlüssel in der Warteschlangen-Map angefordert. Während der Festschreibung wird in derselben Transaktion eine X-Sperre für den Schlüssel in der Warteschlangen-Map angefordert.

Transaktionswiederholungen im Loader

ObjectGrid unterstützt keine zweiphasigen Transaktionen und keine XA-Transaktionen. Der Write-behind-Thread entfernt Datensätze aus der Warteschlangen-Map und aktualisiert die Datensätze im Back-End. Wenn der Server mitten in der Transaktion ausfällt, können einige Back-End-Aktualisierungen verloren gehen.

Der Write-behind-Loader versucht automatisch, fehlgeschlagene Transaktionen erneut zu schreiben, und sendet eine unbestätigte Protokollfolge an das Back-End, um einen Datenverlust zu verhindern. Diese Aktion erfordert, dass der Loader idempotent ist, d. h., wenn Loader.batchUpdate(TxId, LogSequence) zweimal mit demselben Wert aufgerufen wird, liefen diese Aufrufe dasselbe Ergebnis wie ein einmaliger Aufruf. Loader-Implementierungen müssen zum Aktivieren dieses Features die Schnittstelle "RetryableLoader" implementieren. Weitere Einzelheiten finden Sie in der API-Dokumentation.

Ausfall des Loaders

Das Loader-Plug-in kann ausfallen, wenn es nicht mit dem Datenbank-Back-End kommunizieren kann. Dies kann passieren, wenn der Datenbankserver oder die Netzverbindung inaktiv ist. Der Write-behind-Loader reiht die Aktualisierungen in eine Warteschlange ein und versucht anschließend in regelmäßigen Abständen, die Datenänderungen mit Push an den Loader zu übertragen. Der Loader muss die ObjectGrid-Laufzeitumgebung darüber benachrichtigen, dass ein Problem mit der Datenbankkonnektivität vorliegt, indem es eine Ausnahme vom Typ "LoaderNotAvailableException" auslöst.

Deshalb muss die Loader-Implementierung in der Lage sein, einen Datenfehler von einem physischen Ausfall des Loaders zu unterscheiden. Bei Datenfehlern muss eine Ausnahme des Typs "LoaderException" oder "OptimisticCollisionException" ausgelöst bzw. erneut ausgelöst werden, aber beim physischen Ausfall des Loaders muss eine Ausnahme des Typs "LoaderNotAvailableException" ausgelöst werden. ObjectGrid behandelt diese beiden Ausnahmen auf unterschiedliche Weise:

- Wenn der Write-behind-Loader eine Ausnahme vom Typ "LoaderException" abfängt, geht er von einem Datenfehler aus, z. B. von einem doppelten Schlüssel. Der Write-behind-Loader löst den Aktualisierungsstapel auf und versucht, einen Datensatz nach dem anderen zu aktualisieren, um den Datenfehler zu isolieren.

Wird bei dieser Aktualisierung auf Datensatzbasis erneut eine Ausnahme vom Typ "LoaderException" abgefangen, wird ein Datensatz zur fehlgeschlagenen Aktualisierung erstellt und in der Map für fehlgeschlagene Aktualisierungen protokolliert.

- Wenn das Write-Behind-Ladeprogramm eine Ausnahme vom Typ "LoaderNotAvailableException" abfängt, geht es von einem Ausfall aus, weil es keine Verbindung zum Datenbank-Back-End herstellen kann, z. B., weil das Datenbank-Back-End inaktiv ist, keine Datenbankverbindung verfügbar oder das Netz inaktiv ist. Der Write-behind-Loader wartet 15 Sekunden und versucht dann erneut, die Datenbankaktualisierung im Stapelbetrieb durchzuführen.

Häufig wird der Fehler gemacht, eine Ausnahme vom Typ "LoaderException" auszulösen, obwohl eigentlich eine Ausnahme vom Typ "LoaderNotAvailableException" ausgelöst werden müsste. Alle Datensätze, die in die Warteschlange für den Write-behind-Loader eingereiht sind, werden als Datensätze für eine fehlgeschlagene Aktualisierung markiert, was den eigentlich Zweck der Isolierung von Back-End-Fehlern zunichte macht.

Leistungsaspekte

Die Unterstützung des Write-behind-Cachings erhöht die Antwortzeiten, weil die Loader-Aktualisierung aus der Transaktion entfernt wird. Außerdem erhöht sich der Datenbankdurchsatz, weil Datenbankaktualisierungen kombiniert werden. Es ist wichtig, die Kosten zu kennen, die durch den Write-behind-Thread anfallen, der die Daten aus der Warteschlangen-Map extrahiert und mit Push an den Loader überträgt.

Die maximale Aktualisierungsanzahl und die maximale Aktualisierungszeit müssen den erwarteten Verwendungsmustern und der Umgebung entsprechend angepasst werden. Wenn der Wert für die maximale Aktualisierungsanzahl oder der Wert für die maximale Aktualisierungszeit zu klein gewählt wird, kann der Write-behind-Threads mehr Kosten verursachen, als er Vorteile bringt. Wenn ein sehr hoher Wert für diese beiden Parameter festgelegt wird, ist es möglich, dass die Speicherbelegung aufgrund der Einreihung der Daten zunimmt und veraltete Datensätze länger in der Datenbank verbleiben.

Um die beste Leistung zu erzielen, sollten Sie bei der Optimierung der Write-behind-Parameter die folgenden Faktoren berücksichtigen:

- Verhältnis zwischen Lese- und Schreibtransaktionen
- Aktualisierungsintervall für dieselben Datensätze
- Latenzzeit für Datenbankaktualisierung

Loader

Mit einem Loader-Plug-in kann sich eine Datengrid-Map wie ein Speichercache für Daten verhalten, die gewöhnlich in einem persistenten Speicher auf demselben System oder einem anderen System gespeichert werden. Gewöhnlich wird eine Datenbank oder ein Dateisystem als persistenter Speicher verwendet. Es kann auch eine ferne Java Virtual Machine (JVM) als Datenquelle verwendet werden, was die Erstellung Hub-basierter Caches mit eXtreme Scale ermöglicht. Ein Loader enthält die Logik für das Lesen aus einem und das Schreiben in einem persistenten Speicher.

Übersicht

Loader (Ladeprogramme) sind BackingMap-Plug-ins, die aufgerufen werden, wenn Änderungen an der BackingMap vorgenommen werden oder wenn die Backing-Map eine Datenanforderung nicht bedienen kann (Cachefehler). Der Loader wird aufgerufen, wenn der Cache eine Anforderung für einen Schlüssel nicht bedienen kann. Er unterstützt Read-through-Funktionen und eine verzögerte Füllung des Caches. Ein Loader lässt außerdem Aktualisierungen in der Datenbank zu, wenn sich Cachewerte ändern. Alle Änderungen in einer Transaktion werden gruppiert, um die Anzahl der Datenbankinteraktionen zu minimieren. Zusammen mit dem Loader wird ein TransactionCallback-Plug-in verwendet, um die Abgrenzung der Back-End-Transaktion auszulösen. Die Verwendung dieses Plug-ins ist wichtig, wenn mehrere Maps an einer einzelnen Transaktion beteiligt sind oder wenn Transaktionsdaten ohne Festschreibung mit Flush in den Cache übertragen werden.

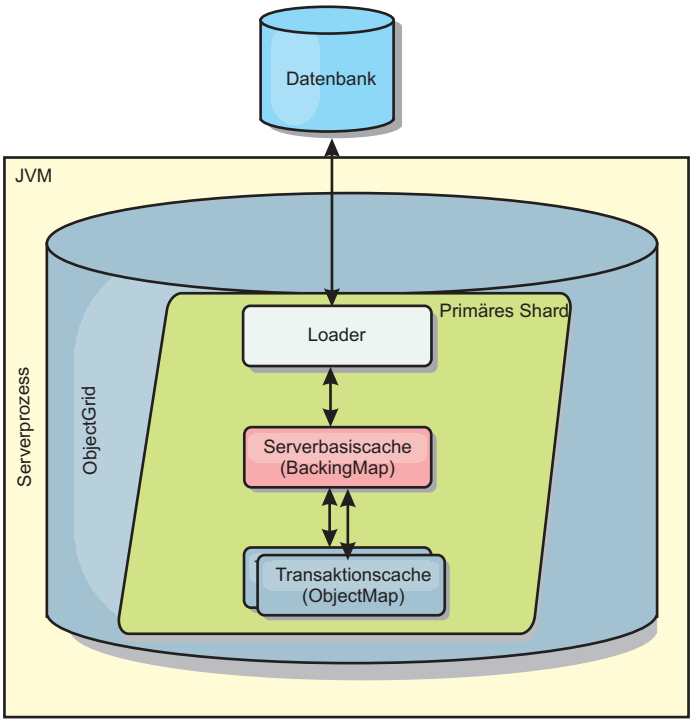


Abbildung 15. Loader

Der Loader kann auch überqualifizierte Aktualisierungen verwenden, um keine Datenbanksperren halten zu müssen. Anhand eines im Cachewert gespeicherten Versionsattributs kann der Loader das Vorher- und Nachher-Abbild des Werts erkennen, wenn dieser im Cache aktualisiert wird. Dieser Wert kann anschließend bei der Aktualisierung der Datenbank bzw. des Back-Ends verwendet werden, um sicherzustellen, dass die Daten nicht aktualisiert wurden. Ein Loader kann auch so konfiguriert werden, dass das Datengrid beim Start vorher geladen wird. Wenn mit Partitionierung gearbeitet wird, wird jeder Partition eine Loader-Instanz zugeordnet. Hat die Map "Company" beispielsweise zehn Partitionen, gibt es zehn Loader-Instanzen, eine für jede primäre Partition. Bei der Aktivierung des primären Shards für die Map wird die Methode "preloadMap" für den Loader synchron oder asynchron aufgerufen. Dies ermöglicht das automatische Laden von Daten aus dem Back-End in die Map-Partition. Wenn die Methode synchron aufgerufen wird, werden alle Clienttransaktionen blockiert, um einen inkonsistenten Zugriff auf das

Datengrid zu verhindern. Alternativ kann ein Client-Preloader zum Laden des vollständigen Datengrids verwendet werden.

Es gibt zwei integrierte Loader, die die Integration mit relationalen Datenbank-Back-Ends erheblich vereinfachen. Die JPA-Loader nutzen die ORM-Funktionen (Object-Relational Mapping, objektrelationale Abbildung) der OpenJPA- und Hibernate-Implementierungen der Spezifikation Java Persistence API (JPA). Weitere Informationen finden Sie unter JPA-Loader.

Wenn Sie Loader in einer Konfiguration mit mehreren Rechenzentren verwenden, müssen Sie berücksichtigen, wie Revisionsinformationen und Cachekonsistenz zwischen den Datengrids verwaltet wird. Weitere Informationen finden Sie im Abschnitt „Hinweise zu Ladeprogrammen in einer Multimastertopologie" auf Seite 41.

Loader-Konfiguration

Wenn Sie der BackingMap-Konfiguration einen Loader hinzufügen möchten, können Sie die programmgesteuerte Konfiguration oder die XML-Konfiguration verwenden. Ein Loader steht mit einer BackingMap in folgender Beziehung.

- Eine BackingMap kann nur einen einzigen Loader haben.
- Eine Client-BackingMap (naher Cache) kann keinen Loader haben.
- Eine Loader-Definition kann auf mehrere BackingMaps angewendet werden, aber jede BackingMap hat eine eigene Loader-Instanz.

Vorheriges Laden von Daten und Vorbereitung

In vielen Szenarien, die die Verwendung eines Loaders (Ladeprogramms) beinhalten, können Sie Ihr Datengrid durch vorheriges Laden von Daten (Preload) vorbereiten.

Wenn das Grid als vollständiger Cache verwendet wird, muss das Datengrid alle Daten aufnehmen und geladen werden, bevor Clients eine Verbindung zum Grid herstellen können. Wenn Sie einen Teilcache verwenden, müssen Sie den Cache mit Daten vorbereiten (Aufwärmphase), so dass Clients sofortigen Zugriff auf die Daten haben, wenn sie eine Verbindung zum Grid herstellen.

Es gibt zwei Methoden für das vorherige Laden von Daten in das Datengrid: Verwendung eines Loader-Plug-ins (Ladeprogramm) oder Verwendung eines Client-Loaders. Diese beiden Methoden werden in den folgenden Abschnitten beschrieben.

Loader-Plug-in

Das Loader-Plug-in wird jeder Map zugeordnet und ist für die Synchronisation eines einzelnen primären Partitions-Shards mit der Datenbank zuständig. Die Methode "preloadMap" des Loader-Plug-ins wird automatisch aufgerufen, wenn ein Shard aktiviert wird. Wenn Sie beispielsweise 100 Partitionen haben, sind 100 Loader-Instanzen vorhanden, die jeweils die Daten für ihre Partition laden. Wenn die Loader-Instanzen synchron ausgeführt werden, werden alle Clients blockiert, bis das vorherige Laden der Daten (der so genannte Preload-Prozess) abgeschlossen ist.

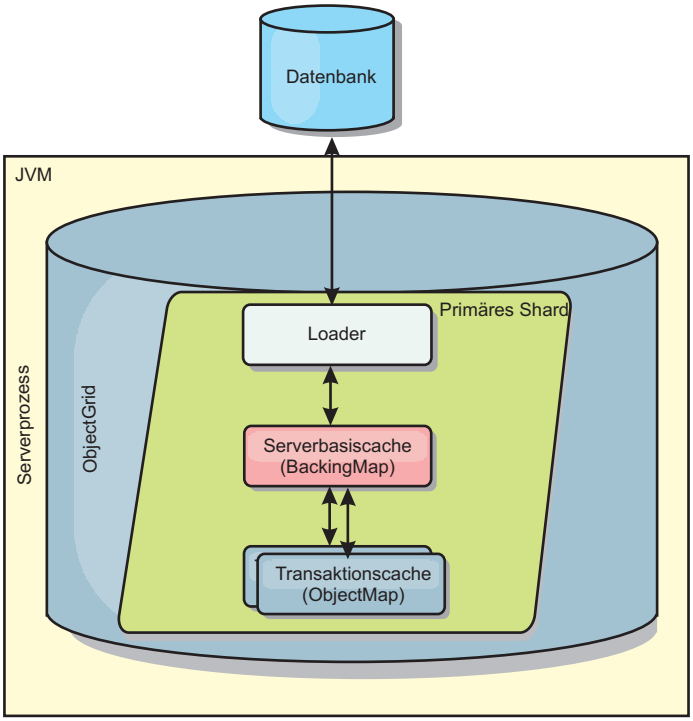


Abbildung 16. Loader-Plug-in

Client-Loader

Ein Client-Loader ist ein Muster für die Verwendung eines oder mehrerer Clients, um Daten in das Grid zu laden. Die Verwendung mehrerer Clients zum Laden von Griddaten kann effektiv sein, wenn das Partitionsschema nicht in der Datenbank gespeichert ist. Sie können Client-Loader manuell oder automatisch aufrufen, wenn das Datengrid gestartet wird. Client-Loader können optional die Schnittstelle "StateManager" verwenden, um den Status des Datengrids auf den Preload-Modus zu setzen, so dass Clients nicht auf das Grid zugreifen können, wenn das vorherige Laden der Daten in das Grid durchgeführt wird. WebSphere eXtreme Scale enthält einen JPA-basierten (Java Persistence API) Loader, den Sie verwenden können, um das Datengrid automatisch über die OpenJPA- oder Hibernate-JPA-Provider zu laden. Weitere Informationen zu Cache-Providern finden Sie unter „JPA-L2-Cache-Plug-in” auf Seite 339.

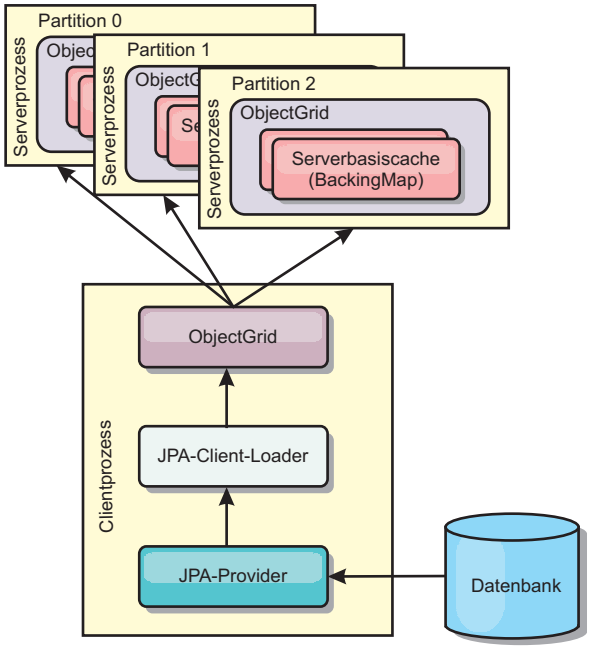


Abbildung 17. Client-Loader

Verfahren für die Datenbanksynchronisation

Wenn WebSphere eXtreme Scale als Cache verwendet wird, müssen Anwendungen so geschrieben werden, dass veraltete Daten toleriert werden, wenn die Datenbank unabhängig von einer eXtreme-Scale-Transaktion aktualisiert werden kann. Für den Einsatz als Verarbeitungsbereich für die synchronisierte speicherinterne Datenbank stellt eXtreme Scale mehrere Methoden für die konstante Aktualisierung des Caches bereit.

Verfahren für die Datenbanksynchronisation

Regelmäßige Aktualisierung

Der Cache kann mit Hilfe der zeitbasierten JPA-Datenbankaktualisierungskomponente (Java Persistence API) automatisch ungültig gemacht oder regelmäßig aktualisiert werden. Die Aktualisierungskomponente fragt die Datenbank in regelmäßigen Abständen über einen JPA-Provider nach Aktualisierungen oder Einfügungen ab, die seit der vorherigen Aktualisierung vorgenommen wurden. Alle gefundenen Änderungen werden automatisch ungültig gemacht oder aktualisiert, wenn ein Teilcache verwendet wird. Wenn ein vollständiger Cache verwendet wird, können die Einträge erkannt und in den Cache eingefügt werden. Es werden keine Einträge aus dem Cache entfernt.

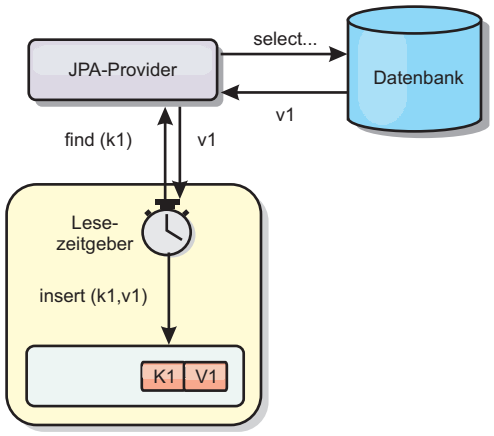


Abbildung 18. Regelmäßige Aktualisierung

Bereinigung

Teilcaches können Bereinigungsrichtlinien verwenden, um Daten ohne Beeinträchtigung der Datenbank automatisch aus dem Cache zu entfernen. Mit eXtreme Scale werden drei integrierte Richtlinien bereitgestellt: Lebensdauer (TTL, Time-to-Live), LRU (least recently used) und LFU (last frequently used). Alle drei Richtlinien können Daten aggressiver entfernen, wenn Speicherengpässe auftreten, indem die Option für speicherbasierte Bereinigung aktiviert wird.

Ereignisbasiertes Ungültigmachen

Teilcaches und vollständige Caches können mit Hilfe eines Ereignisgenerators wie Java Message Service (JMS) ungültig gemacht oder aktualisiert werden. Das Ungültigmachen mit JMS kann manuell an jeden Prozess gebunden werden, der das Back-End über einen Datenbankauslöser aktualisiert. Es wird ein JMS-ObjectGridEventListener-Plug-in in eXtreme Scale bereitgestellt, das Clients benachrichtigen kann, wenn Änderungen im Servercache vorgenommen wurden. Auf diese Weise kann das Zeitfenster, in dem der Client veraltete Daten sieht, verringert werden.

Programmgesteuertes Ungültigmachen

Die APIs von eXtreme Scale unterstützen die manuelle Interaktion zwischen nahem Cache und Servercache über die API-Methoden "Session.beginNoWriteThrough()", "ObjectMap.invalidate()" und "EntityManager.invalidate()". Wenn ein Client- oder Serverprozess einen Teil der Daten nicht mehr benötigt, können Sie mit den Methoden zum Ungültigmachen Daten aus dem nahen Cache bzw. Servercache entfernen. Die Methode "beginNoWriteThrough" gilt für alle ObjectMap- und EntityManager-Operationen im lokalen Cache ohne Aufruf des Loaders. Wenn die Methode von einem Client aufgerufen wird, gilt die Operation nur für den nahen Cache (der ferne Loader wird nicht aufgerufen). Wird die Methode im Server aufgerufen, gilt die Operation nur für den Serverbasiscache ohne Aufruf des Loaders.

Dateninvalidierung

Zum Entfernen von Cachedaten können Sie einen ereignisbasierten oder programmgesteuerten Invalidierungsmechanismus verwenden.

Ereignisgesteuertes Ungültigmachen

Teilcaches und vollständige Caches können mit Hilfe eines Ereignisgenerators wie Java Message Service (JMS) ungültig gemacht oder aktualisiert werden. Das Ungültigmachen mit JMS kann manuell an jeden Prozess gebunden werden, der das Back-End über einen Datenbankauslöser aktualisiert. Es wird ein JMS-ObjectGridEventListener-Plug-in in eXtreme Scale bereitgestellt, das Clients benachrichtigen kann, wenn Änderungen im Servercache vorgenommen wurden. Dieser Typ von Benachrichtigung verkleinert das Zeitfenster, in dem der Client veraltete Daten sieht.

Der ereignisgesteuerte Mechanismus für Ungültigmachen setzt sich gewöhnlich aus den folgenden drei Komponenten zusammen:

- **Ereigniswarteschlange:** In einer Ereigniswarteschlange werden die Datenänderungsereignisse gespeichert. Die Ereigniswarteschlange kann eine JMS-Warteschlange, eine Datenbank, eine speicherinterne FIFO-Warteschlange oder ein beliebiges Manifest sein, das Datenänderungsereignisse verwalten kann.
- **Ereignis-Publisher:** Ein Ereignis-Publisher veröffentlicht die Datenänderungsereignisse in der Ereigniswarteschlange. Ein Ereignis-Publisher ist gewöhnlich eine Anwendung, die Sie erstellen, oder eine Implementierung eines eXtreme-Scale-Plug-ins. Der Ereignis-Publisher weiß, wann die Daten geändert werden, oder ändert die Daten selbst. Wenn eine Transaktion festgeschrieben wird, werden Ereignisse für die geänderten Daten generiert, und der Ereignis-Publisher veröffentlicht diese Ereignisse in der Ereigniswarteschlange.
- **Ereigniskonsument:** Ein Ereigniskonsument konsumiert Datenänderungsereignisse. Der Ereigniskonsument ist gewöhnlich eine Anwendung, die sicherstellt, das die Daten im Ziel-Grid mit den neuesten Änderungen aus anderen Grids aktualisiert werden. Dieser Ereigniskonsument interagiert mit der Ereigniswarteschlange, um die neuesten Datenänderungen abzurufen, und wendet die Datenänderungen auf das Ziel-Grid an. Die Ereigniskonsumenten können APIs von eXtreme Scale verwenden, um veraltete Daten ungültig zu machen oder um das Grid mit den neuesten Daten zu aktualisieren.

JMSObjectGridEventListener hat beispielsweise eine Option für ein Client/Server-Modell, bei der die Ereigniswarteschlange eine festgelegte JMS-Destination ist. Alle Serverprozesse sind Ereignis-Publisher. Wenn eine Transaktion festgeschrieben wird, ruft der Server die Datenänderungen ab und veröffentlicht sie in der festgelegten JMS-Destination. Alle Clientprozesse sind Ereigniskonsumenten. Sie empfangen Datenänderungen von der festgelegten JMS-Destination und wenden die Änderungen auf den nahen Cache des Clients an.

Weitere Informationen finden Sie im Abschnitt zum Aktivieren des Mechanismus für das Ungültigmachen von Clients im *Administratorhandbuch*.

Programmgesteuertes Ungültigmachen

Die APIs von WebSphere eXtreme Scale unterstützen die manuelle Interaktion zwischen nahem Cache und Servercache über die API-Methoden "Session.beginNoWriteThrough()", "ObjectMap.invalidate()" und "EntityManager.invalidate()". Wenn ein Client- oder Serverprozess einen Teil der Daten nicht mehr benötigt, können Sie mit den Methoden zum Ungültigmachen Daten aus dem nahen Cache bzw. Servercache entfernen. Die Methode "beginNoWriteThrough" gilt für alle ObjectMap- und EntityManager-Operationen im lokalen Cache ohne Aufruf des Loaders. Wenn die Methode von einem Client aufgerufen wird, gilt die Operation nur für den nahen

Cache (der ferne Loader wird nicht aufgerufen). Wird die Methode im Server aufgerufen, gilt die Operation nur für den Serverbasiscache ohne Aufruf des Loaders.

Sie können den Mechanismus für programmgesteuertes Ungültigmachen zusammen mit anderen Techniken verwenden, um festzustellen, wann die Daten ungültig gemacht werden müssen. Diese Methode für Ungültigmachen verwendet beispielsweise ereignisgesteuerte Mechanismen für das Ungültigmachen, um die Datenänderungsereignisse zu empfangen, und anschließend APIs, um die veralteten Daten ungültig zu machen.

Indexierung

Verwenden Sie das Plug-in "MapIndexPlugin", um einen Index oder mehrere Indizes in einer BackingMap für die Unterstützung von Datenzugriffen ohne Schlüssel zu erstellen.

Indextypen und Konfiguration

Das Indexierungsfeature wird durch das Plug-in "MapIndexPlugin" oder kurz "Index" dargestellt. Index ist ein BackingMap-Plug-in. Für eine BackingMap können mehrere Index-Plug-ins konfiguriert werden, solange jedes Plug-in den Index-Konfigurationsregeln entspricht.

Sie können das Indexierungsfeature verwenden, um einen oder mehrere Indizes in einer BackingMap zu erstellen. Ein Index wird aus einem Attribut oder einer Liste von Attributen eines Objekts in der BackingMap erstellt. Das Feature bietet Anwendungen eine Möglichkeit, bestimmte Objekte schneller zu finden. Mit dem Indexierungsfeature können Anwendungen mit einem bestimmten Wert oder innerhalb eines bestimmten Wertebereichs indexierter Attribute finden.

Es gibt zwei Typen von Indexierung: statische Indexierung und dynamische Indexierung. Bei der statischen Indexierung müssen Sie das Index-Plug-in in der BackingMap konfigurieren, bevor Sie die ObjectGrid-Instanz initialisieren. Sie können diese Konfiguration durch XML- oder programmgesteuerte Konfiguration der BackingMap vornehmen. Die statische Indexierung beginnt mit der Erstellung eines Index während der ObjectGrid-Initialisierung. Der Index ist immer mit der BackingMap synchronisiert und zur Verwendung bereit. Nach dem Start des statischen Indexierungsprozesses erfolgt die Verwaltung des Index im Rahmen des Transaktionsverwaltungsprozesses von eXtreme Scale. Wenn Transaktionen Änderungen festschreiben, werden diese Änderungen auch im statischen Index durchgeführt, und Indexänderungen werden rückgängig gemacht, wenn die Transaktion rückgängig gemacht wird.

Bei der dynamischen Indexierung können Sie einen Index in einer BackingMap vor oder nach der Initialisierung der übergeordneten ObjectGrid-Instanz erstellen. Anwendungen haben eine Lebenszykluskontrolle über den dynamischen Indexierungsprozess, d. h., Sie können einen dynamischen Index entfernen, wenn er nicht mehr benötigt wird. Wenn eine Anwendung einen dynamischen Index erstellt, ist der Index möglicherweise nicht zur sofortigen Verwendung bereit, weil die Erstellung des Index eine gewisse Zeit dauert. Da die Erstellungsdauer vom Volumen der zu indexierenden Daten abhängig ist, wird die Schnittstelle DynamicIndexCallback für Anwendungen bereitgestellt, die Benachrichtigungen empfangen möchten, wenn bestimmte Indexierungsereignisse eintreten. Zu diesen Ereignissen gehören die Bereitschaft des Index (ready), Fehler (error) und das Löschen des Index (destroy). Anwendungen können diese Callback-Schnittstelle implementieren und sich beim dynamischen Indexierungsprozess registrieren.

Wenn eine BackingMap ein konfiguriertes Index-Plug-in hat, können Sie das Proxy-Objekt für den Anwendungsindex von der entsprechenden ObjectMap abrufen. Wenn Sie die Methode getIndex in der Schnittstelle "ObjectMap" aufrufen und den Namen des Index-Plug-ins übergeben, wird das Index-Proxy-Objekt zurückgegeben. Sie müssen das Index-Proxy-Objekt in die entsprechende Anwendungsindexschnittstelle, z. B. MapIndex, MapRangeIndex oder eine angepasste Indexschnittstelle, umsetzen. Nach dem Abrufen des Index-Proxy-Objekts können Sie in der Anwendungsindexschnittstelle definierte Methoden verwenden, um zwischengespeicherte Objekte zu suchen.

Die Schritte zur Verwendung der Indexierung sind in der folgenden Liste zusammengefasst:

- Fügen Sie statische oder dynamische Index-Plug-ins in der BackingMap hinzu.
- Rufen Sie mit der Methode "getIndex" von ObjectMap ein Proxy-Objekt für den Anwendungsindex ab.
- Setzen Sie das Proxy-Objekt für den Index in eine entsprechende Anwendungsindexschnittstelle um, wie z. B. MapIndex, MapRangeIndex oder eine angepasste Indexschnittstelle.
- Verwenden Sie die in der Anwendungsindexschnittstelle definierten Methoden, um zwischengespeicherte Objekte zu suchen.

Die Klasse HashIndex ist die integrierte Index-Plug-in-Implementierung, die beide integrierten Anwendungsindexschnittstellen, MapIndex und MapRangeIndex, unterstützen kann. Sie können auch eigene Indizes erstellen. Sie können HashIndex als statischen oder dynamischen Index in der BackingMap hinzufügen, ein MapIndex- oder MapRangeIndex-Index-Proxy-Objekt abrufen und das Index-Proxy-Objekt zum Suchen zwischengespeicherter Objekte verwenden.

Standardindex

Wenn Sie durch die Schlüssel in einer lokalen Map iterieren möchten, können Sie den Standardindex verwenden. Dieser Index erfordert keine Konfiguration, aber er muss für das Shard über einen Agenten oder eine ObjectGrid-Instanz, die mit der Methode ShardEvents.shardActivated(ObjectGrid shard) abgerufen wird, verwendet werden.

Hinweis zur Datenqualität

Die Ergebnisse der Indexabfragemethoden stellen nur eine Momentaufnahme der Daten zu einem bestimmten Zeitpunkt dar. Es werden keine Sperren für Dateneinträge angefordert, nachdem die Ergebnisse an die Anwendung zurückgegeben wurden. Die Anwendung muss sich darüber im Klaren sein, dass Datenaktualisierungen für eine zurückgegebene Datengruppe vorgenommen werden können. Beispiel: Die Anwendung ruft den Schlüssel eines zwischengespeicherten Objekts mit der Methode findAll von MapIndex ab. Dieses zurückgegebene Schlüsselobjekt ist einem Dateneintrag im Cache zugeordnet. Die Anwendung muss in der Lage sein, die Methode "get" in ObjectMap auszuführen, um ein Objekt durch Übergabe des Schüsselobjekts zu suchen. Wenn eine andere Transaktion das Datenobjekt aus dem Cache entfernt, kurz bevor die Methode "get" aufgerufen wird, ist das zurückgegebene Ergebnis null.

Hinweise zur Leistung der Indexierung

Eine der Hauptzielsetzungen des Indexierungsfeatures ist die Verbesserung der Gesamtleistung der BackingMap. Wenn die Indexierung nicht ordnungsgemäß ver-

wendet wird, kann dies die Leistung der Anwendung beeinträchtigen. Berücksichtigen Sie vor der Verwendung dieses Features die folgenden Faktoren.

- **Anzahl gleichzeitiger Transaktionen mit Schreibzugriff:** Die Indexverarbeitung kann jedesmal stattfinden, wenn eine Transaktion Daten in eine BackingMap schreibt. Schreiben viele Transaktionen gleichzeitig Daten in die Map, kann es zu Leistungseinbußen kommen, wenn eine Anwendung versucht, Indexabfrageoperationen durchzuführen.
- **Größe der von einer Abfrageoperation zurückgegebenen Ergebnismenge:** Je größer die Ergebnismenge wird, desto mehr nimmt die Abfrageleistung ab. Ab einer Ergebnismengengröße von 15 % der Gesamtgröße der BackingMap beginne sich Leistungseinbußen abzuzeichnen.
- **Anzahl der für dieselbe BackingMap erstellten Indizes:** Jeder Index belegt Systemressourcen. Mit steigender Indexanzahl für die BackingMap nimmt die Leistung ab.

Die Indexierungsfunktion kann die Leistung einer BackingMap erheblich verbessern. Die besten Ergebnisse lassen sich erzielen, wenn hauptsächlich Leseoperationen für die BackingMap durchgeführt werden, wenn die Abfrageergebnismenge nur einen kleinen Prozentsatz der BackingMap-Einträge enthält und wenn nur einige wenige Indizes für die BackingMap erstellt werden.

Topologien mit mehreren Rechenzentren planen

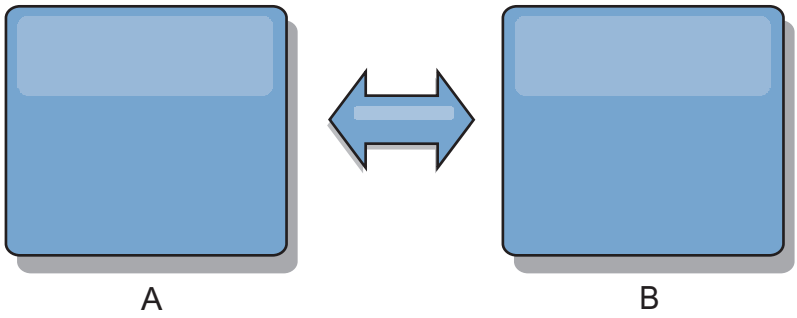
Wenn Sie eine asynchrone Multimasterreplikation verwenden, können zwei oder mehr Datengrids exakte Kopien voneinander werden. Jedes Datengrid ist in einer unabhängigen Katalogservicedomäne mit einem eigenen Katalogservice, eigenen Container-Servern und einem eindeutigen Namen enthalten. Bei asynchroner Multimasterreplikation können Sie Verbindungen verwenden, um eine Sammlung von Katalogservicedomänen zu verbinden. Die Katalogservicedomänen werden anschließend durch Replikation über die Verbindungen synchronisiert. Sie können fast jede Topologie durch die Definition von Verbindungen zwischen den Katalogservicedomänen erstellen.

Topologien für Multimasterreplikation

Sie haben verschiedene Optionen bei der Auswahl der Topologie für Ihre Umgebung mit Multimasterreplikation.

Verbindungen zu Katalogservicedomänen verbinden

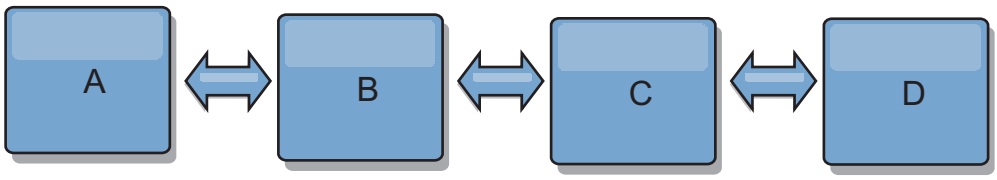
Eine Replikationsdatengridinfrastruktur ist ein verbundener Graph von Katalogservicedomänen mit bidirektionalen Verbindungen zwischen den Domänen. Über ein Verbindung können zwei Katalogservicedomänen Datenänderungen austauschen. Die einfachste Topologie ist beispielsweise ein Katalogservicedomänenpaar mit einer einzigen Verbindung zwischen ihnen. Die Katalogservicedomänen werden alphabetisch von links nach rechts benannt: A, B, C usw. Eine Verbindung kann ein Weitverkehrsnetz (WAN) durchqueren und große Distanzen überwinden. Selbst wenn die Verbindung unterbrochen wird, können Daten in den Katalogservicedomänen trotzdem geändert werden. Die Topologie gleicht die Änderungen ab, sobald die Verbindung die Katalogservicedomänen wieder verbindet. Verbindungen versuchen nach der Unterbrechung der Netzverbindung automatisch, die Verbindung wiederherzustellen.



Nach dem Definieren der Verbindungen versucht eXtreme Scale zuerst, alle Katalogservicedomänen zu synchronisieren. Anschließend versucht eXtreme Scale, die identischen Zustände aufrecht zu erhalten, wenn Änderungen in einer Katalogservicedomäne vorgenommen werden. Das Ziel ist, dass jede Katalogservicedomäne ein exakter Spiegel jeder anderen Katalogservicedomäne ist, mit der sie verbunden ist. Die Replikationsverbindungen zwischen den Katalogservicedomäne stellen sicher, dass alle Änderungen, die in einer Domäne vorgenommen werden, in die anderen Domänen kopiert werden.

Reihentopologien

Obwohl es sich um eine sehr einfache Implementierung handelt, veranschaulicht die Reihentopologie einige Qualitäten der Verbindungen. Zunächst ist es nicht erforderlich, dass eine Katalogservicedomäne direkte mit jeder anderen Katalogservicedomäne verbunden ist, damit sie Änderungen empfängt. Domäne B übernimmt Änderungen von Domäne A. Domäne C empfängt Änderungen von Domäne A über Domäne B, die die Domänen A und B verbindet. Domäne D empfängt Änderungen von den anderen Domänen über Domäne C. Auf diese Weise kann die Quelle der Änderungen von der Verteilung der Änderungen entlastet werden.



Wenn Domäne C ausfällt, werden die folgenden Aktionen ausgeführt:

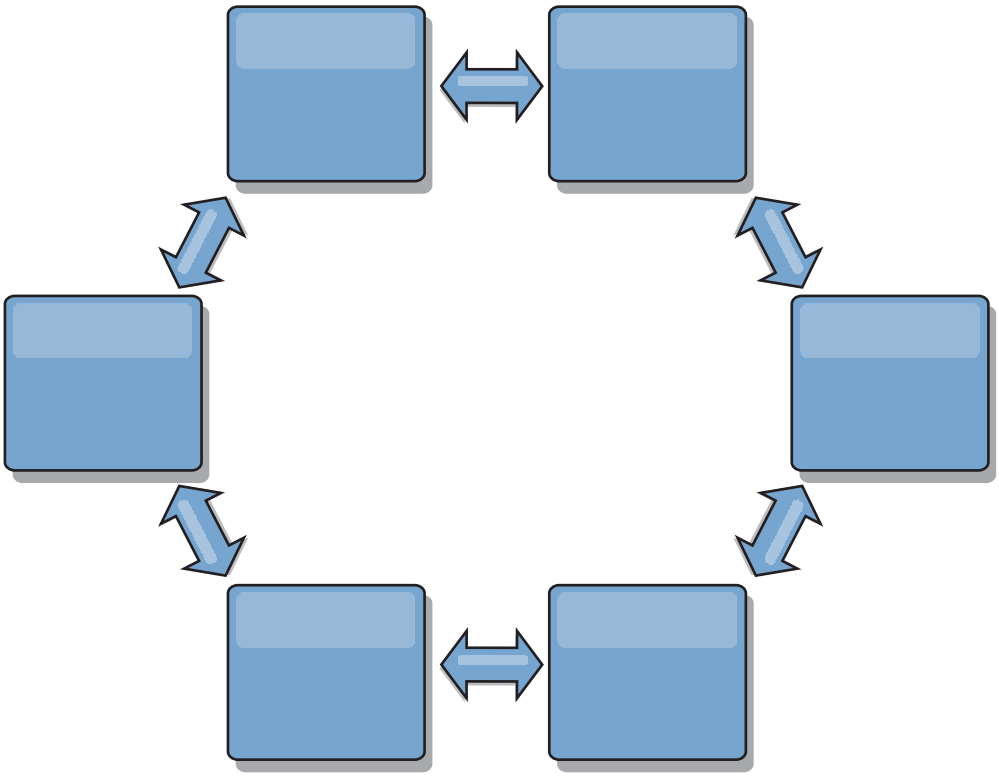
1. Domäne D ist verwaist, bis Domäne C erneut gestartet wird.
2. Domäne C synchronisiert sich selbst mit Domäne B, die eine Kopie von Domäne A ist.
3. Domäne D verwendet Domäne C, um sich mit den Änderungen in den Domänen A und B zu synchronisieren, die vorgenommen wurden, während Domäne D verwaist war (aufgrund des Ausfalls von Domäne C).

Am Ende sind die Domänen A, B, C und D wieder identisch.

Ringtopologien

Ringtopologien sind ein Beispiel für eine Topologie mit erhöhter Ausfallsicherheit. Wenn eine Katalogservicedomäne oder eine einzelne Verbindung ausfällt, können die verbleibenden Katalogservicedomänen trotzdem Änderungen abrufen. Die Katalogservicedomänen bewegen sich ringförmig vom Ausfall weg. Jede Katalogservicedomäne hat maximal zwei Verbindungen, unabhängig davon, wie groß eine

Ringtopologie ist. Die Latenzzeit für die Weitergabe der Änderungen kann hoch sein. Die Änderungen einer bestimmten Katalogservicedomäne müssen möglicherweise über mehrere Verbindungen übertragen werden, bevor sie in allen Katalogservicedomänen vorhanden sind. Eine Reihentopologie weist dasselbe Merkmal auf.

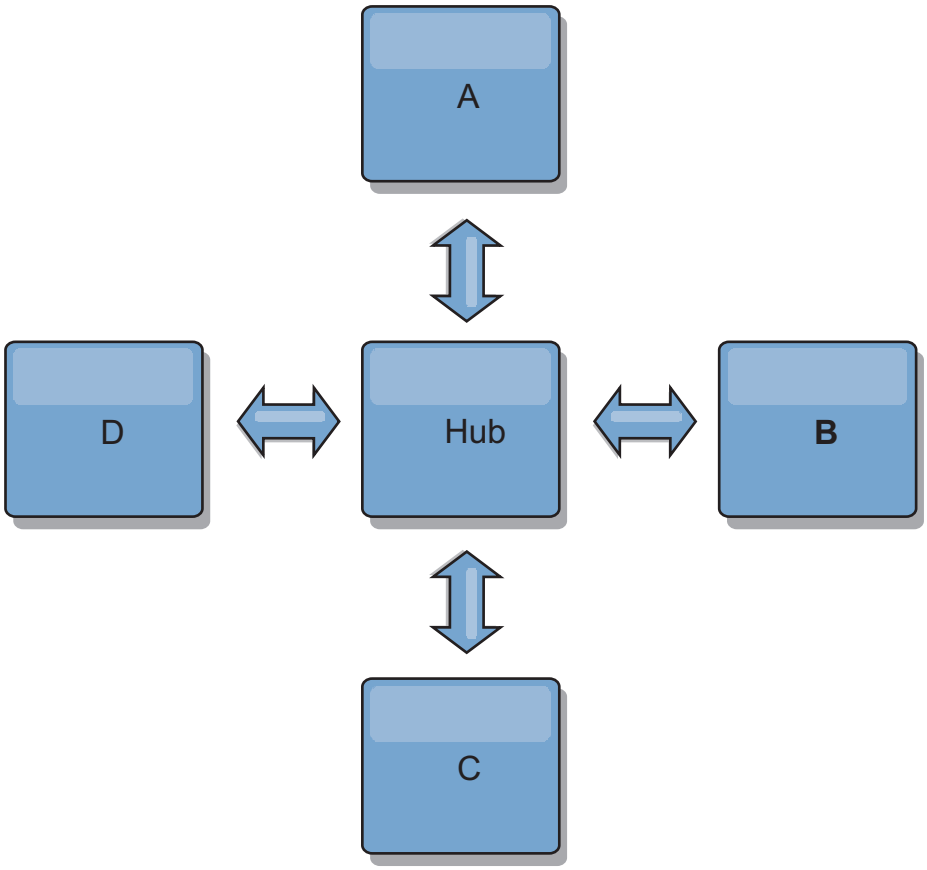


Sie können auch eine fortgeschrittenere Ringtopologie mit einer Stammkatalogservicedomäne in der Mitte des Rings implementieren. Die Stammkatalogservicedomäne dient als zentraler Abgleichspunkt. Die anderen Katalogservicedomänen dienen als ferne Abgleichspunkte für Änderungen, die in der Stammkatalogservicedomäne vorgenommen werden. Die Stammkatalogservicedomäne kann Änderungen unter den Katalogservicedomänen arbitrieren. Wenn eine Ringtopologie mehrere Ringe um eine Stammkatalogservicedomäne herum enthält, kann die Domäne Änderungen nur im inneren Ring arbitrieren. Die Ergebnisse der Arbitrierung werden jedoch über die Katalogservicedomänen in den anderen Ringen verteilt.

Hub- und Peripherietopologien

Mit einer Hub- und Peripherietopologie werden Änderungen über eine Hubkatalogservicedomäne übertragen. Weil der Hub die einzige angegebene zwischengeschaltete Katalogservicedomäne ist, haben Hub- und Peripherietopologien geringere Latenzzeiten. Die Hubdomäne wird über eine Verbindung mit jeder Peripheriedomäne verbunden. Der Hub verteilt Änderungen an die Katalogservicedomänen. Der Hub dient als Abgleichspunkt für Kollisionen. In einer Umgebung mit einer hohen Aktualisierungsrate, muss der Hub im Hinblick auf die Synchronizität möglicherweise auf mehr Hardware als die Peripherie ausgeführt werden. WebSphere eXtreme Scale ist für eine lineare Skalierung konzipiert, d. h., Sie können den Hub bei Bedarf ohne Schwierigkeit vergrößern. Wenn der Hub jedoch ausfällt, werden die Änderungen erst nach einem Neustart des Hubs wieder verteilt.

Alle Änderungen in den Peripheriekatalogservicedomänen werden verteilt, nachdem die Verbindung zum Hub wiederhergestellt wurde.



Sie können auch eine Strategie mit vollständig replizierten Clients verwenden, eine Topologievariante, in der ein Serverpaar von eXtreme Scale als Hub verwendet wird. Jeder Client erstellt ein eigenständiges Einzelcontainer-Datengrid mit einem Katalog in der Client-JVM. Ein Client verwendet sein Datengrid, um die Verbindung zum Hubkatalog herzustellen. Die Verbindung bewirkt, dass sich der Client mit dem Hub synchronisiert, sobald die Verbindung zum Hub hergestellt ist.

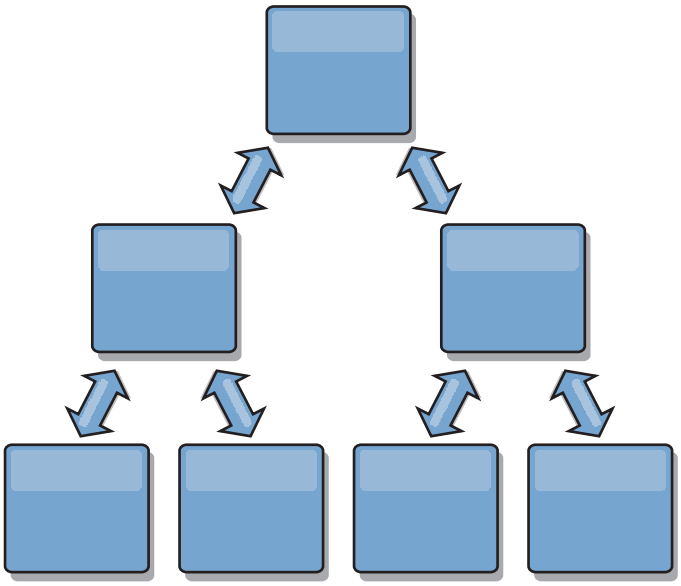
Alle vom Client vorgenommenen Änderungen sind lokal und werden asynchron im Hub repliziert. Der Hub dient als Arbitrierungsdomäne und verteilt Änderungen an alle verbundenen Clients. Die Topologie mit vollständig replizierten Clients ist ein zuverlässiger L2-Cache für einen objektrelationalen Mapper wie OpenJPA. Änderungen werden über den Hub schnell an die Client-JVMs verteilt. Solange die Cachegröße im verfügbaren Heapspeicher untergebracht werden kann, ist die Topologie eine geeignete Architektur für diesen L2-Stil.

Verwenden Sie bei Bedarf mehrere Partitionen für die Skalierung der Hubdomäne in mehreren JVMs. Weil alle Daten immer noch in eine einzige Client-JVM passen müssen, kann die Kapazität des Hubs für die Verteilung und Arbitrierung von Änderungen durch mehrere Partitionen erhöht werden. Die Verwendung mehrerer Partitionen ändert die Kapazität einer einzelnen Domäne jedoch nicht.

Baumtopologien

Sie können auch eine azyklische gerichtete Baumstruktur verwenden. Eine azyklische Baumstruktur hat keine Zyklen oder Schleifen, und ein gerichtetes Setup be-

schränkt Verbindungen auf vorhandene Verbindungen zwischen übergeordneten und untergeordneten Komponenten. Diese Konfiguration kann für Topologien mit vielen Katalogservicedomänen hilfreich sein, und es ist nicht empfehlenswert, einen zentralen Hub einzusetzen, der mit jeder möglichen Peripheriekomponente verbunden ist. Dieser Typ von Topologie kann auch hilfreich sein, wenn Sie untergeordnete Katalogservicedomänen hinzufügen müssen, ohne die Stammkatalogservicedomäne zu aktualisieren.



Eine Baumstrukturtopologie kann einen zentralen Abgleichspunkt in der Stammkatalogservicedomäne haben. Die zweite Ebene kann weiterhin als ferner Abgleichspunkt für Änderungen dienen, die in der darunter liegenden Katalogservicedomäne vorgenommen werden. Die Stammkatalogservicedomäne kann Änderungen zwischen den Katalogservicedomänen nur auf der zweiten Ebene arbitrieren. Sie können auch k-näre Baumstrukturen verwenden, die jeweils N untergeordnete Komponenten auf jeder Ebene haben können. Jede Katalogservicedomäne stellt *n* ausgehende Verbindungen her.

Vollständig replizierte Clients

Diese Topologievariante umfasst ein eXtreme-Scale-Serverpaar, das als Hub ausgeführt wird. Jeder Client erstellt ein eigenständiges Einzelcontainer-Datengrid mit einem Katalog in der Client-JVM. Ein Client verwendet sein Datengrid, um die Verbindung zum Hub-Katalog herzustellen, was bewirkt, dass sich der Client mit dem Hub synchronisiert, sobald die Verbindung zum Hub hergestellt ist.

Alle vom Client vorgenommenen Änderungen sind lokal und werden asynchron im Hub repliziert. Der Hub dient als Arbitrierungsdomäne und verteilt Änderungen an alle verbundenen Clients. Die Topologie mit vollständig replizierten Clients ist ein guter L2-Cache für einen objektrelationalen Mapper wie OpenJPA. Änderungen werden über den Hub schnell an die Client-JVMs verteilt. Solange die Cachegröße vom verfügbaren Heapspeicher der Clients untergebracht werden kann, ist diese Topologie eine geeignete Architektur für diesen L2-Stil.

Verwenden Sie bei Bedarf mehrere Partitionen für die Skalierung der Hub-Domäne in mehreren JVMs. Da alle Daten weiterhin in eine einzige Client-JVM passen müssen, erhöht die Verwendung mehrerer Partitionen die Kapazität des Hubs für die

Verteilung und Arbitrierung von Änderungen, aber nicht die Kapazität einer einzelnen Domäne.

Konfigurationshinweise für Multimastertopologien

Beachten Sie die folgenden Probleme, wenn Sie festlegen, ob und wie Multimasterreplikationstopologien verwendet werden.

- **Voraussetzungen für MapSets**

  MapSets müssen die folgenden Merkmale aufweisen, damit Änderungen über Verbindungen zwischen Katalogservicedomänen repliziert werden können:

  – Der ObjectGrid-Name und der MapSet-Name in einer Katalogservicedomäne müssen mit dem ObjectGrid-Namen und dem MapSet-Namen anderer Katalogservicedomänen übereinstimmen. ObjectGrid "og1" und MapSet "ms1" müssen beispielsweise in Katalogservicedomäne A und Katalogservicedomäne B konfiguriert werden, um die Daten im MapSet zwischen den Katalogservicedomänen zu replizieren.

  – Das Datengrid hat den Typ `FIXED_PARTITION`. Datengrids des Typs `PER_CONTAINER` können nicht repliziert werden.

  – Das MapSet enthält in allen Katalogservicedomänen dieselbe Anzahl von Partitionen. Das MapSet kann dieselbe Anzahl und dieselben Typen von Replikaten haben oder auch nicht.

  – Dieselben Datentypen werden in jeder Katalogservicedomäne für das MapSet repliziert.

  – Das MapSet enthält dieselben Maps und dieselben Schablonen für dynamische Maps in jeder Katalogservicedomäne.

  – Die MapSet verwendet keinen Entitätsmanager. Ein MapSet, das eine Entitäts-Map enthält, wird in Katalogservicedomänen nicht repliziert.

  – Das MapSet verwendet keine Write-behind-Caching-Unterstützung. Ein MapSet, das eine Map enthält, die mit Write-behind-Unterstützung konfiguriert ist, wird in Katalogservicedomänen nicht repliziert.

  Alle MapSets mit den vorherigen Merkmalen werden repliziert, nachdem die Katalogservicedomänen in der Topologie gestartet wurden.

- **Klassenlader mit mehreren Katalogservicedomänen**

  Katalogservicedomänen müssen Zugriff auf alle Klassen haben, die als Schlüssel und Werte verwendet werden. Alle Abhängigkeiten müssen sich in allen Klassenpfaden für Grid-Container-JVMs für alle Domänen widerspiegeln. Wenn ein CollisionArbiter-Plug-in den Wert für einen Cacheeintrag abruft, müssen die Klassen für die Werte für die Domäne vorhanden sein, die den Arbiter startet.

Hinweise zu Ladeprogrammen in einer Multimastertopologie

Wenn Sie Ladeprogramme in einer Multimastertopologie verwenden, müssen Sie die möglichen Anforderungen in Bezug auf die Verwaltung von Kollisions- und Revisionsinformationen berücksichtigen. Das Datengrid verwaltet Revisionsinformationen zu den Elementen im Datengrid, so dass Kollisionen erkannt werden können, wenn andere primäre Shards in der Konfiguration Einträge in das Datengrid schreiben. Wenn Einträge von einem Ladeprogramm hinzugefügt werden, werden diese Revisionsinformationen nicht eingeschlossen, und der Eintrag verwendet eine neue Überarbeitung. Da die Überarbeitung des Eintrags eine neue Einfügung zu sein scheint, könnte eine Fehlkollision auftreten, wenn ein anderes primäres Shard diesen Zustand ebenfalls ändert oder dieselben Daten aus einem Ladeprogramm extrahiert.

Replikationsänderungen rufen die Methode get im Ladeprogramm mit einer Liste der Schlüssel auf, die noch nicht im Datengrid enthalten sind, aber während der

Replikationstransaktion geändert werden. Wenn die Replikation stattfindet, sind diese Einträge Kollisionseinträge. Sind die Kollisionen arbitriert, und die Überarbeitung wird angewendet. Anschließend wird eine Stapelaktualisierung im Ladeprogramm aufgerufen, um die Änderungen auf die Datenbank anzuwenden. Alle Maps, die im Revisionsfenster geändert wurden, werden in derselben Transaktion aktualisiert.

Preload-Rätsel

Stellen Sie sich eine Topologie mit zwei Rechenzentren vor: Rechenzentrum A und Rechenzentrum B. Beide Rechenzentren haben unabhängige Datenbank, aber nur Rechenzentrum A hat ein Datengrid, das aktiv ist. Wenn Sie eine Verbindung zwischen den Rechenzentren für eine Multimasterkonfiguration herstellen, beginnen die Datengrids in Rechenzentrum A, Daten mit Push an die neuen Datengrids im Rechenzentrum B zu übertragen, was bei jedem Eintrag zu einer Kollision führt. Ein weiteres großes Problem tritt bei allen Daten auf, die in der Datenbank in Rechenzentrum B enthalten sind, aber nicht in der Datenbank in Rechenzentrum A. Diese Zeilen werden nicht gefüllt und arbitriert, was zu Inkonsistenzen führt, die nicht aufgelöst werden.

Lösung des Preload-Rätsels

Da die Daten, die nur in der Datenbank enthalten sind, keine Überarbeitungen haben können, müssen Sie das Datengrid immer vollständig aus der lokalen Datenbank laden, bevor Sie die Multimasterverbindung herstellen. Anschließend können beide Datengrids die Daten überarbeiten und arbitrieren und schließlich einen konsistenten Status erreichen.

Teilcache-Rätsel

Mit einem Teilcache versucht die Anwendung zuerst, die Daten im Datengrid zu finden. Wenn die Daten nicht im Datengrid enthalten sind, werden die Daten mithilfe des Loaders in der Datenbank gesucht. Die Einträge im Datengrid werden in regelmäßigen Abständen bereinigt, um die Cachegröße klein zu halten.

Dieser Cachetyp kann in einem Szenario mit einer Multimasterkonfiguration problematisch sein, weil die Einträge im Datengrid Metadaten zur Überarbeitung enthalten, mit deren Hilfe Kollisionen und die Seite, auf der die Änderungen vorgenommen wurden, erkannt werden können. Wenn Verbindungen zwischen den Rechenzentren nicht funktionieren, kann ein Rechenzentrum einen Eintrag aktualisieren und schließlich die Datenbank aktualisieren und den Eintrag im Datengrid ungültig machen. Nach der Wiederherstellung der Verbindung versuchen die Rechenzentren, Überarbeitungen miteinander zu synchronisieren. Da die Datenbank jedoch aktualisiert und der Eintrag im Datengrid ungültig gemacht wurde, geht die Änderung aus der Perspektive des Rechenzentrums, das ausgefallen ist, verloren. Deshalb sind die beiden Seiten des Datengrids nicht mehr synchronisiert und nicht mehr konsistent.

Lösung des Teilcache-Rätsels

Hub- und Peripherietopologie:

Sie können den Loader nur im Hub einer Hub- und Peripherietopologie ausführen, in der die Konsistenz der Daten erhalten bleibt, während das Datengrid horizontal skaliert wird. Wenn Sie diese Implementierung in Erwägung ziehen, müssen Sie jedoch bedenken, dass die Loader ein partielles Laden des Datengrids zulassen kön-

nen, d. h., dass ein Bereinigungsprogramm (Evictor) konfiguriert wurde. Wenn die Peripherie Ihrer Konfiguration aus Teilcaches besteht, die aber keinen Loader haben, besteht bei Cachefehlern keine Möglichkeit, die Daten aus der Datenbank abzurufen. Wegen dieser Einschränkung müssen Sie eine vollständig gefüllte Cachetopologie mit einer Hub- und Peripheriekonfiguration verwenden.

Invalidierungen und Bereinigung

Bei der Invalidierung entstehen Inkonsistenzen zwischen dem Datengrid und der Datenbank. Daten können über das Programm oder durch Bereinigung aus dem Datengrid entfernt werden. Wenn Sie Ihre Anwendung entwickeln, müssen Sie berücksichtigen, dass bei der Behandlung von Überarbeitungen keine Änderungen repliziert werden, die ungültig gemacht wurden, was zu Inkonsistenzen zwischen primären Shards führt.

Invalidierungsereignisse sind keine Cachestatusänderungen und führen nicht zur Replikation. Alle konfigurierten Bereinigungsprogramme werden unabhängig von anderen Bereinigungsprogrammen in der Konfiguration ausgeführt. Es kann beispielsweise ein Bereinigungsprogramm für einen Speicherschwellenwert in der einen Katalogservicedomäne konfiguriert sein, aber ein anderes, weniger aggressives Bereinigungsprogramm in der anderen verbundenen Katalogservicedomäne. Wenn Datengrideinträge aufgrund der Schwellenwertrichtlinie entfernt werden, sind die Einträge in der anderen Katalogservicedomäne nicht betroffen.

Datenbankaktualisierungen und Datengridinvalidierung

Es treten Probleme auf, wenn Sie in einer Multimasterkonfiguration die Datenbank direkt im Hintergrund aktualisieren, während die Invalidierung im Datengrid für die aktualisierten Einträge aufgerufen wird. Dieses Problem tritt auf, weil das Datengrid die Änderung erst dann auf den anderen primären Shards replizieren kann, wenn der Eintrag durch einen Cache in das Datengrid verschoben wird.

Mehrere Ausgabeprogramme für eine einzige logische Datenbank

Wenn Sie eine einzige Datenbank mit mehreren primären Shards verwenden, die über einen Loader verbunden sind, treten Transaktionskonflikte auf. Ihre Loaderimplementierung muss diese Typen von Szenarien in besonderer Weise behandeln.

Daten durch Multimasterreplikation spiegeln

Sie können unabhängige Datenbanken konfigurieren, die mit unabhängigen Katalogservicedomänen verbunden sind. In dieser Konfiguration kann der Loader Änderungen mit Push aus einem Rechenzentrum in das andere Rechenzentrum übertragen.

Designhinweise für die Multimasterreplikation

Wenn Sie die Multimasterreplikation implementieren, müssen Sie Aspekte wie Arbitrierung, Verbindungen und Leistung beim Design berücksichtigen.

Arbitrierungshinweise für das Topologiedesign

Änderungskollisionen können auftreten, wenn dieselben Datensätze gleichzeitig an zwei Stellen geändert werden können. Konfigurieren Sie alle Katalogservicedomänen mit denselben Werten für Prozessor-, Hauptspeicher und Netzressourcen. Sie können beobachten, dass Katalogservicedomänen, die für die Kollisionsbehandlung

(Arbitrierung) zuständig sind, mehr Ressourcen als andere Katalogservicedomänen verbrauchen. Kollisionen werden automatisch erkannt. Sie werden mit einem der folgenden beiden Mechanismen behoben:

- **Standardkollisionsarbiter**: Standardmäßig werden die Änderungen aus der Katalogservicedomäne verwendet, deren Name in der lexikalischer Reihenfolge am niedrigsten steht. Wenn beispielsweise Katalogservicedomäne A und Domäne B einen Konflikt in Bezug auf einen Datensatz verursachen, wird die Änderung aus Katalogservicedomäne B ignoriert. Katalogservicedomäne A behält ihre Version, und der Datensatz in Katalogservicedomäne B wird geändert, so dass er dem Datensatz aus Katalogservicedomäne A entspricht. Dieses Verhalten gilt auch für Anwendungen, in denen Benutzer oder Sitzungen normalerweise gebunden sind oder eine Affinität zu einem der Datengrids haben.
- **Angepasster Kollisionsarbiter**: Anwendungen können einen angepassten Arbiter bereitstellen. Wenn eine Katalogservicedomäne eine Kollision erkennt, ruft sie den Arbiter auf. Informationen zum Entwickeln eines hilfreichen angepassten Arbiters finden Sie unter Angepasste Arbiter für Replikation mehrerer Master entwickeln.

Für Topologien, in denen Kollisionen möglich sind, können Sie eine Hub- und Peripherietopologie oder eine Baumtopologie implementieren. Diese beiden Topologien sind dienlich, um ständige Kollisionen zu vermeiden, die in den folgenden Szenarien auftreten können:

1. In mehreren Katalogservicedomänen tritt eine Kollision auftritt.
2. Jede Katalogservicedomäne behebt die Kollision lokal, was zu Überarbeitungen führt.
3. Die Überarbeitungen kollidieren, was zu Überarbeitungen von Überarbeitungen führt.

Zur Vermeidung von Kollisionen wählen Sie eine bestimmte Katalogservicedomäne, die so genannte *Arbitrierungskatalogservicedomäne*, als Kollisionsarbiter für einen Teil der Katalogservicedomänen aus. In einer Hub- und Peripherietopologie kann der Hub beispielsweise als Kollisionshandler verwendet werden. Der Peripheriekollisionshandler ignoriert alle von den Peripheriekatalogservicedomänen erkannten Kollisionen. Die Hubkatalogservicedomäne erstellt Überarbeitungen, was unerwartete Kollisionsüberarbeitungen verhindert. Die für die Behandlung von Kollisionen zugeordnete Katalogservicedomäne muss eine Verbindung zu allen Domänen haben, für die sie Kollisionen beheben soll. In einer Baumtopologie beheben alle internen übergeordneten Domänen Kollisionen für die ihnen unmittelbar untergeordneten Domänen. Wenn Sie eine Ringtopologie verwendet, ist es nicht möglich, eine einzige Katalogservicedomäne im Ring als Arbiter zu bestimmen.

In der folgenden Tabelle sind die kompatiblen Arbitrierungsansätze für die verschiedenen Topologien zusammengefasst.

Tabelle 1. Arbitrierungsansätze. Der folgenden Tabelle können Sie entnehmen, ob Anwendungsarbitrierung mit den verschiedenen Technologien kompatibel ist.

Topologie	Anwendungs-arbitrierung?	Anmerkungen
Eine Reihe von zwei Katalogservicedomänen	Ja	Wählen Sie eine Katalogservicedomäne als Arbiter aus.
Eine Reihe von drei Katalogservicedomäne	Ja	Die mittlere Katalogservicedomäne muss der Arbiter sein. Stellen Sie sich die mittlere Katalogservicedomäne als Hub in einer einfachen Hub- und Peripherietopologie vor.

Tabelle 1. Arbitrierungsansätze (Forts.). Der folgenden Tabelle können Sie entnehmen, ob Anwendungsarbitrierung mit den verschiedenen Technologien kompatibel ist.

Topologie	Anwendungs-arbitrierung?	Anmerkungen
Eine Reihe von mehr als drei Katalogservicedomänen	Nein	Anwendungsarbitrierung wird nicht unterstützt.
Ein Hub mit N Peripheriedomänen	Ja	Der Hub mit den Verbindungen zu allen Peripheriedomänen muss die Arbitrierungskatalogservicedomäne sein.
Ein Ring mit N Katalogservicedomänen	Nein	Anwendungsarbitrierung wird nicht unterstützt.
Eine azyklische gerichtete Baumstruktur (k-näre Baumstruktur)	Ja	Alle Stammknoten dürfen nur die ihnen direkt untergeordneten Knoten bewerten.

Verbindungshinweise für das Topologiedesign

Im Idealfall enthält eine Topologie die Mindestanzahl an Verbindungen und optimiert gleichzeitig Kompromisse zwischen Latenzzeit für Änderungen, Fehlertoleranz und Leistungsmerkmale.

- **Latenzzeit bei Änderungen**

  Die Latenzzeit bei Änderungen wird durch die Anzahl zwischengeschalteter Katalogservicedomänen bestimmt, die eine Änderung durchlaufen muss, bevor sie in einer bestimmten Katalogservicedomäne ankommt.

  Eine Topologie weist die beste Latenzzeit bei Änderungen auf, wenn zwischengeschaltete Katalogservicedomänen wegfallen, weil jede Katalogservicedomäne mit jeder anderen Katalogservicedomäne verbunden wird. Eine Katalogservicedomäne muss jedoch proportional zur Anzahl ihrer Verbindungen Replikationsarbeiten ausführen. In großen Topologien kann die reine Anzahl zu definierender Verbindungen einen hohen Verwaltungsaufwand darstellen.

  Die Geschwindigkeit, mit der eine Änderung in andere Katalogservicedomänen kopiert wird, richtet sich nach weiteren Faktoren, wie z. B.:

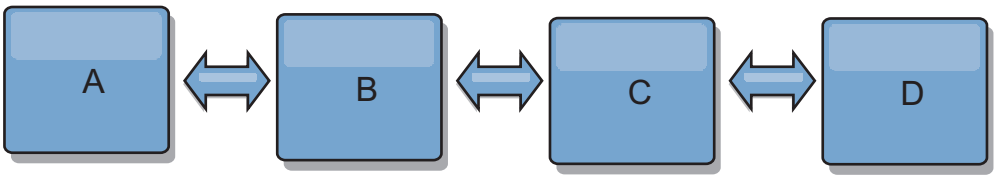
  – Prozessor- und Netzbandbreite in der Quellenkatalogservicedomäne
  – Anzahl zwischengeschalteter Katalogservicedomänen und Verbindungen zwischen der Quellen- und der Zielkatalogservicedomäne
  – Prozessor- und Netzressourcen, die der Quellenkatalogservicedomäne, der Zielkatalogservicedomäne und den zwischengeschalteten Katalogservicedomänen zur Verfügung stehen

- **Fehlertoleranz**

  Die Fehlertoleranz wird durch die Anzahl der existierenden Pfade zwischen zwei Katalogservicedomänen für die Änderungsreplikation bestimmt.

  Wenn Sie nur eine einzige Verbindung zwischen zwei Katalogservicedomänen haben, wird die Weitergabe von Änderungen aufgrund eines Verbindungsfehlers nicht zugelassen. Änderungen zwischen Katalogservicedomänen werden auch nicht weitergegeben, wenn bei einer der zwischengeschalteten Domänen ein Verbindungsfehler auftritt. Ihre Topologie kann eine einzige Verbindung von einer Katalogservicedomäne zu einer anderen Katalogservicedomäne enthalten, der über zwischengeschaltete Domänen führt. In diesem Fall werden Änderungen nicht weitergegeben, wenn eine der zwischengeschalteten Katalogservicedomänen ausfällt.

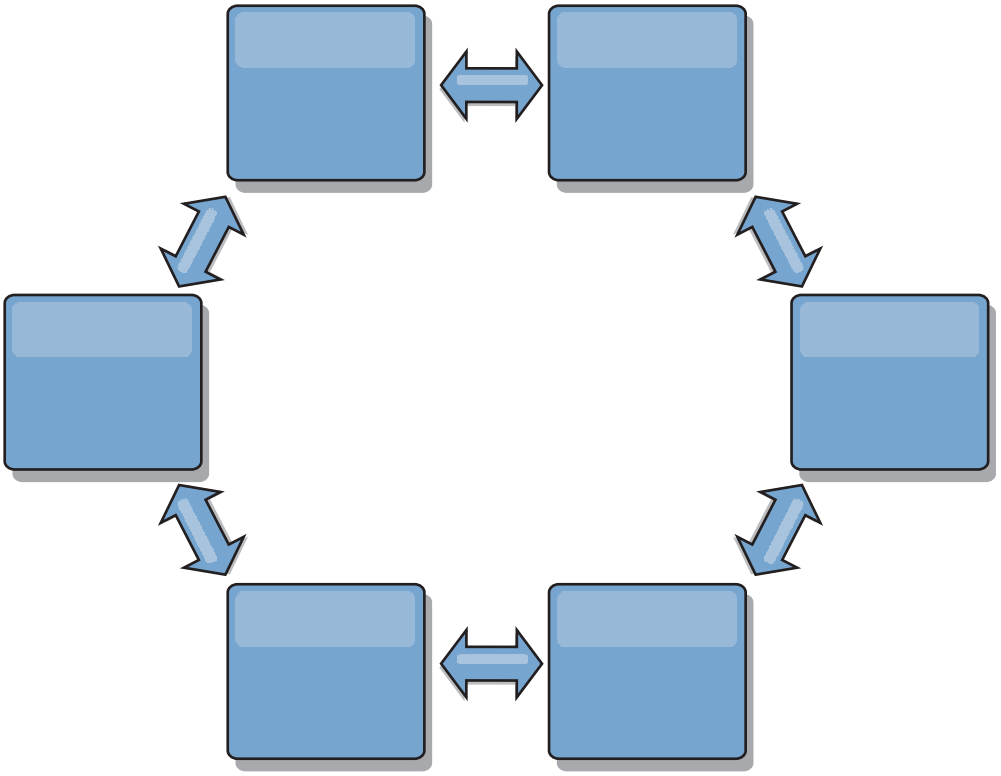
Stellen Sie sich eine Reihentopologie mit vier Katalogservicedomänen, A, B, C und D, vor:



Wenn eine der folgenden Bedingungen zutrifft, sieht die Domäne D Änderungen von Domäne A nicht:

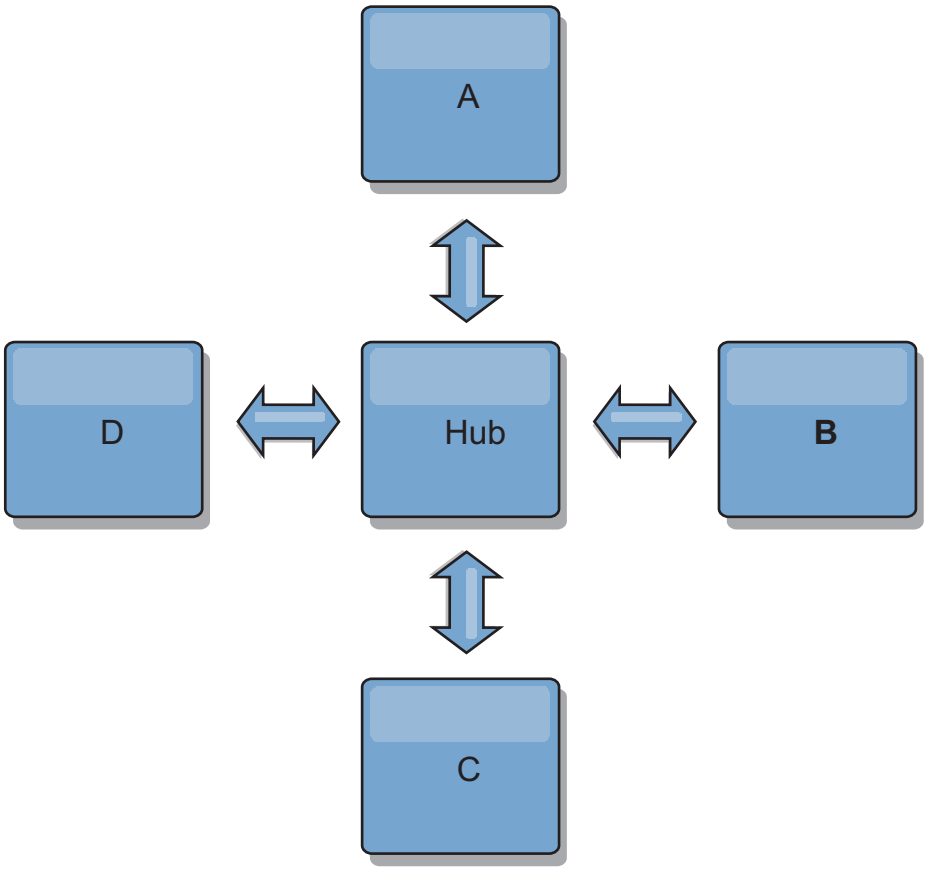
- Die Domäne A ist aktiv, und die Domäne B ist inaktiv.
- Die Domänen A und B sind aktiv, und die Domäne C ist inaktiv.
- Die Verbindung zwischen A und B ist inaktiv.
- Die Verbindung zwischen B und C ist inaktiv.
- Die Verbindung zwischen C und D ist inaktiv.

In einer Ringtopologie hingegen kann jede Katalogservicedomäne Änderungen aus jeder Richtung empfangen.



Wenn beispielsweise ein bestimmter Katalogservice in Ihrer Ringtopologie inaktiv ist, können die beiden benachbarten Domänen weiterhin Änderungen direkt voneinander extrahieren.

Alle Änderungen werden über den Hub weitergegeben. Damit ist das Hub- und Peripheriedesign im Gegensatz zu den Reihen- und Ringtopologien für Störungen anfällig, wenn der Hub ausfällt.



Ein einzige Katalogservicedomäne ist gegen eine bestimmte Anzahl von Serviceausfällen widerstandsfähig. Weiterreichende Ausfälle wie beispielsweise Netzausfälle oder Ausfälle von Verbindungen zwischen physischen Rechenzentren können jedoch den Betrieb Ihrer Katalogservicedomänen unterbrechen.

- **Verbindungen und Leistung**

  Die Anzahl der in einer Katalogservicedomäne definierten Verbindungen wirkt sich auf die Leistung aus. Je mehr Verbindungen definiert werden, desto mehr Ressourcen werden benötigt, und deshalb kann die Replikationsleistung abnehmen. Die Möglichkeit, Änderungen für Domäne A über andere Domänen anzurufen, entlastet die Domäne A effektiv von der Replikation ihrer Transaktionen in allen Domänen. Die Änderungsverteilungslast in einer Domäne wird durch die Anzahl der verwendeten Verbindungen und nicht durch die Anzahl der Domänen in der Topologie beschränkt. Diese Lasteigenschaft unterstützt Skalierbarkeit, damit sich die Domänen in der Topologie die Last der Änderungsverteilung teilen können.

  Eine Katalogservicedomäne kann Änderungen indirekt über andere Katalogservicedomänen abrufen. Stellen Sie sich eine Reihentopologie mit fünf Katalogservicedomänen vor.

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

  – A bezieht Änderungen von B, C, D und E über B.
  – B bezieht Änderungen von A und C direkt und Änderungen von D und E über C.
  – C bezieht Änderungen von B und D direkt und Änderungen von A über B und Änderungen von E über D.
  – D bezieht Änderungen von C und E direkt und Änderungen von A und B über C.

– E bezieht Änderungen von D direkt und Änderungen von A, B und C über D.

Die Verteilungslast ist in den Katalogservicedomänen A und E am geringsten, weil sie jeweils nur eine Verbindung zu einer einzigen Katalogservicedomäne haben. Die Domänen B, C und D haben jeweils eine Verbindung zu zwei Domänen. Somit ist die Verteilungslast in den Domänen B, C und D doppelt so hoch wie die Last in den Domänen A und E. Die Arbeitslast richtet sich nach der Anzahl der Verbindungen in jeder Domäne und nicht nach der Gesamtanzahl der Domänen in der Topologie. Die beschriebene Verteilung der Last bliebe damit auch bei 1000 Domänen in der Reihe konstant.

Leistungshinweise für die Multimasterreplikation

Berücksichtigen Sie bei der Verwendung von Multimaster-Replikationstopologien die folgenden Einschränkungen:

- **Optimierung der Änderungsverteilung**, die im vorherigen Abschnitt beschrieben wurde
- **Leistung der Replikationsverbindungen**: WebSphere eXtreme Scale erstellt einen einzigen TCP/IP-Socket zwischen jedem JVM-Paar. Der gesamte Datenverkehr zwischen den JVMs findet über diesen Socket statt. Dies gilt auch für den Datenverkehr aus der Multimasterreplikation. Die Katalogservicedomänen werden in mindestens *n* Container-JVMs gehostet, wodurch mindestens *n* TCP-Verbindungen zu Peerkatalogservicedomänen bereitgestellt werden. Deshalb haben die Katalogservicedomänen mit einer höheren Anzahl an Containern höhere Replikationsleistungsstufen. Je mehr Container vorhanden sind, desto mehr Prozessor- und Netzressourcen sind erforderlich.
- **Optimierung des TCP-Sliding-Window und RFC 1323**: Die Unterstützung von RFC 1323 auf beiden Seiten einer Verbindung führt zu mehr Daten bei einem Umlauf. Diese Unterstützung führt zu einem höheren Durchsatz, was die Größe des Sliding-Windows um den Faktor 16 erhöht.

  TCP-Sockets verwenden, wie bereits erwähnt, einen Sliding-Window-Mechanismus, um den Fluss von Massendaten zu steuern. Dieser Mechanismus beschränkt den Socket gewöhnlich auf 64 KB in einem Umlaufintervall. Wenn das Umlaufintervall 100 ms lang ist, ist die Bandbreite ohne Optimierung auf 640 KB/Sekunde beschränkt. Um die verfügbare Bandbreite einer Verbindung vollständig nutzen zu können, müssen unter Umständen spezielle Optimierungstasks für ein Betriebssystem ausgeführt werden. Die meisten Betriebssysteme haben Optimierungsparameter, einschließlich Optionen für RFC 1323, um den Durchsatz über Verbindungen mit hoher Latenzzeit zu verbessern.

  Es gibt mehrere Faktoren, die sich auf die Replikationsleistung auswirken können:

  – Geschwindigkeit, mit der eXtreme Scale Änderungen abruft
  – Geschwindigkeit, mit der eXtreme Scale Replikationsanforderungen bei Abruf verarbeiten kann
  – Kapazität des Sliding-Windows
  – Optimierung der Netzpuffer auf beiden Seiten einer Verbindung, was eXtreme Scale ermöglicht, Änderungen effizient über den Socket abzurufen
- **Objektserialisierung**: Alle Daten müssen serialisierbar sein. Wenn eine Domäne `COPY_TO_BYTES` nicht verwendet, muss die Katalogservicedomäne Java-Serialisierung oder ObjectTransformer verwenden, um die Serialisierungsleistung zu optimieren.

- **Komprimierung**: WebSphere eXtreme Scale komprimiert standardmäßig alle Daten, die zwischen Katalogservicedomänen gesendet werden. Die Komprimierung kann momentan nicht inaktiviert werden.
- **Hauptspeicheroptimierung**: Die Speicherbelegung für eine Multimasterreplikationstopologie ist weitgehend unabhängig von der Anzahl der Katalogservicedomänen in der Topologie.

  Bei der Multimasterreplikation entsteht ein fester Verarbeitungsaufwand pro Map-Eintrag für die Versionssteuerung. Außerdem überwacht jeder Container ein festes Datenvolumen für jede Katalogservicedomäne in der Topologie. Eine Topologie mit zwei Katalogservicedomänen belegt ungefähr denselben Speicher wie eine Topologie mit 50 Katalogservicedomänen. WebSphere eXtreme Scale verwendet keine Wiedergabeprotokolle oder ähnlichen Warteschlangen in der Implementierung. Deshalb ist keine Wiederherstellungsstruktur verfügbar, wenn eine Replikationsverbindung über einen längeren Zeitraum hinweg und nach späteren Neustarts nicht verfügbar ist.

Interoperabilität mit anderen WebSphere-Produkten

Sie können WebSphere eXtreme Scale mit anderen Serverprodukten wie WebSphere Application Server und WebSphere Application Server Community Edition integrieren.

WebSphere Application Server

Sie können WebSphere Application Server in verschiedene Aspekte Ihrer Konfiguration von WebSphere eXtreme Scale integrieren. Sie können Datengridanwendungen implementieren und WebSphere Application Server als Host für Container- und Katalogserver verwenden. Außerdem können Sie die Sicherheit von WebSphere Application Server in Ihrer Umgebung von WebSphere eXtreme Scale verwenden.

WebSphere Portal

Sie können HTTP-Sitzungen über WebSphere Portal persistent in einem Datengrid in WebSphere eXtreme Scale speichern.

WebSphere Application Server Community Edition

WebSphere Application Server Community Edition kann den Sitzungsstatus zwar zur gemeinsamen Nutzung bereitstellen, aber nicht auf effiziente und skalierbare Weise. WebSphere eXtreme Scale stellt eine verteilte Persistenzschicht mit hoher Leistung bereit, die zum Replizieren des Status verwendet werden kann, sich aber nicht problemlos mit Anwendungsservern außerhalb von WebSphere Application Server integrieren lässt. Sie können diese beiden Produkte integrieren, um eine skalierbare Lösung für das Sitzungsmanagement zu erhalten.

WebSphere Real Time

Mit der Unterstützung für WebSphere Real Time, dem branchenführenden Java-Angebot für die Echtzeitverarbeitung, ermöglicht WebSphere eXtreme Scale XTP-Anwendungen (Extreme Transaction Processing) konsistentere und voraussagbare Antwortzeiten.

Installation planen

Vor der Installation des Produkts müssen Sie Ihre Umgebung berücksichtigen.

Hardware- und Softwarevoraussetzungen

Dieser Abschnitt enthält eine Übersicht über die Hardware- und Betriebssystemvoraussetzungen. Sie müssen zwar keine bestimmte Version der Hardware oder des Betriebssystems für WebSphere eXtreme Scale verwenden, aber auf der Seite "Systems Requirements" der Produktunterstützungssite sind die formal unterstützten Hardware- und Softwareoptionen beschrieben. Sollte die Informationen im Information Center und auf der Seite "System Requirements" widersprüchlich sein, haben die Informationen auf der Website Vorrang. Die Informationen zu den Voraussetzungen im Information Center werden nur im Hinblick auf die Bedienerfreundlichkeit bereitgestellt.

Die offizielle Beschreibung der Hardware- und Softwarevoraussetzungen finden Sie auf der Webseite System Requirements.

Sie müssen eXtreme Scale nicht unter einer bestimmten Version des Betriebssystems installieren und implementieren. Jede Java-SE- (Java Platform, Standard Edition) und Java-EE-Installation (Java Platform, Enterprise Edition) erfordert andere Betriebssystemversionen oder -fixes.

Sie können das Produkt in Java-EE- und in Java-SE-Umgebungen installieren und implementieren. Außerdem können Sie die Clientkomponente direkt, ohne Integration mit WebSphere Application Server, mit JEE-Anwendungen bündeln. WebSphere eXtreme Scale unterstützt Java SE 5 und höher und WebSphere Application Server Version 6.1 und höher.

Hardwarevoraussetzungen

WebSphere eXtreme Scale setzt keine bestimmte Hardwareversion voraus. Die Hardwarevoraussetzungen richten sich nach der unterstützten Hardware für die Installation der Java Platform, Standard Edition, die Sie für die Ausführung von WebSphere eXtreme Scale verwenden. Wenn Sie eXtreme Scale mit WebSphere Application Server oder einer anderen Java-EE-Implementierung (Java Platform, Enterprise Edition) verwenden, sind die Hardwarevoraussetzungen dieser Plattformen für WebSphere eXtreme Scale ausreichend.

Betriebssystemvoraussetzungen

- **Ohne die Webkonsole**

  eXtreme Scale setzt keine bestimmte Betriebssystemversion voraus. Jede Java-SE- und jede Java-EE-Implementierung setzt verschiedene Betriebssystemversionen oder -Fixes für Probleme voraus, die während des Testens der Java-Implementierung erkannt werden. Die von diesen Implementierungen vorausgesetzten Versionen sind für eXtreme Scale ausreichend.
- **Mit der Webkonsole**

  Die folgenden Voraussetzungen gelten für jedes Betriebssystem, wenn die Konsole verwendet wird:
  - Linux: 32-Bit- oder 64-Bit-JVM
  - Linux PPC: nur 32-Bit-JVM
  - Windows: nur 32-Bit-JVM
  - AIX: nur 32-Bit-JVM

Web-Browser-Voraussetzungen

Die Webkonsole unterstützt die folgenden Web-Browser:

- Mozilla Firefox Version 3.5.x und höher
- Mozilla Firefox Version 3.6.x und höher
- Microsoft Internet Explorer Version 7 oder 8

Voraussetzungen in Bezug auf WebSphere Application Server

- WebSphere Application Server Version 6.1.0.39 oder höher
- WebSphere Application Server Version 7.0.0.19 oder höher
- WebSphere Application Server Version 8.0.0.1 oder höher

Weitere Informationen finden Sie auf der Webseite mit den empfohlenen Fixes für WebSphere Application Server.

Weitere Voraussetzungen für den Anwendungsserver

Andere Java-EE-Implementierungen können die Laufzeitumgebung von eXtreme Scale als lokale Instanz oder als Client für Server von eXtreme Scale verwenden. Zum Implementieren von Java SE müssen Sie Version 5 oder höher verwenden.

Hinweise zu Java SE

WebSphere eXtreme Scale setzt Java SE 5 oder höher voraus. Im Allgemeinen bieten neuere Versionen von Java SE eine bessere Funktionalität und Leistung.

Unterstützte Versionen

Sie können WebSphere eXtreme Scale mit Java SE 5 oder höher verwenden. Die Version, die Sie verwenden, muss vom JRE-Anbieter (Java Runtime Environment) vollständig unterstützt werden.

Eine vollständig unterstützte JRE wird im Rahmen der eigenständigen Installationen von WebSphere eXtreme Scale und WebSphere eXtreme Scale Client im Verzeichnis *WXS-Installationsstammverzeichnis*`/java` installiert und kann von Clients und Servern verwendet werden. Wenn Sie WebSphere eXtreme Scale in WebSphere Application Server installieren, können Sie die JRE verwenden, die in der Installation von WebSphere Application Server enthalten ist.

WebSphere eXtreme Scale nutzt die Funktionalität von Java Development Kit (JDK) 5 oder höher, sobald die diese Komponente verfügbar ist. Im Allgemeinen bieten neuere Versionen von Java Development Kit (JDK) und Java SE eine bessere Leistung und eine bessere Funktionalität.

Weitere Informationen finden Sie auf der Webseite Supported software.

Java-abhängige Features von WebSphere eXtreme Scale

Tabelle 2. Features, die Java SE 5 oder Java SE 6 voraussetzen.

WebSphere eXtreme Scale verwendet Funktionalität, die in Java SE 5 bzw. Java SE 6 eingeführt wurde, um die folgenden Produktfeatures bereitzustellen.

Feature	Unterstützt in Java SE 5 und höher	Unterstützt in Java SE 6 und höher
Annotationen der API "EntityManager" (optional: Sie können auch XML-Dateien verwenden)	X	X
Java Persistence API (JPA): JPA-Loader, JPA-Client-Loader und zeitbasierte JPA-Aktualisierungskomponente	X	X
Speicherbasierte Bereinigung (verwendet MemoryPoolMXBean)	X	X
Instrumentierungsagenten: • `wxssizeagent.jar`: Erhöht die Genauigkeit von Map-Metriken zur Bytebelegung. • `ogagent.jar`: Erhöht die Leistung von Entitäten mit Feldzugriff.	X	X
Webkonsole für die Überwachung		X

Hinweise zu Java EE

Bei der Vorbereitung der Integration von WebSphere eXtreme Scale in eine Java-EE-Umgebung müssen Sie bestimmte Punkte berücksichtigen, wie z. B. Versionen, Konfigurationsoptionen, Voraussetzungen und Einschränkungen sowie Anwendungsimplementierung und -management.

eXtreme-Scale-Anwendungen in einer Java-EE-Umgebung ausführen

Eine Java-EE-Anwendung kann eine Verbindung zu einer fernen eXtreme-Scale-Anwendung herstellen. Außerdem unterstützt die Umgebung von WebSphere Application Server das Starten eines eXtreme-Scale-Servers beim Starten einer Anwendung im Anwendungsserver.

Wenn Sie eine XML-Datei zum Erstellen einer ObjectGrid-Instanz verwenden und die XML-Datei im Modul der EAR-Datei enthalten ist, greifen Sie mit der Methode getClass().getClassLoader().getResource("META-INF/objGrid.xml") auf die Datei zu, um ein URL-Objekt für die Erstellung einer ObjectGrid-Instanz abzurufen. Setzen Sie im Methodenaufruf den Namen der XML-Datei ein, die Sie verwenden.

Sie können Startup-Beans für eine Anwendung verwenden, um eine ObjectGrid-Instanz beim Starten einer Anwendung zu booten und um die Instanz beim Stoppen der Anwendung zu löschen. Eine Startup-Bean ist eine Stateless Session-Bean mit einer fernen com.ibm.websphere.startupservice.AppStartUpHome-Position und

einer fernen com.ibm.websphere.startupservice.AppStartUp-Schnittstelle. Die ferne Schnittstelle hat zwei Methoden: die Methode start und die Methode stop. Verwenden Sie die Methode start, um die Instanz zu booten, und die Methode stop, um die Instanz zu löschen. Die Anwendung verwendet die Methode ObjectGridManager.getObjectGrid, um eine Referenz auf die Instanz zu verwalten. Weitere Einzelheiten finden Sie in den Informationen zum Zugriff auf ein ObjectGrid mit ObjectGridManager in der Veröffentlichung *Programmierung*.

Klassenladeprogramme verwenden

Wenn Anwendungsmodule, die unterschiedliche Klassenladeprogramme verwenden, eine einzige ObjectGrid-Instanz in einer Java-EE-Anwendung gemeinsam nutzen, müssen Sie sicherstellen, dass die Objekte, die in eXtreme Scale gespeichert werden, und die Plug-ins für das Produkt in einem gemeinsamen Loader der Anwendung enthalten sind.

Lebenszyklus von ObjectGrid-Instanzen in einem Servlet verwalten

Für die Verwaltung des Lebenszyklus einer ObjectGrid-Instanz in einem Servlet können Sie die Methode init verwenden, um die Instanz zu erstellen, und die Methode "destroy", um die Instanz zu entfernen. Wenn die Instanz zwischengespeichert ist, wird sie im Servlet-Code abgerufen und bearbeitet. Weitere Einzelheiten finden Sie in den Informationen zum Zugriff auf ein ObjectGrid mit der Schnittstelle ObjectGridManager in der Veröffentlichung *Programmierung*.

Verzeichniskonventionen

Die folgenden Verzeichniskonventionen werden in der Dokumentation verwendet, um auf spezielle Verzeichnisse zu verweisen, wie z.B. *WXS-Installationsstammverzeichnis* und *WXS-Ausgangsverzeichnis*. Sie greifen in verschiedenen Szenarien, wie z. B. während der Installation oder der Verwendung der Befehlszeilenprogramme, auf diese Verzeichnisse zu.

WXS-Installationsstammverzeichnis
Das Verzeichnis *WXS-Installationsstammverzeichnis* ist das Stammverzeichnis, in dem die Produktdateien von WebSphere eXtreme Scale installiert sind. Das Verzeichnis *WXS-Installationsstammverzeichnis* kann das Verzeichnis sein, in dem das Testarchiv entpackt wurde, oder das Verzeichnis, in dem das Produkt WebSphere eXtreme Scale installiert ist.

- Beispiel für die entpackte Testversion:

  Beispiel: `/opt/IBM/WebSphere/eXtremeScale`
- Beispiel, wenn WebSphere eXtreme Scale in einem eigenständigen Verzeichnis installiert ist:

  Beispiel: `/opt/IBM/eXtremeScale`
- Beispiel, wenn WebSphere eXtreme Scale mit WebSphere Application Server integriert ist:

  Beispiel: `/opt/IBM/WebSphere/AppServer`

WXS-Ausgangsverzeichnis
Das Verzeichnis *WXS-Ausgangsverzeichnis* ist das Stammverzeichnis der Produktbibliotheken, Beispiele und Komponenten von WebSphere eXtreme Scale. Dieses Verzeichnis entspricht dem Verzeichnis *WXS-Installationsstammverzeichnis*, wenn die Testversion entpackt wurde. Bei eigenständigen Installationen ist das Verzeichnis *WXS-Ausgangsverzeichnis* das Unterverzeichnis `ObjectGrid` im Verzeichnis *WXS-*

Installationsstammverzeichnis. Bei Installationen, die mit WebSphere Application Server integriert sind, ist dieses Verzeichnis das Verzeichnis `optionalLibraries/ObjectGrid` im Verzeichnis *WXS-Installationsstammverzeichnis*.

- Beispiel für die entpackte Testversion:

  Beispiel: `/opt/IBM/WebSphere/eXtremeScale`
- Beispiel, wenn WebSphere eXtreme Scale in einem eigenständigen Verzeichnis installiert ist:

  Beispiel: `/opt/IBM/eXtremeScale/ObjectGrid`
- Beispiel, wenn WebSphere eXtreme Scale mit WebSphere Application Server integriert ist:

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

WAS-Stammverzeichnis
Das Verzeichnis *WAS-Stammverzeichnis* ist das Stammverzeichnis einer Installation von WebSphere Application Server:

Beispiel: `/opt/IBM/WebSphere/AppServer`

Ausgangsverzeichnis_des_REST-Service
Das Verzeichnis *Ausgangsverzeichnis_des_REST-Service* ist das Verzeichnis, in dem die Bibliotheken und Beispiele des REST-Datenservice von WebSphere eXtreme Scale enthalten sind. Dieses Verzeichnis hat den Namen `restservice` und ist ein Unterverzeichnis im Verzeichnis *WXS-Ausgangsverzeichnis*.

- Beispiel für eigenständige Implementierungen:

  Beispiel: `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`
- Beispiel für integrierte Implementierungen mit WebSphere Application Server:

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

Tomcat-Stammverzeichnis
Das *Tomcat-Stammverzeichnis* ist das Stammverzeichnis der Apache-Tomcat-Installation.

Beispiel: `/opt/tomcat5.5`

WASCE-Stammverzeichnis
Das *WASCE-Stammverzeichnis* ist das Stammverzeichnis der Installation von WebSphere Application Server Community Edition.

Beispiel: `/opt/IBM/WebSphere/AppServerCE`

Java-Ausgangsverzeichnis
Das *Java-Ausgangsverzeichnis* ist das Stammverzeichnis der Installation von Java Runtime Environment (JRE).

Beispiel: `/opt/IBM/WebSphere/eXtremeScale/java`

Ausgangsverzeichnis_für_Beispiele
Das *Ausgangsverzeichnis_für_Beispiele* ist das Verzeichnis, in dem Sie die Beispieldateien entpacken, die für die Lernprogramme verwendet werden.

Beispiel: `/wxs-samples/`

DVD-Stammverzeichnis
Das Verzeichnis *DVD-Stammverzeichnis* ist das Stammverzeichnis der DVD, die das Produkt enthält.

Beispiel: `dvd_root/docs/`

Equinox-Stammverzeichnis
Das Verzeichnis *Equinox-Stammverzeichnis* ist das Stammverzeichnis der Eclipse-Equinox-OSGi-Framework-Installation.

Beispiel:`/opt/equinox`

Benutzerausgangsverzeichnis
Das Verzeichnis *Benutzerausgangsverzeichnis* ist die Position, an der Benutzerdateien gespeichert werden, wie z. B. Sicherheitsprofile.

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

UNIX `/home/`*`Benutzername`*

Umgebungskapazität planen

Wenn Sie eine anfängliche Datasetgröße und eine geplante Datasetgröße haben, können Sie die für die Ausführung von WebSphere eXtreme Scale erforderliche Kapazität planen. Mit diesen Planungsübungen können Sie WebSphere eXtreme Scale effizient für künftige Änderungen planen und die Elastizität des Datengrids maximieren. Diese Möglichkeit haben Sie in einem anderen Szenario, wie z. B. einer speicherinternen Datenbank oder einem anderen Typ von Datenbank, nicht.

Speicher dimensionieren und Partitionsanzahl berechnen

Sie können die für Ihre spezielle Konfiguration benötigte Speicherkapazität und Partitionsanzahl berechnen.

Achtung: Verwenden Sie diesen Artikel, wenn Sie den Kopiermodus COPY_TO_BYTES nicht verwenden. Wenn sie den Modus COPY_TO_BYTES verwenden, ist die Speicherkapazität sehr viel geringer und die Vorgehensweise bei der Berechnung anders.

WebSphere eXtreme Scale speichert Daten im Adressraum von Java Virtual Machines (JVM). Jede JVM stellt Prozessorplatz für die Bearbeitung von Erstellungs-, Abruf-, Aktualisierungs- und Löschaufrufen für Daten bereit, die in der JVM gespeichert sind. Außerdem stellt jede JVM Speicherplatz für Dateneinträge und Replikate bereit. Java-Objekte variieren in ihrer Größe, und deshalb müssen Sie Messungen durchführen, um die benötigte Speicherkapazität zu schätzen.

Zur Dimensionierung des benötigten Speichers laden Sie Ihre Anwendungsdaten in eine einzige JVM. Wenn die Heapspeicherbelegung einen Wert von 60 % erreicht, notieren Sie die Anzahl der verwendeten Objekte. Diese Zahl ist die empfohlene maximale Objektanzahl für jede Ihrer JVMs. Für eine möglichst genaue Dimensionierung sollten Sie realistische Daten verwenden und alle definierten Indizes einbeziehen, weil Indizes auch Speicher belegen. Die beste Methode für die Speicherdimensionierung ist die Durchführung einer ausführlichen Garbage-Collection (mit **verbosegc**), da diese Ausgabe Ihnen die Zahlen nach der Garbage-Collection liefert. Sie können die Heapspeicherbelegung jederzeit über MBeans oder über das Programm abfragen, aber diese Abfragen liefern Ihnen nur eine aktuelle Momentaufnahme des Heapspeichers, die nicht erfassten Garbage (fehlerhafte Daten) enthalten kann. Deshalb ist die Verwendung dieser Methode keine genaue Indikation für den belegten Speicher.

Konfiguration vertikal skalieren

Anzahl der Shards pro Partition (numShardsPerPartition)

Zum Berechnen der Shard-Anzahl pro Partition bzw. des Werts von "numShardsPerPartition" addieren Sie 1 für das primäre Shard und die Gesamtanzahl der gewünschten Replikat-Shards.

```
numShardsPerPartition = 1 + Gesamtanzahl_der_Replikate
```

Anzahl der JVMs (minNumJVMs)

Für die vertikale Skalierung der Konfiguration müssen Sie zuerst die maximale Anzahl an Objekten festlegen, die insgesamt gespeichert werden müssen. Verwenden Sie die folgende Formel, um die Anzahl der benötigten JVMs zu bestimmen:

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

Runden Sie diesen Wert auf die nächst höhere ganze Zahl auf.

Anzahl der Shards (numShards)

Bei der endgültigen Größe sollten Sie 10 Shards für jede JVM verwenden. Wie zuvor beschrieben, hat jede JVM ein primäres Shard und (N-1) Replikat-Shards bzw. in diesem Fall neun Replikate. Da Sie bereits eine Zahl für die Java Virtual Machines haben, in denen die Daten gespeichert werden, können Sie die JVM-Zahl mit 10 multiplizieren, um die Anzahl der Shards zu bestimmen:

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

Anzahl der Partitionen

Wenn eine Partition ein einziges primäres Shard und ein einziges Replikat-Shard hat, hat die Partition zwei Shards (das primäre Shard und das Replikat-Shard). Die Anzahl der Partitionen entspricht der Shard-Anzahl, geteilt durch 2, aufgerundet auf die nächst höhere Primzahl. Wenn die Partition ein primäres Shard und zwei Replikat-Shards hat, entspricht die Anzahl der Partitionen der Shard-Anzahl, geteilt durch 3, aufgerundet auf die nächst höhere Primzahl.

```
numPartitions = numShards / numShardsPerPartition
```

Skalierungsbeispiel

In diesem Beispiel wird von einer anfänglichen Eintragsanzahl von 250 Millionen ausgegangen. Jedes Jahr nimmt die Anzahl der Einträge um etwa 14 % zu. Nach sieben Jahren sind insgesamt 500 Millionen Einträge vorhanden. Deshalb müssen Sie Ihre Kapazität entsprechend planen. Für die hohe Verfügbarkeit wird ein einziges Replikat benötigt. Mit einem Replikat verdoppelt sich die Anzahl der Einträge, d. h. auf 1.000.000.000 Einträge. Zu Testzwecken können 2 Millionen Einträge in jeder JVM gespeichert werden. Laut den Berechnungen wird dann in diesem Szenario die folgende Konfiguration benötigt:

- 500 JVMs zum Speichern der endgültigen Anzahl an Einträgen
- 5000 Shards (500 Java Virtual Machines mal 10)
- 2500 Partitionen bzw. 2503, da dies die nächst höhere Primzahl ist (5000 Shards, geteilt durch zwei für primäre Shards und Replikat-Shards)

Anfangskonfiguration

Basierend auf den vorherigen Berechnungen beginnen Sie mit 250 Java Virtual Machines und steigern sich dann im Lauf von fünf Jahren auf 500 Java Virtual Machines. Mit dieser Konfiguration können Sie ein inkrementelles Wachstum verwalten, bis Sie die endgültige Anzahl von Einträgen erreichen.

In dieser Konfiguration werden ungefähr 200.000 Einträge pro Partition (500 Millionen Einträge, geteilt durch 2503 Partitionen) gespeichert. Setzen Sie den Parameter **numberOfBuckets** in der Map, die die Einträge enthält, auf die nächst höhere Primzahl setzen (in diesem Beispiel 70887), die das Verhältnis bei ungefähr drei hält.

Erreichen der maximalen Anzahl an Java Virtual Machines

Wenn Sie die maximale Anzahl von 500 Java Virtual Machines erreichen, können Sie Ihr Datengrid trotzdem weiter vergrößern. Da die Anzahl der Java Virtual Machines über 500 steigt, fällt die Shard-Anzahl für jede JVM unter 10, was unter der empfohlenen Zahl liegt. Die Shards werden größer, was zu Problemen führen kann. Wiederholen Sie den Dimensionierungsprozess unter Berücksichtigung des künftigen Wachstums, und setzen Sie die Partitionsanzahl zurück. Dieses Verfahren erfordert einen vollständigen Neustart bzw. die Außerbetriebnahme des Datengrids.

Anzahl der Server

Achtung: Verwenden Sie unter keinen Umständen Paging auf einem Server.

Die Speicherbelegung einer einzigen JVM ist höher als die Größe des Heapspeichers. Bei einem Heapspeicher mit einer Größe von 1 GB für eine JVM werden tatsächlich 1,4 GB Realspeicher belegt. Bestimmen Sie den verfügbaren freien Arbeitsspeicher des Servers. Teilen Sie die Arbeitsspeicherkapazität durch den Speicher pro JVM, um die maximale Anzahl an Java Virtual Machines auf dem Server zu erhalten.

CPU-Dimensionierung pro Partition für Transaktionen

Obwohl die Hauptfunktionalität von eXtreme Scale die elastische Skalierung ist, ist es auch wichtig, die optimale Anzahl an CPUs zu dimensionieren und anzupassen, um eine vertikale Skalierung zu erreichen.

Die Prozessorkosten setzen sich wie folgt zusammen:

- Kosten für die Verarbeitung von Erstellungs-, Abruf-, Aktualisierungs- und Löschoperationen von Clients,
- Kosten für die Replikation von Daten anderer Java Virtual Machines,
- Kosten für die Invalidierung,
- Kosten für die Bereinigungsrichtlinie,
- Kosten für die Garbage-Collection,
- Kosten für die Anwendungslogik,
- Kosten für die Serialisierung.

Java Virtual Machines pro Server

Verwenden Sie zwei Server, und starten Sie die maximale Anzahl an JVMs pro Server. Verwenden Sie die im vorherigen Abschnitt berechnete Partitionsanzahl. Laden Sie vorab nur so viele Daten in die Java Virtual Machines, wie auf diese beiden Computer passen. Verwenden Sie einen gesonderten Server als Client. Führen Sie eine realistische Transaktionssimulation für das Datengrid der beiden Server durch.

Versuchen Sie die Prozessorauslastung zu sättigen, um das Ausgangsniveau zu berechnen. Sollte dies nicht möglich sein, ist das Netz wahrscheinlich gesättigt. Wenn das Netz gesättigt ist, fügen Sie weitere Netzkarten hinzu, und verteilen Sie die Java Virtual Machines auf die verfügbaren Netzkarten.

Führen Sie die Computer mit einer Prozessorauslastung von 60 % aus, und messen Sie die Raten für die Erstellungs- (create), Abruf- (retrieve), Aktualisierungs- (update) und Löschtransaktionen (delete). Diese Messung liefert Ihnen den Durchsatz auf den beiden Servern. Diese Zahl verdoppelt sich bei vier Servern und verdoppelt sich dann nochmal bei 8 Servern usw. Bei dieser Skalierung wird davon ausgegangen, dass die Netzkapazität und die Clientkapazität ebenfalls skalierbar sind.

Die Antwortzeiten von eXtreme Scale sollten mit zunehmender Anzahl an Servern somit stabil bleiben. Der Transaktionsdurchsatz sollte linear steigen, wenn dem Datengrid Computer hinzugefügt werden.

CPU-Dimensionierung für parallele Transaktionen

Einzelpartitionstransaktionen haben einen Durchsatz, der linear zum Wachstum des Datengrids steigt. Parallele Transaktionen unterscheiden sich von Einzelpartitionstransaktionen, weil sie einen Teil der Server (oder auch alle Server) betreffen.

Wenn eine Transaktion alle Server betrifft, ist der Durchsatz auf den Durchsatz des Clients, der die Transaktion einleitet, bzw. auf den Durchsatz des langsamsten betroffenen Servers beschränkt. In größeren Datengrids werden die Daten breiter verteilt. Diese Grids stellen mehr Prozessorplatz, Speicherplatz, Netzkapazität usw. bereit. Der Client muss jedoch auf die Antwort des langsamsten Servers warten, und der Client muss die Ergebnisse der Transaktion konsumieren.

Wenn eine Transaktion einen Teil der Server betrifft, erhalten M von N Servern eine Anforderung. Der Durchsatz ist dann N/M-Mal so hoch wie der Durchsatz des langsamsten Servers. Wenn Sie beispielsweise 20 Server und eine Transaktion haben, die 5 Server betrifft, ist der Durchsatz 4 Mal so hoch wie der Durchsatz des langsamsten Servers im Datengrid.

Nach Abschluss einer parallelen Transaktion werden die Ergebnisse an den Client-Thread gesendet, der die Transaktion gestartet hat. Dieser Client muss daraufhin die Ergebnisse in Einzel-Threads zusammenfassen. Die Dauer dieser Zusammenfassung (oder Aggregation) erhöht sich mit zunehmender Anzahl der von der Transaktion betroffenen Server. Diese Dauer hängt jedoch von der Anwendung ab, da es möglich ist, dass jeder Server bei zunehmender Größe des Datengrids ein kleineres Ergebnis zurückgibt.

Gewöhnlich betreffen parallele Transaktionen alle Server im Datengrid, weil Partitionen gleichmäßig auf das Grid verteilt werden. In diesem Fall ist der Durchsatz, wie im ersten Fall beschrieben, beschränkt.

Zusammenfassung

Für diese Dimensionierung stehen Ihnen drei Messwerte zur Verfügung:

- Anzahl der Partitionen,
- Anzahl der Server, die für den erforderlichen Speicherbedarf benötigt werden,
- Anzahl der Server, die für den erforderlichen Durchsatz benötigt werden.

Wenn Sie 10 Server für den Speicherbedarf benötigen, aber aufgrund der Prozessorsättigung nur 50 % des erforderlichen Durchsatzes erzielen, benötigen Sie doppelt so viele Server wie vorhanden.

Die höchste Stabilität erzielen Sie, wenn Sie Ihre Server mit einer Prozessorauslastung von 60 % und Ihre JVMs mit einer Heapspeicherauslastung von 60 % betreiben. So können Lastspitzen die Prozessorauslastung auf 80–90 % hochtreiben. Ein

dauerhafter Betrieb der Server mit diesen Ständen oder höher sollte aber vermieden werden.

Kapazitätsplanung für den dynamischen Cache

Die Anwendungsprogrammierschnittstelle (API, Application Programming Interface) für dynamischen Cache steht Java-EE-Anwendungen zur Verfügung, die in WebSphere Application Server implementiert sind. Der dynamische Cache kann genutzt werden, um Geschäftsdaten und generierte HTML zwischenzuspeichern oder um die zwischengespeicherten Daten in der Zelle über den Datenreplikationsservice (DRS) zu synchronisieren.

Übersicht

Alle dynamischen Cacheinstanzen, die mit dem dynamischen Cache-Provider von WebSphere eXtreme Scale erstellt werden, sind standardmäßig hoch verfügbar. Die Stufe und die Speicherkosten der hohen Verfügbarkeit sind von der verwendeten Topologie abhängig.

Wenn Sie die integrierte Topologie verwenden, ist die Cachegröße auf den freien Speicher in einem einzelnen Serverprozess beschränkt, und in jedem Serverprozess wird eine vollständige Kopie des Caches gespeichert. Solange auch nur ein einziger Serverprozess aktiv ist, bleibt auch der Cache aktiv. Die Cachedaten gehen nur dann verloren, wenn alle Server, die auf den Cache zugreifen, beendet werden.

Beim Caching mit der integrierten partitionierten Topologie ist die Cachegröße auf den summierten freien Speicher aller Serverprozesse beschränkt. Standardmäßig verwendet der dynamische Cache-Provider von eXtreme Scale ein Replikat für jedes primäre Shard, d. h., jedes einzelne zwischengespeicherte Datenelement wird zweimal gespeichert.

Verwenden Sie die folgende Formel A, um die Kapazität eines integrierten partitionierten Caches zu bestimmen:

Formel A

F * C / (1 + R) = M

Für diese Formel gilt Folgendes:

- F = Freier Speicher pro Containerprozess
- C = Anzahl der Container
- R = Anzahl der Replikate
- M = Gesamtgröße des Caches

Für ein Datengrid von WebSphere Application Server Network Deployment mit 256 MB verfügbarem Speicher in jedem Prozess und 4 Serverprozessen insgesamt können in einer Cacheinstanz über alle diese Server verteilt 512 Megabyte Daten gespeichert werden. In diesem Modus kann der Cache einen Serverabsturz ohne Datenverlust "überleben". Es könnten sogar nacheinander zwei Server beendet werden, ohne dass irgendwelche Daten verloren gehen. Für das vorherige Beispiel lautet die Formel deshalb wie folgt:

256 MB * 4 Container/ (1 primäres Shard + 1 Replikat) = 512 MB

Caches, die die ferne Topologie verwenden, haben ähnliche Größenmerkmale wie Caches, die die integrierte partitionierte Topologie verwenden, sind jedoch auf den summierten verfügbaren Speicher aller Containerprozesse von eXtreme Scale beschränkt.

In fernen Topologien kann die Anzahl der Replikate erhöht werden, um eine höhere Stufe der Verfügbarkeit zu erreichen, wodurch sich jedoch der Speicheraufwand erhöht. In den meisten dynamischen Cacheanwendungen ist dies in der Regel nicht erforderlich, aber Sie können die Datei `dynacache-remote-deployment.xml` bearbeiten, um die Anzahl der Replikate zu erhöhen.

Verwenden Sie die folgenden Formeln (B und C), um zu berechnen, welche Auswirkungen das Hinzufügen weiterer Replikate auf die hohe Verfügbarkeit des Caches hat.

Formel B

N = Minimum(T -1, R)

Für diese Formel gilt Folgendes:
- N = Anzahl der Prozesse, die gleichzeitig abstürzen
- T = Gesamtanzahl der Container
- R = Gesamtanzahl der Replikate

Formel C

Obere Grenze(T/ (1+N)) = m

Für diese Formel gilt Folgendes:
- T = Gesamtanzahl der Container
- N = Gesamtanzahl der Replikate
- m = Erforderliche Mindestanzahl an Containern für die Unterstützung der Cachedaten

Informationen zur Leistungsoptimierung mit dem dynamischen Cache-Provider finden Sie im Abschnitt „Dynamischen Cache-Provider optimieren" auf Seite 513.

Cachegröße festlegen

Bevor eine Anwendung, die den dynamischen Cache-Provider von WebSphere eXtreme Scale verwendet, implementiert werden kann, müssen die allgemeinen Principals, die im vorherigen Abschnitt beschrieben wurden, mit den Umgebungsdaten für die Produktionssysteme kombiniert werden. Der erste zu ermittelnde Wert ist die Gesamtanzahl der Containerprozesse und der verfügbare Hauptspeicher jedes einzelnen Prozesses zum Speichern der Cachedaten. Wenn Sie die integrierte Topologie verwenden, befinden sich die Cachecontainer in den Prozessen von WebSphere Application Server. d. h., es gibt einen Container pro Server, der den Cache verwendet. Die Bestimmung der Speicherkosten der Anwendung ohne aktiviertes Caching und ohne WebSphere Application Server ist die beste Methode zu ermitteln, wie viel Speicher im Prozess verfügbar ist. Zur Ermittlung dieses Werts bietet sich die Analyse der Daten einer ausführlichen Garbage-Collection an. Wenn Sie die ferne Topologie verwenden, können diese Informationen anhand der Ausgabe der ausführlichen Garbage-Collection für einen neu gestarteten eigenständigen Container ermittelt werden, der noch nicht mit Cachedaten gefüllt wurde. Ein letz-

ter Aspekt, der bei der Ermittlung des für Cachedaten verfügbaren Speichers pro Prozess berücksichtigt werden muss, ist die Reservierung einer gewissen Menge des Heapspeichers für die Garbage-Collection. Die Summe aus Containerkosten (Umgebung mit WebSphere Application Server oder eigenständige Umgebung) und reservierter Größe für den Cache sollte nicht mehr als 70 % der Gesamtgröße des Heapspeichers betragen.

Nachdem Sie diese Informationen zusammengestellt haben, können Sie die Werte in die zuvor beschriebene Formel A eintragen, um die maximale Größe für den partitionierten Cache zu bestimmen. Sobald die maximale Größe bekannt ist, müssen Sie im nächsten Schritt die Gesamtanzahl der Cacheeinträge bestimmen, die unterstützt werden können. Hierfür muss zunächst die durchschnittliche Größe pro Cacheeintrag bestimmt werden. Eine einfache Methode zur Bestimmung dieses Werts ist, 10 % auf die Größe des Kundenobjekts aufzuschlagen. Ausführlichere Informationen zur Festlegung der Größe von Cacheeinträgen bei der Verwendung des dynamischen Caches finden Sie in der Veröffentlichung "Tuning guide for dynamic cache and data replication service".

Wenn die Komprimierung aktiviert ist, wirkt diese sich auf die Größe des Kundenobjekts aus, aber nicht auf die Kosten des Caching-Systems. Verwenden Sie die folgende Formel, um die Größe eines zwischengespeicherten Objekts zu bestimmen, wenn Sie mit Komprimierung arbeiten:

S = O * C + O * 0.10

Für diese Formel gilt Folgendes:

- S = Durchschnittliche Größe eines zwischengespeicherten Objekts
- O = Durchschnittliche Größe eines nicht komprimierten Kundenobjekts
- C = Komprimierungsfaktor als Bruchzahl

Komprimierung: Ein Komprimierungsfaktor von 2 zu 1 ist 1/2 = 0,50. Je kleiner der Faktor ist, desto besser. Wenn das zu speichernde Objekt ein normales POJO mit überwiegend primitiven Datentypen ist, können Sie von einem Komprimierungsfaktor von 0,60 bis 0,70 ausgehen. Wenn das zwischengespeicherte Objekte ein Servlet, eine JSP-Datei oder ein Web-Service-Objekt ist, ist die optimale Methode für die Bestimmung des Komprimierungsfaktors, ein repräsentatives Beispielobjekt mit einem Komprimierungsdienstprogramm zu komprimieren. Sollte dies nicht möglich sein, können Sie im Allgemeinen von einem Komprimierungsfaktor von 0,2 bis 0,35 für diesen Typ von Daten ausgehen.

Anschließend verwenden Sie diese Informationen, um die Gesamtanzahl der Cacheeinträge zu bestimmen, die unterstützt werden können. Verwenden Sie die folgende Formel D:

Formel D

T = S / A

Für diese Formel gilt Folgendes:

- T = Gesamtanzahl der Cacheeinträge
- S = Verfügbare Gesamtkapazität für Cachedaten laut Berechnung mit Formel A
- A = Durchschnittliche Größe eines Cacheeintrags

Abschließend müssen Sie die Cachegröße in der dynamischen Cacheinstanz festlegen, damit dieser Grenzwert wirksam wird. Der dynamische Cache-Provider von

WebSphere eXtreme Scale unterscheidet sich in dieser Beziehung vom dynamischen Standardcacheprovider. Verwenden Sie die folgende Formel E, um den Wert zu bestimmen, der für die Cachegröße in der dynamischen Cacheinstanz festzulegen ist:

Formel E

Cs = Ts / Np

Für diese Formel gilt Folgendes:

- Ts = Gesamtzielgröße für den Cache
- Cs = In der dynamischen Cacheinstanz festzulegende Cachegröße
- Np = Anzahl der Partitionen (der Standardwert ist 47)

Setzen Sie die Größe der dynamischen Cacheinstanz in allen Servern, die die Cacheinstanz gemeinsam nutzen, auf den mit der Formel E berechneten Wert.

Konfiguration planen

Bevor Sie die Hardware oder Software konfigurieren, müssen Sie Folgendes berücksichtigen.

Prüfliste für die Betriebsbereitschaft

Verwenden Sie die Prüfliste für die Betriebsbereitschaft, um Ihre Umgebung für die Implementierung von WebSphere eXtreme Scale vorzubereiten.

Tabelle 3. Prüfliste für die Betriebsbereitschaft

Prüflistenpunkt	Weiterführende Informationen

Wenn Sie AIX verwenden, optimieren Sie die folgenden Betriebssystemeinstellungen:
 	
TCP_KEEPINTVL
Die Einstellung TCP_KEEPINTVL gehört zu einem Socket-Keepalive-Protokoll, das die Erkennung von Netzausfällen ermöglicht. Die Eigenschaft gibt das Intervall an, in dem Pakete zum Validieren der Verbindung gesendet werden. Wenn Sie WebSphere eXtreme Scale verwenden, setzen Sie die Eigenschaft auf `10`. Zum Überprüfen der aktuellen Einstellung führen Sie den folgenden Befehl aus:
`# no –o tcp_keepintvl`
Führen Sie den folgenden Befehl aus, um die aktuelle Einstellung zu ändern:
`# no –o tcp_keepintvl=10`
Der Wert für die Einstellung TCP_KEEPINTVL wird in Halbsekunden angegeben.
 	
TCP_KEEPINIT
Die Einstellung TCP_KEEPINIT gehört zu einem Socket-Keepalive-Protokoll, das die Erkennung von Netzausfällen ermöglicht. Die Eigenschaft gibt das Anfangszeitlimit für die TCP-Verbindung an. Wenn Sie WebSphere eXtreme Scale verwenden, setzen Sie die Eigenschaft auf `40`. Zum Überprüfen der aktuellen Einstellung führen Sie den folgenden Befehl aus:
`# no –o tcp_keepinit`
Führen Sie den folgenden Befehl aus, um die aktuelle Einstellung zu ändern:
`# no –o tcp_keepinit=40`
Der Wert für die Einstellung TCP_KEEPINIT wird in Halbsekunden angegeben.
	• Informationen zur Optimierung von AIX finden Sie unter "AIX-Systeme optimieren".

Aktualisieren Sie die Datei `orb.properties`, um das Transportverhalten des Grids zu ändern. Sie finden die Datei `orb.properties` im Verzeichnis `java/jre/lib`.	„ORB-Eigenschaften” auf Seite 500

Tabelle 3. Prüfliste für die Betriebsbereitschaft (Forts.)

Prüflistenpunkt	Weiterführende Informationen
Verwenden Sie Parameter im Script **startOgServer**, insbesondere die folgenden: • Legen Sie die Einstellungen für den Heapspeicher mit dem Parameter **-jvmArgs** fest. • Legen Sie den Anwendungsklassenpfad und Anwendungseigenschaften mit dem Parameter **-jvmArgs** fest. • Setzen Sie den Parameter **-jvmArgs** für die Konfiguration der Agentenüberwachung. **Porteinstellungen** WebSphere eXtreme Scale muss für einige Transporte Kommunikationsports öffnen. Diese Ports werden alle dynamisch definiert. Wenn jedoch eine Firewall zwischen den Containern verwendet wird, müssen Sie die Ports angeben. Verwenden Sie die folgenden Portinformationen: **Listener-Port** Sie können das Argument **-listenerPort** verwenden, um den Port anzugeben, der für die Kommunikation zwischen Prozessen verwendet wird. **Stammgruppenport** Sie können das Argument **-haManagerPort** verwenden, um den Port anzugeben, der für die Fehlererkennung verwendet wird. Dieses Argument ist mit "peerPort" identisch. Beachten Sie, dass Stammgruppen nicht zonenübergreifend kommunizieren müssen. Deshalb müssen Sie diesen Port unter Umständen nicht setzen, wenn die Firewall für alle Member einer einzigen Zone offen ist. **JMX-Serviceport** Sie können das Argument **-JMXServicePort** verwenden, um den Port anzugeben, den der JMX-Service verwenden soll. **SSL-Port** Wenn Sie `-Dcom.ibm.CSI.SSLPort=1234` als Argument mit **-jvmArgs** angeben, wird der SSL-Port auf `1234` gesetzt. Der SSL-Port ist der sichere Port-Peer zum Listener-Port. **Clientport** Wird nur im Katalogservice verwendet. Sie können diesen Wert mit dem Argument **-catalogServiceEndPoints** angeben. Der Wert für diesen Parameter muss im folgenden Format angegeben werden: `Servername:Hostname:Clientport:Peerport`	„Script **startOgServer**" auf Seite 409
Stellen Sie sicher, dass die Sicherheitseinstellungen ordnungsgemäß konfiguriert sind: • Transport (SSL) • Anwendung (Authentifizierung und Berechtigung) Zum Überprüfen Ihrer Sicherheitseinstellungen können Sie versuchen, einen zerstörerischen Client zu verwenden, der eine Verbindung zu Ihrer Konfiguration herstellt. Wenn beispielsweise die Einstellung "SSL-Required" konfiguriert ist, sollte ein Client, der die Einstellung TCP_IP hat, oder ein Client mit dem falschen Truststore nicht in der Lage sein, eine Verbindung zum Server herzustellen. Wenn eine Authentifizierung erforderlich ist, sollte ein Client ohne Berechtigungsnachweis (z. B. ohne Benutzer-ID und Kennwort) nicht in der Lage sein, eine Verbindung zum Server herzustellen. Wenn die Berechtigung zwingend erforderlich ist, sollte einem Client ohne Zugriffsberechtigung der Zugriff auf die Serverressourcen nicht erteilt werden.	„Sicherheitsintegration mit externen Providern" auf Seite 528

Tabelle 3. Prüfliste für die Betriebsbereitschaft (Forts.)

Prüflistenpunkt	Weiterführende Informationen
Legen Sie fest, wie die Umgebung überwacht werden soll. • Tool **xscmd**: – Die JMX-Ports der Katalogserver müssen für das Tool **xscmd** sichtbar sein. Auch die Container-Server-Ports müssen für einige Befehle, die Informationen von den Containern erfassen, zugänglich sein. • Überwachungskonsole: Mit der Überwachungskonsole können Sie aktuelle Statistiken und Protokollstatistiken in einem Diagramm darstellen. • Überwachungstools anderer Anbieter: – Tivoli Enterprise Monitoring Agent – CA Wily Introscope – Hyperic HQ	• „Überwachung mit dem Dienstprogramm **xscmd**" auf Seite 470 • „JMX-Sicherheit (Java Management Extensions)" auf Seite 526 • „Überwachung mit der Webkonsole" auf Seite 451 • „Überwachung mit IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale" auf Seite 485 • „eXtreme Scale mit Hyperic HQ überwachen" auf Seite 494 • „eXtreme-Scale-Anwendungen mit CA Wily Introscope überwachen" auf Seite 491

Netzports planen

WebSphere eXtreme Scale ist ein verteilter Cache, der das Öffnen von Ports für die Kommunikation mit dem ORB (Object Request Broker) und dem TCP-Stack (Transmission Control Protocol) zwischen Java Virtual Machines voraussetzt. Planen und steuern Sie Ihre Ports, insbesondere in einer Umgebung, die eine Firewall hat, und wenn Sie einen Katalogservice und Container an mehreren Ports verwenden.

Wichtig: Wenn Sie Portnummern angeben, vermeiden Sie die Festlegung von Ports, die in einem ephemeren Bereich für Ihr Betriebssystem enthalten sind. Wenn Sie einen Port verwenden, der in einem ephemeren Bereich enthalten ist, können Portkonflikte auftreten.

Katalogservicedomäne

Eine Katalogservicedomäne erfordert die Definition der folgenden Ports:

peerPort
Gibt den Port für den High-Availability Manager an, über den Peerkatalogserver über einen TCP-Stack miteinander kommunizieren. In WebSphere Application Server wird diese Einstellung von der Portkonfiguration des High Availability Manager übernommen.

clientPort
Gibt den Port für den Zugriff auf Katalogservicedaten für Katalogserver an. In WebSphere Application Server wird dieser Port über die Katalogservicedomänenkonfiguration festgelegt.

listenerPort
Gibt die Portnummer an, an die der Object Request Broker (ORB) gebunden wird. Diese Einstellung konfiguriert Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. In WebSphere Application Server wird der Listener-Port von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.

Standardeinstellung: 2809

JMXConnectorPort
Definiert den SSL-Port (Secure Sockets Layer), an den der JMX-Service (Java Management Extensions) gebunden wird.

Container-Server

Die Container-Server von WebSphere eXtreme Scale erfordern für den Betrieb ebenfalls verschiedene Ports. Standardmäßig generiert der Container-Server von eXtreme Scale seinen HA-Manager-Port und seinen ORB-Listener-Port automatisch mit dynamischen Ports. Für eine Umgebung mit einer Firewall empfiehlt es sich, Ports zu planen und zu steuern. Damit Container-Server an bestimmten Ports gestartet werden, können Sie die folgenden Optionen im Befehl **startOgServer** verwenden.

haManagerPort
Synonym mit Peer-Port. Gibt die Nummer des vom High Availability Manager verwendeten Ports an. Wenn Sie diese Eigenschaft nicht definieren, generiert der Katalogservice automatisch einen verfügbaren Port. Diese Eigenschaft gilt für den Container-Server und den Katalogservice. (Nur erforderlich für Umgebungen von WebSphere Application Server.)

listenerPort
Gibt die Portnummer an, an die der Object Request Broker (ORB) gebunden wird. Diese Einstellung konfiguriert Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. In WebSphere Application Server wird der Listener-Port von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.

Standardeinstellung: 2809

JMXConnectorPort
Definiert den SSL-Port (Secure Sockets Layer), an den der JMX-Service (Java Management Extensions) gebunden wird.

7.1.1+ **xioChannel.xioContainerTCPSecure.Port**
Gibt die SSL-Portnmmer von eXtremeIO auf dem Server an. Diese Eigenschaft wird nur verwendet, wenn die Eigenschaft **transportType** auf `SSL-Supported` oder `SSL-Required` gesetzt ist.

7.1.1+ **xioChannel.xioContainerTCPNonSecure.Port**
Gibt die Nummer des nicht sicheren Listener-Ports von eXtremeIO auf dem Server an. Wenn Sie keinen Wert festlegen, wird ein ephemerer Port verwendet. Diese Eigenschaft wird nur verwendet, wenn die Eigenschaft **transportType** auf `TCP/IP` gesetzt ist.

Eine ordnungsgemäße Planung der Portsteuerung ist von entscheidender Bedeutung, wenn Hunderte von Java Virtual Machines in einem Server gestartet werden. Wenn ein Portkonflikt vorliegt, werden die Container-Server nicht gestartet.

Clients

Clients von WebSphere eXtreme Scale können Rückrufe von Servern empfangen, wenn Sie die API DataGrid oder mehrere andere Befehle verwenden. Verwenden Sie die Eigenschaft **listenerPort** in der Clienteigenschaftendatei, um den Port anzugeben, an dem der Client auf Rückrufe vom Server wartet.

haManagerPort
Synonym mit Peer-Port. Gibt die Nummer des vom High Availability Manager verwendeten Ports an. Wenn Sie diese Eigenschaft nicht definieren, generiert der Katalogservice automatisch einen verfügbaren Port. Diese Eigenschaft gilt für den Container-Server und den Katalogservice. (Nur erforderlich für Umgebungen von WebSphere Application Server.)

jvmArgs (optional)
Gibt eine Liste mit JVM-Argumenten (Java Virtual Machine) an. Wenn die Sicherheit aktiviert ist, müssen Sie das folgende Argument verwenden, um den SSL-Port (Secure Socket Layer) zu konfigurieren: `-jvmArgs -Dcom.ibm.CSI.SSLPort=<SSL-Port>`.

listenerPort
Gibt die Portnummer an, an die der Object Request Broker (ORB) gebunden wird. Diese Einstellung konfiguriert Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. In WebSphere Application Server wird der Listener-Port von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.

Standardeinstellung: 2809

Ports in WebSphere Application Server

- Der Wert von **listenerPort** wird aus dem **BOOTSTRAP_ADDRESS**-Wert jedes Anwendungsservers von WebSphere Application Server übernommen.
- Die Werte von **haManagerPort** und **peerPort** werden aus dem **DCS_UNICAST_ADDRESS**-Wert jedes Anwendungsservers von WebSphere Application Server übernommen.

Sie können eine Katalogservicedomäne gemäß der Beschreibung im Artikel „Katalogservicedomänen in WebSphere Application Server erstellen" auf Seite 264 in der Administrationskonsole definieren.

Sie können die Ports für einen bestimmten Server anzeigen, indem Sie auf einen der folgenden Pfade in der Administrationskonsole klicken:

- WebSphere Application Server Network Deployment Version 6.1: **Server** > **Anwendungsserver** > *Servername* > **Ports** > *Endpunktname*
- WebSphere Application Server Network Deployment Version 7.0: **Server** > **Servertypen** > **WebSphere-Anwendungsserver** > *Servername* > **Ports** > ***Portname***

Übersicht über die Sicherheit

WebSphere eXtreme Scale kann den Datenzugriff sichern, unter anderem durch Integration mit externen Sicherheitsprovidern.

Anmerkung: In einem vorhandenen nicht zwischengespeicherten Datenspeicher, z. B. einer Datenbank, haben Sie wahrscheinlich integrierte Sicherheitsfeatures, die Sie nicht aktiv konfigurieren oder aktivieren müssen. Nachdem Sie Ihre Daten jedoch mit eXtreme Scale zwischengespeichert haben, müssen Sie die daraus resultierende wichtige Tatsache berücksichtigen, dass die Sicherheitsfeatures Ihres Back-Ends nicht mehr wirksam sind. Sie können die Sicherheit von eXtreme Scale auf den erforderlichen Stufen konfigurieren, so dass Ihre neue zwischengespeicherte Datenarchitektur ebenfalls sicher ist.
Es folgt eine kurze Zusammenfassung der Sicherheitsfeatures von eXtreme Scale. Ausführlichere Informationen zur Konfiguration der Sicherheit finden Sie in der Veröffentlichung *Verwaltung* und in der Veröffentlichung *Programmierung*.

Grundlegende Informationen zur verteilten Sicherheit

Die verteilte Sicherheit von eXtreme Scale basiert auf drei Schlüsselkonzepten:

Vertrauenswürdige Authentifizierung
Die Möglichkeit, die Identität des Anforderers zu bestimmen. WebSphere eXtreme Scale unterstützt Client/Server- und Server/Server-Authentifizierung.

Berechtigung
Die Möglichkeit, dem Anforderer Zugriffsberechtigungen zu erteilen. WebSphere eXtreme Scale unterstützt verschiedene Berechtigungen für verschiedene Operationen.

Sicherer Transport
Die sichere Übertragung von Daten über ein Netz. WebSphere eXtreme Scale unterstützt die Protokolle Layer Security/Secure Sockets Layer (TLS/SSL).

Authentifizierung

WebSphere eXtreme Scale unterstützt ein verteiltes Client/Server-Framework. Eine Client/Server-Sicherheitsinfrastruktur ist verfügbar, um den Zugriff auf Server von eXtreme Scale zu sichern. Wenn der Server von eXtreme Scale beispielsweise eine Authentifizierung erfordert, muss ein Client von eXtreme Scale Berechtigungsnachweise für die Authentifizierung beim Server vorlegen. Diese Berechtigungsnachweise können eine Kombination von Benutzername und Kennwort, ein Clientzertifikat, ein Kerberos-Ticket oder Daten sein, die in einem zwischen Client und Server vereinbarten Format präsentiert werden.

Berechtigung

Berechtigungen von WebSphere eXtreme Scale basieren auf Subject-Objekten und Berechtigungen. Sie können Java Authentication and Authorization Services (JAAS) für die Berechtigung des Zugriffs verwenden, oder Sie können eine angepasste Lösung wie Tivoli Access Manager (TAM) für die Behandlung der Berechtigungen integrieren. Die folgenden Berechtigungen können einem Client oder einer Gruppe erteilt werden:

Map-Berechtigung
Berechtigung zum Durchführen von Einfüge-, Lese-, Aktualisierungs-, Bereinigungs- oder Löschoperationen in Maps.

ObjectGrid-Berechtigung
Berechtigung zum Ausführen von Objekt- oder Entitätsabfragen und Datenstromabfragen für ObjectGrid-Objekte.

DataGrid-Agentenberechtigung
Berechtigung für die Implementierung von DataGrid-Agenten in einem ObjectGrid.

Serverseitige Map-Berechtigung
Berechtigung zum Replizieren einer Server-Map auf der Clientseite oder zum Erstellen eines dynamischen Index für die Server-Map.

Verwaltungsberechtigung
Berechtigung für die Ausführung von Verwaltungs-Tasks.

Transportsicherheit

Zum Sichern der Client/Server-Kommunikation unterstützt WebSphere eXtreme Scale TLS/SSL. Diese Protokolle bieten Sicherheit auf Transportebene mit Authentizität, Integrität und Vertraulichkeit für eine sichere Verbindung zwischen einem

Client und einem Server von eXtreme Scale.

Gridsicherheit

In einer sicheren Umgebung muss ein Server in der Lage sein, die Authentizität eines anderen Servers zu prüfen. WebSphere eXtreme Scale verwendet für diesen Zweck einen Mechanismus mit Shared-Secret-Schlüsselzeichenfolgen. Dieser Shared-Secret-Schlüsselmechanismus gleicht einem gemeinsam genutzten Kennwort. Alle Server von eXtreme Scale stimmen der Verwendung einer gemeinsamen Shared-Secret-Zeichenfolge zu. Wenn ein Server dem Datengrid beitritt, wird er aufgefordert, diese Shared-Secret-Zeichenfolge vorzulegen. Wenn die Shared-Secret-Zeichenfolge des beitretenden Servers der Zeichenfolge im Masterserver entspricht, kann der Server dem Grid beitreten. Andernfalls wird die Beitrittsanforderung zurückgewiesen.

Das Senden einer Shared-Secret-Zeichenfolge als Klartext ist nicht sicher. Die Sicherheitsinfrastruktur von eXtreme Scale stellt ein SecureTokenManager-Plug-in bereit, über das der Server den geheimen Schlüssel vor dem Senden sichern kann. Sie können festlegen, wie die Sicherungsoperation implementiert wird. WebSphere eXtreme Scale stellt eine Implementierung bereit, in der die Sicherungsoperation so implementiert ist, dass das Shared Secret verschlüsselt und signiert wird.

JMX-Sicherheit (Java Management Extensions) in einer dynamischen Implementierungstopologie

Die JMX-MBean-Sicherheit wird in allen Versionen von eXtreme Scale unterstützt. Clients der Katalogserver-MBeans und Container-Server-MBeans können authentifiziert werden und auf die MBean-Operationen zugreifen.

Lokale Sicherheit von eXtreme Scale

Die lokale Sicherheit von eXtreme Scale unterscheidet sich vom verteilten eXtreme-Scale-Modell, weil die Anwendung direkt instanziiert wird und eine ObjectGrid-Instanz verwendet. Ihre Anwendung und eXtreme-Scale-Instanzen befinden sich in derselben Java Virtual Machine (JVM). Da es in diesem Modell kein Client/Server-Konzept gibt, wird die Authentifizierung nicht unterstützt. Ihre Anwendungen müssen ihre Authentifizierung selbst verwalten und anschließend das authentifizierte Subject-Objekt an eXtreme Scale übergeben. Der Berechtigungsmechanismus, der für das lokale Programmiermodell von eXtreme Scale verwendet wird, ist jedoch dasselbe wie beim Client/Server-Modell.

Konfiguration und Programmierung

Weitere Informationen zum Konfigurieren und Programmieren der Sicherheit finden Sie in den Abschnitten „Sicherheitsintegration mit externen Providern” auf Seite 528 und Sicherheits-API.

Kapitel 3. Lernprogramme

Sie können Lernprogramme verwenden, um sich mit den Einsatzszenarien für das Produkt, einschließlich Entitätsmanager, Abfragen und Sicherheit vertraut zu machen.

Lernprogramm: Sicherheit von Java SE konfigurieren

Mit dem folgenden Lernprogramm können Sie eine verteilte eXtreme-Scale-Umgebung in einer Java-SE-Umgebung erstellen.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie mit den Grundlagen einer verteilten eXtreme-Scale-Konfiguration vertraut sind.

Informationen zu diesem Vorgang

In diesem Lernprogramm werden der Katalogserver, der Container-Server und der Client alle in einer Java-SE-Umgebung ausgeführt. Jeder Schritt im Lernprogramm baut auf dem vorherigen Schritt auf. Führen Sie jeden der Schritte aus, um eine verteilte eXtreme-Scale-Konfiguration zu sichern und eine einfache Java-SE-Anwendung für den Zugriff auf eine gesicherte eXtreme-Scale-Konfiguration zu entwickeln.

Lernprogramm starten

Vorgehensweise

1. „Lernprogramm zur Java-SE-Sicherheit - Schritt 1" auf Seite 72
   - Nicht gesicherten Katalogserver starten
   - Nicht gesicherten Container-Server starten
   - Client für den Zugriff auf die Daten starten
   - Map-Größe mit dem Dienstprogramm **xscmd** anzeigen
   - Server stoppen
2. „Lernprogramm zur Java-SE-Sicherheit - Schritt 2" auf Seite 75
   - CredentialGenerator verwenden
   - Authentifikator verwenden
   - Sicheren Katalogserver starten
   - Sicheren Container-Server starten
   - Client für den Zugriff auf ein gesichertes ObjectGrid starten
   - Map-Größe mit dem Dienstprogramm **xscmd** anzeigen
   - Sicheren Server stoppen
3. „Lernprogramm zur Java-SE-Sicherheit - Schritt 3" auf Seite 81
   - JAAS-Berechtigungsrichtlinie verwenden
4. „Lernprogramm zur Java-SE-Sicherheit - Schritt 4" auf Seite 85
   - Keystore und Truststore erstellen
   - SSL-Eigenschaften für den Server konfigurieren

- SSL-Eigenschaften für den Client konfigurieren
- Map-Größe mit dem Dienstprogramm **xscmd** anzeigen
- Sicheren Server stoppen

Lernprogramm zur Java-SE-Sicherheit - Schritt 1

In diesem Abschnitt wird ein *einfaches nicht gesichertes Beispiel* beschrieben. In den nachfolgenden Schritten des Lernprogramms werden nach und nach weitere Sicherheitsfeatures hinzugefügt, um die verfügbare integrierte Sicherheit zu erhöhen.

Vorbereitende Schritte

Anmerkung: Alle für diesen Schritt des Lernprogramms erforderlichen Dateien sind im folgenden Abschnitt beschrieben.

Vorgehensweise

Beispiel ausführen
Starten Sie den Katalogservice mit den folgenden Scripts. Weitere Informationen zum Starten des Katalogservice finden Sie unter „Eigenständigen Katalogservice starten" auf Seite 403.

1. Navigieren Sie wie folgt zum Verzeichnis "bin": `cd ObjectGrid-Stammverzeichnis/bin`
2. Starten Sie einen Katalogserver mit dem Namen "catalogServer":
   - UNIX Linux `startOgServer.sh catalogServer`
   - Windows `startOgServer.bat catalogServer`
3. Navigieren Sie wie folgt zum Verzeichnis "bin": `cd ObjectGrid-Stammverzeichnis/bin`
4. Starten Sie anschließend mit dem folgenden Script einen Container-Server mit dem Namen "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
     ```

Beispiel

Weitere Informationen zum Starten von Container-Servern finden Sie unter „Container-Server starten" auf Seite 406.

Nach dem Starten des Katalogservers und des Container-Servers starten Sie den Client wie folgt:

1. Navigieren Sie wieder zum Verzeichnis "bin".
2. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SimpleApp`

Die Datei `secsample.jar` enthält die Klasse "SimpleApp".

Die Ausgabe dieses Programms ist wie folgt:

```
Der Kundenname für ID 0001 ist fName lName
```

Sie können auch das Dienstprogramm **xscmd** verwenden, um die Map-Größen des Grids "accounting" anzuzeigen.

- Navigieren Sie zum Verzeichnis `ObjectGrid-Stammverzeichnis/bin`.
- Map-Größen mit dem Dienstprogramm **xscmd** anzeigen
  - UNIX Linux `xscmd.sh -c showMapSizes -g accounting -ms mapSet1`
  - Windows `xscmd.bat -c showMapSizes -g accounting -ms mapSet1`

Server stoppen

Container-Server

Verwenden Sie den folgenden Befehl, um den Container-Server c0 zu stoppen.

UNIX Linux
```
stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809
```

Windows
```
stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809
```

Die folgende Nachricht wird angezeigt.

```
CWOBJ2512I: ObjectGrid-Server c0 wurde gestoppt.
```

Katalogserver

Sie können einen Katalogserver mit dem folgenden Befehl stoppen.

UNIX Linux
```
stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809
```

Windows
```
stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809
```

Wenn Sie den Katalogserver beenden, wird die folgende Nachricht angezeigt.

```
CWOBJ2512I: ObjectGrid-Server catalogServer wurde gestoppt.
```

Erforderliche Dateien

Die folgende Datei ist die Java-Klasse für die Anwendung "SimpleApp".

SimpleApp.java
```
// Dieses Beispielprogramm wird ohne Wartung (auf "as-is"-Basis)
// bereitgestellt und kann vom Kunden (a) zu Schulungs- und Studienzwecken,
// (b) zum Entwickeln von Anwendungen für ein IBM WebSphere-Produkt zur
// internen Nutzung beim Kunden oder Weitergabe im Rahmen einer solchen
// Anwendung in kundeneigenen Produkten gebührenfrei genutzt, ausgeführt,
// kopiert und geändert werden.
// Lizenziertes Material - Eigentum von 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");

        System.out.println("The customer name for ID 0001 is " + customer);
    }

    /**
     * Get the ObjectGrid
     * @return Eine ObjectGrid-Instanz.
     * @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;

    }

}
```

Die Methode "getObjectGrid" in dieser Klasse ruft ein ObjectGrid ab, und die Methode "run" liest einen Datensatz aus der Map "Customer" und aktualisiert den Wert.

Wenn Sie diesen Beispielcode in einer verteilten Umgebung ausführen möchten, müssen Sie eine ObjectGrid-XML-Deskriptordatei `SimpleApp.xml` und eine XML-Implementierungsdatei `SimpleDP.xml` erstellen. Diese Dateien werden im folgenden Beispiel gezeigt:

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>
```

Die folgende XML-Datei konfiguriert die Implementierungsumgebung.

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>
```

Dies ist eine einfache ObjectGrid-Konfiguration mit einer einzigen ObjectGrid-Instanz mit dem Namen "accounting" und einer einzigen Map mit dem Namen "customer" (im MapSet "mapSet1"). Die Datei `SimpleDP.xml` enthält ein einziges MapSet mit einer Partition und einer erforderlichen Mindestanzahl von 0 Replikaten.

Nächster Schritt des Lernprogramms

Lernprogramm zur Java-SE-Sicherheit - Schritt 2

Aufbauend auf dem vorherigen Schritt, zeigt der folgende Abschnitt, wie die Clientauthentifizierung in einer verteilten eXtreme-Scale-Umgebung implementiert wird.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie die Anweisungen im Abschnitt „Lernprogramm zur Java-SE-Sicherheit - Schritt 1” auf Seite 72 ausgeführt haben.

Informationen zu diesem Vorgang

Wenn die Clientauthentifizierung aktiviert ist, wird ein Client authentifiziert, bevor eine Verbindung zum eXtreme-Scale-Server hergestellt wird. In diesem Abschnitt wird anhand von Beispielcode und Scripts veranschaulicht, wie die Clientauthentifizierung in einer eXtreme-Scale-Serverumgebung durchgeführt werden kann.

Wie jedes andere Authentifizierungsverfahren setzt sich diese minimale Authentifizierung aus den folgenden Schritten zusammen:

1. Der Administrator ändert Konfigurationen, um die Authentifizierung als Voraussetzung festzulegen.
2. Der Client übergibt einen Berechtigungsnachweis an den Server.
3. Der Server authentifiziert den Berechtigungsnachweis anhand der Registry.

Vorgehensweise

1. **Clientberechtigungsnachweis**

   Ein Clientberechtigungsnachweis wird durch eine Schnittstelle des Typs "com.ibm.websphere.objectgrid.security.plugins.Credential" dargestellt. Gültige Clientberechtigungsnachweise sind eine Kombination von Benutzername und Kennwort, ein Kerberos-Ticket, ein Clientzertifikat oder Daten in einem beliebigen Format, auf das sich Client und Server geeinigt haben. Weitere Einzelheiten finden Sie in der Dokumentation der API "Credential".

   Diese Schnittstelle definiert explizit die Methoden "equals(Object)" und "hashCode()". Diese beiden Methoden sind wichtig, weil die authentifizierten Subject-Objekte mit dem Credential-Objekt als Schlüssel auf der Serverseite zwischengespeichert werden.

   eXtreme Scale stellt auch ein Plug-in für die Generierung eines Berechtigungsnachweises bereit. Dieses Plug-in wird durch die Schnittstelle "com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator" dargestellt und wird verwendet, um einen Clientberechtigungsnachweis zu generieren. Dies ist hilfreich,

wenn der Berechtigungsnachweis eine Verfallszeit hat. In diesem Fall wird die Methode "getCredential()" aufgerufen, um einen Berechtigungsnachweis zu erneuern. Weitere Einzelheiten finden Sie in der Dokumentation zur API "CredentialGenerator".

Sie können diese beiden Schnittstellen für die eXtreme-Scale-Clientumgebung implementieren, um Clientberechtigungsnachweise abzurufen.

In diesem Beispielcode werden die folgenden beiden Beispiel-Plug-in-Implementierungen verwendet, die von eXtreme Scale bereitgestellt werden.

```
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredential
```

```
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
```

Weitere Informationen zu diesen Plug-ins finden Sie unter Programmierung der Clientauthentifizierung.

2. **Serverauthentifikator** Nachdem der eXtreme-Scale-Client das Credential-Objekt mit dem CredentialGenerator-Objekt abgerufen hat, wird dieses Client-Credential-Objekt zusammen mit der Clientanforderung an den eXtreme-Scale-Server gesendet. Der eXtreme-Scale-Server authentifiziert das Credential-Objekt, bevor er die Anforderung verarbeitet. Bei erfolgreicher Authentifizierung des Credential-Objekts wird ein Subject-Objekt zurückgegeben, dass diesen Client repräsentiert.

   Dieses Subject-Objekt wird zwischengespeichert und verfällt erst, wenn seine Lebensdauer das festgelegte Sitzungszeitlimit erreicht. Das Zeitlimit für die Anmeldesitzung kann mit der Eigenschaft "loginSessionExpirationTime" in der XML-Datei des Clusters definiert werden. Wenn Sie beispielsweise `loginSessionExpirationTime="300"` definieren, verfällt das Subject-Objekt nach 300 Sekunden.Dieses Subject-Objekt wird anschließend für die Berechtigung der Anforderung verwendet, was später noch erläutert wird.

   Ein eXtreme-Scale-Server verwendet das Authenticator-Plug-in, um das Credential-Objekt zu authentifizieren. Weitere Einzelheiten finden Sie in der Dokumentation zur API "Authenticator".

   In diesem Beispiel wird eine integrierte eXtreme-Scale-Implementierung verwendet, die Implementierung "KeyStoreLoginAuthenticator", die für Test- und Beispielzwecke bestimmt ist (ein Keystore ist eine einfache Benutzerregistry und sollte nicht für eine Produktionsumgebung verwendet werden). Weitere Informationen finden Sie im Abschnitt zum Authentifikator-Plug-in unter Programmierung der Clientauthentifizierung.

   Dieser KeyStoreLoginAuthenticator verwendet ein KeyStoreLoginModule, um den Benutzer über das JAAS-Anmeldemodul "KeyStoreLogin" für den Keystore zu authentifizieren. Der Keystore kann als Option für die Klasse "KeyStoreLoginModule" konfiguriert werden. Im folgenden Beispiel sehen Sie den keyStoreLogin-Alias, der in der JAAS-Konfigurationsdatei "og_jaas.config" konfiguriert ist:

   ```
   KeyStoreLogin{
   com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
        keyStoreFile="../security/sampleKS.jks" debug = true;
   };
   ```

   Die folgenden Befehle erstellen einen Keystore "sampleKS.jks" im Verzeichnis "%OBJECTGRID_HOME%/security" mit dem Kennwort "sampleKS1". Außerdem werden drei Benutzerzertifikate mit eigenen Kennwörtern erstellt, die den Benutzer "administrator", den Benutzer "manager" und den Benutzer "cashier" darstellen.

   a. Navigieren Sie zum Stammverzeichnis von eXtreme Scale:

      ```
      cd objectgridRoot
      ```

   b. Erstellen Sie ein Verzeichnis mit dem Namen "security":

      ```
      mkdir security
      ```

c. Navigieren Sie zum neu erstellten Verzeichnis "security":

```
cd security
```

d. Verwenden Sie keytool (im Verzeichnis javaHOME/bin), um einen Benutzer "administrator" mit dem Kennwort "administrator1" im Keystore "sampleKS-.jks" zu erstellen:

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias administrator -keypass administrator1
-dname CN=administrator,O=acme,OU=OGSample -validity 10000
```

e. Verwenden Sie keytool (im Verzeichnis javaHOME/bin), um einen Benutzer "manager" mit dem Kennwort "manager1" im Keystore "sampleKS.jks" zu erstellen:

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias manager -keypass manager1
-dname CN=manager,O=acme,OU=OGSample -validity 10000
```

f. Verwenden Sie keytool (im Verzeichnis javaHOME/bin), um einen Benutzer "cashier" mit dem Kennwort "cashier1" im Keystore "sampleKS.jks" zu erstellen:

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias cashier -keypass cashier1 -dname CN=cashier,O=acme,OU=OGSample
-validity 10000
```

Die Clientsicherheitskonfiguration wird in der Clienteigenschaftendatei konfiguriert. Verwenden Sie die folgenden Befehle, um eine Kopie im Verzeichnis %OBJECTGRID_HOME%/security zu erstellen:

a. Wechseln Sie in das Verzeichnis "security":

```
cd objectgridRoot/security
```

b. Kopieren Sie die Datei "sampleClient.properties" in die Datei "client.properties":

```
cp ../properties/sampleClient.properties client.properties
```

Die folgenden Eigenschaften sind in der Datei "client.properties" im Verzeichnis "security" hervorgehoben:

a. **securityEnabled:** Wenn Sie "securityEnabled" auf "true" (Standardwert) setzen, wird die Clientsicherheit aktiviert, die die Authentifizierung umfasst.

b. **credentialAuthentication:** Setzen Sie "credentialAuthentication" auf "Supported" (Standardwert), d. h., der Client unterstützt die Authentifizierung von Berechtigungsnachweisen.

c. **transportType:** Setzen Sie "transportType" auf "TCP/IP", d. h., es wird kein SSL verwendet.

d. **singleSignOnEnabled:** Setzen Sie diese Eigenschaft auf "false" (Standardwert). Single Sign-on (SSO) ist nicht verfügbar.

3. **Serversicherheitskonfiguration**

Die Serversicherheitskonfiguration wird in der XML-Sicherheitsdeskriptordatei und in der Servereigenschaftendatei angegeben.Die XML-Sicherheitsdeskriptordatei beschreibt die Sicherheitseigenschaften, die für alle Server (einschließlich Katalogservern und Container-Servern) gelten. Ein Beispiel für eine solche Eigenschaft ist die Authentifikatorkonfiguration, die die Benutzerregistry und das Authentifizierungsverfahren darstellt.

Im Folgenden sehen Sie die Datei security.xml, die für diesen Beispielcode verwendet wird:

```
<?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>
```

a. **securityEnabled:** Wenn diese Einstellung den Wert "true" hat, wird die Serversichrheit, einschließlich Authentifizierung aktiviert.

b. **loginSessionExpirationTime:** Setzen Sie diese Eigenschaft auf 300 (Standardwert).

c. **authenticator:** Fügen Sie die Authentifikatorklasse "KeyStoreLoginAuthenticator" der XML-Custerdatei wie folgt hinzu:

```
<authenticator className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
        </authenticator>
```

d. **credentialAuthentication:** Setzen Sie das Attribut "credentialAuthentication" auf "Required", damit der Server eine Authentifizierung erfordert.

Eine ausführlichere Erläuterung der Datei `security.xml` finden Sie unter XML-Sicherheitsdeskriptordatei.

Kopieren Sie die Servereigenschaftendatei in das Verzeichnis "security". Dieses Mal müssen Sie keine Änderungen in der Datei vornehmen.

a. Navigieren Sie zum Verzeichnis "security":

```
cd objectgridRoot/security
```

b. Kopieren Sie die Beispiel-ObjectGrid-Datei `sampleServer.properties` aus dem Verzeichnis "properties" in die neue Datei `server.properties`:

```
cp ../properties/containerServer.properties server.properties
```

Nehmen Sie die folgenden Änderungen in der Datei `server.properties` vor:

a. **securityEnabled:** Setzen Sie das Attribut **`securityEnabled`** auf "true".

b. **transportType:** Setzen Sie das Attribut **`transportType`** auf "TCP/IP", d. h., es wird kein SSL verwendet.

c. **secureTokenManagerType:** Setzen Sie das Attribut **`secureTokenManagerType`** auf "none", damit der Manager für sichere Token nicht konfiguriert wird.

4. **Sicherer Client** Verbinden Sie die Clientanwendung, wie im folgenden Beispiel gezeigt, sicher mit dem Server:

```
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 Eine ObjectGrid-Instanz.
     * @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");

        // Creates a ClientSecurityConfiguration object using the specified file
        ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
                .getClientSecurityConfiguration(args[0]);
```

```
        // Creates a CredentialGenerator using the passed-in user and password.
        CredentialGenerator credGen = new UserPasswordCredentialGenerator(args[1], args[2]);
        clientSC.setCredentialGenerator(credGen);

        // Create an ObjectGrid by connecting to the catalog server
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", clientSC, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;

    }

}
```

Es gibt drei Punkte, in der sich die sichere Anwendung von der nicht gesicherten Anwendung unterscheidet:

a. Es wurde ein ClientSecurityConfiguration-Objekt durch Übergabe der konfigurierten Datei `client.properties` erstellt.
b. Es wurde ein UserPasswordCredentialGenerator-Objekt durch Verwendung der übergebenen Benutzer-ID/Kennwort-Kombination erstellt.
c. Es wurde eine Verbindung zum Katalogserver hergestellt, um ein ObjectGrid vom ClientClusterContext durch Übergabe eines ClientSecurityConfiguration-Objekts abzurufen.

5. **Führen Sie die Anwendung aus.**

   Zum Ausführen der Anwendung starten Sie den Katalogserver. Setzen Sie die Befehlszeilenoptionen "-clusterFile" und -serverProps" zur Übergabe der Sicherheitseigenschaften ab:

   a. Navigieren Sie wie folgt zum Verzeichnis "bin":

      ```
      cd ObjectGrid-Stammverzeichnis/bin
      ```

   b. Starten Sie den Katalogserver:

      - 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"
        ```

   Starten Sie anschließend einen sicheren Container-Server mit dem folgenden Script:

   a. Navigieren Sie erneut zum Verzeichnis "bin":

      ```
      cd ObjectGrid-Stammverzeichnis/bin
      ```

   b. Starten Sie einen sicheren Container-Server:

      - Linux UNIX

        ```
        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"
        ```

   Die Servereigenschaftendatei wird mit der Option "-serverProps" übergeben.

Nachdem Sie den Server gestartet haben, starten Sie den Client mit dem folgenden Befehl:

a. `cd ObjectGrid-Stammverzeichnis/bin`

b.
```
java -classpath ../lib/objectgrid.jar;../applib/secsample.jar
  com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties manager manager1
```

Linux Verwenden Sie einen Doppelpunkt (:) als Klassenpfadtrennzeichen anstelle des im vorherigen Beispiel verwendeten Semikolons (;).

Die Datei `secsample.jar` enthält die Klasse "SimpleApp".

SecureSimpleApp verwendet drei Parameter, die in der folgenden Liste aufgeführt sind:

a. Die Datei `../security/client.properties` ist die Datei mit den Clientsicherheitseigenschaften.

b. `manager` ist die Benutzer-ID.

c. `manager1` ist das Kennwort.

Nachdem Sie die Klasse angegeben haben, werden die folgenden Ergebnisse ausgegeben:

Der Kundenname für `ID 0001` ist `fName lName`.

Sie können auch das Dienstprogramm **xscmd** verwenden, um die Map-Größen des Grids "accounting" anzuzeigen.

- Navigieren Sie zum Verzeichnis `ObjectGrid-Stammverzeichnis/bin`.
- Verwenden Sie den Befehl **xscmd** mit der Option "-c showMapSizes" wie folgt.
  - UNIX Linux `xscmd.sh -c showMapSizes -g accounting -m mapSet1 -username manager -password manager1`
  - Windows `xscmd.bat -c showMapSizes -g accounting -m mapSet1 -username manager -password manager1`

Jetzt können Sie den Befehl **stopOgServer** verwenden, um den Container-Server- oder Katalogserverprozess zu stoppen. Sie müssen jedoch eine Sicherheitskonfigurationsdatei angeben. In der Beispielclienteigenschaftendatei werden die folgenden beiden Eigenschaften für die Generierung eines Benutzer-ID/Kennwort-Berechtigungsnachweises (manager/manager1) definiert.

```
credentialGeneratorClass=com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
credentialGeneratorProps=manager manager1
```

Stoppen Sie den Container "c0" mit dem folgenden Befehl:

- UNIX Linux `stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- Windows `stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Wenn Sie die Option **-clientSecurityFile** nicht angeben, wird eine Ausnahme mit der folgenden Nachricht angezeigt:

```
>> SERVER (id=39132c79, host=9.10.86.47) TRACE START:
>> org.omg.CORBA.NO_PERMISSION: Server requires credential
authentication but there is no security context from the client. This
usually happens when the client does not pass a credential the server.
vmcid: 0x0
minor code: 0
completed: No
```

Sie können den Katalogserver auch mit dem folgenden Befehl beenden. Wenn Sie jedoch den nächsten Schritt des Lernprogramms ausführen möchten, können Sie den Katalogserver aktiviert lassen.

- UNIX Linux `stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- Windows `stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Wenn Sie den Katalogserver beenden, wird die folgende Nachricht angezeigt.

```
CWOBJ2512I: ObjectGrid-Server catalogServer wurde gestoppt
```

Sie haben Ihr System jetzt erfolgreich teilweise gesichert, indem Sie die Authentifizierung aktiviert haben. Die haben den Server für die Integration der Benutzerregistry konfiguriert, den Client für die Bereitstellung von Clientberechtigungsnachweisen konfiguriert und die Clienteigenschaftendatei und die XML-Clusterdatei für die Aktivierung der Authentifizierung geändert.

Wenn Sie ein ungültiges Kennwort angeben, wird eine Ausnahme angezeigt, in der Sie darauf hingewiesen werden, dass der Benutzername oder das Kennwort nicht korrekt ist.

Weitere Einzelheiten zur Clientauthentifizierung finden Sie unter „Anwendungsclientauthentifizierung” auf Seite 515.

Nächster Schritt des Lernprogramms

Lernprogramm zur Java-SE-Sicherheit - Schritt 3

Nach der Authentifizierung eines Clients (wie im vorherigen Schritt) können Sie über die Berechtigungsmechanismen von eXtreme Scale Sicherheitsberechtigungen erteilen.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie die Anweisungen im Abschnitt „Lernprogramm zur Java-SE-Sicherheit - Schritt 2” auf Seite 75 ausgeführt haben, bevor Sie mit dieser Task fortfahren.

Informationen zu diesem Vorgang

Im vorherigen Schritt dieses Lernprogramms wurde veranschaulicht, wie die Authentifizierung in einem eXtreme-Scale-Grid aktiviert wird. Aufgrund der Aktivierung kann kein nicht authentifizierter Client eine Verbindung zu Ihrem Server mehr herstellen und Anforderungen an Ihr System übergeben. Jeder authentifizierte Client hat jedoch dieselben Berechtigungen oder Privilegien beim Server, z. B. Lesen, Schreiben oder Löschen von Daten, die in ObjectGrid-Maps gespeichert sind. Clients können auch jeden Typ von Abfrage absetzen. In diesem Abschnitt wird gezeigt, wie Sie über eXtreme-Scale-Berechtigungen verschiedenen authentifizierten Benutzern unterschiedliche Privilegien erteilen.

Ähnlich wie viele andere Systeme verwendet eXtreme Scale einen rechtebasierten Berechtigungsmechanismus. WebSphere eXtreme Scale hat verschiedene Berechtigungskategorien, die von verschiedenen Berechtigungsklassen dargestellt werden. Hier wird die Berechtigungsklasse "MapPermission" verwendet. Informationen zur vollständigen Berechtigungskategorie finden Sie unter Programmierung der Clientberechtigung.

In WebSphere eXtreme Scale stellt die Klasse "com.ibm.websphere.objectgrid.security.MapPermission" Berechtigungen für die eXtreme-Scale-Ressourcen dar, insbesondere die Methoden der Schnittstellen "ObjectMap" und "JavaMap". WebSphere eXtreme Scale definiert die folgenden Berechtigungszeichenfolgen für den Zugriff auf die Methoden der Schnittstellen "ObjectMap" und "JavaMap":

- read: Erteilt die Berechtigung zum Lesen der Daten aus der Map.
- write: Erteilt die Berechtigung zum Aktualisieren der Daten in der Map.
- insert: Erteilt die Berechtigung zum Einfügen der Daten in die Map.
- remove: Erteilt die Berechtigung zum Entfernen der Daten aus der Map.
- invalidate: Erteilt die Berechtigung zum Ungültigmachen der Daten in der Map.
- all: Erteilt alle zuvor beschriebenen Berechtigungen: read, write, insert, remote und invalidate.

Die Berechtigung findet statt, wenn ein Client eine Methode von ObjectMap oder JavaMap aufruft. Der Laufzeitumgebung von eXtreme Scale prüft verschiedene Map-Berechtigungen für verschiedene Methoden. Wenn dem Client die erforderlichen Berechtigungen nicht erteilt wurden, wird eine Ausnahme des Typs "AccessControlException" ausgegeben.

Dieses Lernprogramm veranschaulicht, wie über JAAS-Berechtigung verschiedenen Benutzern Berechtigungen für Map-Zugriffe erteilt werden.

Vorgehensweise

1. **Aktivieren Sie die eXtreme-Scale-Berechtigung**. Zum Aktivieren der Berechtigung im ObjectGrid müssen Sie das Attribut "securityEnabled" für das gewünschte ObjectGrid in der XML-Datei auf `true` setzen. Die Sicherheit im ObjectGrid zu aktivieren, bedeutet, dass Sie die Berechtigung aktivieren. Verwenden Sie die folgenden Befehle, um eine neue ObjectGrid-XML-Dadtei mit aktivierter Sicherheit zu erstellen.
   a. Navigieren Sie zum Verzeichnis `xml`.
      ```
      cd objectgridRoot/xml
      ```
   b. Kopieren Sie die Datei `SimpleApp.xml` in die Datei `SecureSimpleApp.xml`.
      ```
      cp SimpleApp.xml SecureSimpleApp.xml
      ```
   c. Öffnen Sie die Datei `SecureSimpleApp.xml`, und fügen Sie, wie in der folgenden XML gezeigt, `securityEnabled="true"` auf ObjectGrid-Ebene hinzu.
      ```
      <?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. **Definieren Sie die Berechtigungsrichtlinie.** Im vorherigen Schritt haben Sie zur Vorbereitung der Clientauthentifizierung drei Benutzer im Keystore erstellt: cashier, manager und administrator. In diesem Beispiel wird gezeigt, dass der Benutzer "cashier" nur Leseberechtigungen für alle Maps und der Benutzer "manager" alle Berechtigungen besitzt. In diesem Beispiel wird die JAAS-Berechtigung verwendet. Die JAAS-Berechtigung verwendet eine Berechtigungsrichtliniendatei, um Principals Berechtigungen zu erteilen. Die folgende datei wird im Verzeichnis "security" definiert:
   ```
   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";
};
```

Anmerkung:

- `codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"` ist ein speziell reservierter URL für ObjectGrid. Alle ObjectGrid-Berechtigungen, die Principals erteilt werden, müssen diese spezielle Codebasis verwenden.
- Die erste grant-Anweisung erteilt dem Principal `"CN=cashier,O=acme,OU=OGSample"` die Map-Berechtigung "read", so dass der Benutzer "cashier" ausschließlich Leseberechtigung für alle Maps im ObjectGrid "accounting" hat.
- Die zweite grant-Anweisung erteilt dem Principal `"CN=manager,O=acme,OU=OGSample"` die Map-Berechtigung "all", so dass der Benutzer "manager" alle Berechtigungen für die Maps im ObjectGrid "accounting" hat.

Jetzt können Sie einen Server mit einer Berechtigungsrichtlinie starten. Die JAAS-Berechtigungsrichtliniendatei kann mit der Standardeigenschaft "-D" definiert werden: -Djava.security.auth.policy=../security/ogAuth.policy

3. **Führen Sie die Anwendung aus.**

   Nachdem Sie die zuvor beschriebenen Dateien erstellt haben, können Sie die Anwendung ausführen.

   Verwenden Sie die folgenden Befehle, um den Katalogserver zu starten. Weitere Informationen zum Starten des Katalogservice finden Sie unter „Eigenständigen Katalogservice starten” auf Seite 403.

   a. Navigieren Sie wie folgt zum Verzeichnis "bin": `cd ObjectGrid-Stammverzeichnis/bin`

   b. Starten Sie den Katalogserver:

      - 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"
        ```

      Die Dateien `security.xml` und `server.properties` wurden im vorherigen Schritt dieses Lernprogramms erstellt.

      T

   c. Anschließend können Sie einen sicheren Container-Server mit dem folgenden Script starten. Führen Sie das Script im Verzeichnis "bin" aus:

      - 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.auth.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.auth.policy="../security/og_auth.policy"
```

Beachten Sie die folgenden Unterschiede zum vorherigen Startbefehl für den Katalogserver:

- Verwenden Sie die Datei `SecureSimpleApp.xml` anstelle der Datei `SimpleApp.xml`.
- Es wird eine weitere Eigenschaft `-Djava.security.auth.policy` hinzugefügt, um die JAAS-Berechtigungsrichtliniendatei für den Container-Serverprozess zu definieren.

Verwenden Sie denselben Befehl wie im vorherigen Schritt des Lernprogramms.

a. Navigieren Sie zum Verzeichnis "bin".

b. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp ../security/client.properties manager manager1`

   Da der Benutzer "manager" alle Berechtigungen für die Maps im ObjectGrid "accounting" hat, wird die Anwendung ordnungsgemäß ausgeführt.

   Jetzt verwenden Sie an Stelle des Benutzers "manager" den Benutzer "cashier", um die Clientanwendung zu starten.

c. Navigieren Sie zum Verzeichnis "bin".

d. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp ../security/client.properties cashier cashier1`

Die folgende Ausnahme wird ausgegeben:

```
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
```

Diese Ausnahme tritt ein, weil der Benutzer "cashier" keine Schreibberechtigung hat und deshalb die Map "Customer" nicht aktualisieren kann.

Jetzt unterstützt Ihr System Berechtigung. Sie können Berechtigungsrichtlinien definieren, um unterschiedlichen Benutzern unterschiedliche Berechtigungen zu erteilen. Weitere Informationen zur Berechtigung finden Sie unter „Anwendungsclientberechtigung” auf Seite 517.

Nächste Schritte

Führen Sie den nächsten Schritt des Lernprogramms aus (siehe „Lernprogramm zur Java-SE-Sicherheit - Schritt 4”).

Lernprogramm zur Java-SE-Sicherheit - Schritt 4

Im folgenden Schritt wird erläutert, wie Sie eine Sicherheitsschicht für die Kommunikation zwischen den Endpunkten Ihrer Umgebung aktivieren.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie die Anweisungen im Abschnitt „Lernprogramm zur Java-SE-Sicherheit - Schritt 3” auf Seite 81 ausgeführt haben, bevor Sie mit dieser Task fortfahren.

Informationen zu diesem Vorgang

Die eXtreme-Scale-Topologie unterstützt Transport Layer Security/Secure Sockets Layer (TLS/SSL) für die sichere Kommunikation zwischen ObjectGrid-Endpunkten (Client, Container-Server und Katalogserver). Dieser Schritt des Lernprogramms zum Aktivieren der Transportsicherheit baut auf den vorherigen Schritten auf.

Vorgehensweise

1. **Erstellen Sie TLS/SSL-Schlüssel und Keystores.**

   Zum Akivieren der Transportsicherheit müssen Sie einen Keystore und einen Truststore erstellen. In dieser Übung wird nur ein einziges Keystore/Truststore-Paar erstellt. Diese Speicher werden für ObjectGrid-Clients, Container-Server und Katalogserver verwendet und mit keytool von JDK erstellt.

   - *Privaten Schlüssel im Keystore erstellen*

     ```
     keytool -genkey -alias ogsample -keystore key.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
     ```

     Mit diesem Befehl wird ein Keystore "key.jks" erstellt, in dem ein Schlüssel "ogsample" gespeichert ist. Dieser Keystore "key.jks" wird als SSL-Keystore verwendet.

   - *Öffentliches Zertifikat exportieren*

```
keytool -export -alias ogsample -keystore key.jks -file temp.key
-storepass ogpass
```

Mit diesem Befehl wird das öffentliche Zertifikat des Schlüssels "ogsample" extrahiert und in der Datei "temp.key" gespeichert.

- *Öffentliches Zertifikat des Clients in den Truststore importieren*

```
keytool -import -noprompt -alias ogsamplepublic -keystore trust.jks
-file temp.key -storepass ogpass
```

Mit diesem Befehl wird das öffentliche Zertifikat dem Keystore "trust.jks" hinzugefügt. Dieser Keystore "trust.jks" wird als SSL-Truststore verwendet.

2. **ObjectGrid-Eigenschaftendateien konfigurieren**

   In diesem Schritt müssen Sie die ObjectGrid-Eigenschaftendateien für die Aktivierung der Transportsicherheit konfigurieren.

   Kopieren Sie zuerst die Dateien "key.jks" und "trust.jks" in das Verzeichnis "Objectgrid-Stammverzeichnis/security".

   Definieren Sie die folgenden Eigenschaften in den Dateien `client.properties` und `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: Die Eigenschaft "transportType" wird auf "SSL-Required" gesetzt, d. h., für den Transport ist SSL erforderlich. Deshalb muss für alle ObjectGrid-Endpunkte (Clients, Katalogserver und Container-Server) SSL konfiguriert werden, und die gesamte Transportkommunikation wird verschlüsselt.

   Die anderen Eigenschaften werden zum Definieren der SSL-Konfigurationen verwendet. Eine ausführlicher Erläuterung finden Sie unter „Transport Layer Security und Secure Sockets Layer” auf Seite 523. Stellen Sie sicher, dass Sie die Anweisungen in diesem Abschnitt befolgen, um Ihre Datei `orb.properties` zu aktualisieren.

   Stellen Sie sicher, dass Sie den folgenden Anweisungen folgen, um Ihre Datei `orb.properties` zu aktualisieren.

   In der Datei `server.properties` müssen Sie eine zusätzliche Eigenschaft "clientAuthentication" hinzufügen und diese auf "false" setzen. Auf der Serverseite müssen Sie den Client nicht anerkennen.

```
clientAuthentication=false
```

3. **Anwendung ausführen**

   Die Befehle sind dieselben wie im Abschnitt „Lernprogramm zur Java-SE-Sicherheit - Schritt 3” auf Seite 81.

   Verwenden Sie die folgenden Befehle, um einen Katalogserver zu starten:

   a. Navigieren Sie wie folgt zum Verzeichnis "bin": `cd ObjectGrid-Stammverzeichnis/bin`

   b. Starten Sie den Katalogserver:

      - 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"
```

Die Dateien `security.xml` und `server.properties` wurden im Abschnitt „Lernprogramm zur Java-SE-Sicherheit - Schritt 2" auf Seite 75 erstellt.

Verwenden Sie die Option **-JMXServicePort**, um den JMX-Port für den Server explizit anzugeben. Diese Option ist für die Verwendung des Dienstprogramms **xscmd** erforderlich.

Führen Sie einen sicheren ObjectGrid-Container-Server aus:

c. Navigieren Sie erneut zum Verzeichnis "bin": `cd ObjectGrid-Stammverzeichnis/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.auth.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.auth.policy="../security/og_auth.policy"
```

Beachten Sie die folgenden Unterschiede zum vorherigen Startbefehl für den Katalogserver:

- Verwenden Sie die Datei `SecureSimpleApp.xml` anstelle der Datei `SimpleApp.xml`.
- Es wird eine weitere Eigenschaft "-Djava.security.auth.policy" hinzugefügt, um die JAAS-Berechtigungsrichtliniendatei für den Container-Serverprozess zu definieren.

Führen Sie den folgenden Befehl für die Clientauthentifizierung aus:

a. `cd ObjectGrid-Stammverzeichnis/bin`

b.

```
javaHome/java -classpath ../lib/objectgrid.jar;../applib/secsample.jar
com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties manager manager1
```

Da der Benutzer "manager" die Berechtigung für alle Maps im ObjectGrid "accounting" hat, wird die Anwendung erfolgreich ausgeführt.

Sie können das Dienstprogramm **xscmd** verwenden, um die Map-Größen des Grids "accounting" anzuzeigen.

- Navigieren Sie zum Verzeichnis `ObjectGrid-Stammverzeichnis/bin`.
- Verwenden Sie den Befehl **xscmd**, um die Map-Größen anzuzeigen:
  - UNIX Linux

```
xscmd.sh -c showMapsizes -g accounting -m mapSet1  -jp 11001 -ssl
-ts ..\security\trust.jks -tsp ogpass -tst jks
-user manager -pwd manager1
```

  - Windows

```
xscmd.bat -c showMapsizes -g accounting -m mapSet1  -jp 11001 -ssl
-ts ..\security\trust.jks -tsp ogpass -tst jks
-user manager -pwd manager1
```

Hier geben Sie den JMX-Port des Katalogservice mit "-p 11001" an.

Sie sehen die folgende Ausgabe.

```
This administrative utility is provided as a sample only and is not to
be considered a fully supported component of the WebSphere eXtreme Scale product.
Connecting to Catalog service at localhost:1099
********** Displaying Results for Grid - accounting, MapSet - mapSet1 **********
*** Listing Maps for c0 ***
Map Name: customer Partition #: 0 Map Size: 1 Shard Type: Primary
Server Total: 1
Total Domain Count: 1
```

Anwendung mit einem ungültigen Keystore ausführen

Wenn Ihr Truststore das öffentliche Zertifikat zum privaten Schlüssel im Keystore nicht enthält, wird eine Ausnahme angezeigt, in der erläutert wird, dass der Schlüssel nicht anerkannt werden kann.

Zur Demonstration erstellen Sie einen weiteren Keystore mit dem Namen "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
```

Ändern Sie anschließend die Datei "server.properties" so, dass "keyStore" auf diesen neuen Keystore "key2.jks" zeigt:

```
keyStore=../security/key2.jks
```

Führen Sie den folgenden Befehl aus, um den Katalogserver zu starten:

a. Navigieren Sie wie folgt zum Verzeichnis "bin": `cd ObjectGrid-Stammverzeichnis/bin`

b. Starten Sie den Katalogserver:

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.auth.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.auth.policy="../security/og_auth.policy"
```

Sie sehen die folgende Ausgabe:

```
Caused by: com.ibm.websphere.objectgrid.ObjectGridRPCException:
    com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
        SSL connection fails and plain socket cannot be used.
```

Ändern Sie abschließend die Datei `server.properties` wieder so zurück, dass sie die Datei `key.jks` verwendet.

Lernprogramm: Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server integrieren

Dieses Lernprogramm veranschaulicht, wie eine Serverimplementierung von WebSphere eXtreme Scale in einer Umgebung von WebSphere Application Server gesichert wird.

Lernziele

Im Folgenden sind die Lernziele für dieses Lernprogramm aufgeführt:

- WebSphere eXtreme Scale für die Verwendung der Authentifizierungs-Plug-ins von WebSphere Application Server konfigurieren
- Transportsicherheit von WebSphere eXtreme Scale für die Verwendung der CSIv2-Konfiguration von WebSphere Application Server konfigurieren

- JAAS-Berechtigung (Java Authentication and Authorization Service) in WebSphere Application Server verwenden
- Angepasstes Anmeldemodul für gruppenbasierte JAAS-Berechtigung verwenden
- Dienstprogramm **xscmd** von WebSphere eXtreme Scale in einer Umgebung von WebSphere Application Server verwenden

Erforderliche Zeit

Das Durcharbeiten dieses Lernprogramms dauert insgesamt ungefähr 4 Stunden.

Einführung: Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server über Authentifizierungs-Plug-ins von WebSphere Application Server integrieren

In diesem Lernprogramm integrieren Sie die Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server. Zuerst konfigurieren Sie die Authentifizierung mit einer einfachen Webanwendung, die authentifizierte Benutzerberechtigungsnachweise aus dem aktuellen Thread verwendet, um die Verbindung zum ObjectGrid herzustellen. Anschließend überprüfen Sie die Verschlüsselung der Daten, die zwischen dem Client und dem Server mit Transportsicherheit übertragen werden. Wenn Sie Benutzern variierende Berechtigungsstufen erteilen möchten, können Sie Java Authentication and Authorization Service (JAAS) konfigurieren. Nach Abschluss der Konfiguration können Sie das Dienstprogramm **xscmd** verwenden, um Ihre Datengrids und Maps zu überwachen.

In diesem Lernprogramm wird davon ausgegangen, dass alle Clients, Container-Server und Katalogserver von WebSphere eXtreme Scale in der Umgebung von WebSphere Application Server implementiert sind.

Lernziele

Im Folgenden sind die Lernziele für dieses Lernprogramm aufgeführt:

- WebSphere eXtreme Scale für die Verwendung der Authentifizierungs-Plug-ins von WebSphere Application Server konfigurieren
- Transportsicherheit von WebSphere eXtreme Scale für die Verwendung der CSIv2-Konfiguration von WebSphere Application Server konfigurieren
- JAAS-Berechtigung (Java Authentication and Authorization Service) in WebSphere Application Server verwenden
- Angepasstes Anmeldemodul für gruppenbasierte JAAS-Berechtigung verwenden
- Dienstprogramm **xscmd** von WebSphere eXtreme Scale in einer Umgebung von WebSphere Application Server verwenden

Erforderliche Zeit

Das Durcharbeiten dieses Lernprogramms dauert insgesamt ungefähr 4 Stunden.

Kenntnisstufe

Fortgeschrittener Anfänger.

Zielgruppe

Entwickler und Administratoren, die an der Sicherheitsintegration zwischen WebSphere eXtreme Scale und WebSphere Application Server interessiert sind.

Systemvoraussetzungen und Topologie

- WebSphere Application Server Version 6.1 oder Version 7.0.0.11 oder höher
- Aktualisieren Sie die Java Runtime, um den folgenden Fix anzuwenden: IZ79819: IBMJDK FAILS TO READ PRINCIPAL STATEMENT WITH WHITESPACE FROM SECURITY FILE

In diesem Lernprogramm werden vier Anwendungsserver und ein Deployment Manager von WebSphere Application Server verwendet, um das Beispiel zu veranschaulichen.

Voraussetzungen

Ein grundlegendes Verständnis der folgenden Punkte ist hilfreich, bevor Sie mit diesem Lernprogramm beginnen:

- Programmiermodell von WebSphere eXtreme Scale
- Grundlegende Sicherheitskonzepte von WebSphere eXtreme Scale
- Grundlegende Sicherheitskonzepte von WebSphere Application Server

Hintergrundinformationen zur Sicherheitsintegration von WebSphere eXtreme Scale und WebSphere Application Server finden Sie unter „Integration der Sicherheit mit WebSphere Application Server" auf Seite 533.

Modul 1: WebSphere Application Server vorbereiten

Bevor Sie das Lernprogramm für die Integration mit WebSphere eXtreme Scale starten, müssen Sie eine Basissicherheitskonfiguration in WebSphere Application Server erstellen.

Lernziele

Anhand der Übungen in diesem Modul lernen Sie Folgendes:

- Konfigurieren Sie die Sicherheit von WebSphere Application Server für die Verwendung eines internen dateibasierten eingebundenen Repositorys als Benutzer-Account-Registry.
- Erstellen Sie Benutzergruppen und Benutzer.
- Erstellen Sie Cluster für die Awnendung und die Server von WebSphere eXtreme Scale.

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Lerneinheit 1.1: Topologie verstehen und Lernprogrammdateien abrufen

Um die Umgebung für das Lernprogramm vorzubereiten, müssen Sie die Sicherheit von WebSphere Application Server konfigurieren. Sie konfigurieren die Verwaltungs- und Anwendungssicherheit mit internen dateibasierten eingebundenen Repositorys als Benutzer-Account-Registry.

Diese Lerneinheit führt Sie durch die Beispieltopologie und Anwendungen, die im Lernprogramm verwendet werden. Um mit der Ausführung des Lernprogramms zu beginnen, müssen Sie die Anwendungen herunterladen und die Konfigurationsdateien an den richtigen Positionen für Ihre Umgebung ablegen. Sie können die Beispielanwendung aus dem Wiki zu WebSphere eXtreme Scale herunterladen.

Beispieltopologie mit WebSphere Application Server: Dieses Lernprogramm führt Sie durch die Erstellung von vier Anwendungsservern von WebSphere Application Server, um die Verwendung von Beispielanwendungen mit aktivierter Sicherheit zu demonstrieren. Diese Anwendungsserver werden in zwei Clustern mit jeweils zwei Servern gruppiert:

- **Cluster appCluster**: Hostet die Beispielunternehmensanwendung "EmployeeManagement". Dieser Cluster hat zwei Anwendungsserver: s1 und s2.
- **Cluster xsCluster**: Hostet die Container-Server von eXtreme Scale. Dieser Cluster hat zwei Anwendungsserver: xs1 und xs2.

In dieser Implementierungstopologie sind die Anwendungsserver "s1" und "s2" die Client-Server, die auf Daten zugreifen, die im Datengrid gespeichert werden. Die Server "xs1" und "xs2" sind die Container-Server, die das Datengrid hosten.

Der Katalogserver wird standardmäßig im Deployment-Manager-Prozess implementiert. In diesem Lernprogramm wird das Standardverhalten verwendet. Das Hosten des Katalogservers im Deployment Manager wird für eine Produktionsumgebung nicht empfohlen. In einer Produktionsumgebung sollten Sie eine Katalogservicedomäne erstellen, um festzulegen, wo Katalogserver gestartet werden. Weitere Informationen finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen” auf Seite 264.

Alternative Konfiguration: Sie können alle Anwendungsserver in einem einzigen Cluster, z. B. im Cluster "appCluster", hosten. Bei dieser Konfiguration sind alle Server im Cluster sowohl Clients als auch Container-Server. In diesem Lernprogramm werden zwei Cluster verwendet, um eine Unterscheidung zwischen den Anwendungsservern zu treffen, die die Clients und Container-Server hosten.

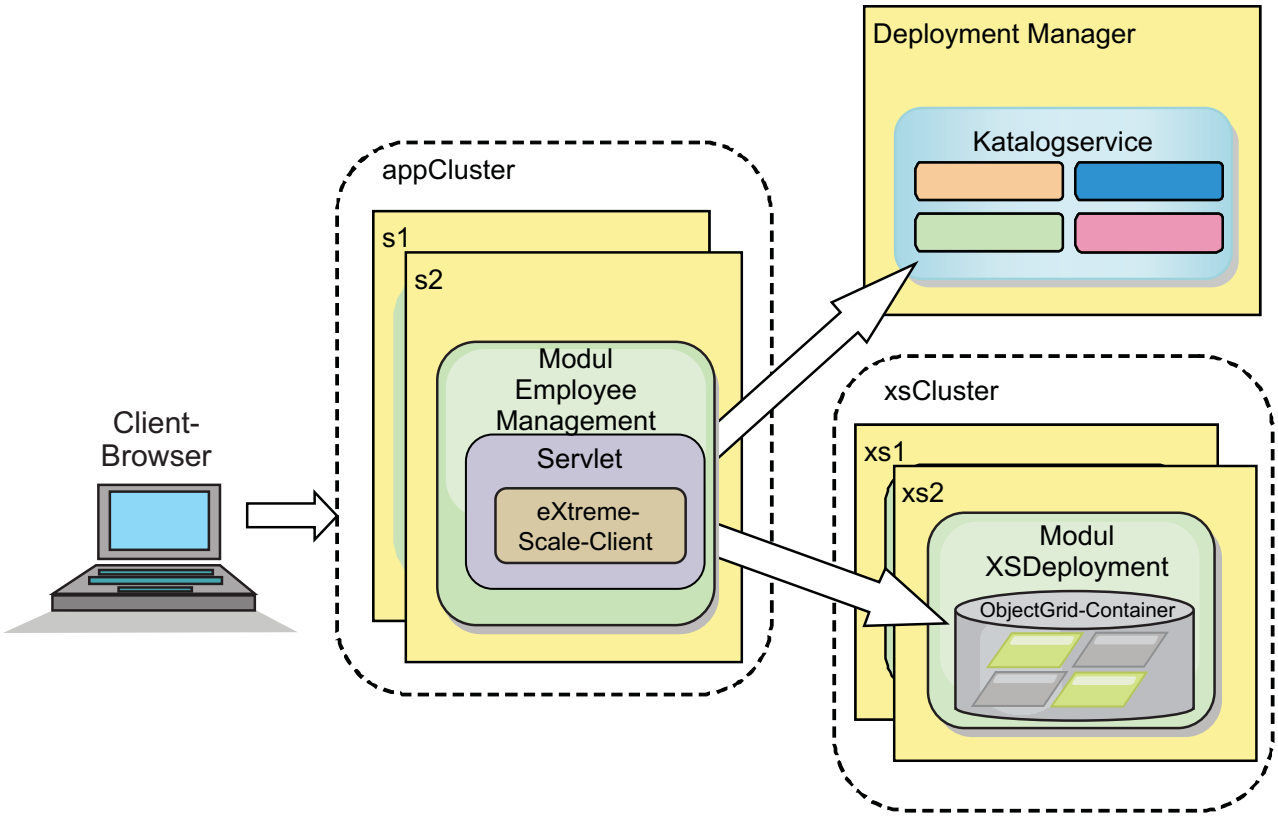


Abbildung 19. Lernprogrammtopologie

Anwendungen: In diesem Lernprogramm verwenden Sie zwei Anwendungen und eine gemeinsam genutzte Bibliotheksdatei:

- **`EmployeeManagement.ear`**: Die Anwendung `EmployeeManagement.ear` ist eine vereinfachte J2EE-Unternehmensanwendung (Java 2 Platform, Enterprise Edition). Sie enthält ein Webmodul für die Verwaltung der Mitarbeiterprofile. Das Webmodul enthält die Datei `management.jsp`, die Mitarbeiterprofile anzeigt, einfügt, aktualisiert und löscht, die in den Container-Servern gespeichert sind.
- **`XSDeployment.ear`**: Diese Anwendung enthält ein Unternehmensanwendungsmodul ohne Anwendungsartefakte. Die Cacheobjekte sind in der Datei `EmployeeData.jar` gepackt. Die Datei `EmployeeData.jar` ist als gemeinsam genutzte Bibliothek für die Datei `XSDeployment.ear` implementiert, so dass die Datei `XSDeployment.ear` auf die Klassen zugreifen kann. Der Zweck dieser Anwendung ist, die Konfigurationsdateien von eXtreme Scale zu packen. Wenn diese Unternehmensanwendung gestartet wird, werden die Konfigurationsdateien von eXtreme Scale automatisch von der Laufzeitumgebung von eXtreme Scale erkannt, sodass die Container-Server erstellt werden. Diese Konfigurationsdateien enthalten die Dateien `objectGrid.xml` und `objectGridDeployment.xml`.
- **`EmployeeData.jar`**: Diese JAR-Datei enthält eine einzige Klasse: com.ibm.websphere.sample.xs.data.EmployeeData. Diese Klasse stellt Mitarbeiterdaten dar, die im Grid gespeichert sind. Diese JAR-Datei wird mit den Dateien `EmployeeManagement.ear` und `XSDeployment.ear` als gemeinsam genutzte Bibliothek implementiert.

Lernprogrammdateien abrufen:

1. Laden Sie die Dateien `WASSecurity.zip` und `security.zip` herunter. Sie können die Beispielanwendung aus dem Wiki zu WebSphere eXtreme Scale herunterladen.

2. Entpacken Sie die Datei WASSecurity.zip in einem Verzeichnis wie /wxs_samples/, um die Binärdateien und Qellenartefakte anzuzeigen. Im verbleibenden Teil des Lernprogramms wird *Ausgangsverzeichnis_für_Beispiele* für dieses Verzeichnis verwendet. Eine Beschreibung des Inhalts der Datei WASSecurity.zip und des Ladens der Quelle in den Eclipse-Arbeitsbereichs finden Sie in der Datei README.txt im Paket.
3. Entpacken Sie die Datei security.zip im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*. Die Datei security.zip enthält die folgenden Sicherheitskonfigurationsdateien, die in diesem Lernprogramm verwendet werden:
   - catServer2.props
   - server2.props
   - client2.props
   - securityWAS2.xml
   - xsAuth2.props

Informationen zu den Konfigurationsdateien:

Die Dateien objectGrid.xml und objectGridDeployment.xml erstellen die Datengrids und Maps, in denen die Anwendungsdaten gespeichert werden.

Diese Konfigurationsdateien müssen objectGrid.xml und objectGridDeployment.xml genannt werden. Wenn der Anwendungsserver gestartet wird, erkennt eXtreme Scale diese Dateien im Verzeichnis META-INF der EJB- und Webmodule. Werden diese Dateien gefunden, wird angenommen, dass die Java Virtual Machine (JVM) als Container-Server für die in den Konfigurationsdateien definierten Datengrids dient.

Datei objectGrid.xml

Die Datei objectGrid.xml definiert ein ObjectGrid mit dem Namen Grid. Das Datengrid Grid enthält eine einzige Map mit dem Namen Map1, in der das Mitarbeiterprofil für die Anwendung gespeichert wird.

```
<?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>
```

Datei objectGridDeployment.xml

Die Datei objectGridDeployment.xml gibt an, wie das Datengrid Grid implementiert wird. Wenn das Grid implementiert wird, hat es fünf Partitionen und ein synchrones Replikat.

```
<?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>
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie sich mit der Topologie für das Lernprogramm vertraut gemacht und die Konfigurationsdateien und Beispielanwendungen in Ihrer Umgebung hinzugefügt.

Wenn Sie mehr über das automatische Starten von Container-Servern wissen möchten, lesen Sie den Artikel „Anwendungen von WebSphere Application Server für den automatischen Start von Container-Servern konfigurieren" auf Seite 282.

Lerneinheit 1.2: Umgebung von WebSphere Application Server konfigurieren

Um die Umgebung für das Lernprogramm vorzubereiten, müssen Sie die Sicherheit von WebSphere Application Server konfigurieren. Aktivieren Sie die Verwaltungs- und Anwendungssicherheit mit internen dateibasierten eingebundenen Repositorys als Benutzer-Account-Registry. Anschließend können Sie Server-Cluster zum Hosten der Clientanwendung und der Container-Server erstellen.

Die folgenden Schritte wurden mit WebSphere Application Server Version 7.0 geschrieben. Sie können die Konzepte aber auch auf frühere Versionen von WebSphere Application Server anwenden.

Sicherheit von WebSphere Application Server konfigurieren:

1. Konfigurieren Sie die Sicherheit von WebSphere Application Server.
   a. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Sicherheit** > **Globale Sicherheit**.
   b. Wählen Sie **Eingebundene Repositorys** als **Benutzer-Account-Repository** aus. Klicken Sie auf **Als aktuelles Repository festlegen**.
   c. Klicken Sie auf **Konfigurieren...**, um die Anzeige **Eingebundene Repositorys** aufzurufen.
   d. Geben Sie einen Wert im Feld **Name des primären Benutzers mit Verwaltungsaufgaben** ein, z. B. `admin`. Klicken Sie auf **Anwenden**.
   e. Geben Sie auf Anforderung das Kennwort des Benutzers mit Verwaltungsaufgaben ein, und klicken Sie auf **OK**. Speichern Sie Ihre Änderungen.
   f. Vergewissern Sie sich, dass die Einstellung **Eingebundene Repositorys** auf der Seite **Globale Sicherheit** auf die aktuelle Benutzer-Account-Registry gesetzt ist.
   g. Wählen Sie die folgenden Einträge aus: **Verwaltungssicherheit aktivieren**, **Anwendungssicherheit aktivieren** und **Java-2-Sicherheit verwenden, um den Anwendungszugriff auf lokale Ressourcen zu beschränken**. Klicken Sie auf **Anwenden**, und speichern Sie Ihre Änderungen.
   h. Starten Sie den Deployment Manager und alle aktiven Anwendungsserver erneut.

   Die Verwaltungssicherheit von WebSphere Application Server wird über die internen dateibasierten eingebundenen Repositorys als Benutzer-Account-Registry aktiviert.
2. Erstellen Sie zwei Benutzergruppen: adminGroup und operatorGroup.
   a. Klicken Sie auf **Benutzer und Gruppen** > **Gruppen verwalten** > **Erstellen...**.

b. Geben Sie adminGroup als Gruppennamen ein. Geben Sie Verwaltungsgruppe als Beschreibung ein. Klicken Sie auf **Erstellen**.
c. Klicken Sie auf **Create alike**. Geben Sie operatorGroup als Gruppennamen ein. Geben Sie Operator group als Beschreibung ein. Klicken Sie auf **Erstellen**.
d. Klicken Sie auf **Schließen**.

3. Erstellen Sie die Benutzer admin1 und operator1.
   a. Klicken Sie auf **Benutzer und Gruppen** > **Benutzer verwalten** > **Erstellen...**
   b. Erstellen Sie einen Benutzer mit dem Namen admin1, dem Vornamen Joe und dem Nachnamen Doe mit dem Kennwort admin1. Klicken Sie auf **Erstellen**.
   c. Erstellen Sie einen zweiten Benutzer. Klicken Sie auf **Create alike**, um einen Benutzer mit dem Namen operator1, dem Vornamen Jane und dem Nachnamen Doe mit dem Kennwort operator1 zu erstellen. Klicken Sie auf **Erstellen**. Klicken Sie auf **Schließen**.
4. Fügen Sie den Benutzergruppen Benutzer hinzu. Fügen Sie den Benutzer admin1 der Gruppe adminGroup und den Benutzer operator1 der Gruppe operatorGroup hinzu.
   a. Klicken Sie auf **Benutzer und Gruppen** > **Benutzer verwalten**.
   b. Suchen Sie die Benutzer, die Sie den Gruppen hinzufügen möchten. Klicken Sie auf **Suchen...**, und setzen Sie den Suchbegriff auf einen Stern (*), um alle Benutzer anzuzeigen.
   c. Wählen Sie in den Suchergebnissen den Benutzer admin1 aus, und klicken Sie dann auf das Register **Gruppen**. Klicken Sie auf **Hinzufügen**, um die Gruppe hinzuzufügen.
   d. Suchen Sie die verfügbaren Gruppen. Klicken Sie auf adminGroup, und klicken Sie auf **Hinzufügen**.
   e. Wiederholen Sie diese Schritte, um den Benutzer operator1 der Benutzergruppe operatorGroup hinzuzufügen.
5. Speichern Sie Ihre Änderungen, melden Sie sich von der Administrationskonsole ab, und starten Sie den Deployment Manager und den Node Agent erneut, um die Sicherheitseinstellungen zu aktivieren.

Sie haben die Sicherheit aktiviert und Benutzer und Benutzergruppen erstellt, die Verwaltungs- und Bedienerzugriff auf Ihre Konfiguration von WebSphere Application Server haben.

Server-Cluster erstellen:

Erstellen Sie zwei Server-Cluster in Ihrer Konfiguration von WebSphere Application Server, den Cluster appCluster zum Hosten der Beispielanwendung für das Lernprogramm und den Cluster xsCluster zum Hosten des Datengrids.

1. Öffnen Sie in der Administrationskonsole von WebSphere Application Server die Anzeige "Cluster". Klicken Sie auf **Server** > **Cluster** > **WebSphere-Anwendungsservercluster** > **Neu**.
2. Geben Sie appCluster als Clusternamen ein, lassen Sie die Option **Prefer local** ausgewählt, und klicken Sie auf **Weiter**.
3. Erstellen Sie Server im Cluster. Erstellen Sie einen Server mit dem Namen s1, und übernehmen Sie dabei die Standardoptionen. Fügen Sie ein weiteres Cluster-Member mit dem Namen s2 hinzu.
4. Führen Sie die verbleibenden Schritte im Assistenten aus, um den Cluster zu erstellen. Speichern Sie die Änderungen.

5. Wiederholen Sie diese Schritte, um den Cluster `xsCluster` zu erstellen. Dieser Cluster hat zwei Server, `xs1` und `xs2`.

Prüfpunkt der Lerneinheit:

Sie haben die globale Sicherheit für die Zelle von WebSphere Application Server aktiviert, Benutzer und Benutzergruppen sowie Cluster zum Hosten der Anwendung und des Datengrids erstellt.

Modul 2: WebSphere eXtreme Scale für die Verwendung der Authentifizierungs-Plug-ins von WebSphere Application Server konfigurieren

Nachdem Sie die Konfiguration von WebSphere Application Server erstellt haben, können Sie die Authentifizierung von WebSphere eXtreme Scale mit WebSphere Application Server integrieren.

Wenn ein Client von WebSphere eXtreme Scale eine Verbindung zu einem Container-Server herstellt, der eine Authentifizierung erfordert, muss der Client einen Berechtigungsnachweisgenerator bereitstellen, der von der Schnittstelle com.ibm-.websphere.objectgrid.security.plugins.CredentialGenerator dargestellt wird. Ein Berechtigungsnachweisgenerator ist eine Factory für die Erstellung von Clientberechtigungsnachweisen. Gültige Clientberechtigungsnachweise sind Benutzername/Kennwort-Paare, Kerberos-Tickets, Clientzertifikate und Clientidentifikationsdaten in einem beliebigen Format, auf das sich Client und Server einigen. Weitere Einzelheiten finden Sie in der Dokumentation zur API Credential. In diesem Beispiel ist der Client von WebSphere eXtreme Scale die Webanwendung "EmployeeManagment", die im Cluster "appCluster" implementiert ist. Der Clientberechtigungsnachweis ist ein WebSphere-Sicherheitstoken, das die Identität des Webbenutzers darstellt.

Lernziele

Anhand der Übungen in diesem Modul lernen Sie Folgendes:

- Client-Server-Sicherheit konfigurieren
- Katalogserversicherheit konfigurieren
- Container-Server-Sicherheit konfigurieren
- Installieren Sie die Beispielanwendung, und führen Sie sie aus.

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Lerneinheit 2.1: Client/Server-Sicherheit konfigurieren

Die Clienteigenschaftendatei gibt die zu verwendende CredentialGenerator-Implementierungsklasse an.

Konfigurieren Sie die Clienteigenschaftendatei mit der JVM-Eigenschaft **`-Dobjectgrid.client.props`**. Der Dateiname, der mit dieser Eigenschaft angegeben wird, ist ein absoluter Dateipfad, wie z. B. *Ausgangsverzeichnis_für_Beispiele*`/security/client2.props`. Weitere Informationen zur Clienteigenschaftendatei finden Sie unter Clienteigenschaftendatei.

Inhalt der Clienteigenschaftendatei:

In diesem Beispiel werden Sicherheitstoken von WebSphere Application Server als Clientberechtigungsnachweis verwendet. Die Datei `client2.props` befindet sich im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*`/security`. Die Datei `client2.props` enthält die folgenden Einstellungen:

securityEnabled
: Wenn diese Eigenschaft auf `true` gesetzt wird, muss der Client verfügbare Sicherheitsinformationen an den Server senden.

credentialAuthentication
: Wenn Sie diese Eigenschaft auf `Supported` setzen, unterstützt der Client die Authentifizierung der Berechtigungsnachweise.

credentialGeneratorClass
: Gibt die Klasse com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator an, damit der Client die Sicherheitstoken aus dem Thread abruft. Weitere Informationen zum Abrufen von Sicherheitstoken finden Sie unter „Integration der Sicherheit mit WebSphere Application Server" auf Seite 533.

Clienteigenschaftendatei mit JVM-Eigenschaften definieren:

Führen Sie in der Administrationskonsole die folgenden Schritte für die Server `s1` und `s2` im Cluster `appCluster` aus. Wenn Sie eine andere Topologie verwenden, führen Sie die folgenden Schritte für alle Anwendungsserver aus, in denen die Anwendung `EmployeeManagement` implementiert ist.

1. Klicken Sie auf **Server** > **WebSphere-Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Erstellen Sie die folgende generische JVM-Eigenschaft, um die Position der Clienteigenschaftendatei festzulegen:

   `-Dobjectgrid.client.props=`*Ausgangsverzeichnis_für_Beispiele*`/security/client2.props`
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Prüfpunkt der Lerneinheit:

Sie haben die Clienteigenschaftendatei bearbeitet und die Server im Cluster `appCluster` für die Verwendung der Clienteigenschaftendatei konfiguriert. Diese Eigenschaftendatei gibt die zu verwendende CredentialGenerator-Implementierungsklasse an.

Lerneinheit 2.2: Sicherheit des Katalogservers konfigurieren

Ein Katalogserver enthält zwei Stufen von Sicherheitsinformationen: Die Sicherheitseigenschaften gelten für alle Server von WebSphere eXtreme Scale, einschließlich des Katalogservice und der Container-Server, und die Sicherheitseigenschaften, die speziell für den Katalogserver gelten.

Die Sicherheitseigenschaften, die für die Katalogserver und Container-Server gelten, werden in der XML-Sicherheitsdeskriptordatei konfiguriert. Ein Beispiel für allgemeine Eigenschaften ist die Authentifikatorkonfiguration, die die Benutzerregistry und den Authentifizierungsmechanismus darstellt. Weitere Informationen zu den Sicherheitseigenschaften finden Sie unter XML-Sicherheitsdeskriptordatei.

Zum Konfigurieren der XML-Sicherheitsdeskriptordatei erstellen Sie eine Eigenschaft "-Dobjectgrid.cluster.security.xml.url" im JVM-Argument (Java Virtual Machine). Der für diese Eigenschaft angegebene Dateiname hat ein URL-Format, wie z. B. `file:///`*Ausgangsverzeichnis_für_Beispiele*`/security/securityWAS2.xml`.

Datei securityWAS2.xml:

In diesem Lernprogramm befindet sich die Datei securityWAS2.xml im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/security. Im Folgenden sehen Sie den Inhalt der Datei securityWAS2.xml, im dem die Kommentare entfernt wurden:

```
<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>
```

Die folgenden Eigenschaften sind in der Datei securityWAS2.xml definiert:

securityEnabled
Die Eigenschaft "securityEnabled" ist auf true gesetzt. Dies zeigt dem Katalogserver an, dass die globale Sicherheit von WebSphere eXtreme Scale aktiviert ist.

authenticator
Der Authentifikator ist als Klasse com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenAuthenticator konfiguriert. Mit dieser integrierten Implementierung des Authenticator-Plug-ins kann der Server von WebSphere eXtreme Scale die Sicherheitstoken in ein Subject-Objekt konvertieren. Weitere Informationen zur Konvertierung von Sicherheitstoken finden Sie unter „Integration der Sicherheit mit WebSphere Application Server" auf Seite 533.

Datei catServer2.props:

In der Servereigenschaftendatei werden die serverspezifischen Eigenschaften gespeichert, zu denen auch die serverspezifischen Sicherheitseigenschaften gehören. Weitere Informationen finden Sie unter Servereigenschaftendatei. Sie können die Servereigenschaftendatei mit der Eigenschaft -Dobjectgrid.server.props im JVM-Argument konfigurieren. Geben Sie den Dateinamen für diese Eigenschaft in Form eines absoluten Pfads an, z. B. *Ausgangsverzeichnis_für_Beispiele*/security/catServer2.props. In diesem Lernprogramm ist die Datei catServer2.props im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/security enthalten. Im Folgenden sehen Sie den Inhalt der Datei catServer2.props, in dem die Kommentare entfernt wurden:

securityEnabled
Die Eigenschaft "securityEnabled" ist auf true gesetzt. Damit wird angezeigt, dass dieser Katalogserver ein sicherer Server ist.

credentialAuthentication
Die Eigenschaft "credentialAuthentication" wird auf Required gesetzt, so dass jeder Client, der eine Verbindung zum Server herstellt, einen Berechtigungsnachweis bereitstellen muss.

secureTokenManagerType
Die Eigenschaft "secureTokenManagerType" wird auf none gesetzt, um anzuzeigen, dass der geheime Schlüssel für die Authentifizierung nicht verschlüsselt ist, wenn die vorhandenen Server verbunden werden.

authenticationSecret
Die Eigenschaft "authenticationSecret" wird auf ObjectGridDefaultSecret gesetzt. Diese geheime Zeichenfolge wird für die Einbindung in den eXtre-

me-Scale-Server-Cluster verwendet. Wenn ein Server in das Datengrid eingebunden wird, wird er aufgefordert, diese Shared-Secret-Zeichenfolge vorzulegen. Wenn die geheime Zeichenfolge des einzubindenden Servers mit der Zeichenfolge im Katalogserver übereinstimmt, wird der Server akzeptiert. Stimmen die Zeichenfolgen nicht überein, wird die Anforderung zur Einbindung des Servers zurückgewiesen.

transportType
Die Eigenschaft "transportType" wird zunächst auf `TCP/IP` gesetzt. Später in diesem Lernprogramm wird die Transportsicherheit aktiviert.

Servereigenschaftendatei mit JVM-Eigenschaften definieren:

Definieren Sie die Servereigenschaftendatei im Deployment-Manager-Server. Wenn Sie eine andere Topologie als die für dieses Lernprogramm verwendete Topologie verwenden, definieren Sie die Servereigenschaftendatei in allen Anwendungsservern, die Sie zum Hosten von Container-Servern verwenden.

1. Öffnen Sie die JVM-Konfiguration (Java Virtual Machine) für den Server. Klicken Sie in der Administrationskonsole auf **Systemverwaltung** > **Deployment Manager** > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Fügen Sie die folgenden generischen JVM-Argumente hinzu:
   ```
   -Dobjectgrid.cluster.security.xml.url=file:///Ausgangsverzeichnis_für_Beispiele/security/securityWAS2.xml
   -Dobjectgrid.server.props=Ausgangsverzeichnis_für_Beispiele/security/catServer2.props
   ```
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Prüfpunkt der Lerneinheit:

Sie haben die Katalogserversicherheit konfiguriert, indem Sie dem Deployment Manager, in dem der Katalogserverprozess in der Konfiguration von WebSphere Application Server ausgeführt wird, die Dateien `securityWAS2.xml` und `catServer2.props` zugeordnet.

Lerneinheit 2.3: Sicherheit des Container-Servers konfigurieren

Wenn ein Container-Server eine Verbindung zum Katalogservice herstellt, ruft der Container-Server alle Sicherheitskonfigurationen ab, die in der ObjectGrid-XML-Sicherheitsdatei konfiguriert sind, wie z. B. die Authentifikatorkonfiguration, das Zeitlimit für Anmeldesitzungen und weitere Konfigurationsdaten. Außerdem hat ein Container-Server eigene serverspezifische Sicherheitseigenschaften in der Servereigenschaftendatei.

Konfigurieren Sie die Servereigenschaftendatei mit der JVM-Eigenschaft `-Dobjectgrid.server.props`. Der Dateiname, der mit dieser Eigenschaft angegeben wird, ist ein absoluter Dateipfad, wie z. B. *Ausgangsverzeichnis_für_Beispiele*/`security/server2.props`.

In diesem Lernprogramm sind die Container-Server in den Servern `xs1` und `xs2` im Cluster `xsCluster` gehostet.

Datei `server2.props`:

Die Datei `server2.props` befindet sich im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/`security` im Verzeichnis `WASSecurity`. Im Folgenden sind die in der Datei `server2.props` definierten Eigenschaften beschrieben:

securityEnabled
Die Eigenschaft "securityEnabled" wird auf `true` gesetzt, um Container-Server als sicheren Server zu kennzeichnen.

credentialAuthentication
Die Eigenschaft "credentialAuthentication" wird auf `Required` gesetzt, so dass jeder Client, der eine Verbindung zum Server herstellt, einen Berechtigungsnachweis bereitstellen muss.

secureTokenManagerType
Die Eigenschaft "secureTokenManagerType" wird auf `none` gesetzt, um anzuzeigen, dass der geheime Schlüssel für die Authentifizierung nicht verschlüsselt ist, wenn die vorhandenen Server verbunden werden.

authenticationSecret
Die Eigenschaft "authenticationSecret" wird auf `ObjectGridDefaultSecret` gesetzt. Diese geheime Zeichenfolge wird für die Einbindung in den eXtreme-Scale-Server-Cluster verwendet. Wenn ein Server dem Datengrid beitritt, wird er aufgefordert, diese Shared-Secret-Zeichenfolge vorzulegen. Wenn die geheime Zeichenfolge des einzubindenden Servers mit der Zeichenfolge im Katalogserver übereinstimmt, wird der Server akzeptiert. Stimmen die Zeichenfolgen nicht überein, wird die Anforderung zur Einbindung des Servers zurückgewiesen.

Servereigenschaftendatei mit JVM-Eigenschaften definieren:

Definieren Sie die Servereigenschaftendatei in den Servern "xs1" und "xs2". Wenn Sie die Topologie für dieses Lernprogramm nicht verwenden, definieren Sie die Servereigenschaftendatei in allen Anwendungsservern, die Sie zum Hosten von Container-Servern verwenden.

1. Öffnen Sie die Seite "Java Virtual Machine" für den Server. Klicken Sie auf **Server** > **Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Fügen Sie die folgenden generischen JVM-Argumente hinzu:
   `-Dobjectgrid.server.props=`*`Ausgangsverzeichnis_für_Beispiele`*`/security/server2.props`
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Prüfpunkt der Lerneinheit:

Jetzt ist die Serverauthentifizierung von WebSphere eXtreme Scale gesichert. Wenn Sie diese Sicherheit konfigurieren, müssen alle Anwendungen, die versuchen, eine Verbindung zu Servern von WebSphere eXtreme Scale herzustellen, einen Berechtigungsnachweis bereitstellen. In diesem Lernprogramm ist WSTokenAuthenticator der Authentifikator. Deshalb muss der Client ein Sicherheitstoken von WebSphere Application Server bereitstellen.

Lerneinheit 2.4: Beispiel installieren und ausführen

Nach der Konfiguration der Authentifizierung können Sie die Beispielanwendung installieren und ausführen.

Gemeinsam genutzte Bibliothek für die Datei `EmployeeData.jar` erstellen:

1. Öffnen Sie in der Administrationskonsole von WebSphere Application Server die Seite **Gemeinsam genutzte Bibliotheken**. Klicken Sie auf **Umgebung** > **Gemeinsam genutzte Bibliotheken**.
2. Wählen Sie den Geltungsbereich **Zelle** aus.

3. Erstellen Sie die gemeinsam genutzte Bibliothek. Klicken Sie auf **Neu**. Geben Sie `EmployeeManagementLIB` im Feld **Name** ein. Geben Sie den Pfad zur Datei `EmployeeData.jar` im Klassenpfad ein, z. B. *Ausgangsverzeichnis_für_Beispiele*/`WASSecurity/EmployeeData.jar`.
4. Klicken Sie auf **Anwenden**.

Beispiel installieren:

1. Installieren Sie die Datei `EmployeeManagement.ear`.
   a. Klicken Sie zum Starten der Installation auf **Anwendungen** > **Neue Anwendung** > **Neue Unternehmensanwendung**. Wählen Sie den detaillierten Pfad für die Installation der Anwendung aus.
   b. Geben Sie im Schritt **Servern Module zuordnen** den Cluster `appCluster` für die Installation des Moduls "EmployeeManagementWeb" an.
   c. Wählen Sie im Schritt **Gemeinsam genutzte Bibliotheken zuordnen** das Modul `EmployeeManagementWeb` aus.
   d. Klicken Sie auf **Gemeinsam genutzte Bibliotheken referenzieren**. Wählen Sie die Bibliothek `EmployeeManagementLIB` aus.
   e. Ordnen Sie **Alle Authentifizierten im Anwendungsrealm** die Rolle `webUser` zu.
   f. Klicken Sie auf **OK**.

   Die Clients werden in den Servern `s1` und `s2` in diesem Cluster ausgeführt.
2. Installieren Sie die Beispieldatei `XSDeployment.ear`.
   a. Klicken Sie zum Starten der Installation auf **Anwendungen** > **Neue Anwendung** > **Neue Unternehmensanwendung**. Wählen Sie den detaillierten Pfad für die Installation der Anwendung aus.
   b. Geben Sie im Schritt **Servern Module zuordnen** den Cluster `xsCluster` für die Installation des Webmoduls `XSDeploymentWeb` an.
   c. Wählen Sie im Schritt **Gemeinsam genutzte Bibliotheken zuordnen** das Modul `XSDeploymentWeb` aus.
   d. Klicken Sie auf **Gemeinsam genutzte Bibliotheken referenzieren**. Wählen Sie die Bibliothek `EmployeeManagementLIB` aus.
   e. Klicken Sie auf **OK**.

   Die Server `xs1` und `xs2` in diesem Cluster hosten die Container-Server.
3. Starten Sie den Deployment Manager erneut. Wenn der Deployment Manager gestartet wird, wird auch der Katalogserver gestartet. Wenn Sie sich die Datei `SystemOut.log` des Deployment Manager ansehen, können Sie die folgende Nachricht sehen, die darauf hinweist, dass die Eigenschaftendatei des eXtreme-Scale-Servers geladen wurde.

   ```
   CWOBJ0913I: Die Servereigenschaftendateien wurden geladen:
   /wxs_samples/security/catServer2.props.
   ```
4. Starten Sie den Cluster "xsCluster" erneut. Wenn der Cluster "xsCluster" gestartet wird, wird die Anwendung `XSDeployment` und jeweils ein Container-Server auf den Servern "xs1" und "xs2" gestartet. Wenn Sie sich die Datei `SystemOut.log` der Server "xs1" und "xs2" ansehen, wird die folgende Nachricht angezeigt, die darauf hinweist, dass die Servereigenschaftendatei geladen wurde:

   ```
   CWOBJ0913I: Die Servereigenschaftendateien wurden geladen:
   /wxs_samples/security/server2.props.
   ```
5. Starten Sie den Cluster "appClusters" erneut. Wenn der Cluster "appCluster" gestartet wird, wird auch die Anwendung "EmployeeManagement" gestartet.

Wenn Sie sich die Datei SystemOut.log der Server "s1" und "s2" ansehen, können Sie die folgende Nachricht sehen, die darauf hinweist, dass die Clienteigenschaftendatei geladen wurde.

```
CWOBJ0924I: Die Clienteigenschaftendatei {0} wurde geladen.
```

Sie können die Warnung bezüglich der Eigenschaften "authenticationRetryCount", "transportType" und "clientCertificateAuthentication" ignorieren. Es werden die Standardwerte verwendet weil die Werte nicht in der Eigenschaftendatei angegeben wurden.Wenn Sie WebSphere eXtreme Scale Version 7.0 verwenden, wird die Nachricht CWOBJ9000I angezeigt, die nur in Englisch verfügbar ist und darauf hinweist, dass die Clienteigenschaftendatei geladen wurde. Wenn Sie die erwartete Nachricht nicht sehen, vergewissern Sie sich, dass Sie die Eigenschaft "-Dobjectgrid.server.props" bzw. "-Dobjectgrid.client.props" im JVM-Argument konfiguriert haben. Ist die Eigenschaft konfiguriert, stellen Sie sicher, dass der Strich (-) ein UTF-Zeichen ist.

Beispielanwendung ausführen:

1. Führen Sie die Datei management.jsp aus. Greifen Sie in einem Web-Browser auf den URL http://*<Ihr_Servername>*:*<Port>*/EmployeeManagementWeb/management.jsp auf. Sie können beispielsweise den folgenden URL verwenden: http://localhost:9080/EmployeeManagementWeb/management.jsp.
2. Authentifizierung für die Anwendung bereitstellen. Geben Sie die Berechtigungsnachweise des Benutzers ein, den Sie der Rolle "webUser" zugeordnet haben. Standardmäßig wird diese Benutzerrolle allen authentifizierten Benutzern zugeordnet. Geben Sie admin1 als Benutzer-ID und admin1 als Kennwort ein. Es erscheint eine Seite, auf der Sie Mitarbeiter anzeigen, hinzufügen, aktualisieren und löschen können.
3. Mitarbeiter anzeigen. Klicken Sie auf **Display an Employee**. Geben Sie emp1@acme.com als E-Mail-Adresse ein, und klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer nicht gefunden wurde.
4. Mitarbeiter hinzufügen. Klicken Sie auf **Add an Employee**. Geben Sie emp1@acme.com als E-Mail-Adresse, Joe als Vornamen und Doe als Nachnamen ein. Klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass ein Mitarbeiter mit der Adresse emp1@acme.com hinzugefügt wurde.
5. Neuen Mitarbeiter anzeigen. Klicken Sie auf **Display an Employee**. Geben Sie emp1@acme.com als E-Mail-Adresse mit leeren Feldern für den Vor- und Nachnamen ein, und klicken Sie dann auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Mitarbeiter gefunden wurde, und die richtigen Namen werden in den Feldern für den Vornamen und den Nachnamen angezeigt.
6. Mitarbeiter löschen. Klicken Sie auf **Delete an employee**. Geben Sie emp1@acme.com ein, und klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer gelöscht wurde.

Prüfpunkt der Lerneinheit:

Sie haben die Beispielanwendung installiert und ausgeführt. Da in diese Lernprogramm die Integration von WebSphere Application Server verwendet wird, können Sie das Szenario, in dem sich ein Client nicht beim eXtreme-Scale-Server authentifizieren kann, nicht sehen. Wenn sich der Benutzer erfolgreich bei WebSphere Application Server authentifiziert, wird auch eXtreme Scale erfolgreich authentifiziert.

Modul 3: Transportsicherheit konfigurieren

Konfigurieren Sie die Transportsicherheit, um die Datenübertragung zwischen den Clients und den Servern in der Konfiguration zu sichern.

Im vorherigen Modul des Lernprogramms haben Sie die eXtreme-Scale-Authentifizierung aktiviert. Mit Authentifizierung muss jede Anwendung, die versucht, eine Verbindung zum Server von WebSphere eXtreme Scale herzustellen, einen Berechtigungsnachweis bereitstellen. Deshalb kann kein nicht authentifizierter Client eine Verbindung zum Server von WebSphere eXtreme Scale herstellen. Die Clients müssen eine authentifizierte Anwendung sein, die in einer Zelle von WebSphere Application Server ausgeführt wird.

Mit der Konfiguration, die Sie bis zu diesem Modul hin vorgenommen haben, ist die Datenübertragung zwischen den Clients im Cluster "appCluster" und den Servern im Cluster "xsCluster" nicht verschlüsselt. Diese Konfiguration kann akzeptabel sein, wenn Ihre Cluster von WebSphere Application Server auf Servern hinter einer Firewall installiert sind. In manchen Szenarien wird ein nicht verschlüsselter Datenverkehr jedoch aus bestimmten Gründen nicht akzeptiert, selbst wenn die Topologie durch eine Firewall geschützt ist. Der verschlüsselte Datenverkehr kann beispielsweise durch eine behördliche Richtlinie vorgegeben werden. WebSphere eXtreme Scale unterstützt Transport Layer Security/Secure Sockets Layer (TLS/SSL) für die sichere Kommunikation zwischen ObjectGrid-Endpunkten, zu denen Client-Server, Container-Server und Katalogserver gehören.

In dieser Beispielimplementierung werden die Clients und die Container-Server von eXtreme Scale alle in der Umgebung von WebSphere Application Server ausgeführt werden. Es sind keine Client- und Servereigenschaften zum Konfigurieren der SSL-Einstellungen erforderlich, weil die Transportsicherheit von eXtreme Scale durch die CSIv2-Transporteinstellungen (Common Secure Interoperability Protocol Version 2) des Anwendungsservers verwaltet wird. Die Server von WebSphere eXtreme Scale verwenden dieselbe ORB-Instanz (Object Request Broker) wie die Anwendungsserver, in denen sie ausgeführt werden. Legen Sie alle SSL-Einstellungen für die Clients und Container-Server mit diesen CSIv2-Transporteinstellungen in der Konfiguration von WebSphere Application Server fest. Der Katalogserver hat eigene proprietäre Transportpfade, die weder Internet Inter-ORB Protocol (IIOP) noch Remote Method Invocation (RMI) verwenden. Wegen dieser proprietären Transportpfade kann der Katalogserver nicht über die CSIv2-Transporteinstellungen von WebSphere Application Server verwaltet werden. Deshalb müssen Sie die SSL-Einstellungen in der Servereigenschaftendatei für den Katalogserver konfigurieren.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie in der Lage, die folgenden Aufgaben auszuführen:

- Eingehende und abgehende CSIv2-Transporte konfigurieren
- SSL-Eigenschaften zur Eigenschaftendatei des Katalogservers hinzufügen
- ORB-Eigenschaftendatei überprüfen
- Beispiel ausführen

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Voraussetzungen

Dieser Schritt des Lernprogramms baut auf den vorherigen Modulen auf. Arbeiten Sie die vorherigen Module in diesem Lernprogramm durch, bevor Sie die Transportsicherheit konfigurieren.

Lerneinheit 3.1: Eingehenden und abgehenden CSIv2-Transport konfigurieren

Zum Konfigurieren von Transport Layer Security/Secure Sockets Layer (TLS/SSL) für den Servertransport setzen Sie den eingehenden CSIv2-Transport (Common Secure Interoperability Protocol Version 2) und den abgehenden CSIv2-Transport für alle Server von WebSphere Application Server, die Clients, Katalogserver und Container-Server hosten, auf `SSL-Required`.

In der Beispieltopologie des Lernprogramms müssen Sie diese Eigenschaften für die Anwendungsserver s1, s2, xs1 und xs2 definieren. Mit den folgenden Schritten konfigurieren Sie die eingehenden und abgehenden Transporte für alle Server in der Konfiguration.

Definieren Sie die eingehenden und abgehenden Transporte in der Administrationskonsole. Stellen Sie sicher, dass die Verwaltungssicherheit aktiviert ist.

- **WebSphere Application Server Version 6.1**: Klicken Sie auf **Sicherheit** > **Sichere Verwaltung** > **Anwendung** > **RMI/IIOP-Sicherheit**, und ändern Sie den Transporttyp in **SSL-Required**.
- **WebSphere Application Server Version 7.0**: Klicken Sie auf **Sicherheit** > **Globale Sicherheit** > **RMI/IIOP-Sicherheit** > **Eingehende CSIv2-Kommunikation**. Ändern Sie den Transporttyp unter der CSIv2-Transportschicht in **SSL-Required**. Wiederholen Sie diesen Schritt zum Konfigurieren der abgehenden CSIv2-Kommunikation.

Sie können zentral verwaltete Endpunktsicherheitseinstellungen verwenden oder SSL-Repositorys konfigurieren. Weitere Informationen finden Sie unter Common Secure Interoperability Version 2 transport inbound settings.

Lerneinheit 3.2: SSL-Eigenschaften der Eigenschaftendatei des Katalogservers hinzufügen

Der Katalogserver hat eigene proprietäre Transportpfade, die nicht über die CSIv2-Transporteinstellungen (WebSphere Application Server Common Secure Interoperability Protocol Version 2) verwaltet werden können. Deshalb müssen Sie die SSL-Eigenschaften in der Servereigenschaftendatei für den Katalogserver konfigurieren.

Zum Konfigurieren der Katalogserversicherheit sind weitere Schritte erforderlich, weil der Katalogserver eigene proprietäre Transportpfade hat. Diese Transportpfade können nicht mit den CSIv2-Transporteinstellungen des Anwendungsservers verwaltet werden.

1. Editieren Sie die SSL-Eigenschaften in der Datei `catServer2.props`. Zum Konfigurieren der Katalogserversicherheit entfernen Sie die Kommentarzeichen aus den Zeilen mit den folgenden SSL-Eigenschaften in der Eigenschaftendatei des Katalogservers. Für dieses Lernprogramm sind die Eigenschaften des Katalogservers in der Datei `catServer2.props` enthalten. Aktualisieren Sie die Eigenschaften "keyStore" und "trustStore" so, dass sie auf die richtige Position in Ihrer Umgebung verweisen.

```
#alias=default
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR-NAME>/config/
```

```
cells/<ZELLENNAME>/nodes/<KNOTENNAME>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR-NAME>/config/
cells/<ZELLENNAME>/nodes/<KNOTENNAME>/trust.p12
#trustStorePassword=WebAS
#clientAuthentication=false
```

Die Datei `catServer2.props` verwendet den Standardkeystore und -truststore von WebSphere Application Server auf Knotenebene. Wenn Sie die Implementierung in einer komplexeren Implementierungsumgebung durchführen, müssen Sie den richtigen Keystore und Truststore auswählen. In manchen Fällen müssen Sie einen Keystore und einen Truststore erstellen und die Schlüssel aus Keystores anderer Server importieren. Die Zeichenfolge `WebAS` ist das Standardkennwort des Keystores und des Truststores von WebSphere Application Server. Weitere Einzelheiten finden Sie unter Konfiguration mit selbst signiertem Standardzertifikat.

2. Aktualisieren Sie in der Datei `catServer2.props` den Wert der Eigenschaft "transportType". In den vorherigen Schritten des Lernprogramms wurde der Wert auf `TCP/IP` gesetzt. Ändern Sie den Wert in `SSL-Required`.
3. Starten Sie den Deployment Manager erneut, um die Änderungen an den Sicherheitseinstellungen des Katalogservers zu aktivieren.

Prüfpunkt der Lerneinheit:

Sie haben die SSL-Eigenschaften für den Katalogserver konfiguriert.

Lerneinheit 3.3: Beispiel ausführen

Starten Sie alle Server erneut, und führen Sie die Beispielanwendung erneut aus. Sie sollten die Schritte ohne Probleme ausführen können.

Weitere Informationen zum Ausführen und Installieren der Beispielanwendung finden Sie unter „Lerneinheit 2.4: Beispiel installieren und ausführen” auf Seite 100.

Prüfpunkt der Lerneinheit:

Sie haben die Beispielanwendung mit aktivierter Transportsicherheit ausgeführt.

Modul 4: JAAS-Berechtigung (Java Authentication and Authorization Service) in WebSphere Application Server verwenden

Nachdem Sie nun die Authentifizierung für Clients konfiguriert haben, können Sie die Authentifizierung weiter konfigurieren, um verschiedenen Benutzern verschiedene Berechtigungen zuzuordnen. Ein Bediener kann beispielsweise nur in der Lage sein, Daten anzuzeigen, während ein Administrator alle Operationen ausführen kann.

Nach der Authentifizierung eines Clients können Sie wie im vorherigen Modul dieses Lernprogramms Sicherheitsberechtigungen über den Berechtigungsmechanismus von eXtreme Scale erteilen. Das vorherige Modul dieses Lernprogramms hat veranschaulicht, wie die Authentifizierung für ein Datengrid durch Integration mit WebSphere Application Server aktiviert wird. Deswegen kann kein nicht authentifizierter Client eine Verbindung zu den Servern von eXtreme Scale herstellen oder Anforderungen an Ihr System übergeben. Jeder authentifizierte Client hat jedoch dieselben Berechtigungen oder Privilegien beim Server, z. B. Lesen, Schreiben oder Löschen von Daten, die in ObjectGrid-Maps gespeichert sind. Clients können auch jeden Typ von Abfrage absetzen.

In diesem Teil des Lernprogramms wird gezeigt, wie Sie über eXtreme-Scale-Berechtigungen authentifizierten Benutzern unterschiedliche Privilegien erteilen. WebSphere eXtreme Scale verwendet einen auf Berechtigungen basierenden Berechtigungsmechanismus. Sie können verschiedene Berechtigungskategorien zuweisen, die durch verschiedene Berechtigungsklassen dargestellt werden. In diesem Modul wird die Klasse MapPermission verwendet. Eine Liste der möglichen Berechtigungen finden Sie unter Programmierung der Clientberechtigung.

In WebSphere eXtreme Scale stellt die Klasse `com.ibm.websphere.objectgrid.security.MapPermission` Berechtigungen für die Ressourcen von eXtreme Scale dar, insbesondere die Methoden der Schnittstelle ObjectMap oder JavaMap. WebSphere eXtreme Scale definiert die folgenden Berechtigungszeichenfolgen für den Zugriff auf die Methoden der Schnittstellen "ObjectMap" und "JavaMap":

- **read**: Erteilt die Berechtigung zum Lesen der Daten aus der Map.
- **write**: Erteilt die Berechtigung zum Aktualisieren der Daten in der Map.
- **insert**: Erteilt die Berechtigung zum Einfügen der Daten in die Map.
- **remove**: Erteilt die Berechtigung zum Entfernen der Daten aus der Map.
- **invalidate**: Erteilt die Berechtigung zum Ungültigmachen der Daten in der Map.
- **all**: Erteilt alle zuvor beschriebenen Berechtigungen: read, write, insert, remote und invalidate.

Die Berechtigung findet statt, wenn ein eXtreme-Scale-Client eine Datenzugriffs-API wie die API ObjectMap, JavaMap oder EntityManager verwendet. Die Laufzeitumgebung von eXtreme Scale überprüft die entsprechenden Map-Berechtigungen, wenn die Methode aufgerufen wird. Wenn dem Client die erforderlichen Berechtigungen nicht erteilt wurden, wird eine Ausnahme des Typs "AccessControlException" ausgegeben. Dieses Lernprogramm veranschaulicht, wie über JAAS-Berechtigung verschiedenen Benutzern Berechtigungen für Map-Zugriffe erteilt werden.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie in der Lage, die folgenden Aufgaben auszuführen:

- Berechtigung für WebSphere eXtreme Scale aktivieren
- Benutzerbasierte Berechtigung aktivieren
- Gruppenbasierte Berechtigung konfigurieren

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Voraussetzungen

Sie müssen die vorherigen Module dieses Lernprogramms durcharbeiten, bevor Sied die Authentifizierung konfigurieren.

Lerneinheit 4.1: eXtreme-Scale-Berechtigung aktivieren

Zum Aktivieren der Berechtigung in WebSphere eXtreme Scale müssen Sie die Sicherheit in einem bestimmten ObjectGrid aktivieren.

Zum Aktivieren der Berechtigung im ObjectGrid müssen Sie das Attribut **`securityEnabled`** für das entsprechende ObjectGrid in der XML-Datei auf "true"

setzen. Für dieses Lernprogramm können Sie die Datei XSDeployment_sec.ear im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/WASSecurity verwenden, in der die Sicherheit bereits in der Datei objectGrid.xml definiert ist, oder Sie können die vorhandene Datei objectGrid.xml bearbeiten, um die Sicherheit zu aktivieren. Diese Lerneinheit veranschaulicht, wie die Datei bearbeitet wird, um die Sicherheit zu aktivieren.

1. Extrahieren Sie die Dateien aus der Datei XSDeployment.ear, und entpacken Sie dann die Datei XSDeploymentWeb.war.
2. Öffnen Sie die Datei objectGrid.xml, und setzen Sie das Attribut "securityEnabled" auf ObjectGrid-Ebene auf true. Ein Beispiel für dieses Attribut folgt:

```
<?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>
```

   Wenn Sie mehrere ObjectGrids definiert haben, müssen Sie dieses Attribut in jedem Datengrid definieren.
3. Packen Sie die Datei XSDeploymentWeb.war und die Datei XSDeployment.ear erneut, um Ihre Änderungen einzuschließen. Nennen Sie die Datei XSDeployment_sec.ear, damit Sie das ursprüngliche Paket nicht überschreiben.
4. Deinstallieren Sie die vorhandene Anwendung XSDeployment, und installieren Sie die Datei XSDeployment_sec.ear. Weitere Informationen zum Implementieren von Anwendungen finden Sie unter „Lerneinheit 2.4: Beispiel installieren und ausführen” auf Seite 100.

Prüfpunkt der Lerneinheit:

Sie haben die Sicherheit im ObjectGrid und damit auch die Berechtigung im Datengrid aktiviert.

Lerneinheit 4.2: Benutzerbasierte Berechtigung aktivieren

IM Authentifizierungsmodul dieses Lernprogramms haben Sie zwei Benutzer erstellt: operator1 und admin1Sie können diesen Benutzern mit JAAS-Berechtigung (Java Authentication and Authorization Service) verschiedene Berechtigungen zuordnen.

JAAS-Berechtigungsrichtlinie (Java Authentication and Authorization Service) mit Benutzerprincipals definieren:

Sie können den Benutzern, die Sie zuvor erstellt haben, Berechtigungen zuordnen. Ordnen Sie dem Benutzer operator1 nur Leseberechtigungen für alle Maps zu. Ordnen Sie dem Benutzer admin1 alle Berechtigungen zu. Verwenden Sie die JAAS-Berechtigungsrichtliniendatei, um Principals Berechtigungen zu erteilen.

Bearbeiten Sie die JAAS-Berechtigungsdatei. Die Datei xsAuth2.policy befindet sich im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/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";
};
```

In dieser Datei ist die Codebasis `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` ein speziell reservierter URL für ObjectGrid. Alle ObjectGrid-Berechtigungen, die Principals erteilt werden, verwenden diese spezielle Codebasis. Die folgenden Berechtigungen werden in dieser Datei zugeordnet:

- Die erste grant-Anweisung erteilt dem Principal `operator1` die Map-Berechtigung `read`. Der Benutzer `operator1` hat nur Leseberechtigung für die Map "Map1" in der ObjectGrid-Instanz "Grid".
- Die zweite grant-Anweisung erteilt dem Principal "admin1" alle Map-Berechtigungen. Der Benutzer `admin1` hat alle Berechtigungen für die Map "Map1" in der ObjectGrid-Instanz "Grid".
- Der Name des Principals ist `defaultWIMFileBasedRealm/operator1`, aber nicht `Operator1`. WebSphere Application Server fügt den Realmnamen dem Name des Principals automatisch hinzu, wenn eingebundene Repositorys als Benutzer-Account-Registry verwendet werden. Passen Sie diesen Wert bei Bedarf an.

JAAS-Berechtigungsrichtliniendatei mit JVM-Eigenschaften definieren:

Verwenden Sie die folgenden Schritte, um JVM-Eigenschaften für die Server "xs1" und "xs2" im Cluster "xsCluster" zu definieren. Wenn Sie eine Topologie verwenden, die von der Beispieltopologie abweicht, die in diesem Lernprogramm verwendet wird, definieren Sie die Datei in allen Container-Servern.

1. Klicken Sie in der Administrationskonsole auf **Server** > **Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine.**.
2. Fügen Sie die folgenden generischen JVM-Argumente hinzu:
   `-Djava.security.auth.policy=`*`Ausgangsverzeichnis_für_Beispiele`*`/security/xsAuth2.policy`
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Beispielanwendung zu Testen der Berechtigung ausführen:

Sie können die Beispielanwendung verwenden, um die Berechtigungseinstellungen zu testen. Der Benutzer mit Administratorberechtigung hat weiterhin alle Berechtigungen in der Map "Map1", einschließlich der Berechtigungen zum Anzeigen und Hinzufügen von Mitarbeitern. Der Benutzer mit Bedienerberechtigung darf Mitarbeiter nur anzeigen, weil diesem Benutzer nur Leseberechtigung erteilt wurde.

1. Starten Sie alle Anwendungsserver, in denen Container-Server ausgeführt werden, erneut.
2. Öffnen Sie die Anwendung `EmployeeManagementWeb`. Öffnen Sie in einem Webbrowser `http://<Host>:<Port>/EmployeeManagermentWeb/management.jsp`.
3. Melden Sie sich als Administrator an der Anwendung an. Verwenden Sie den Benutzernamen `admin1` und das Kennwort `admin1`.
4. Versuchen Sie, einen Mitarbeiter anzuzeigen. Klicken Sie auf **Display an Employee**, und suchen Sie die E-Mail-Adresse `authemp1@acme.com`. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer nicht gefunden wurde.
5. Fügen Sie einen Mitarbeiter hinzu. Klicken Sie auf **Add an Employee**. Fügen Sie die E-Mail-Adresse `authemp1@acme.com`, den Vornamen `Joe` und den Nach-

namen Doe hinzu. Klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer hinzugefügt wurde.

6. Melden Sie sich als Benutzer mit Bedienerberechtigung an. Öffnen Sie in einem zweiten Web-Browser-Fenster den URL `http://<Host>:<Port>/EmployeeManagermentWeb/management.jsp`. Verwenden Sie den Benutzernamen `operator1` und das Kennwort `operator1`.
7. Versuchen Sie, einen Mitarbeiter anzuzeigen. Klicken Sie auf **Display an Employee**, und suchen Sie die E-Mail-Adresse `authemp1@acme.com`. Der Mitarbeiter wird angezeigt.
8. Fügen Sie einen Mitarbeiter hinzu. Klicken Sie auf **Add an Employee**. Fügen Sie die E-Mail-Adresse `authemp2@acme.com`, den Vornamen `Joe` und den Nachnamen `Doe` hinzu. Klicken Sie auf **Submit**. Die folgende Nachricht wird angezeigt:

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   Die folgende Ausnahme ist in der Ausnahmekette enthalten:

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Diese Nachricht wird angezeigt, weil der Benutzer `operator1` nicht berechtigt ist, Daten in die Map "Map1" einzufügen.

Wenn Sie mit einer Version von WebSphere Application Server vor Version 7.0.0.11 arbeiten, wird im Container-Server unter Umständen ein Fehler des Typs "java.lang.StackOverflowError" angezeigt. Dieser Fehler ist auf ein Problem mit IBM Developer Kit zurückzuführen. Das Problem ist in dem IBM Developer Kit, das mit WebSphere Application Server Version 7.0.0.11 und höher bereitgestellt wird, behoben.

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Berechtigung konfiguriert, indem Sie bestimmten Benutzern Berechtigungen erteilt haben.

Lerneinheit 4.3: Gruppenbasierte Berechtigung konfigurieren

In der vorherigen Lerneinheit haben Sie Benutzerprincipals in der JAAS-Berechtigungsrichtlinie (Java Authentication and Authorization Service) einzelne benutzerbasierte Berechtigungen zugeordnet. Wenn Sie jedoch Hunderte oder Tausende von Benutzern haben, verwenden Sie die gruppenbasierte Berechtigung, die den Zugriff auf der Basis von Gruppen und nicht auf der Basis einzelner Benutzer autorisiert.

Leider enthält das Subject-Objekt, das über WebSphere Application Server authentifiziert wird, nur einen Benutzerprincipal. Dieses Objekt enthält keinen Gruppenprincipal. Sie können ein angepasstes Anmeldemodul hinzufügen, um den Gruppenprincipal in das Subject-Objekt einzutragen.

In diesem Lernprogramm hat das angepasste Anmeldemodul den Namen com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule. Das Modul ist in der Datei `groupLM.jar` enthalten. Speichern Sie diese JAR-Datei im Verzeichnis `WAS-INSTALL/lib/ext`.

WASAddGroupLoginModule ruft den öffentlichen Gruppenberechtigungsnachweis aus dem Subject-Objekt von WebSphere Application Server ab und erstellt einen Gruppenprincipal com.ibm.websphere.samples.objectgrid.security.WSGroupPrinci-

pal für die Darstellung der Gruppe. Dieser Gruppenprincipal kann dann für die Gruppenberechtigung verwendet werden. Die Gruppen sind in der Datei `xsAuthGroup2.policy` definiert:

```
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";
};
```

Der Principalname ist `WSGroupPrincipal` und stellt die Gruppe dar.

Angepasstes Anmeldemodul hinzufügen:

Das angepasste Anmeldemodul muss jedem der folgenden Systemanmeldemoduleinträge hinzugefügt werden. Wenn Sie Lightweight Third Party Authentication (LTPA) verwenden, fügen Sie den Eintrag den RMI_INBOUND-Anmeldemodulen hinzu. LTPA ist der Standardauthentifizierungsmechanismus für WebSphere Application Server Version 7.0. Für eine Konfiguration von WebSphere Application Server Network Deployment müssen Sie nur die Konfigurationseinträge für den LTPA-Authentifizierungsmechanismus konfigurieren.

Verwenden Sie die folgenden Schritte, um das bereitgestellte Anmeldemodul "com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule" zu konfigurieren:

1. Klicken Sie in der Administrationskonsole auf **Sicherheit** > **Globale Sicherheit** > **Java Authentication and Authorization Service** > **Systemanmeldungen** > *Name_des_Anmeldemoduls* > **JAAS-Anmeldemodule** > **Neu**.
2. Geben Sie als Klassennamen `com.ibm.websphere.sample.xs.security.lm.WASAddGroupLoginModule` ein.
3. Optional: Fügen Sie die Eigenschaft `debug` hinzu, und setzen Sie sie auf den Wert `true`.
4. Klicken Sie auf **Anwenden**, um der Anmeldmodulliste das neue Modul hinzuzufügen.

JAAS-Berechtigungsrichtliniendatei mit JVM-Eigenschaften definieren:

Führen Sie in der Administrationskonsole die folgenden Schritte für die Server "xs1" und "xs2" im Cluster "xsCluster" aus. Wenn eine andere Implementierungstopologie verwendet wird, führen Sie die folgenden Schritte für die Anwendungsserver aus, die die Container-Server hosten.

1. Klicken Sie in der Administrationskonsole auf **Server** > **Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Geben Sie die folgenden generischen JVM-Argumente ein, oder ersetzen Sie den Eintrag `-Djava.security.auth.policy` durch den folgenden Text:

   `-Djava.security.auth.policy=`*`Ausgangsverzeichnis_für_Beispiele`*`/security/xsAuthGroup2.policy`
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Gruppenberechtigung mit der Beispielanwendung testen:

Mit der Beispielanwendung können Sie prüfen, ob die Gruppenberechtigung vom Anmeldemodul konfiguriert wird.

1. Starten Sie die Container-Server erneut. In diesem Lernprogramm sind die Server "xs1" und "xs2" die Container-Server.
2. Melden Sie sich an der Beispielanwendung an. Öffnen Sie in einem Web-Browser den URL `http://<Host>:<Port>/EmployeeManagementWeb/management.jsp`, und melden Sie sich mit dem Benutzernamen `admin1` und dem Kennwort `admin1` an.
3. Mitarbeiter anzeigen. Klicken Sie auf **Display an Employee**, und suchen Sie die E-Mail-Adresse `authemp2@acme.com`. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer nicht gefunden wurde.
4. Mitarbeiter hinzufügen. Klicken Sie auf **Add an Employee**. Fügen Sie die E-Mail-Adresse `authemp2@acme.com`, den Vornamen `Joe` und den Nachnamen `Doe` hinzu. Klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer hinzugefügt wurde.
5. Melden Sie sich als Benutzer mit Bedienerberechtigung an. Öffnen Sie in einem zweiten Web-Browser-Fenster den URL `http://<Host>:<Port>/EmployeeManagermentWeb/management.jsp`. Verwenden Sie den Benutzernamen `operator1` und das Kennwort `operator1`.
6. Versuchen Sie, einen Mitarbeiter anzuzeigen. Klicken Sie auf **Display an Employee**, und suchen Sie die E-Mail-Adresse `authemp2@acme.com`. Der Mitarbeiter wird angezeigt.
7. Mitarbeiter hinzufügen. Klicken Sie auf **Add an Employee**. Fügen Sie die E-Mail-Adresse `authemp3@acme.com`, den Vornamen `Joe` und den Nachnamen `Doe` hinzu. Klicken Sie auf **Submit**. Die folgende Nachricht wird angezeigt:

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   Die folgende Ausnahme ist in der Ausnahmekette enthalten:

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Diese Nachricht wird angezeigt, weil der Benutzer mit Bedieneraufgaben nicht berechtigt ist, Daten in die Map "Map1" einzufügen.

Prüfpunkt der Lerneinheit:

Sie haben Gruppen konfiguriert, um die Zuordnung von Berechtigungen zu den Benutzern Ihrer Anwendung zu vereinfachen.

Modul 5: Tool `xscmd` für die Überwachung von Datengrids und Maps verwenden

Sie können das Tool **`xscmd`** verwenden, um die Größen der primären Datengrids und Maps des Datengrids `Grid` anzuzeigen. Das Tool **`xscmd`** verwendet die MBean, um alle Artefakte des Datengrids, wie z. B. primäre Shards, Replikat-Shards, Container-Server, Map-Größen usw., abzufragen.

In diesem Lernprogramm werden die Container- und Katalogserver in Anwendungsservern von WebSphere Application Server ausgeführt. Die Laufzeitumgebung von WebSphere eXtreme Scale registriert die Managed Beans (MBean) bei dem MBean-Server, der von der Laufzeitumgebung von WebSphere Application Server erstellt wird. Die vom Tool **`xscmd`** verwendete Sicherheit wird von der MBean-Sicherheit von WebSphere Application Server bereitgestellt. Deshalb ist keine spezielle Sicherheitskonfiguration in WebSphere eXtreme Scale erforderlich.

1. Wechseln Sie über ein Befehlszeilentool in das Verzeichnis *DMGR-PROFIL*/bin.
2. Führen Sie das Tool **xscmd** aus.

   Verwenden Sie den Befehl **-c listObjectGridPlacement -sf P**, um die Verteilung der primären Shards aufzulisten. Linux UNIX

   ```
   xscmd.sh -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Windows

   ```
   xscmd.bat -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Bevor Sie die Ausgabe anzeigen können, werden Sie aufgefordert, sich mit Ihrer ID und Ihrem Kennwort bei WebSphere Application Server anzumelden.

Prüfpunkt der Lerneinheit

Sie haben das Tool **xscmd** in WebSphere Application Server verwendet.

Lernprogramm: Sicherheit von WebSphere eXtreme Scale mit einem externen Authentifikator in eine heterogene Umgebung integrieren

Dieses Lernprogramm veranschaulicht, wie Server von WebSphere eXtreme Scale gesichert werden, die teilweise in einer Umgebung von WebSphere Application Server implementiert sind.

In der Implementierung für dieses Lernprogramm sind die Container-Server in WebSphere Application Server implementiert. Der Katalogserver ist als eigenständiger Server implementiert und wird in einer Java-SE-Umgebung (Java Standard Edition) gestartet.

Da der Katalogserver nicht in WebSphere Application Server implementiert ist, können Sie die Authentifizierungs-Plug-ins von WebSphere Application Server nicht verwenden. Weitere Informationen zum Vorgehensweise beim Konfigurieren der Authentifizierungs-Plug-ins von WebSphere Application Server finden Sie unter „Lernprogramm: Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server integrieren” auf Seite 88. In diesem Lernprogramm ist ein anderer Authentifikator für die Katalogserverauthentifizierung erforderlich. Sie konfigurieren einen Keystoreauthentifikator, um die Clients zu authentifizieren.

Lernziele

Im Folgenden sind die Lernziele für dieses Lernprogramm aufgeführt:

- WebSphere eXtreme Scale für die Verwendung des KeyStoreLoginAuthenticator-Plug-ins konfigurieren
- Transportsicherheit von WebSphere eXtreme Scale für die Verwendung der CSIv2-Konfiguration von WebSphere Application Server und der Eigenschaftendatei von WebSphere eXtreme Scale konfigurieren
- JAAS-Berechtigung (Java Authentication and Authorization Service) in WebSphere Application Server verwenden
- Dienstprogramm **xscmd** für die Überwachung der Datengrids und Maps, die Sie im Lernprogramm erstellt haben, verwenden

Erforderliche Zeit

Das Durcharbeiten dieses Lernprogramms dauert insgesamt ungefähr 4 Stunden.

Einführung: Sicherheit in einer heterogenen Umgebung

In diesem Lernprogramm integrieren Sie die Sicherheit von WebSphere eXtreme Scale in einer heterogenen Umgebung. Die Container-Server werden in WebSphere Application Server ausgeführt, und der Katalogservice wird im eigenständigen Modus ausgeführt. Da der Katalogserver im eigenständigen Modus ausgeführt wird, müssen Sie einen externen Authentifikator konfigurieren.

Wichtig: Wenn Ihre Container-Server und Ihr Katalogserver in WebSphere Application Server ausgeführt werden, können Sie die Authentifizierungs-Plug-ins von WebSphere Application Server oder einen externen Authentifikator verwenden. Weitere Informationen zur Verwendung der Authentifizierungs-Plug-ins von WebSphere Application Server finden Sie unter „Lernprogramm: Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server integrieren" auf Seite 88.

Lernziele

Im Folgenden sind die Lernziele für dieses Lernprogramm aufgeführt:

- WebSphere eXtreme Scale für die Verwendung des KeyStoreLoginAuthenticator-Plug-ins konfigurieren
- Transportsicherheit von WebSphere eXtreme Scale für die Verwendung der CSIv2-Konfiguration von WebSphere Application Server und der Eigenschaftendatei von WebSphere eXtreme Scale konfigurieren
- JAAS-Berechtigung (Java Authentication and Authorization Service) in WebSphere Application Server verwenden
- Dienstprogramm **xscmd** für die Überwachung der Datengrids und Maps, die Sie im Lernprogramm erstellt haben, verwenden

Erforderliche Zeit

Das Durcharbeiten dieses Lernprogramms dauert insgesamt ungefähr 4 Stunden.

Kenntnisstufe

Fortgeschrittener Anfänger.

Zielgruppe

Entwickler und Administratoren, die an der Sicherheitsintegration zwischen WebSphere eXtreme Scale und WebSphere Application Server interessiert sind und externe Authentifikatoren konfigurieren.

Systemvoraussetzungen

- WebSphere Application Server Version 6.1 oder Version 7.0.0.11 oder höher mit den folgenden angewendeten Fixes: vorläufiger Fix PM20613 und vorläufiger Fix PM15818.
- Der Katalogserver muss in einer eigenständigen Installation und nicht in einer Installation, die mit WebSphere Application Server integriert ist, ausgeführt werden.
- Aktualisieren Sie die Java Runtime, um den folgenden Fix anzuwenden: IZ79819: IBMJDK FAILS TO READ PRINCIPAL STATEMENT WITH WHITESPACE FROM SECURITY FILE
- Der eigenständige Knoten, auf dem der Katalogservice ausgeführt wird, muss IBM Software Development Kit Version 1.6 J9 verwenden. Dieses Software Deve-

lopment Kit ist in der Installation von WebSphere Application Server enthalten. Der Katalogserverknoten muss eine eigenständige Installation sein, weil Sie den Befehl **startOgServer** nicht in einer Installation von WebSphere eXtreme Scale in WebSphere Application Server ausführen können.

In diesem Lernprogramm werden vier Anwendungsserver und ein Deployment Manager von WebSphere Application Server verwendet, um das Beispiel zu veranschaulichen.

Voraussetzungen

Ein grundlegendes Verständnis der folgenden Punkte ist hilfreich, bevor Sie mit diesem Lernprogramm beginnen:

- Programmiermodell von WebSphere eXtreme Scale
- Grundlegende Sicherheitskonzepte von WebSphere eXtreme Scale
- Grundlegende Sicherheitskonzepte von WebSphere Application Server

Hintergrundinformationen zur Sicherheitsintegration von WebSphere eXtreme Scale und WebSphere Application Server finden Sie unter „Integration der Sicherheit mit WebSphere Application Server" auf Seite 533.

Modul 1: Heterogene Umgebung von WebSphere Application Server und eigenständige Umgebung vorbereiten

Bevor Sie mit dem Lernprogramm beginnen, müssen Sie eine Basistopologie erstellen, die Container-Server enthält, die in WebSphere Application Server ausgeführt werden. In diesem Lernprogramm werden die Katalogserver im eigenständigen Modus ausgeführt.

Lernziele

Anhand der Übungen in diesem Modul lernen Sie Folgendes:

- Verständnis der heterogenen Topologie und der Dateien, die für das Lernprogramm erforderlich sind
- WebSphere Application Server für die Ausführung der Container-Server konfigurieren

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Lerneinheit 1.1: Topologie verstehen und Lernprogrammdateien abrufen

Um die Umgebung für das Lernprogramm vorzubereiten, müssen Sie die Katalog- und Container-Server für die Topologie konfigurieren.

Diese Lerneinheit führt Sie durch die Beispieltopologie und Anwendungen, die im Lernprogramm verwendet werden. Um mit der Ausführung des Lernprogramms zu beginnen, müssen Sie die Anwendungen herunterladen und die Konfigurationsdateien an den richtigen Positionen für Ihre Umgebung ablegen. Sie können die Beispielanwendung aus dem Wiki zu WebSphere eXtreme Scale herunterladen.

Topologie: In diesem Lernprogramm erstellen Sie die folgenden Cluster in der Zelle von WebSphere Application Server:

- **Cluster appCluster**: Hostet die Beispielunternehmensanwendung "EmployeeManagement". Dieser Cluster hat zwei Anwendungsserver: s1 und s2.
- **Cluster xsCluster**: Hostet die Container-Server von eXtreme Scale. Dieser Cluster hat zwei Anwendungsserver: xs1 und xs2.

In dieser Implementierungstopologie sind die Anwendungsserver "s1" und "s2" die Client-Server, die auf Daten zugreifen, die im Datengrid gespeichert werden. Die Server "xs1" und "xs2" sind die Container-Server, die das Datengrid hosten.

Alternative Konfiguration: Sie können alle Anwendungsserver in einem einzigen Cluster, z. B. im Cluster "appCluster", hosten. Bei dieser Konfiguration sind alle Server im Cluster sowohl Clients als auch Container-Server. In diesem Lernprogramm werden zwei Cluster verwendet, um eine Unterscheidung zwischen den Anwendungsservern zu treffen, die die Clients und Container-Server hosten.

In diesem Lernprogramm konfigurieren Sie eine Katalogservicedomäne, die sich aus einem fernen Server zusammensetzt, der nicht in der Zelle von WebSphere Application Server enthalten ist. Diese Konfiguration ist nicht die Standardkonfiguration und führt dazu, dass die Katalogserver im Deployment Manager und andere Prozesse in der Zelle von WebSphere Application Server ausgeführt werden. Weitere Informationen zum Erstellen einer Katalogservicedomäne, die sich aus fernen Servern zusammensetzt, finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen” auf Seite 264.

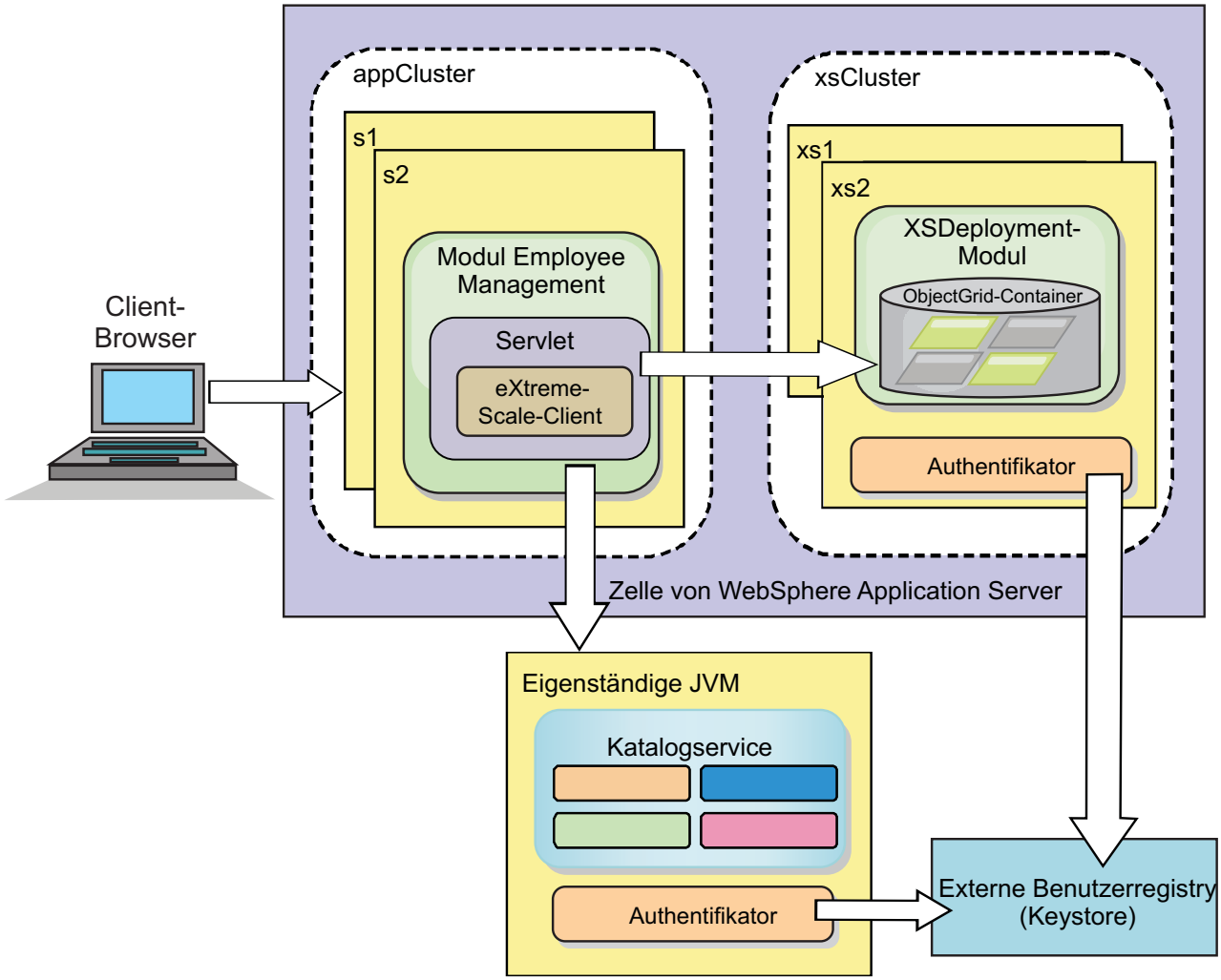


Abbildung 20. Lernprogrammtopologie

Anwendungen: In diesem Lernprogramm verwenden Sie zwei Anwendungen und eine gemeinsam genutzte Bibliotheksdatei:

- **`EmployeeManagement.ear`**: Die Anwendung `EmployeeManagement.ear` ist eine vereinfachte J2EE-Unternehmensanwendung (Java 2 Platform, Enterprise Edition). Sie enthält ein Webmodul für die Verwaltung der Mitarbeiterprofile. Das Webmodul enthält die Datei `management.jsp`, die Mitarbeiterprofile anzeigt, einfügt, aktualisiert und löscht, die in den Container-Servern gespeichert sind.
- **`XSDeployment.ear`**: Diese Anwendung enthält ein Unternehmensanwendungsmodul ohne Anwendungsartefakte. Die Cacheobjekte sind in der Datei `EmployeeData.jar` gepackt. Die Datei `EmployeeData.jar` ist als gemeinsam genutzte Bibliothek für die Datei `XSDeployment.ear` implementiert, so dass die Datei `XSDeployment.ear` auf die Klassen zugreifen kann. Der Zweck dieser Anwendung ist, die Konfigurationsdatei und die Eigenschaftsdatei von eXtreme Scale zu packen. Wenn diese Unternehmensanwendung gestartet wird, werden die Konfigurationsdateien von eXtreme Scale automatisch von der Laufzeitumgebung von eXtreme Scale erkannt, sodass die Container-Server erstellt werden. Diese Konfigurationsdateien enthalten die Dateien `objectGrid.xml` und `objectGridDeployment.xml`.
- **`EmployeeData.jar`**: Diese JAR-Datei enthält eine einzige Klasse: com.ibm.websphere.sample.xs.data.EmployeeData. Diese Klasse stellt Mitarbeiterdaten dar, die

im Grid gespeichert sind. Diese JAR-Datei wird mit den Dateien `EmployeeManagement.ear` und `XSDeployment.ear` als gemeinsam genutzte Bibliothek implementiert.

Lernprogrammdateien abrufen:

1. Laden Sie die Dateien `WASSecurity.zip` und `security_extauth.zip` aus dem Wiki von WebSphere eXtreme Scale herunter.
2. Entpacken Sie die Datei `WASSecurity.zip` in einem Verzeichnis, wie z. B. `wxs_samples/`, um die Binärdateien und Quellenartefakte anzuzeigen. Im verbleibenden Teil des Lernprogramms wird *Ausgangsverzeichnis_für_Beispiele* für dieses Verzeichnis verwendet. In der Datei `README.txt` des Pakets finden Sie eine Beschreibung des Inhalts und Informationen zum Laden der Quelle in den Eclipse-Arbeitsbereich. Die folgenden ObjectGrid-Konfigurationsdateien befinden sich im Verzeichnis `META-INF`:
   - `objectGrid.xml`
   - `objectGridDeployment.xml`
3. Erstellen Sie ein Verzeichnis zum Speichern der Eigenschaftendateien, die zum Sichern dieser Umgebung verwendet werden. Sie könnten beispielsweise das Verzeichnis `/opt/wxs/security` verwenden.
4. Entpacken Sie die Datei `security_extauth.zip` im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*. Die Datei `security_extauth.zip` enthält die folgenden Sicherheitskonfigurationsdateien, die in diesem Lernprogramm verwendet werden:
   - `catServer3.props`
   - `server3.props`
   - `client3.props`
   - `security3.xml`
   - `xsAuth3.props`
   - `xsjaas3.config`
   - `sampleKS3.jks`

Informationen zu den Konfigurationsdateien:

Die Dateien `objectGrid.xml` und `objectGridDeployment.xml` erstellen die Datengrids und Maps, in denen die Anwendungsdaten gespeichert werden.

Diese Konfigurationsdateien müssen `objectGrid.xml` und `objectGridDeployment.xml` genannt werden. Wenn der Anwendungsserver gestartet wird, erkennt eXtreme Scale diese Dateien im Verzeichnis `META-INF` der EJB- und Webmodule. Werden diese Dateien gefunden, wird angenommen, dass die Java Virtual Machine (JVM) als Container-Server für die in den Konfigurationsdateien definierten Datengrids dient.

Datei `objectGrid.xml`

Die Datei `objectGrid.xml` definiert ein ObjectGrid mit dem Namen `Grid`. Das Datengrid `Grid` enthält eine einzige Map mit dem Namen `Map1`, in der das Mitarbeiterprofil für die Anwendung gespeichert wird.

```
<?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>
```

Datei objectGridDeployment.xml

Die Datei objectGridDeployment.xml gibt an, wie das Datengrid Grid implementiert wird. Wenn das Grid implementiert wird, hat es fünf Partitionen und ein synchrones Replikat.

```
<?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>
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie sich mit der Topologie für das Lernprogramm vertraut gemacht und die Konfigurationsdateien und Beispielanwendungen in Ihrer Umgebung hinzugefügt.

Lerneinheit 1.2: Umgebung von WebSphere Application Server konfigurieren

Um die Umgebung für das Lernprogramm vorzubereiten, müssen Sie die Sicherheit von WebSphere Application Server konfigurieren. Aktivieren Sie die Verwaltungs- und Anwendungssicherheit mit internen dateibasierten eingebundenen Repositorys als Benutzer-Account-Registry. Anschließend können Sie Server-Cluster zum Hosten der Clientanwendung und der Container-Server erstellen. Außerdem müssen Sie die Katalogserver erstellen und starten.

Die folgenden Schritte wurden mit WebSphere Application Server Version 7.0 geschrieben. Sie können die Konzepte aber auch auf frühere Versionen von WebSphere Application Server anwenden.

Sicherheit von WebSphere Application Server konfigurieren:

Erstellen und erweitern Sie Profile für den Deployment Manager und die Knoten mit WebSphere eXtreme Scale. Weitere Informationen finden Sie im Abschnitt „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren” auf Seite 167.

Konfigurieren Sie die Sicherheit von WebSphere Application Server.

1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Sicherheit** > **Globale Sicherheit**.
2. Wählen Sie **Eingebundene Repositorys** als **Benutzer-Account-Repository** aus. Klicken Sie auf **Als aktuelles Repository festlegen**.
3. Klicken Sie auf **Konfigurieren...**, um die Anzeige **Eingebundene Repositorys** aufzurufen.
4. Geben Sie einen Wert im Feld **Name des primären Benutzers mit Verwaltungsaufgaben** ein, z. B. admin. Klicken Sie auf **Anwenden**.

5. Geben Sie auf Anforderung das Kennwort des Benutzers mit Verwaltungsaufgaben ein, und klicken Sie auf **OK**. Speichern Sie Ihre Änderungen.
6. Vergewissern Sie sich, dass die Einstellung **Eingebundene Repositorys** auf der Seite **Globale Sicherheit** auf die aktuelle Benutzer-Account-Registry gesetzt ist.
7. Wählen Sie die folgenden Einträge aus: **Verwaltungssicherheit aktivieren**, **Anwendungssicherheit aktivieren** und **Java-2-Sicherheit verwenden, um den Anwendungszugriff auf lokale Ressourcen zu beschränken**. Klicken Sie auf **Anwenden**, und speichern Sie Ihre Änderungen.
8. Starten Sie den Deployment Manager und alle aktiven Anwendungsserver erneut.

Die Verwaltungssicherheit von WebSphere Application Server wird über die internen dateibasierten eingebundenen Repositorys als Benutzer-Account-Registry aktiviert.

Server-Cluster erstellen:

Erstellen Sie zwei Server-Cluster in Ihrer Konfiguration von WebSphere Application Server, den Cluster `appCluster` zum Hosten der Beispielanwendung für das Lernprogramm und den Cluster `xsCluster` zum Hosten des Datengrids.

1. Öffnen Sie in der Administrationskonsole von WebSphere Application Server die Anzeige "Cluster". Klicken Sie auf **Server** > **Cluster** > **WebSphere-Anwendungsservercluster** > **Neu**.
2. Geben Sie `appCluster` als Clusternamen ein, lassen Sie die Option **Prefer local** ausgewählt, und klicken Sie auf **Weiter**.
3. Erstellen Sie Server im Cluster. Erstellen Sie einen Server mit dem Namen `s1`, und übernehmen Sie dabei die Standardoptionen. Fügen Sie ein weiteres Cluster-Member mit dem Namen `s2` hinzu.
4. Führen Sie die verbleibenden Schritte im Assistenten aus, um den Cluster zu erstellen. Speichern Sie die Änderungen.
5. Wiederholen Sie diese Schritte, um den Cluster `xsCluster` zu erstellen. Dieser Cluster hat zwei Server, `xs1` und `xs2`.

Katalogservicedomäne erstellen:

Nach der Konfiguration der Server-Cluster und der Sicherheit müssen Sie festlegen, wo die Katalogserver gestartet werden sollen.

Katalogservicedomäne in WebSphere eXtreme Scale definieren

1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen**.
2. Erstellen Sie die Katalogservicedomäne. Klicken Sie auf **Neu**. Erstellen Sie die Katalogservicedomäne mit dem Namen `catalogService1`, und legen Sie diese Katalogservicedomäne als Standarddomäne fest.
3. Fügen Sie der Katalogservicedomäne ferne Server hinzu. Wählen Sie **Ferner Server** aus. Geben Sie den Namen des Hosts an, auf dem der Katalogserver ausgeführt wird. Verwenden Sie für dieses Beispiel den Listener-Port-Wert `16809`.
4. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Prüfpunkt der Lerneinheit:

Sie haben die Sicherheit in WebSphere Application Server aktiviert und die Servertopologie für WebSphere eXtreme Scale erstellt.

Modul 2: eXtreme-Scale-Authentifizierung in einer heterogenen Umgebung konfigurieren

Durch die Konfiguration der Authentifizierung können Sie die Identität des Anfordernden zuverlässig bestimmen. WebSphere eXtreme Scale unterstützt Client/Server- und Server/Server-Authentifizierung.

Authentifizierungsablauf

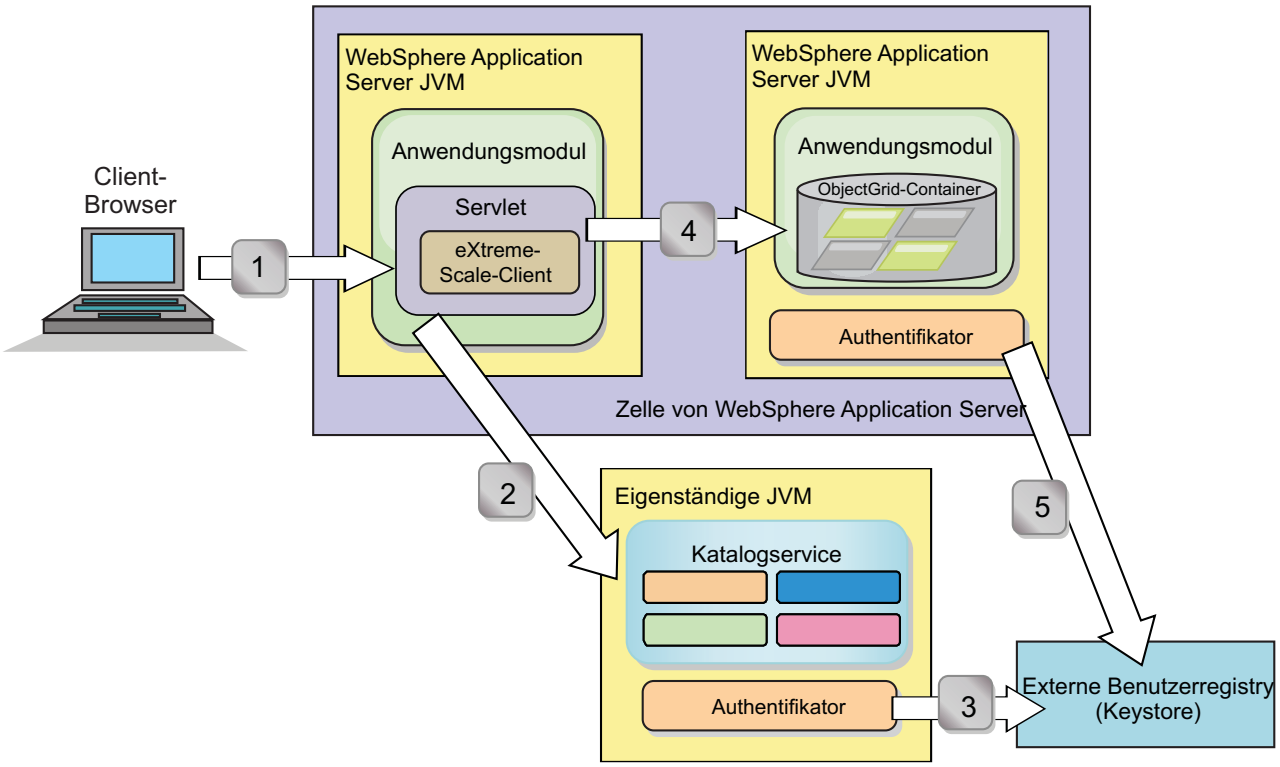


Abbildung 21. Authentifizierungsablauf

Das vorherige Diagramm zeigt zwei Anwendungsserver. Der erste Anwendungsserver hostet die Webanwendung, die auch ein Client von WebSphere eXtreme Scale ist. Der zweite Anwendungsserver hostet einen Container-Server. Der Katalogserver wird in einer eigenständigen Java Virtual Machine (JVM) und nicht in WebSphere Application Server ausgeführt.

Die mit Zahlen markierten Pfeile im Diagramm veranschaulichen den Authentifizierungsablauf:

1. Der Benutzer der Unternehmensanwendung greift auf den Web-Browser zu und meldet sich mit einem Benutzernamen und einem Kennwort beim ersten Anwendungsserver an. Der erste Anwendungsserver sendet den Benutzernamen und das Kennwort des Clients zur Authentifizierung in der Benutzerregistry an die Sicherheitsinfrastruktur. Diese Benutzerregistry ist ein Keystore. Deshalb sind die Sicherheitsinformationen im Thread von WebSphere Application Server gespeichert.
2. Die JSP-Datei (JavaServer Pages) tritt als eXtreme-Scale-Client auf, um die Sicherheitsinformationen aus der Clienteigenschaftendatei abzurufen. Die JSP-Anwendung, die als Client von WebSphere eXtreme Scale auftritt, sendet den Sicherheitsberechtigungsnachweis des Clients von WebSphere eXtreme Scale zusammen mit der Anforderung an den Katalogserver. Das Senden des Sicher-

heitsberechtigungsnachweises zusammen mit der Anforderung wird als *RunAs*-Modell betrachtet. In einem RunAs-Modell wird der Web-Browser-Client als Client von WebSphere eXtreme Scale ausgeführt, um auf die im Container-Server gespeicherten Daten zuzugreifen. Der Client verwendet einen JVM-weiten Clientberechtigungsnachweis für die Herstellung der Verbindung zu den eXtreme-Scale-Servern. Die Verwendung des RunAs-Modells gleicht dem Herstellen einer Verbindung zu einer Datenbank mit einer Benutzer-ID und einem Kennwort auf Datenquellenebene.

3. Der Katalogserver empfängt den eXtreme-Scale-Clientberechtigungsnachweis, der die Sicherheitstoken von WebSphere Application Server enthält. Anschließend ruft der Katalogserver das Authentifikator-Plug-in für die Authentifizierung des Clientberechtigungsnachweises auf. Der Authentifikator stellt die Verbindung zur externen Benutzerregistry her und sendet den Clientberechtigungsnachweis zur Authentifizierung an die Benutzerregistry.
4. Der Client sendet die Benutzer-ID und das Kennwort an den Container-Server, der vom Anwendungsserver gehostet wird.
5. Der im Anwendungsserver gehostete Container-Service empfängt den eXtreme-Scale-Clientberechtigungsnachweis, der aus einer Benutzer-ID und einem Kennwort besteht. Anschließend ruft der Container-Server das Authentifikator-Plug-in zur Authentifizierung des Clientberechtigungsnachweises auf. Der Authentifikator stellt die Verbindung zur Benutzerregistry des Keystores her und sendet den Clientberechtigungsnachweis zur Authentifizierung an die Benutzerregistry.

Lernziele

Anhand der Übungen in diesem Modul lernen Sie Folgendes:

- eXtreme-Scale-Clientsicherheit konfigurieren
- eXtreme-Scale-Katalagserversicherheit konfigurieren
- eXtreme-Scale-Container-Server-Sicherheit konfigurieren
- Installieren Sie die Beispielanwendung, und führen Sie sie aus.

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Lerneinheit 2.1: Clientsicherheit in WebSphere eXtreme Scale konfigurieren

Sie konfigurieren die Clienteigenschaften mit einer Eigenschaftendatei. Die Clienteigenschaftendatei gibt die zu verwendende CredentialGenerator-Implementierungsklasse an.

Inhalt der Clienteigenschaftendatei:

In diesem Lernprogramm werden Sicherheitstoken von WebSphere Application Server als Clientberechtigungsnachweis verwendet. Das Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/`security_extauth` enthält die Datei `client3.props`.

Die Datei `client3.props` enthält die folgenden Einstellungen:

securityEnabled
: Aktiviert die Sicherheit für eXtreme-Scale-Clients. Wenn diese Eigenschaft auf `true` gesetzt wird, muss der Client verfügbare Sicherheitsinformationen an den Server senden.

credentialAuthentication
Gibt die Unterstützung für die Authentifizierung von Clientberechtigungsnachweisen an. Wenn Sie diese Eigenschaft auf `Supported` setzen, unterstützt der Client die Authentifizierung der Berechtigungsnachweise.

credentialGeneratorClass
Gibt den Namen der Klasse an, die die Schnittstelle "com.ibm.websphere-.objectgrid.security.plugins.CredentialGenerator" implementiert. Die Eigenschaft wird auf die Klasse com.ibm.websphere.objectgrid.security.plugins-.builtins. UserPasswordCredentialGenerator gesetzt, damit der Client die Sicherheitsinformationen aus der Klasse UserPasswordCredentialGenerator abruft.

credentialGeneratorProps
Gibt den Benutzernamen und das Kennwort an: `manager manager1`. Der Benutzername ist `manager`, und das Kennwort ist `manager1`. Zum Codieren dieser Eigenschaft mit einem exklusiven oder (xor)-Algorithmus können Sie auch den Befehl **`FilePasswordEncoder.bat|sh`** verwenden.

Clienteigenschaftendatei mit JVM-Eigenschaften definieren:

Führen Sie in der Administrationskonsole die folgenden Schritte für die Server `s1` und `s2` im Cluster `appCluster` aus. Wenn Sie eine andere Topologie verwenden, führen Sie die folgenden Schritte für alle Anwendungsserver aus, in denen die Anwendung `EmployeeManagement` implementiert ist.

1. Klicken Sie auf **Server** > **WebSphere-Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Erstellen Sie die folgende generische JVM-Eigenschaft, um die Position der Clienteigenschaftendatei festzulegen:
   `-Dobjectgrid.client.props=`*`Ausgangsverzeichnis_für_Beispiele`*`/security_extauth/client3.props`
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Prüfpunkt der Lerneinheit:

Sie haben die Clienteigenschaftendatei bearbeitet und die Server im Cluster `appCluster` für die Verwendung der Clienteigenschaftendatei konfiguriert. Diese Eigenschaftendatei gibt die zu verwendende CredentialGenerator-Implementierungsklasse an.

Lerneinheit 2.2: Sicherheit des Katalogservers konfigurieren

Ein Katalogserver enthält zwei verschiedene Stufen von Sicherheitsinformationen: Die erste Stufe enthält die Sicherheitseigenschaften, für alle Server von WebSphere eXtreme Scale, einschließlich des Katalogservice und der Container-Server, gelten. Die zweite Stufe enthält die Sicherheitseigenschaften, die speziell für den Katalogserver gelten.

Die Sicherheitseigenschaften, die für die Katalogserver und Container-Server gelten, werden in der XML-Sicherheitsdeskriptordatei konfiguriert. Ein Beispiel für allgemeine Eigenschaften ist die Authentifikatorkonfiguration, die die Benutzerregistry und den Authentifizierungsmechanismus darstellt. Weitere Informationen zu den Sicherheitseigenschaften finden Sie unter XML-Sicherheitsdeskriptordatei.

Zum Konfigurieren der XML-Sicherheitsdeskriptordatei in einer Java-SE-Umgebung verwenden Sie die Option **`-clusterSecurityFile`**, wenn Sie den Befehl **`startOgServer`** ausführen. Geben Sie einen Wert in einem Dateiformat an, z. B. *Ausgangsverzeichnis_für_Beispiele*`/security_extauth/security3.xml`.

Datei security3.xml:

In diesem Lernprogramm befindet sich die Datei security3.xml im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/security_extauth. Im Folgenden sehen Sie den Inhalt der Datei security3.xml, im dem die Kommentare entfernt wurden:

```
<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>
```

Die folgenden Eigenschaften sind in der Datei security3.xml definiert:

securityEnabled
: Die Eigenschaft "securityEnabled" ist auf true gesetzt. Dies zeigt dem Katalogserver an, dass die globale Sicherheit von WebSphere eXtreme Scale aktiviert ist.

authenticator
: Der Authentifikator ist als Klasse com.ibm.websphere.objectgrid.security.plugins.builtins. KeyStoreLoginAuthenticator konfiguriert. Bei dieser integrierten Implementierung des Authenticator-Plug-ins werden die Benutzer-ID und das Kennwort übergeben, um sicherzustellen, dass sie in der Keystore-Datei konfiguriert sind. Die Klasse KeyStoreLoginAuthenticator verwendet einen Alias für das KeyStoreLogin-Anmeldemodul. Deshalb ist eine JAAS-Anmeldekonfiguration (Java Authentication and Authorization Service) erforderlich.

Datei catServer3.props:

In der Servereigenschaftendatei werden die serverspezifischen Eigenschaften gespeichert, zu denen auch die serverspezifischen Sicherheitseigenschaften gehören. Weitere Informationen finden Sie unter Servereigenschaftendatei. Sie können die Option **-serverProps** verwenden, um die Katalogservereigenschaft bei der Ausführung des Befehls **startOgServer** anzugeben. Für dieses Lernprogramm befindet sich eine Datei catServer3.props im Verzeichnis c. Im Folgenden sehen Sie den Inhalt der Datei catServer3.props, in dem die Kommentare entfernt wurden:

```
securityEnabled=true
credentialAuthentication=Required
transportType=TCP/IP
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
: Die Eigenschaft "securityEnabled" ist auf true gesetzt. Damit wird angezeigt, dass dieser Katalogserver ein sicherer Server ist.

credentialAuthentication
: Die Eigenschaft "credentialAuthentication" wird auf Required gesetzt, so dass jeder Client, der eine Verbindung zum Server herstellt, einen Berechtigungsnachweis bereitstellen muss. In der Clienteigenschaftendatei ist die Eigenschaft "credentialAuthentication" auf den Wert Supported gesetzt, damit der Server die vom Client gesendeten Berechtigungsnachweise empfängt.

secureTokenManagerType
: Die Eigenschaft "secureTokenManagerType" wird auf none gesetzt, um an-

zuzeigen, dass der geheime Schlüssel für die Authentifizierung nicht verschlüsselt ist, wenn die vorhandenen Server verbunden werden.

authenticationSecret
Die Eigenschaft "authenticationSecret" wird auf `ObjectGridDefaultSecret` gesetzt. Diese geheime Zeichenfolge wird für die Einbindung in den eXtreme-Scale-Server-Cluster verwendet. Wenn ein Server dem Datengrid beitritt, wird er aufgefordert, diese Shared-Secret-Zeichenfolge vorzulegen. Wenn die geheime Zeichenfolge des einzubindenden Servers mit der Zeichenfolge im Katalogserver übereinstimmt, wird der Server akzeptiert. Stimmen die Zeichenfolgen nicht überein, wird die Anforderung zur Einbindung des Servers zurückgewiesen.

transportType
Die Eigenschaft "transportType" wird zunächst auf `TCP/IP` gesetzt. Später in diesem Lernprogramm wird die Transportsicherheit aktiviert.

Datei `xsjaas3.config`:

Da die KeyStoreLoginAuthenticator-Implementierung ein Anmeldemodul verwendet, müssen Sie das Anmeldemodell mit einer Anmeldekonfigurationsdatei für die JAAS-Authentifizierung konfigurieren. Im Folgenden sehen Sie den Inhalt der Datei `xsjaas3.config`:

```
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
     keyStoreFile="Ausgangsverzeichnis_für_Beispiele/security_extauth/sampleKS3.jks" debug = true;
};
```

Wenn Sie für *Ausgangsverzeichnis_für_Beispiele* eine andere Position als `/wxs_samples/` verwendet haben, müssen Sie die Position der Keystore-Datei aktualisieren. Diese Anmeldekonfiguration zeigt an, dass das Modul "com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule" als Anmeldemodul verwendet wird. Als Keystore-Datei ist die Datei `sampleKS3.jks` definiert.

In der Beispiel-Keystore-Datei `sampleKS3.jks` sind zwei Benutzer-IDs und die zugehörigen Kennwörter gespeichert: manager/manager1 und cashier/cashier1.

Sie können die folgenden **`keytool`**-Befehle verwenden, um diesen Keystore zu erstellen:

- `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`

Katalogserver mit aktivierter Sicherheit starten:

Zum Starten des Katalogservers setzen Sie den Befehl **`startOgServer`** mit den Parametern **`-clusterFile`** und **`-serverProps`** ab, um die Sicherheitseigenschaften zu übergeben.

Verwenden Sie eine eigenständige Installation von WebSphere eXtreme Scale, um den Katalogserver auszuführen. Wenn Sie das eigenständige Installationsimage verwenden, müssen Sie das IBM SDK verwenden. Sie können das mit WebSphere Application Server bereitgestellte SDK verwenden, indem Sie die Variable *JA-*

VA_HOME so setzen, dass sie auf das IBM SDK verweist, z. B. `set JAVA_HOME=`*`WAS-Stammverzeichnis`*`/IBM/WebSphere/AppServer/java/`.

1. Wechseln Sie in das Verzeichnis `bin`.

   ```
   cd WXS-Ausgangsverzeichnis/bin
   ```

2. Führen Sie den Befehl **startOgServer** aus.

   Linux UNIX

   ```
   ./startOgServer.sh cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
   cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile Ausgangsverzeichnis_für_Beispiele/security_extauth/security3.xml
   -serverProps Ausgangsverzeichnis_für_Beispiele/security_extauth/catServer3.props -jvmArgs
   -Djava.security.auth.login.config="Ausgangsverzeichnis_für_Beispiele/security_extauth/xsjaas3.config"
   ```

   Windows

   ```
   startOgServer.bat cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
   cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile Ausgangsverzeichnis_für_Beispiele/security_extauth/security3.xml
   -serverProps Ausgangsverzeichnis_für_Beispiele/security_extauth/catServer3.props -jvmArgs
   -Djava.security.auth.login.config="Ausgangsverzeichnis_für_Beispiele/security_extauth/xsjaas3.config"
   ```

Nach der Ausführung des Befehls **startOgServer** wird ein sicherer Server mit dem Listener-Port 16809, dem Client-Port 16601, dem Peer-Port 16602 und dem JMX-Port 16099 gestartet. Wenn ein Portkonflikt vorliegt, ändern Sie die Portnummer in eine Portnummer, die nicht im Gebrauch ist.

Katalogserver stoppen, in dem die Sicherheit aktiviert ist:

Sie können den Befehl **stopOgServer** verwenden, um den Katalogserver zu stoppen.

1. Wechseln Sie in das Verzeichnis `bin`.

   ```
   cd WXS-Ausgangsverzeichnis/bin
   ```

2. Führen Sie den Befehl **stopOgServer** aus. Linux UNIX

   ```
   stopOgServer.sh cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   Ausgangsverzeichnis_für_Beispiele/security_extauth/client3.props
   ```

   Windows

   ```
   stopOgServer.bat cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   Ausgangsverzeichnis_für_Beispiele/security_extauth/client3.props
   ```

Prüfpunkt der Lerneinheit:

Sie haben die Sicherheit des Katalogservers konfiguriert, indem Sie dem Katalogservice die Dateien `security3.xml`, `catServer3.props` und `xsjaas3.config` zugeordnet haben.

Lerneinheit 2.3: Sicherheit des Container-Servers konfigurieren

Wenn ein Container-Server eine Verbindung zum Katalogservice herstellt, ruft der Container-Server alle Sicherheitskonfigurationen ab, die in der ObjectGrid-XML-Sicherheitsdatei konfiguriert sind. Die ObjectGrid-XML-Sicherheitsdatei definiert die Authentifikatorkonfiguration, das Zeitlimit für Anmeldesitzungen und weitere Konfigurationsinformationen. Außerdem hat ein Container-Server eigene serverspezifische Sicherheitseigenschaften in der Servereigenschaftendatei.

Konfigurieren Sie die Servereigenschaftendatei mit der JVM-Eigenschaft "-Dobjectgrid.server.props". Der Dateiname, der mit dieser Eigenschaft angegeben wird, ist ein absoluter Dateipfad, wie z. B. *Ausgangsverzeichnis_für_Beispiele*`/security_extauth/server3.props`.

In diesem Lernprogramm sind die Container-Server in den Servern `xs1` und `xs2` im Cluster `xsCluster` gehostet.

Datei server3.props:

Die Datei `server3.props` befindet sich im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*`/security_extauth/`. Im Folgenden sehen Sie den Inhalt der Datei `server3.props`:

```
securityEnabled=true
credentialAuthentication=Required
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
Die Eigenschaft "securityEnabled" wird auf `true` gesetzt, um Container-Server als sicheren Server zu kennzeichnen.

credentialAuthentication
Die Eigenschaft "credentialAuthentication" wird auf `Required` gesetzt, so dass jeder Client, der eine Verbindung zum Server herstellt, einen Berechtigungsnachweis bereitstellen muss. In der Clienteigenschaftendatei wird die Eigenschaft "credentialAuthentication" auf `Supported` gesetzt, damit der Server den vom Client gesendeten Berechtigungsnachweis empfängt.

secureTokenManagerType
Die Eigenschaft "secureTokenManagerType" wird auf `none` gesetzt, um anzuzeigen, dass der geheime Schlüssel für die Authentifizierung nicht verschlüsselt ist, wenn die vorhandenen Server verbunden werden.

authenticationSecret
Die Eigenschaft "authenticationSecret" wird auf `ObjectGridDefaultSecret` gesetzt. Diese geheime Zeichenfolge wird für die Einbindung in den eXtreme-Scale-Server-Cluster verwendet. Wenn ein Server in das Datengrid eingebunden wird, wird er aufgefordert, diese Shared-Secret-Zeichenfolge vorzulegen. Wenn die geheime Zeichenfolge des einzubindenden Servers mit der Zeichenfolge im Katalogserver übereinstimmt, wird der Server akzeptiert. Stimmen die Zeichenfolgen nicht überein, wird die Anforderung zur Einbindung des Servers zurückgewiesen.

Servereigenschaftendatei mit JVM-Eigenschaften definieren:

Definieren Sie die Servereigenschaftendatei in den Servern "xs1" und "xs2". Wenn Sie die Topologie für dieses Lernprogramm nicht verwenden, definieren Sie die Servereigenschaftendatei in allen Anwendungsservern, die Sie zum Hosten von Container-Servern verwenden.

1. Öffnen Sie die Seite "Java Virtual Machine" für den Server. Klicken Sie auf **Server** > **WebSphere-Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Fügen Sie das folgende generische JVM-Argument hinzu:
   `-Dobjectgrid.server.props=`*Ausgangsverzeichnis_für_Beispiele*`/security_extauth/server3.props`
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Angepasstes Anmeldemodul hinzufügen:

Der Container-Server verwendet dieselbe KeyStoreAuthenticator-Implementierung wie der Katalogserver. Die KeyStoreAuthenticator-Implementierung verwendet einen **`KeyStoreLogin`**-Anmeldemodulalias. Deshalb müssen Sie den Modelleinträgen für die Anwendungsanmeldung ein angepasstes Anmeldemodul hinzufügen.

1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Sicherheit** > **Globale Sicherheit** > **Java Authentication and Authorization Service**.
2. Klicken Sie auf **Anwendungsanmeldungen**.
3. Klicken Sie auf **Neu**, und fügen Sie einen `KeyStoreLogin`-Alias hinzu. Klicken Sie auf **Anwenden**.
4. Klicken Sie unter **JAAS-Anmeldemodule** auf **Neu**.
5. Geben Sie `com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule` als Modulklassennamen ein, und wählen Sie **SUFFICIENT** als Authentifizierungsstrategie aus. Klicken Sie auf **Anwenden**.
6. Fügen Sie die angepasste Eigenschaft `keyStoreFile` mit dem Wert *Ausgangsverzeichnis_für_Beispiele*`/security_extauth/sampleKS.jks` hinzu.
7. Optional: Fügen Sie die angepasste Eigenschaft `debug` mit dem Wert `true` hinzu.
8. Speichern Sie die Konfiguration.

Prüfpunkt der Lerneinheit:

Jetzt ist die Serverauthentifizierung von WebSphere eXtreme Scale gesichert. Wenn Sie diese Sicherheit konfigurieren, müssen alle Anwendungen, die versuchen, eine Verbindung zu Servern von WebSphere eXtreme Scale herzustellen, einen Berechtigungsnachweis bereitstellen. In diesem Lernprogramm ist KeyStoreLoginAuthenticator der Authentifikator. Deshalb muss der Client einen Benutzernamen und ein Kennwort angeben.

Lerneinheit 2.4: Beispiel installieren und ausführen

Nach der Konfiguration der Authentifizierung können Sie die Beispielanwendung installieren und ausführen.

Gemeinsam genutzte Bibliothek für die Datei `EmployeeData.jar` erstellen:

1. Öffnen Sie in der Administrationskonsole von WebSphere Application Server die Seite **Gemeinsam genutzte Bibliotheken**. Klicken Sie auf **Umgebung** > **Gemeinsam genutzte Bibliotheken**.
2. Wählen Sie den Geltungsbereich **Zelle** aus.
3. Erstellen Sie die gemeinsam genutzte Bibliothek. Klicken Sie auf **Neu**. Geben Sie `EmployeeManagementLIB` im Feld **Name** ein. Geben Sie den Pfad zur Datei `EmployeeData.jar` im Klassenpfad ein, z. B. *Ausgangsverzeichnis_für_Beispiele*`/WASSecurity/EmployeeData.jar`.
4. Klicken Sie auf **Anwenden**.

Beispiel installieren:

1. Installieren Sie die Datei `EmployeeManagement_extauth.ear` im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*`/security_extauth`.

   Wichtig: Die Datei `EmployeeManagement_extauth.ear` und die Datei *Ausgangsverzeichnis_für_Beispiele*`/WASSecurity/EmployeeManagement.ear` sind verschieden. Die Art und Weise, in der die ObjectGrid-Sitzung abgerufen wird, wurde so aktualisiert, dass der in der Clienteigenschaftendatei in der Anwendung `EmployeeManagement_extauth.ear` zwischengespeicherte Berechtigungsnachweis verwendet wird. Sehen Sie sich die Kommentare in der Klasse com.ibm.websphere.sample.xs.DataAccessor im Projekt *Ausgangsverzeichnis_für_Beispiele*`/WASSecurity/EmployeeManagementWeb` an, um den Code zu überprüfen, der für diese Änderung aktualisiert wurde.

a. Klicken Sie zum Starten der Installation auf **Anwendungen** > **Neue Anwendung** > **Neue Unternehmensanwendung**. Wählen Sie den detaillierten Pfad für die Installation der Anwendung aus.
b. Geben Sie im Schritt **Servern Module zuordnen** den Cluster `appCluster` für die Installation des Moduls "EmployeeManagementWeb" an.
c. Wählen Sie im Schritt **Gemeinsam genutzte Bibliotheken zuordnen** das Modul `EmployeeManagementWeb` aus.
d. Klicken Sie auf **Gemeinsam genutzte Bibliotheken referenzieren**. Wählen Sie die Bibliothek `EmployeeManagementLIB` aus.
e. Ordnen Sie **Alle Authentifizierten im Anwendungsrealm** die Rolle `webUser` zu.
f. Klicken Sie auf **OK**.

Die Clients werden in den Servern `s1` und `s2` in diesem Cluster ausgeführt.

2. Installieren Sie die Beispieldatei `XSDeployment.ear`, die im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*`/WASSecurity` enthalten ist.
   a. Klicken Sie zum Starten der Installation auf **Anwendungen** > **Neue Anwendung** > **Neue Unternehmensanwendung**. Wählen Sie den detaillierten Pfad für die Installation der Anwendung aus.
   b. Geben Sie im Schritt **Servern Module zuordnen** den Cluster `xsCluster` für die Installation des Webmoduls `XSDeploymentWeb` an.
   c. Wählen Sie im Schritt **Gemeinsam genutzte Bibliotheken zuordnen** das Modul `XSDeploymentWeb` aus.
   d. Klicken Sie auf **Gemeinsam genutzte Bibliotheken referenzieren**. Wählen Sie die Bibliothek `EmployeeManagementLIB` aus.
   e. Klicken Sie auf **OK**.

   Die Server `xs1` und `xs2` in diesem Cluster hosten die Container-Server.
3. Vergewissern Sie sich, dass der Katalogserver gestartet ist. Weitere Informationen zum Starten eines Katalogservers für dieses Lernprogramm finden Sie unter „Katalogserver mit aktivierter Sicherheit starten” auf Seite 124.
4. Starten Sie den Cluster "xsCluster" erneut. Wenn der Cluster "xsCluster" gestartet wird, wird die Anwendung `XSDeployment` und jeweils ein Container-Server auf den Servern "xs1" und "xs2" gestartet. Wenn Sie sich die Datei `SystemOut.log` der Server "xs1" und "xs2" ansehen, wird die folgende Nachricht angezeigt, die darauf hinweist, dass die Servereigenschaftendatei geladen wurde:

   ```
   CWOBJ0913I: Die Servereigenschaftendateien wurden geladen:
   Ausgangsverzeichnis_für_Beispiele/security_extauth/server3.props.
   ```
5. Starten Sie den Cluster "appClusters" erneut. Wenn der Cluster "appCluster" gestartet wird, wird auch die Anwendung "EmployeeManagement" gestartet. Wenn Sie sich die Datei `SystemOut.log` der Server "s1" und "s2" ansehen, können Sie die folgende Nachricht sehen, die darauf hinweist, dass die Clienteigenschaftendatei geladen wurde.

   ```
   CWOBJ0924I: Die Clienteigenschaftendatei {0} wurde geladen.
   ```

   Wenn Sie WebSphere eXtreme Scale Version 7.0 verwenden, wird die Nachricht CWOBJ9000I angezeigt, die nur in Englisch verfügbar ist und darauf hinweist, dass die Clienteigenschaftendatei geladen wurde. Wenn Sie die erwartete Nachricht nicht sehen, vergewissern Sie sich, dass Sie die Eigenschaft "-Dobjectgrid-.server.props" bzw. "-Dobjectgrid.client.props" im JVM-Argument konfiguriert haben. Ist die Eigenschaft konfiguriert, stellen Sie sicher, dass der Strich (-) ein UTF-Zeichen ist.

Beispielanwendung ausführen:

1. Führen Sie die Datei `management.jsp` aus. Greifen Sie in einem Web-Browser auf den URL `http://<Ihr_Servername>:<Port>/EmployeeManagementWeb/management.jsp` auf. Sie können beispielsweise den folgenden URL verwenden: `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Authentifizierung für die Anwendung bereitstellen. Geben Sie die Berechtigungsnachweise des Benutzers ein, den Sie der Rolle "webUser" zugeordnet haben. Standardmäßig wird diese Benutzerrolle allen authentifizierten Benutzern zugeordnet. Geben Sie einen gültigen Benutzernamen und ein gültiges Kennwort an, z. B. den Namen und das Kennwort des Benutzers mit Verwaltungsaufgaben. Es erscheint eine Seite, auf der Sie Mitarbeiter anzeigen, hinzufügen, aktualisieren und löschen können.
3. Mitarbeiter anzeigen. Klicken Sie auf **Display an Employee**. Geben Sie `emp1@acme.com` als E-Mail-Adresse ein, und klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer nicht gefunden wurde.
4. Mitarbeiter hinzufügen. Klicken Sie auf **Add an Employee**. Geben Sie `emp1@acme.com` als E-Mail-Adresse, `Joe` als Vornamen und `Doe` als Nachnamen ein. Klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass ein Mitarbeiter mit der Adresse `emp1@acme.com` hinzugefügt wurde.
5. Neuen Mitarbeiter anzeigen. Klicken Sie auf **Display an Employee**. Geben Sie `emp1@acme.com` als E-Mail-Adresse mit leeren Feldern für den Vor- und Nachnamen ein, und klicken Sie dann auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Mitarbeiter gefunden wurde, und die richtigen Namen werden in den Feldern für den Vornamen und den Nachnamen angezeigt.
6. Mitarbeiter löschen. Klicken Sie auf **Delete an employee**. Geben Sie `emp1@acme.com` ein, und klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer gelöscht wurde.

Da der Transporttyp des Katalogservers auf `TCP/IP` gesetzt ist, vergewissern Sie sich, dass die Einstellung für abgehende Transporte von Server "s1" und "s2" nicht auf `SSL-Required` gesetzt ist. Andernfalls tritt eine Ausnahme ein. In der Systemausgabedatei des Katalogservers, `logs/cs1/SystemOut.log`, sehen Sie die folgende Debugausgabe, die auf die Keystoreauthentifizierung hinweist:

```
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:
...
```

Prüfpunkt der Lerneinheit:

Sie haben die Beispielanwendung installiert und ausgeführt.

Modul 3: Transportsicherheit konfigurieren

Konfigurieren Sie die Transportsicherheit, um die Datenübertragung zwischen den Clients und den Servern in der Konfiguration zu sichern.

Im vorherigen Modul des Lernprogramms haben Sie die eXtreme-Scale-Authentifizierung aktiviert. Mit Authentifizierung muss jede Anwendung, die versucht, eine Verbindung zum Server von WebSphere eXtreme Scale herzustellen, einen Berechtigungsnachweis bereitstellen. Deshalb kann kein nicht authentifizierter Client eine

Verbindung zum Server von WebSphere eXtreme Scale herstellen. Die Clients müssen eine authentifizierte Anwendung sein, die in einer Zelle von WebSphere Application Server ausgeführt wird.

Mit der Konfiguration, die Sie bis zu diesem Modul hin vorgenommen haben, ist die Datenübertragung zwischen den Clients im Cluster "appCluster" und den Servern im Cluster "xsCluster" nicht verschlüsselt. Diese Konfiguration kann akzeptabel sein, wenn Ihre Cluster von WebSphere Application Server auf Servern hinter einer Firewall installiert sind. In manchen Szenarien wird ein nicht verschlüsselter Datenverkehr jedoch aus bestimmten Gründen nicht akzeptiert, selbst wenn die Topologie durch eine Firewall geschützt ist. Der verschlüsselte Datenverkehr kann beispielsweise durch eine behördliche Richtlinie vorgegeben werden. WebSphere eXtreme Scale unterstützt Transport Layer Security/Secure Sockets Layer (TLS/SSL) für die sichere Kommunikation zwischen ObjectGrid-Endpunkten, zu denen Client-Server, Container-Server und Katalogserver gehören.

In dieser Beispielimplementierung werden die Clients und die Container-Server von eXtreme Scale alle in der Umgebung von WebSphere Application Server ausgeführt werden. Es sind keine Client- und Servereigenschaften zum Konfigurieren der SSL-Einstellungen erforderlich, weil die Transportsicherheit von eXtreme Scale durch die CSIv2-Transporteinstellungen (Common Secure Interoperability Protocol Version 2) des Anwendungsservers verwaltet wird. Die Server von WebSphere eXtreme Scale verwenden dieselbe ORB-Instanz (Object Request Broker) wie die Anwendungsserver, in denen sie ausgeführt werden. Legen Sie alle SSL-Einstellungen für die Clients und Container-Server mit diesen CSIv2-Transporteinstellungen in der Konfiguration von WebSphere Application Server fest. Sie müssen die SSL-Einstellungen in der Servereigenschaftendatei für den Katalogserver konfigurieren.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie in der Lage, die folgenden Aufgaben auszuführen:

- Eingehende und abgehende CSIv2-Transporte konfigurieren
- SSL-Eigenschaften zur Eigenschaftendatei des Katalogservers hinzufügen
- ORB-Eigenschaftendatei überprüfen
- Beispiel ausführen

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Voraussetzungen

Dieser Schritt des Lernprogramms baut auf den vorherigen Modulen auf. Arbeiten Sie die vorherigen Module in diesem Lernprogramm durch, bevor Sie die Transportsicherheit konfigurieren.

Lerneinheit 3.1: Eingehenden und abgehenden CSIv2-Transport konfigurieren

Zum Konfigurieren von Transport Layer Security/Secure Sockets Layer (TLS/SSL) für den Servertransport setzen Sie den eingehenden CSIv2-Transport (Common Secure Interoperability Protocol Version 2) und den abgehenden CSIv2-Transport für alle Server von WebSphere Application Server, die Clients, Katalogserver und Container-Server hosten, auf `SSL-Required`.

In der Beispieltopologie des Lernprogramms müssen Sie diese Eigenschaften für die Anwendungsserver s1, s2, xs1 und xs2 definieren. Mit den folgenden Schritten konfigurieren Sie die eingehenden und abgehenden Transporte für alle Server in der Konfiguration.

Definieren Sie die eingehenden und abgehenden Transporte in der Administrationskonsole. Stellen Sie sicher, dass die Verwaltungssicherheit aktiviert ist.

- **WebSphere Application Server Version 6.1**: Klicken Sie auf **Sicherheit** > **Sichere Verwaltung** > **Anwendung** > **RMI/IIOP-Sicherheit**, und ändern Sie den Transporttyp in **SSL-Required**.
- **WebSphere Application Server Version 7.0**: Klicken Sie auf **Sicherheit** > **Globale Sicherheit** > **RMI/IIOP-Sicherheit** > **Eingehende CSIv2-Kommunikation**. Ändern Sie den Transporttyp unter der CSIv2-Transportschicht in **SSL-Required**. Wiederholen Sie diesen Schritt zum Konfigurieren der abgehenden CSIv2-Kommunikation.

Sie können zentral verwaltete Endpunktsicherheitseinstellungen verwenden oder SSL-Repositorys konfigurieren. Weitere Informationen finden Sie unter Common Secure Interoperability Version 2 transport inbound settings.

Lerneinheit 3.2: SSL-Eigenschaften der Eigenschaftendatei des Katalogservers hinzufügen

Der Katalogserver wird außerhalb von WebSphere Application Server ausgeführt. Deshalb müssen Sie die SSL-Eigenschaften in der Servereigenschaftendatei konfigurieren.

Der andere Grund für die Konfiguration der SSL-Eigenschaften in der Servereigenschaftendatei ist der, dass der Katalogserver eigene proprietäre Transportpfade hat, die nicht mit den CSIv2-Transporteinstellungen (Common Secure Interoperability Protocol Version 2) von WebSphere Application Server verwaltet werden können. Deshalb müssen Sie die SSL-Eigenschaften in der Servereigenschaftendatei für den Katalogserver konfigurieren.

SSL-Eigenschaften in der Datei `catServer3.props`:

```
alias=default
contextProvider=IBMJSSE2
protocol=SSL
keyStoreType=PKCS12
keyStore=/WAS-Stammverzeichnis/IBM/WebSphere/AppServer/profiles/
<Deployment-Manager-Name>/config/cells/<Zellenname>/nodes/
<Knotenname>/key.p12
keyStorePassword=WebAS
trustStoreType=PKCS12
trustStore=/WAS-Stammverzeichnis/IBM/WebSphere/AppServer/profiles/
<Deployment-Manager-Name>/config/cells/<Zellenname>/nodes/
<Knotenname>/trust.p12
trustStorePassword=WebAS
clientAuthentication=false
```

Die Datei `catServer3.props` verwendet den Standardkeystore und -truststore von WebSphere Application Server auf Knotenebene. Wenn Sie die Implementierung in einer komplexeren Implementierungsumgebung durchführen, müssen Sie den richtigen Keystore und Truststore auswählen. In manchen Fällen müssen Sie einen Keystore und einen Truststore erstellen und die Schlüssel aus Keystores anderer Server importieren. Die Zeichenfolge `WebAS` ist das Standardkennwort des Keystores und des Truststores von WebSphere Application Server. Weitere Einzelheiten finden Sie unter Konfiguration mit selbst signiertem Standardzertifikat.

Diese Einträge sind bereits als Kommentare in der Datei *Ausgangsverzeichnis_für_Beispiele*/security_extauth/catServer3.props enthalten. Sie können die Kommentarzeichen aus diesen Einträgen entfernen und die entsprechenden Aktualisierungen für Ihre Installation an den Variablen *WAS-Stammverzeichnis*, *<Deployment-Manager-Name>*, *<Zellenname>* und *<Knotenname>* vornehmen.

Nach der Konfiguration der SSL-Eigenschaften ändern Sie den Wert der Eigenschaft "transportType" von TCP/IP in SSL-Required.

SSL-Eigenschaften in der Datei client3.props:

Sie müssen die SSL-Eigenschaften auch in der Datei client3.props konfigurieren, weil diese Datei verwendet wird, wenn Sie den Katalogserver stoppen, der außerhalb von WebSphere Application Server ausgeführt wird.

Diese Eigenschaften haben keine Auswirkung auf die Client-Server, die in WebSphere Application Server ausgeführt werden, weil sie die CSIv2-Transporteinstellungen (Common Security Interoperability Protocol Version 2) von WebSphere Application Server verwenden. Wenn Sie den Katalogserver stoppen, müssen Sie jedoch eine Clienteigenschaftendatei mit dem Befehl **stopOgServer** angeben. Setzen Sie die folgenden Eigenschaften in der Datei "<SAMPLES_HOME>/security_extauth/client3.props" so, dass deren Werte den Werten entsprechen, die zuvor in der Datei "catServer3.props" angegeben wurden:

```
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/WAS-Stammverzeichnis/IBM/WebSphere/AppServer/profiles/
<Deployment-Manager-Name>/config/cells/<Zellenname>/nodes/
<Knotenname>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/WAS-Stammverzeichnis/IBM/WebSphere/AppServer/profiles/
<Deployment-Manager-Name>/config/cells/<Zellenname>/nodes/
<Knotenname>/trust.p12
#trustStorePassword=WebAS
```

Wie bei der Datei catServer3.props können Sie die bereits in der Datei *Ausgangsverzeichnis_für_Beispiele*/security_extauth/client3.props enthaltenen Kommentare verwenden und die entsprechenden Aktualisierungen an den Variablen *WAS-Stammverzeichnis*, *<Deployment-Manager-Name>*, *<Zellenname>* und *<Knotenname>* vornehmen, so dass sie Ihrer Umgebung entsprechen.

Prüfpunkt der Lerneinheit:

Sie haben die SSL-Eigenschaften für den Katalogserver konfiguriert.

Lerneinheit 3.3: Beispiel ausführen

Starten Sie alle Server erneut, und führen Sie die Beispielanwendung erneut aus. Sie sollten die Schritte ohne Probleme ausführen können.

Weitere Informationen zum Ausführen und Installieren der Beispielanwendung finden Sie unter „Lerneinheit 2.4: Beispiel installieren und ausführen” auf Seite 127.

Modul 4: JAAS-Berechtigung (Java Authentication and Authorization Service) in WebSphere Application Server verwenden

Nachdem Sie nun die Authentifizierung für Clients konfiguriert haben, können Sie die Berechtigung weiter konfigurieren, um verschiedenen Benutzern verschiedene Berechtigungen zuzuordnen. Ein Bediener kann beispielsweise nur in der Lage sein, Daten anzuzeigen, während ein Manager alle Operationen ausführen kann.

Nach der Authentifizierung eines Clients können Sie wie im vorherigen Modul dieses Lernprogramms Sicherheitsberechtigungen über den Berechtigungsmechanismus von eXtreme Scale erteilen. Das vorherige Modul dieses Lernprogramms hat veranschaulicht, wie die Authentifizierung für ein Datengrid durch Integration mit WebSphere Application Server aktiviert wird. Deswegen kann kein nicht authentifizierter Client eine Verbindung zu den Servern von eXtreme Scale herstellen oder Anforderungen an Ihr System übergeben. Jeder authentifizierte Client hat jedoch dieselben Berechtigungen oder Privilegien beim Server, z. B. Lesen, Schreiben oder Löschen von Daten, die in ObjectGrid-Maps gespeichert sind. Clients können auch jeden Typ von Abfrage absetzen.

In diesem Teil des Lernprogramms wird gezeigt, wie Sie über eXtreme-Scale-Berechtigungen authentifizierten Benutzern unterschiedliche Privilegien erteilen. WebSphere eXtreme Scale verwendet einen auf Berechtigungen basierenden Berechtigungsmechanismus. Sie können verschiedene Berechtigungskategorien zuweisen, die durch verschiedene Berechtigungsklassen dargestellt werden. In diesem Modul wird die Klasse MapPermission verwendet. Eine Liste der möglichen Berechtigungen finden Sie unter Programmierung der Clientberechtigung.

In WebSphere eXtreme Scale stellt die Klasse `com.ibm.websphere.objectgrid.security.MapPermission` Berechtigungen für die Ressourcen von eXtreme Scale dar, insbesondere die Methoden der Schnittstelle ObjectMap oder JavaMap. WebSphere eXtreme Scale definiert die folgenden Berechtigungszeichenfolgen für den Zugriff auf die Methoden der Schnittstellen "ObjectMap" und "JavaMap":

- **read**: Erteilt die Berechtigung zum Lesen der Daten aus der Map.
- **write**: Erteilt die Berechtigung zum Aktualisieren der Daten in der Map.
- **insert**: Erteilt die Berechtigung zum Einfügen der Daten in die Map.
- **remove**: Erteilt die Berechtigung zum Entfernen der Daten aus der Map.
- **invalidate**: Erteilt die Berechtigung zum Ungültigmachen der Daten in der Map.
- **all**: Erteilt alle zuvor beschriebenen Berechtigungen: read, write, insert, remote und invalidate.

Die Berechtigung findet statt, wenn ein eXtreme-Scale-Client eine Datenzugriffs-API wie die API ObjectMap, JavaMap oder EntityManager verwendet. Die Laufzeitumgebung von eXtreme Scale überprüft die entsprechenden Map-Berechtigungen, wenn die Methode aufgerufen wird. Wenn dem Client die erforderlichen Berechtigungen nicht erteilt wurden, wird eine Ausnahme des Typs "AccessControlException" ausgegeben. Dieses Lernprogramm veranschaulicht, wie über JAAS-Berechtigung verschiedenen Benutzern Berechtigungen für Map-Zugriffe erteilt werden.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie in der Lage, die folgenden Aufgaben auszuführen:

- Berechtigung für WebSphere eXtreme Scale aktivieren
- Benutzerbasierte Berechtigung aktivieren

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Lerneinheit 4.1: eXtreme-Scale-Berechtigung aktivieren

Zum Aktivieren der Berechtigung in WebSphere eXtreme Scale müssen Sie die Sicherheit in einem bestimmten ObjectGrid aktivieren.

Zum Aktivieren der Berechtigung im ObjectGrid müssen Sie das Attribut **securityEnabled** für das entsprechende ObjectGrid in der XML-Datei auf "true" setzen. Für dieses Lernprogramm können Sie die Datei `XSDeployment_sec.ear` im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*`/WASSecurity` verwenden, in der die Sicherheit bereits in der Datei `objectGrid.xml` definiert ist, oder Sie können die vorhandene Datei `objectGrid.xml` bearbeiten, um die Sicherheit zu aktivieren. Diese Lerneinheit veranschaulicht, wie die Datei bearbeitet wird, um die Sicherheit zu aktivieren.

1. Optional: Extrahieren Sie die Dateien aus der Datei `XSDeployment.ear`, und entpacken Sie dann die Datei `XSDeploymentWeb.war`.
2. Optional: Öffnen Sie die Datei `objectGrid.xml`, und setzen Sie das Attribut **securityEnabled** auf ObjectGrid-Ebene auf `true`. Ein Beispiel für dieses Attribut folgt:

   ```
   <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>
   ```

   Wenn Sie mehrere ObjectGrids definiert haben, müssen Sie dieses Attribut in jedem Grid definieren.
3. Optional: Packen Sie die Datei `XSDeploymentWeb.war` und die Datei `XSDeployment.ear` erneut, um Ihre Änderungen einzuschließen.
4. Erforderlich: Deinstallieren Sie die Datei `XSDeployment.ear`, und installieren Sie anschließend die aktualisierte Datei `XSDeployment.ear`. Sie können die Datei verwenden, die Sie in den vorherigen Schritten bearbeitet haben, oder Sie können die Datei `XSDeployment_sec.ear` installieren, die im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*`/WASSecurity` bereitgestellt wird. Weitere Informationen zum Installieren der Anwendung finden Sie unter „Lerneinheit 2.4: Beispiel installieren und ausführen” auf Seite 127.
5. Starten Sie alle Anwendungsserver erneut, um die Berechtigung in WebSphere eXtreme Scale zu aktivieren.

Prüfpunkt der Lerneinheit:

Sie haben die Sicherheit im ObjectGrid und damit auch die Berechtigung im Datengrid aktiviert.

Lerneinheit 4.2: Benutzerbasierte Berechtigung aktivieren

IM Authentifizierungsmodul dieses Lernprogramms haben Sie zwei Benutzer erstellt: `operator` und `manager`Sie können diesen Benutzern mit JAAS-Berechtigung (Java Authentication and Authorization Service) verschiedene Berechtigungen zuordnen.

JAAS-Berechtigungsrichtlinie (Java Authentication and Authorization Service) mit Benutzerprincipals definieren:

Sie können den Benutzern, die Sie zuvor erstellt haben, Berechtigungen zuordnen. Ordnen Sie dem Benutzer operator nur Leseberechtigungen für alle Maps zu. Ordnen Sie dem Benutzer manager alle Berechtigungen zu. Verwenden Sie die JAAS-Berechtigungsrichtliniendatei, um Principals Berechtigungen zu erteilen.

Bearbeiten Sie die JAAS-Berechtigungsdatei. Die Datei xsAuth3.policy befindet sich im Verzeichnis *Ausgangsverzeichnis_für_Beispiele*/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";
};
```

In dieser Datei ist die Codebasis http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction ein speziell reservierter URL für ObjectGrid. Alle ObjectGrid-Berechtigungen, die Principals erteilt werden, verwenden diese spezielle Codebasis. Die folgenden Berechtigungen werden in dieser Datei zugeordnet:

- Die erste grant-Anweisung erteilt dem Principal "CN=operator,O=acme,OU=OGSample" die Map-Berechtigung read. Der Benutzer "CN=operator,O=acme,OU=OGSample" hat nur Leseberechtigung für die Map "Map1" in der ObjectGrid-Instanz "Grid".
- Die zweite grant-Anweisung erteilt dem Principal "CN=manager,O=acme,OU=OGSample" alle Map-Berechtigungen. Der Benutzer "CN=manager,O=acme,OU=OGSample" hat alle Berechtigungen für die Map "Map1" in der ObjectGrid-Instanz "Grid".

JAAS-Berechtigungsrichtliniendatei mit JVM-Eigenschaften definieren:

Verwenden Sie die folgenden Schritte, um JVM-Eigenschaften für die Server "xs1" und "xs2" im Cluster "xsCluster" zu definieren. Wenn Sie eine Topologie verwenden, die von der Beispieltopologie abweicht, die in diesem Lernprogramm verwendet wird, definieren Sie die Datei in allen Container-Servern.

1. Klicken Sie in der Administrationskonsole auf **Server** > **Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Fügen Sie die folgenden generischen JVM-Argumente hinzu:

   -Djava.security.auth.policy=*Ausgangsverzeichnis_für_Beispiele*/security_extauth/xsAuth3.policy
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Beispielanwendung zu Testen der Berechtigung ausführen:

Sie können die Beispielanwendung verwenden, um die Berechtigungseinstellungen zu testen. Der Managerbenutzer mit Administratorberechtigung hat weiterhin alle Berechtigungen in der Map "Map1", einschließlich der Berechtigungen zum Anzeigen und Hinzufügen von Mitarbeitern. Der Benutzer mit Bedienerberechtigung darf Mitarbeiter nur anzeigen, weil diesem Benutzer nur Leseberechtigung erteilt wurde.

1. Starten Sie alle Anwendungsserver, in denen Container-Server ausgeführt werden, erneut. Für dieses Lernprogramm starten Sie die Server xs1 und xs2 erneut.
2. Öffnen Sie die Anwendung EmployeeManagementWeb. Öffnen Sie in einem Webbrowser http://<Host>:<Port>/EmployeeManagermentWeb/management.jsp.

3. Melden Sie sich mit einem gültigen Benutzernamen und einem Kennwort an der Anmeldung an.
4. Versuchen Sie, einen Mitarbeiter anzuzeigen. Klicken Sie auf **Display an Employee**, und suchen Sie die E-Mail-Adresse `authemp1@acme.com`. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer nicht gefunden wurde.
5. Fügen Sie einen Mitarbeiter hinzu. Klicken Sie auf **Add an Employee**. Fügen Sie die E-Mail-Adresse `authemp1@acme.com`, den Vornamen `Joe` und den Nachnamen `Doe` hinzu. Klicken Sie auf **Submit**. Es erscheint eine Nachricht, in der Ihnen mitgeteilt wird, dass der Benutzer hinzugefügt wurde.
6. Editieren Sie die Datei *Ausgangsverzeichnis_für_Beispiele*`/security_extauth/client3.props`. Ändern Sie den Wert der Eigenschaft "credentialGeneratorProps" von `manager manager1` in `operator operator1`. Nach dem Editieren der Datei verwendet das Servlet den Benutzernamen "operator" und das Kennwort "operator1" für die Authentifizierung bei den Servern von WebSphere eXtreme Scale.
7. Starten Sie den Cluster "appCluster" erneut, damit die Änderungen in der Datei *Ausgangsverzeichnis_für_Beispiele*`/security_extauth/client3.props` übernommen werden.
8. Versuchen Sie, einen Mitarbeiter anzuzeigen. Klicken Sie auf **Display an Employee**, und suchen Sie die E-Mail-Adresse `authemp1@acme.com`. Der Mitarbeiter wird angezeigt.
9. Fügen Sie einen Mitarbeiter hinzu. Klicken Sie auf **Add an Employee**. Fügen Sie die E-Mail-Adresse `authemp2@acme.com`, den Vornamen `Joe` und den Nachnamen `Doe` hinzu. Klicken Sie auf **Submit**. Die folgende Nachricht wird angezeigt:

   ```
   An exception occurs when Add the employee. See below for detailed exception messages.
   ```

   Der detaillierte Ausnahmetext folgt:

   ```
   java.security.AccessControlException: Access denied
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Diese Nachricht wird angezeigt, weil der Benutzer `operator` nicht berechtigt ist, Daten in die Map "Map1" einzufügen.

Wenn Sie mit einer Version von WebSphere Application Server vor Version 7.0.0.11 arbeiten, wird im Container-Server unter Umständen ein Fehler des Typs "java.lang.StackOverflowError" angezeigt. Dieser Fehler ist auf ein Problem mit IBM Developer Kit zurückzuführen. Das Problem ist in dem IBM Developer Kit, das mit WebSphere Application Server Version 7.0.0.11 und höher bereitgestellt wird, behoben.

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Berechtigung konfiguriert, indem Sie bestimmten Benutzern Berechtigungen erteilt haben.

Modul 5: Dienstprogramm `xscmd` für die Überwachung von Datengrids und Maps verwenden

Sie können das Dienstprogramm **`xscmd`** verwenden, um die Größen der primären Datengrids und Maps des Datengrids `Grid` anzuzeigen. Das Tool **`xscmd`** verwendet die MBean, um alle Artefakte des Datengrids, wie z. B. primäre Shards, Replikat-Shards, Container-Server, Map-Größen und andere Daten, abzufragen.

In diesem Lernprogramm wird der Katalogserver als eigenständiger Java-SE-Server ausgeführt. Die Container-Server werden in Anwendungsservern von WebSphere Application Server ausgeführt.

Für den Katalogserver wird ein MBean-Server in der eigenständigen Java Virtual Machine (JVM) erstellt. Wenn Sie das Tool **xscmd** im Katalogserver verwenden, wird die Sicherheit von WebSphere eXtreme Scale verwendet.

Für die Container-Server registriert die Laufzeitumgebung von WebSphere eXtreme Scale die Managed Beans (MBean) bei dem MBean-Server, der von der Laufzeitumgebung von WebSphere Application Server erstellt wird. Die vom Tool **xscmd** verwendete Sicherheit wird von der MBean-Sicherheit von WebSphere Application Server bereitgestellt.

1. Wechseln Sie über ein Befehlszeilentool in das Verzeichnis *DMGR-PROFIL*/bin.
2. Führen Sie das Tool **xscmd** aus. Verwenden Sie die Parameter **-c showPlacement -st P**, wie in den folgenden Beispielen gezeigt:

   Linux UNIX

   ```
   xscmd.sh -c listObjectGridPlacement -cep localhost:16099 -g Grid -ms mapSet -sf P
   -user manager -pwd manager1
   ```

   Windows

   ```
   xscmd.bat -c listObjectGridPlacement -cep localhost:16099 -g Grid -m mapSet -sf P
   -user manager -pwd manager1
   ```

   Der Benutzername und das Kennwort werden zur Authentifizierung an den Katalogserver übergeben.
3. Sehen Sie sich die Befehlsergebnisse an.

   ```
   *** 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. Führen Sie das Tool **xscmd** aus. Verwenden Sie den Parameter **-c showMapSizes** wie in den folgenden Beispielen:

   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
   ```

   Der Benutzername und das Kennwort werden zur Authentifizierung an den Katalogserver übergeben. Nach der Ausführung des Befehls werden Sie zur Eingabe der Benutzer-ID und des Kennworts von WebSphere Application Server für die Authentifizierung bei WebSphere Application Server aufgefordert. Sie müssen diese Anmeldeinformationen angeben, weil die Option **-c showMapSizes** die Map-Größe von jedem Container-Server abruft. Hierfür ist die Sicherheit von WebSphere Application Server erforderlich.
5. Optional: Sie können die Datei PROFIL/properties/sas.client.props so ändern, dass der Befehl ausgeführt wird, ohne dass die Benutzer-ID und das Kennwort eingegeben werden müssen. Ändern Sie die Eigenschaft "com.ibm-.CORBA.loginSource" von prompt in properties, und geben Sie dann die Benutzer-ID und das Kennwort an. Im Folgenden sehen Sie ein Beispiel für die Eigenschaften in der Datei 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. Optional: Wenn Sie den Befehl **xscmd** in einer eigenständigen Installation von WebSphere eXtreme Scale verwenden, müssen Sie die folgenden Optionen hinzufügen:
   - Bei Verwendung der Sicherheit von WebSphere eXtreme Scale:

     ```
     -user
     -pwd
     ```

   - Bei Verwendung der Sicherheit von WebSphere eXtreme Scale mit Generierung angepasster Berechtigungsnachweise:

     ```
     -user
     -pwd
     -cgc
     -cgp
     ```

   - Bei Aktivierung von SSL:

     ```
     -tt
     -cxpv
     -prot
     -ks
     -ksp
     -kst
     -ts
     -tsp
     -tst
     ```

     Wenn die Sicherheit von WebSphere eXtreme Scale und SSL aktiviert sind, sind beide Gruppen von Parametern erforderlich.

Prüfpunkt der Lerneinheit

Sie haben das Tool **xscmd** verwendet, um Datengrids und Maps in Ihrer Konfiguration zu überwachen.

Lernprogramm: eXtreme-Scale-Bundles im OSGi-Framework ausführen

Das OSGi-Beispiel baut auf den Serializer-Beispielen für Google Protocol Buffers auf. Nach dem Durcharbeiten dieser Lerneinheiten haben Sie die Serializer-Beispiel-Plug-ins im OSGi-Framework ausgeführt.

Lernziele

Dieses Beispiel veranschaulicht die OSGi-Bundles. Das Serializer-Plug-in ist nebensächlich und nicht erforderlich. Das OSGi-Beispiel ist in der Beispielsammlung von WebSphere eXtreme Scale enthalten. Sie müssen das Beispiel herunterladen und im Verzeichnis *WXS-Ausgangsverzeichnis*/samples entpacken. Das Stammverzeichnis für das OSGi-Beispiel ist wxs_home/samples/OSGiProto.

Das Beispiel für die Google-Protokollpuffer befindet sich im Verzeichnis *WXS-Ausgangsverzeichnis*/samples/SerializerProto.

Das Beispiel für die BSON-Serialisierungsmethode (Binary JSON) befindet sich im Verzeichnis *WXS-Ausgangsverzeichnis*/samples/SerializerBSON.

In den Befehlsbeispielen in diesem Lernprogramm wird angenommen, dass Sie mit dem Betriebssystem UNIX arbeiten. Sie müssen das Befehlsbeispiel auf einem Windows-Betriebssystem anpassen.

Nach der Ausführung der Lerneinheiten in diesem Lernprogramm sind Sie mit den OSGi-Beispielkonzepten vertraut und wissen, wie die folgenden Ziele erreicht werden:

- Server-Bundle von WebSphere eXtreme Scale im OSGi-Container installieren, um den eXtreme-Scale-Server zu starten
- Entwicklungsumgebung von eXtreme Scale zum Ausführen des Beispielclients einrichten
- Befehl xscmd verwenden, um das Service-Ranking des Beispielbundles abzufragen, das Bundle auf ein neues Service-Ranking zu aktualisieren und das neue Service-Ranking zu überprüfen

Erforderliche Zeit

Das Durcharbeiten dieses Moduls dauert ungefähr 60 Minuten.

Voraussetzungen

Für dieses Lernprogramm müssen Sie nicht nur die Serializer-Beispiele herunterladen und entpacken, sondern auch die folgenden vorausgesetzten Aufgaben ausführen:

- Produkt eXtreme Scale installieren und entpacken
- Eclipse-Equinox-Umgebung einrichten

Einführung: eXtreme-Scale-Server und -Container für die Ausführung von Plug-ins im OSGi-Framework starten und konfigurieren

In diesem Lernprogramm starten Sie einen eXtreme-Scale-Server im OSGi-Framework, starten einen eXtreme-Scale-Container und verbinden die Beispiel-Plug-ins mit der eXtreme-Scale-Laufzeitumgebung.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Lernprogramm sind Sie mit den OSGi-Beispielkonzepten vertraut und wissen, wie die folgenden Ziele erreicht werden:

- Server-Bundle von WebSphere eXtreme Scale im OSGi-Container installieren, um den eXtreme-Scale-Server zu starten
- Entwicklungsumgebung von eXtreme Scale zum Ausführen des Beispielclients einrichten
- Befehl xscmd verwenden, um das Service-Ranking des Beispielbundles abzufragen, das Bundle auf ein neues Service-Ranking zu aktualisieren und das neue Service-Ranking zu überprüfen

Erforderliche Zeit

Das Durcharbeiten dieses Lernprogramms dauert ungefähr 60 Minuten. Wenn Sie sich sich mit weiteren Konzepten vertraut machen, die sich auf dieses Lernprogramm beziehen, dauert es unter Umständen noch länger.

Kenntnisstufe

Fortgeschrittener Anfänger.

Zielgruppe

Entwickler und Administratoren, die eXtreme-Scale-Bundles im OSGi-Framework erstellen, installieren und ausführen möchten.

Systemvoraussetzungen

- Befehlszeilenclient Luminis OSGi Configuration Admin Version 0.2.5
- Apache Felix File Install Version 3.0.2
- Wenn Sie Eclipse Gemini als Blueprint-Container-Provider verwenden möchten, sind folgende Komponenten erforderlich:
  - 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
- Wenn Sie Apache Aries als Blueprint-Container-Provider verwenden, müssen die folgenden Voraussetzungen erfüllt sein:
  - Apache Aries (letzte Momentaufnahme)
  - ASM-Bibliothek
  - PAX-Protokollierung

Voraussetzungen

Zum Durcharbeiten dieses Lernprogramms müssen Sie das Beispiel herunterladen und im Verzeichnis `wxs_home/samples` entpacken. Das Stammverzeichnis für das OSGi-Beispiel ist `wxs_home/samples/OSGiProto`.

Erwartete Ergebnisse

Nach dem Durcharbeiten dieses Lernprogramms haben Sie die Beispielbundles installiert und einen eXtreme-Scale-Client ausgeführt, um Daten in das Grid einzufügen. Sie können diese Beispielbundles auch mit den dynamischen Funktionen abfragen und aktualisieren, die der OSGi-Container bereitstellt.

Modul 1: Installation und Konfiguration von Serverbundles von eXtreme Scale vorbereiten

Arbeiten Sie dieses Modul durch, um sich mit den OSGi-Beispielbundles und den Konfigurationsdateien vertraut zu machen, die Sie für die Konfiguration des eXtreme-Scale-Servers verwenden.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie mit den Konzepten vertraut und wissen, wie die folgenden Ziele erreicht werden:

- Im OSGi-Beispiel enthaltene Bundles suchen und untersuchen
- Konfigurationsdateien untersuchen, die für die Konfiguration des eXtreme-Scale-Grids- und -Servers verwendet werden

Lerneinheit 1.1: OSGi-Beispielbundles kennenlernen

Arbeiten Sie diese Lerneinheit durch, und untersuchen Sie die Bundles, die im OSGi-Beispiel bereitgestellt werden.

OSGi-Beispielbundles:

Neben den Bundles, die in der Datei `config.ini` konfiguriert sind und deren Konfiguration im Artikel zum Einrichten der Eclipse-Equinox-Umgebung beschrieben wird, werden die folgenden zusätzlichen Bundles im OSGi-Beispiel verwendet:

objectgrid.jar
Das Laufzeitbundle für den Server von WebSphere eXtreme Scale. Dieses Bundle befindet sich im Verzeichnis *wxs_home*/lib.

com.google.protobuf_2.4.0a.jar
Das Bundle für Google Protocol Buffers Version 2.4.0a. Dieses Bundle befindet sich im Verzeichnis *wxs_sample_osgi_root*/lib.

ProtoBufSamplePlugins-1.0.0.jar
Version 1.0.0 des Benutzer-Plug-in-Bundles mit den Beispiel-Plug-in-Implementierungen ObjectGridEventListener und MapSerializerPlugin. Dieses Bundle befindet sich im Verzeichnis *wxs_sample_osgi_root*/lib. Die Services werden mit dem Service-Ranking 1 konfiguriert.

Diese Version verwendet die Blueprint-Standard-XML für die Konfiguration der Plug-in-Services von eXtreme Scale. Die Serviceklasse ist eine vom Benutzer implementierte Klasse für die Schnittstelle com.ibm.websphere-.objectgrid.plugins.osgi.PluginServiceFactory von WebSphere eXtreme Scale. Die vom Benutzer implementierte Klasse erstellt eine Bean für jede Anforderung und arbeitet ähnlich wie eine Prototyp-Bean.

ProtoBufSamplePlugins-2.0.0.jar
Version 2.0.0 des Benutzer-Plug-in-Bundles mit den Beispiel-Plug-in-Implementierungen ObjectGridEventListener und MapSerializerPlugin. Dieses Bundle befindet sich im Verzeichnis *wxs_sample_osgi_root*/lib. Die Services werden mit dem Service-Ranking 2 konfiguriert.

Diese Version verwendet die Blueprint-Standard-XML für die Konfiguration der Plug-in-Services von eXtreme Scale. Die Serviceklasse verwendet die integrierte Klasse com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl von von WebSphere eXtreme Scale, die den Service BlueprintContainer verwendet. Mithilfe der Blueprint-XML-Standardkonfiguration können die Beans als Prototyp- oder Singleton-Bean konfiguriert werden. Die Bean wird nicht als Shard-Bean konfiguriert.

ProtoBufSamplePlugins-Gemini-3.0.0.jar
Version 3.0.0 des Benutzer-Plug-in-Bundles mit den Beispiel-Plug-in-Implementierungen ObjectGridEventListener und MapSerializerPlugin. Dieses Bundle befindet sich im Verzeichnis *wxs_sample_osgi_root*/lib. Die Services werden mit dem Service-Ranking 3 konfiguriert.

Diese Version verwendet die Eclipse-Gemini-spezifische Blueprint-XML für die Konfiguration der Plug-in-Services von eXtreme Scale. Die Serviceklasse verwendet die integrierte Klasse com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl von von WebSphere eXtreme Scale, die den Service BlueprintContainer verwendet. Für die Konfiguration einer Shard-Bean wird ein Gemini-spezifischer Ansatz verwendet. Diese Version konfiguriert die Bean myShardListener als ShardBean, indem `{http://www.ibm.com/schema/objectgrid}shard` als Wert für den Geltungsbereich angegeben und ein Pseudoattribut konfiguriert wird, damit der angepasste Geltungsbereich von Gemini erkannt wird. Dies ist auf das folgende Eclipse-Problem zurückzuführen: https://bugs.eclipse.org/bugs/show_bug.cgi?id=348776

ProtoBufSamplePlugins-Aries-4.0.0.jar
Version 4.0.0 des Benutzer-Plug-in-Bundles mit den Beispiel-Plug-in-Imple-

mentierungen ObjectGridEventListener und MapSerializerPlugin. Dieses Bundle befindet sich im Verzeichnis *wxs_sample_osgi_root*/lib. Die Services werden mit dem Service-Ranking 4 konfiguriert.

Diese Version verwendet Blueprint-Standard-XML für die Konfiguration der Plug-in-Services von eXtreme Scale. Die Serviceklasse verwendet die integrierte Klasse com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl von von WebSphere eXtreme Scale, die den Service Blueprint-Container verwendet. Mithilfe der Blueprint-XML-Standardkonfiguration können die Beans mit einem angepassten Geltungsbereich konfiguriert werden. Diese Version konfiguriert myShardListenerbean als Shard-Bean, indem {http://www.ibm.com/schema/objectgrid}shard als Wert für den Geltungsbereich angegeben wird.

ProtoBufSamplePlugins-Activator-5.0.0.jar
Version 5.0.0 des Benutzer-Plug-in-Bundles mit den Beispiel-Plug-in-Implementierungen ObjectGridEventListener und MapSerializerPlugin. Dieses Bundle befindet sich im Verzeichnis *wxs_sample_osgi_root*/lib. Die Services werden mit dem Service-Ranking 5 konfiguriert.

Diese Version verwendet gar keine Blueprint-Container. In dieser Version werden die Services mithilfe der OSGi-Serviceregistrierung registriert. Die Serviceklasse ist eine vom Benutzer implementierte Klasse für die Schnittstelle "com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory" von WebSphere eXtreme Scale. Die vom Benutzer implementierte Klasse erstellt eine Bean für jede Anforderung. Sie funktioniert ähnlich wie eine Prototyp-Bean.

Prüfpunkt der Lerneinheit:

Indem Sie die Bundles, die mit dem OSGi-Beispiel bereitgestellt werden, untersucht werden, verstehen Sie besser, wie eigene Implementierungen entwickelt werden, die im OSGi-Container ausgeführt werden.

Sie haben Folgendes gelernt:

- Bundles, die mit dem OSGi-Beispiel bereitgestellt werden
- Position dieser Bundles
- Service-Ranking, mit dem jedes Bundle konfiguriert wurde

Lerneinheit 1.2: OSGi-Konfigurationsdateien verstehen

Das OSGi-Beispiel enthält drei Konfigurationsdateien. Sie verwenden diese Dateien, um das Grid und den Server von WebSphere eXtreme Scale zu starten und zu konfigurieren.

OSGi-Konfigurationsdateien:

In dieser Lerneinheit machen Sie sich mit den folgenden Konfigurationsdateien vertraut:

- collocated.server.properties
- protoBufObjectGrid.xml
- protoBufDeployment.xml

collocated.server.properties

Eine Serverkonfiguration ist zum Starten eines Servers erforderlich. Wenn das Server-Bundle von eXtreme Scale gestartet wird, wird kein Server gestartet. Das Bund-

le wartet, bis die Konfigurations-PID, com.ibm.websphere.xs.server, mit einer Servereigenschaftendatei erstellt wird. Diese Servereigenschaftendatei gibt den Servernamen, die Portnummer und weitere Servereigenschaften an.

In den meisten Fällen erstellen Sie eine Konfiguration, um die Servereigenschaftendatei zu definieren. In seltenen Fällen wird ein Server mit den Standardwerten der Eigenschaften gestartet. Wenn Sie einen Server mit den Standardwerten starten möchten, können Sie eine Konfiguration mit dem Namen com.ibm.websphere.xs.server erstellen, die den Wert default hat.

Weitere Einzelheiten zur Servereigenschaftendatei finden Sie unter Servereigenschaftendatei.

Das OSGi-Beispiel enthält die Beispielservereigenschaftendatei *wxs_sample_osgi_root*/server/properties/collocated.server.properties. Diese Beispieleigenschaftendatei startet einen einzigen Katalogservice und einen Container-Server im OSGi-Frameworkprozess. eXtreme-Scale-Clients stellen eine Verbindung zu Port 2809 und JMX-Clients eine Verbindung zu Port 1099 her. Im Folgenden sehen Sie den Inhalt der Beispielservereigenschaftendatei:

```
serverName=collocatedServer
isCatalog=true
catalogClusterEndPoints=collocatedServer:localhost:6601:6602
traceSpec=ObjectGridOSGi=all=enabled
traceFile=logs/trace.log
listenerPort=2809
JMXServicePort=1099
```

protoBufObjectGrid.xml

Im Folgenden sehen Sie den Inhalt der Beispiel-ObjectGrid-XML-Deskriptordatei protoBufObjectGrid.xml, in dem die Kommentare entfernt wurden:

```
<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>
```

Es sind zwei Plug-ins in dieser ObjectGrid-XML-Deskriptordatei konfiguriert:

ObjectGridEventListener
Das Plug-in auf Shard-Ebene. Für jede ObjectGrid-Instanz gibt es eine Instanz von ObjectGridEventListener. Die Instanz ist für die Verwendung des OSGi-Service myShardListener konfiguriert, d. h., wenn das Grid erstellt

wird, verwendet das Plug-in ObjectGridEventListener den OSGi-Service myShardListener mit dem höchsten verfügbaren Service-Ranking.

MapSerializerPlugin
Das Plug-in auf Map-Ebene. Für die BackingMap `Map` ist ein Plug-in MapSerializerPlugin konfiguriert. Dieses Plug-in ist für die Verwendung des OSGI-Service myProtoBufSerializer konfiguriert, d. h., wenn die Map erstellt wird, verwendet das Plug-in "MapSerializerPlugin" den Service myProtoBufSerializer mit dem höchsten verfügbaren Service-Ranking.

protoBufDeployment.xml

Die XML-Implementierungsdeskriptordatei beschreibt die Implementierungsrichtlinie für das Grid `Grid`, das fünf Partitionen verwendet. Im Folgenden sehen Sie ein Codebeispiel für diese XML-Datei:

```
<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

Anstelle der Datei `collocated.server.properties` in Verbindung mit der Konfigurations-PID `com.ibm.websphere.xs.server` können Sie alternativ die ObjectGrid-XML- und XML-Implementierungsdateien zusammen mit einer Blueprint-XML-Datei in ein OSGi-Bundle packen, wie im folgenden Beispiel gezeigt wird:

```
<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>
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie sich mit den Konfigurationsdateien vertraut gemacht, die im OSGi-Beispiel verwendet werden. Wenn Sie das eXtreme-Scale-Grid und den eXtreme-Scale-Server jetzt starten und konfigurieren, verstehen Sie, welche Dateien in diesen Prozessen verwendet werden und wie diese Dateien mit Ihren Plug-ins im OSGi-Framework interagieren.

Modul 2: eXtreme-Scale-Bundles im OSGi-Framework installieren und starten

Verwenden Sie die Module in diesen Lerneinheiten, um das eXtreme-Scale-Server-Bundle im OSGi-Container zu installieren und den Server von WebSphere eXtreme Scale zu starten.

Das Starten der Server im OSGi-Framework bedeutet nicht, dass Ihre OSGi-Bundles ausgeführt werden können. Sie müssen die Servereigenschaften und Container so konfigurieren, dass die OSGi-Bundles, die Sie installieren, erkannt werden und ordnungsgemäß ausgeführt werden können.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie mit den Konzepten vertraut und wissen, wie die folgenden Aufgaben ausgeführt werden:

- eXtreme-Scale-Bundles über die Equinox-OSGi-Konsole installieren
- eXtreme-Scale-Server konfigurieren
- eXtreme-Scale-Container konfigurieren
- eXtreme-Scale-Beispielbundles starten

Voraussetzungen

Zum Durcharbeiten dieses Moduls müssen vorher die folgenden Aufgaben ausgeführt werden:

- Produkt eXtreme Scale installieren und entpacken
- Eclipse-Equinox-Umgebung einrichten

Außerdem müssen Sie für die Lerneinheiten in diesem Modul den Zugriff auf die folgenden Dateien vorbereiten:

- Bundle `objectgrid.jar`. Sie installieren dieses eXtreme-Scale-Bundle.
- Datei `collocated.server.properties`. Sie fügen der Konfigurationsdatei die Servereigenschaften hinzu:
- Sie können mit der Installation und dem Start der folgenden Bundles rechnen:
- `protobuf-java-2.4.0a-bundle.jar`
- `ProtoBufSamplePlugins-1.0.0.jar`
- `ProtoBufSamplePlugins-2.0.0.jar`

Lerneinheit 2.1: Konsole starten und das Server-Bundle von eXtreme Scale installieren

In dieser Lerneinheit verwenden Sie die Equinox-OSGi-Konsole, um WebSphere eXtreme Scale starten und installieren.

1. Verwenden Sie den folgenden Befehl, um die Equinox-OSGi-Konsole zu starten:

```
cd equinox_root

java -jar
plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
```

2. Nach dem Start der OSGI-Konsole setzen Sie den Befehl `ss` in der Konsole ab, und die folgenden Bundles werden gestartet:

Eclipse-Gemini-Ausgabe:

```
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
```

Apache-Aries-Ausgabe:

```
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. Installieren Sie das Bundle `objectgrid.jar`. Zum Starten eines Servers in der Java Virtual Machine (JVM) müssen Sie ein Server-Bundle von eXtreme Scale installieren. Dieses Server-Bundle von eXtreme Scale kann einen Server starten und Container erstellen. Verwenden Sie den folgenden Befehl, um die Datei `objectgrid.jar` zu installieren:

   ```
   osgi> install file:///wxs_home/lib/objectgrid.jar
   ```

   Sehen Sie sich das folgende Beispiel an:

   ```
   osgi> install
   file:///opt/wxs/ObjectGrid/lib/objectgrid.jar
   ```

   Equinox zeigt seine Bundle-ID an, z. B.:

   ```
   Bundle id is 19
   ```

   Hinweis: Ihre Bundle-ID kann anders sein. Der Dateipfad muss ein absoluter URL zum Bundlepfad sein. Relative Pfade werden nicht unterstützt.

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Equinox-OSGi-Konsole verwendet, um das Bundle `objectgrid.jar` zu installieren, das Sie später in diesem Lernprogramm verwenden, um einen Server zu starten und einen Container zu erstellen.

Lerneinheit 2.2: eXtreme-Scale-Server anpassen und konfigurieren

Verwenden Sie diese Lerneinheit, um die Servereigenschaften anzupassen und dem Server von WebSphere eXtreme Scale hinzuzufügen.

1. Bearbeiten Sie die Datei wxs_sample_osgi_root/server/properties/collocated.server.properties.
   a. Ändern Sie die Eigenschaft "workingDirectory" in equinox_root.
   b. Ändern Sie die Eigenschaft "traceFile" in equinox_root/logs/trace.log.
2. Speichern Sie die Datei.
3. Geben Sie die folgenden Codezeilen in der OSGI-Konsole ein, um die Serverkonfiguration aus der Datei zu erstellen:

```
osgi> cm create com.ibm.websphere.xs.server

osgi> cm put com.ibm.websphere.xs.server
objectgrid.server.props
wxs_sample_osgi_root/server/properties/collocated.serve
r.properties
```

4. Zum Anzeigen der Konfiguration führen Sie den folgenden Befehl aus:

```
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 objectgrid.server.props
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Datei wxs_sample_osgi_root/server/properties/collocated.server.properties bearbeitet, um Servereinstellungen wie das Arbeitsverzeichnis und die Position für die Traceprotokolldateien festzulegen.

Lerneinheit 2.3: eXtreme-Scale-Container konfigurieren

Führen Sie diese Übung aus, um einen Container zu konfigurieren, der die ObjectGrid-XML-Deskriptordatei und die ObjectGrid-XML-Implementierungsdatei von WebSphere eXtreme Scale enthält. Diese Dateien enthalten die Konfiguration für das Grid und dessen Topologie.

Zum Erstellen eines Containers erstellen Sie zuerst unter Verwendung der Prozessidentifikationsnummer (PID) der Managed-Service-Factory, com.ibm.websphere.xs.container, einen Konfigurationsservice. Die Servicekonfiguration ist eine Managed-Service-Factory. Deshalb können Sie mehrere Service-PIDs aus der Factory-PID erstellen. Anschließend setzen Sie zum Starten des Container-Service die objectgridFile- und deploymentPolicyFile-PIDs auf die jeweilige Service-PID.

Führen Sie die folgenden Schritte aus, um die Servereigenschaften anzupassen und dem OSGi-Framework hinzuzufügen:

1. Geben Sie in der OSGI-Konsole den folgenden Befehl ein, um den Container aus der Datei zu erstellen:

```
osgi> cm createf com.ibm.websphere.xs.container
PID: com.ibm.websphere.xs.container-1291179621421-0
```

2. Geben Sie den folgenden Befehl ein, um die neu erstellte PID an die ObjectGrid-XML-Dateien zu binden.

   Hinweis: Die PID-Nummer ist eine andere als die in diesem Beispiel verwendete.

```
osgi> cm put com.ibm.websphere.xs.container-1291179621421-0
objectgridFile wxs_sample_osgi_root/server/META-INF/protoBufObjectgrid.xml

osgi> cm put com.ibm.websphere.xs.container-1291179621421-0
deploymentPolicyFile wxs_sample_osgi_root/server/META-INF/protoBufDeployment.xml
```

3. Verwenden Sie den folgenden Befehl, um die Konfiguration anzuzeigen:

```
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
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie einen Konfigurationsservice erstellt, den Sie verwendet haben, um einen eXtreme-Scale-Container zu erstellen. Da die ObjectGrid-XML-Dateien die Konfiguration für das Grid und dessen Topologie enthält, mussten Sie den Container, den Sie erstellt haben, an diese ObjectGrid-XML-Dateien binden. Mit dieser Konfiguration kann der eXtreme-Scale-Container die OSGi-Bundles erkennen, die Sie später in diesem Lernprogramm ausführen.

Lerneinheit 2.4: Google Protocol Buffers und Beispiel-Plug-in-Bundles installieren

Arbeiten Sie dieses Lernprogramm durch, um das Bundle `protobuf-java-2.4.0a-bundle.jar` und das Plug-in-Bundle `ProtoBufSamplePlugins-1.0.0.jar` über die Equinox-OSGi-Konsole zu installieren.

Führen Sie die folgenden Schritte aus, um das Bundle für Google Protocol Buffers zu installieren.

Geben Sie in der OSGI-Konsole den folgenden Befehl ein, um das Bundle zu installieren:

```
osgi> install file:///wxs_sample_osgi_root/common/lib/com.google.protobuf_2.4.0a.jar
```

Die folgende Ausgabe wird angezeigt:

```
Bundle ID is 21
```

Übersicht über die Beispiel-Plug-in-Bundles:

Das OSGi-Beispiel enthält fünf Beispielbundles mit eXtreme-Scale-Plug-ins, einschließlich angepasster ObjectGridEventListener- und MapSerializerPlugin-Plug-ins. Das MapSerializerPlugin-Plug-in verwendet das Beispiel für Google Protocol Buffers und Nachrichten, die vom MapSerializerPlugin-Beispiel bereitgestellt werden.

Die folgenden Bundles befinden sich im Verzeichnis *wxs_sample_osgi_root*`/lib`: `ProtoBufSamplePlugins-1.0.0.jar` und `ProtoBufSamplePlugins-2.0.0.jar`.

Im Folgenden sehen Sie den Inhalt der Datei `blueprint.xml`, in dem die Kommentare entfernt wurden:

```
<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>
```

Die Blueprint-XML-Datei exportiert zwei Services, myShardListener und myProtoBufSerializer. Diese beiden Services werden in der Datei `protoBufObjectgrid.xml` referenziert.

Beispiel-Plug-in-Bundle installieren:

Führen Sie die folgenden Schritte aus, um das Bundle `ProtoBufSamplePlugins-1.0.0.jar` zu installieren.

Führen Sie den folgenden Befehl in der Equinox-OSGi-Konsole aus, um das Plug-in-Bundle `ProtoBufSamplePlugins-1.0.0.jar` zu installieren:

```
osgi> install file:///wxs_sample_osgi_root/common/lib/ProtoBufSamplePlugins-1.0.0.jar
```

Die folgende Ausgabe wird angezeigt:

```
Bundle ID is 22
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie das Bundle `protobuf-java-2.4.0a-bundle.jar` und das Plug-in-Bundle `ProtoBufSamplePlugins-1.0.0.jar` installiert.

Lerneinheit 2.5: OSGi-Bundles starten

Der Server von WebSphere eXtreme Scale wird als OSGi-Server-Bundle gepackt. Arbeiten Sie diese Lerneinheit durch, um das Server-Bundle von eXtreme Scale sowie andere OSGi-Bundles, die Sie installiert haben, zu installieren.

1. Starten Sie das Beispiel-Plug-in-Bundle. Führen Sie den folgenden Befehl in der Equinox-OSGi-Konsole aus, um das Bundle zu starten. In diesem Beispiel ist die Bundle-ID des Beispiel-Plug-ins 22.

   ```
   osgi> start 22
   ```

2. Starten Sie das Bundle für Google Protocol Buffers. Führen Sie den folgenden Befehl in der Equinox-OSGi-Konsole aus, um das Bundle zu starten. In diesem Beispiel ist die Bundle-ID des Plug-ins für Google Protocol Buffers 21.

   ```
   osgi> start 21
   ```

3. Starten Sie das Server-Bundle. Führen Sie den folgenden Befehl in der OSGi-Konsole aus, um den Server zu starten. In diesem Beispiel ist die Bundle-ID des Server-Bundles von eXtreme Scale 19.

   ```
   osgi> start 19
   ```

Nachdem Sie den Server gestartet haben, wird der Ereignislistener MyShardListener gestartet und ist danach für das Einfügen oder Aktualisieren von Datensätzen bereit. Sie können die folgende Ausgabe in der OSGI-Konsole suchen, um sicherzustellen, dass das Plug-in-Bundle erfolgreich gestartet wurde:

```
SystemOut O MyShardListener@1253853884(version=1.0.0) order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Order$Builder
@1aba1aba(22) inserted
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie zwei Plug-in-Bundles und das Server-Bundle in dem eXtreme-Scale-Container gestartet, den Sie für das OSGi-Framework konfiguriert haben.

Modul 3: Beispielclient von eXtreme Scale ausführen

Der Server von WebSphere eXtreme Scale wird jetzt in einer OSGi-Umgebung ausgeführt. Führen Sie die Schritte in diesem Modul aus, um einen Client von WebSphere eXtreme Scale auszuführen, der Daten in das Grid einfügt.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie in der Lage, die folgenden Aufgaben auszuführen:

- Clientanwendung ausführen, die eine Verbindung zum Grid herstellt und Daten in das Grid einfügt und Daten aus dem Grid abruft
- Auftrag über eine Nicht-OSGi-Clientanwendung starten

Voraussetzungen

Arbeiten Sie das Modul 2: eXtreme-Scale-Bundles im OSGi-Framework installieren und starten durch.

Lerneinheit 3.1: Eclipse für die Ausführung des Clients und die Erstellung der Beispiele einrichten

Arbeiten Sie diese Lerneinheit durch, um das Eclipse-Projekt zu importieren, das Sie verwenden, um den Client auszuführen und die Beispiel-Plug-ins zu erstellen.

Das Beispiel enthält ein Java-SE-Clientprogramm, das eine Verbindung zum Grid herstellt und Daten in das Grid einfügt und Daten aus dem Grid abruft. Außerdem enthält es Projekte, die Sie verwenden können, um die OSGi-Bundles zu erstellen und erneut zu implementieren.

Das bereitgestellte Projekt wurde mit Eclipse 3.x und höher getestet und erfordert nur die Perspektive für Java-Standardentwicklungsprojekte. Führen Sie die folgenden Schritte aus, um Ihre Entwicklungsumgebung von WebSphere eXtreme Scale einzurichten.

1. Öffnen Sie einen neuen oder vorhandenen Arbeitsbereich in Eclipse.
2. Wählen Sie im Menü "File" die Option **Import** aus.
3. Erweitern Sie den Ordner "General". Wählen Sie **Existing Projects into Workspace** aus, und klicken Sie auf **Next**.
4. Geben Sie im Feld **Select root directory** das Verzeichnis *wxs_sample_osgi_root* ein, bzw. navigieren Sie dorthin. Klicken Sie auf **Finish**. Es werden mehrere neue Projekte im Arbeitsbereich angezeigt. Sie müssen mehrere Buildfehler beheben, indem Sie die Benutzerbibliothek von eXtreme Scale definieren. Führen Sie die nächsten Schritte aus, um die Benutzerbibliothek zu definieren.
5. Wählen Sie im Menü "Window" die Option **Preferences** aus.
6. Erweitern Sie den Zweig **Java** > **Build Path**, und wählen Sie **User Libraries** aus.
7. Klicken Sie auf **New**.
8. Geben Sie `eXtremeScale` im Feld **User Library Name** ein, und klicken Sie auf **OK**.
9. Wählen Sie neue Benutzerbibliothek aus, und klicken Sie auf **Add JARs**.
   a. Navigieren Sie zur Datei `objectgrid.jar` im Verzeichnis *WXS-Installationsstammverzeichnis*`/lib`, und wählen Sie sie aus. Klicken Sie auf **OK**.

b. Wenn Sie die API-Dokumentation für die ObjectGrid-APIs einschließen möchten, wählen Sie die Position der API-Dokumentation für die Datei `objectgrid.jar` aus, die Sie im vorherigen Schritt hinzugefügt haben. Klicken Sie auf **Edit**.
c. Wählen Sie im Verzeichnispfadfeld für die API-Dokumentation die Datei `Javadoc.zip` aus die im folgenden Verzeichnis enthalten ist: *WXS-Installationsstammverzeichnis*`/docs/javadoc.zip`.

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie das Eclipse-Beispielprojekt importiert, die Benutzerbibliothek von eXtreme Scale definiert und unterstützende API-Dokumentation für das Beispielprojekt eingeschlossen. Jetzt können Sie die Beispielclientanwendung starten.

Lerneinheit 3.2: Client starten und Daten in das Grid einfügen

Arbeiten Sie diese Lerneinheit durch, um einen Nicht-OSGi-Client zu starten und eine Clientanwendung auszuführen.

Die Java-Clientanwendung ist `com.ibm.websphere.samples.xs.proto.client.Client`.

Dieser Client verwendet eine ObjectGrid-XML-Deskriptor (Clientüberschreibung), um die OSGi-Konfiguration zu überschreiben, so dass der Client in einer Nicht-OSGi-Umgebung ausgeführt werden kann. Im Folgenden sehen Sie den Inhalt der Datei, in dem Kommentare und Header entfernt wurden. Einige Codezeilen werden aus Formatierungsgründen in mehreren Zeilen angezeigt.

```
<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>
```

Führen Sie die folgenden Schritte aus, um die Clientanwendung zu starten.

1. Verwenden Sie das folgende Codebeispiel, um die Attribute der Client-Klasse an Ihre Umgebung anzupassen.

```
private String catHost = "localhost";
private int catListenerPort = 2809;
private String clientOGXML = "wxs_sample_osgi_root/client/META-INF/
clientProtoBufObjectgrid.xml";
private String gridName = "Grid";
private String mapName = "Map";
```

2. Führen Sie die Anwendung aus.

Wenn Sie die Anwendung ausführen, erscheint die folgende Nachricht. Die Nachricht gibt an, dass ein Auftrag eingefügt wurde:

```
order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Or
der$Builder@5d165d16(5000000) inserted
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Anwendung `com.ibm.websphere.samples.xs.proto.client.Client` gestartet, die einen Auftrag erstellt.

Modul 4: Beispielbundle abfragen und aktualisieren

Arbeiten Sie die Lerneinheiten in diesem Modul durch, um mit dem Befehl **xscmd** das Service-Ranking des Beispielbundles abzufragen, das Bundle auf ein neues Service-Ranking zu aktualisieren und das neue Service-Ranking zu überprüfen

Um die Ausführung der Beispielanwendungen zu vereinfachen wird ein Eclipse-Projekt bereitgestellt.

Lernziele

Nach der Ausführung der Lerneinheiten in diesem Modul sind Sie in der Lage, die folgenden Aufgaben auszuführen:

- Aktuelles Service-Ranking für einen Service abfragen
- Aktuelles Ranking für alle Services abfragen
- Alle verfügbaren Rankings für einen Service abfragen
- Alle verfügbaren Service-Rankings abfragen
- Tool "xscmd" verwenden, um festzustellen, ob bestimmte Service-Rankings verfügbar sind
- Service-Rankings für OSGi-Beispielservices aktualisieren

Voraussetzungen

Arbeiten Sie Modul 3: eXtreme-Scale-Beispielclient ausführen durch.

Lerneinheit 4.1: Service-Rankings für Abfragen

Arbeiten Sie diese Lerneinheit durch, um aktuelle Service-Rankings und Service-Rankings abzufragen, die für ein Upgrade verfügbar sind.

- Aktuelles Service-Ranking für einen Service abfragen. Geben Sie den folgenden Befehl ein, um das aktuelle Service-Ranking abzufragen, das für den Service myShardListener verwendet wird, der vom ObjectGrid `Grid` und dem MapSet `MapSet` verwendet wird.
  1. Wechseln Sie in das folgende Verzeichnis:
     ```
     cd wxs_home/bin
     ```

2. Geben Sie den folgenden Befehl ein, um das aktuelle Service-Ranking für den Service myShardListener abzufragen.

   ```
   ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet -sn myShardListener
   ```

   Die folgende Ausgabe wird angezeigt:

   ```
   OSGi Service Name: myShardListener
   ObjectGrid Name MapSet Name Server Name        Current Ranking
   --------------- ----------- -----------        ---------------
   Grid            MapSet      collocatedServer   1

   CWXSI0040I: Der Befehl osgiCurrent wurde erfolgreich ausgeführt.
   ```

- Aktuelles Ranking für alle Services abfragen. Geben Sie den folgenden Befehl ein, um das aktuelle Service-Ranking für alle Services abzufragen, die vom ObjectGrid `Grid` und vom MapSet `MapSet` verwendet werden.
  1. Wechseln Sie in das folgende Verzeichnis:

     ```
     cd wxs_home/bin
     ```

  2. Geben Sie den folgenden Befehl ein, um das aktuelle Service-Ranking für alle Services abzufragen.

     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
     ```

     Die folgende Ausgabe wird angezeigt:

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- ---------------- --------------- ----------- -----------
myProtoBufSerializer 1              Grid            MapSet      collocatedServer
myShardListener      1              Grid            MapSet      collocatedServer

CWXSI0040I: Der Befehl osgiCurrent wurde erfolgreich ausgeführt.
```

- Alle verfügbaren Rankings für einen Service abfragen. Geben Sie den folgenden Befehl ein, um alle verfügbaren Service-Rankings für den Service myShardListener abzufragen.
  1. Wechseln Sie in das folgende Verzeichnis:

     ```
     cd wxs_home/bin
     ```

  2. Geben Sie den folgenden Befehl ein, um alle verfügbaren Rankings für einen Service abzufragen.

     ```
     ./xscmd.sh -c osgiAll -sn myShardListener
     ```

     Die folgende Ausgabe wird angezeigt:

     ```
     Server: collocatedServer
        OSGi Service Name Available Rankings
        ----------------- ------------------
        myShardListener   1

        Summary - All servers have the same service rankings.

     CWXSI0040I: The command osgiAll has completed successfully.
     ```

     Die Ausgabe ist nach Server gruppiert. In diesem Beispiel ist nur der folgende Server vorhanden: collocatedServer.

- Alle verfügbaren Service-Rankings abfragen. Geben Sie den folgenden Befehl ein, um alle verfügbaren Service-Rankings für alle Services abzufragen.
  1. Wechseln Sie in das folgende Verzeichnis:

     ```
     cd wxs_home/bin
     ```

  2. Geben Sie den folgenden Befehl ein, um alle verfügbaren Service-Rankings abzufragen.

     ```
     ./xscmd.sh -c osgiAll
     ```

Die folgende Ausgabe wird angezeigt:

```
Server: collocatedServer
   OSGi Service Name          Available Rankings
   -----------------    ------------------
   myProtoBufSerializer 1
   myShardListener      1

   Summary - All servers have the same service rankings.
```

- Installieren und starten Sie Version 2 des Plug-in-Bundles. Installieren Sie in der OSGi-Konsole des Servers ein neues Bundle, das eine neue Version der Klasse "Order" und das Plug-in MapSerializerPlugin enthält. Einzelheiten zum Installieren des Bundles `ProtoBufSamplePlugins-2.0.0.jar` finden Sie in Lerneinheit 2.4: Bundles für Google Protocol Buffers und Beispiel-Plug-ins installieren.
  1. Starten Sie nach der Installation das neue Bundle. Die Services für Ihr neues Bundle sind verfügbar, aber sie werden noch nicht vom eXtreme-Scale-Server verwendet. Sie müssen eine Serviceaktalisierungsanforderung ausführen, um einen Service mit einer bestimmten Version verwenden zu können.
- Wenn Sie alle verfügbaren Service-Rankings erneut abfragen, wird das Service-Ranking 2 der Ausgabe hinzugefügt.
  1. Wechseln Sie in das folgende Verzeichnis:

     ```
     cd wxs_home/bin
     ```

  2. Geben Sie den folgenden Befehl ein, um alle verfügbaren Service-Rankings abzufragen.

     ```
     ./xscmd.sh -c osgiAll
     ```

     Die folgende Ausgabe wird angezeigt:

     ```
     Server: collocatedServer
        OSGi Service Name          Available Rankings
        -----------------    ------------------
        myProtoBufSerializer 1, 2
        myShardListener      1, 2

     Summary - All servers have the same service rankings.
     ```

Prüfpunkt der Lerneinheit:

In diesem Lernprogramm haben Sie momentan angegebene und alle verfügbaren Service-Rankings abgefragt. Außerdem haben Sie das Service-Ranking für ein neues Bundle angezeigt, das Sie installiert und gestartet haben.

Lerneinheit 4.2: Bestimmen, ob bestimmte Service-Rankings verfügbar sind.

Arbeiten Sie dieses Lernprogramm durch, um festzustellen, ob bestimmte Service-Rankings für die angegebenen Servicenamen verfügbar sind.

1. Geben Sie den folgenden Befehl ein, um festzustellen, ob der Service myShardListener mit dem Service-Ranking 2 und der Service mit dem Namen myProtoBufSerializer mit dem Service-Ranking 2 verfügbar sind. Sie Liste der Service-Rankings wird mit der Option `-sr` übergeben.
   a. Wechseln Sie in das folgende Verzeichnis:

      ```
      cd wxs_home/bin
      ```

   b. Geben Sie den folgenden Befehl ein, um festzustellen, ob die Services verfügbar sind:

      ```
      ./xscmd.sh -c osgiCheck -g Grid -ms MapSet -sr
      "myShardListener;2,myProtoBufSerializer;2"
      ```

Die folgende Ausgabe wird angezeigt:

```
CWXSI0040I: The command osgiCheck has completed successfully.
```

2. Geben Sie den folgenden Befehl ein, um festzustellen, ob der Service myShardListener mit dem Service-Ranking 2 und der Service mit dem Namen myProtoBufSerializer mit dem Service-Ranking 3 verfügbar sind.
   a. Wechseln Sie in das folgende Verzeichnis:

      ```
      cd wxs_home/bin
      ```

   b. Geben Sie den folgenden Befehl ein, um festzustellen, ob die Services verfügbar sind:

      ```
      ./xsadmin.sh -c osgiCheck -g Grid -ms MapSet -sr
      "myShardListener;2,myProtoBufSerializer;3"
      ```

      Die folgende Ausgabe wird angezeigt:

      ```
      Server  OSGi Service Unavailable Rankings
      ------ ------------ --------------------
      collocatedServer myProtoBufSerializer 3
      ```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Services myShardListener und myProtoBufSerializer zusammen mit bestimmten Service-Rankings angegeben, um festzustellen, ob diese Rankings verfügbar sind.

Lerneinheit 4.3: Service-Rankings aktualisieren

Arbeiten Sie diese Lerneinheit durch, um aktuelle Service-Rankings zu aktualisieren, die Sie abgefragt haben.

1. Geben Sie die folgenden Befehl ein, um die Service-Rankings der Services myShardListener und myProtoBufSerializer auf Service-Ranking 2 zu aktualisieren. Die Liste der Service-Rankings wird mit der Option `-sr` übergeben.
   a. Wechseln Sie in das folgende Verzeichnis:

      ```
      cd wxs_home/bin
      ```

   b. Geben Sie den folgenden Befehl ein, um die Service-Rankings zu aktualisieren:

      ```
      ./xscmd.sh -c osgiUpdate -g Grid -ms MapSet
      -sr "myShardListener;2,myProtoBufSerializer;2"
      ```

      Die folgende Ausgabe wird angezeigt:

      ```
      Update succeeded for the following service rankings:
      Service Ranking
      ------- -------
      myProtoBufSerializer 2
      myShardListener 2

      CWXSI0040I: Der Befehl osgiUpdate wurde erfolgreich ausgeführt.
      ```

      Die folgende Ausgabe wird in der OSGi-Konsole angezeigt:

      ```
      SystemOut O MyShardListener@326505334(version=2.0.0) order
      com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order$Builder@
      22342234(34) updated
      ```

      Sie sehen, dass der Service MyShardListener jetzt Version 2.0.0 hat, die Service-Ranking 2 hat.
2. Wenn Sie den Befehl **`xscmd`** ausführen, um das aktuelle Service-Ranking abzufragen, die für alle vom ObjectGrid `Grid` und vom MapSet `MapSet` verwendeten Services verwendet wird.

a. Wechseln Sie in das folgende Verzeichnis:

```
cd wxs_home/bin
```

b. Geben Sie den folgenden Befehl ein, um die Service-Rankings für alle Services abzufragen, die von `Grid` und `MapSet` verwendet werden:

```
./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
```

Die folgende Ausgabe wird angezeigt:

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
---------------- -------------- -------------- ----------- -----------
myProtoBufSerializer 2 Grid MapSet collocatedServer
myShardListener 2 Grid MapSet collocatedServer

CWXSI0040I: Der Befehl osgiCurrent wurde erfolgreich ausgeführt.
```

Prüfpunkt der Lerneinheit:

In dieser Lerneinheit haben Sie die Service-Rankings für die Services myShardListener und myProtoBufSerializer aktualisiert.

.

Kapitel 4. Installation

WebSphere eXtreme Scale ist ein speicherinternes Datengrid, das Sie verwenden können, um Anwendungsdaten und Geschäftslogik in mehreren Servern zu partitionieren, zu replizieren und zu verwalten. Nachdem Sie den Zweck und die Anforderungen Ihrer Implementierung bestimmt haben, installieren Sie eXtreme Scale auf Ihrem System.

Vorbereitungen

- Bevor Sie mit der Installation beginnen, müssen Sie mit den Cachingarchitekturen von WebSphere eXtreme Scale, Cache- und Datenbankintegration, Serialisierung, Skalierbarkeit und Verfügbarkeit vertraut sein. Weitere Informationen finden Sie in Produktübersicht.
- Planen Sie ihre Implementierung von WebSphere eXtreme Scale. Weitere Informationen zu den verschiedenen Cachingtopologien, zur Kapazitätsplanung usw. finden Sie unter Kapitel 2, „Planung", auf Seite 9.
- Vergewissern Sie sich, dass Ihre Umgebung die Voraussetzungen für die Installation von eXtreme Scale erfüllt. Weitere Informationen finden Sie im Abschnitt „Hardware- und Softwarevoraussetzungen" auf Seite 50.
- Weitere Informationen zu Umgebungen und anderen Voraussetzungen finden Sie unter „Installation planen" auf Seite 50.
- Wenn Sie ein Upgrade in einer früheren Version von WebSphere eXtreme Scale installieren, führen Sie die unter „eXtreme-Scale-Server aktualisieren" auf Seite 217 beschriebenen Schritte aus.

Installationsübersicht

Sie können die vollständige Installation oder die Clientinstallation verwenden, u WebSphere eXtreme Scale in einer eigenständigen Umgebung oder in einer Umgebung von WebSphere Application Server zu installieren.

Installationstypen

Das Installationsprogramm für die vollständige Installation und das gesonderte Clientinstallationsprogramm, das Sie von der Unterstützungssite herunterladen können, bieten Ihnen eine Vielzahl von Installationsoptionen. Wenn Sie das Installationsprogramm für die vollständige Installation verwenden, können Sie Katalogserver und Container-Server ausführen. In den Servern, in denen Clientanwendungen ausgeführt werden, die auf das Datengrid zugreifen, können Sie eine reine Clientinstallation verwenden. Verwenden Sie die Serverinstallation oder die Client/Server-Installation auf Knoten, auf denen Katalogserver oder Container-Server ausgeführt werden.

- **Vollständige Installation:**
  - Wenn Sie die Installation in WebSphere Application Server durchführen, können Sie nur den Client oder den Server und den Client installieren.
  - Wenn Sie die Installation in einer eigenständigen Umgebung durchführen, können Sie Client und Server installieren. Verwenden Sie die Installation von WebSphere eXtreme Scale Client, wenn Sie nur den Client installieren möchten.
- **Clientinstallation:**

Die können die reine Clientinstallation auf Knoten verwenden, auf denen die Clientanwendungen ausgeführt werden. Zu Installieren des Clients können Sie das Installationsprogramm für die reine Clientinstallation für die entsprechende Plattform über den Downloadabschnitt auf der Unterstützungssite herunterladen.

Umgebungsoptionen

Sie können WebSphere eXtreme Scale in einer eigenständigen Umgebung von WebSphere Application Server installieren.

- **Umgebung von WebSphere Application Server:**

  Durch die Installation von WebSphere eXtreme Scale auf den Knoten in Ihrer Umgebung von WebSphere Application Server können Sie Katalogserver und Container-Server automatisch in derselben Zelle wie Ihren Deployment Manager und Ihre anderen Anwendungsserver starten.
- **Eigenständige Umgebung:**

  In einer eigenständigen Installation installieren Sie WebSphere eXtreme Scale in einer Umgebung, in der WebSphere Application Server nicht installiert ist. Bei einer eigenständigen Umgebung konfigurieren und starten Sie die Katalogserver- und Container-Server-Prozesse manuell.

Installation planen

Vor der Installation des Produkts müssen Sie Ihre Umgebung berücksichtigen.

Installationstopologien

Mit WebSphere eXtreme Scale können Sie zahlreiche Installationstopologien erstellen, die eigenständige Server und/oder WebSphere Application Server enthalten. Die folgenden Beispiele stellen mögliche Topologien dar, die Sie erstellen können.

Entwicklungsknoten

Das einfachste Installationsszenario ist die Erstellung eines Entwicklungsknotens. In diesem Szenario installieren Sie die Client/Server-Installation von WebSphere eXtreme Scale einmal auf dem Knoten, auf dem Sie Ihre Anwendung entwickeln möchten.

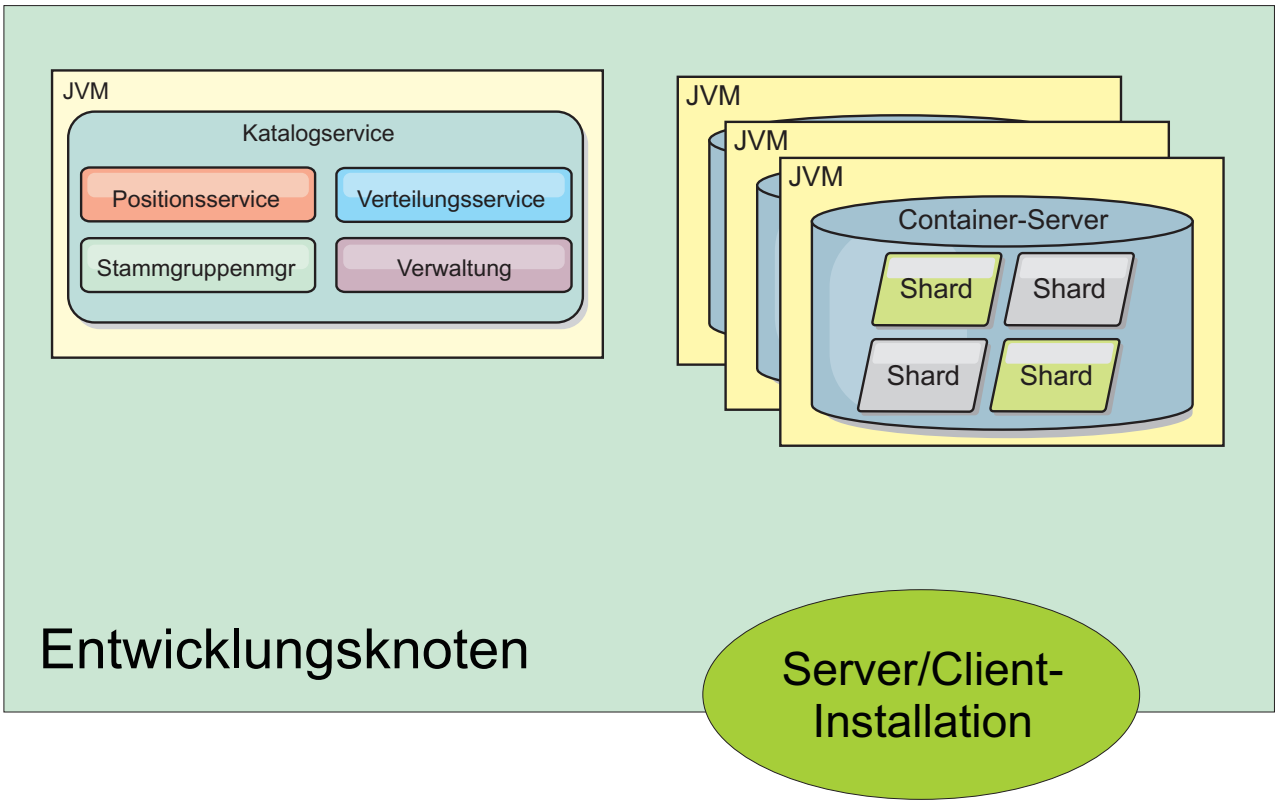


Abbildung 22. Entwicklungsknoten

Nach Abschluss der Installation auf Ihrem Entwicklungsknoten können Sie Ihre Entwicklungsumgebung konfigurieren und mit dem Schreiben Ihrer Anwendung beginnen.

Eigenständige Topologie

Eine eigenständige Topologie setzt sich aus Servern zusammen, die nicht in WebSphere Application Server ausgeführt werden. Sie können viele verschiedene eigenständige Topologien erstellen, aber die folgende Topologie wird als Beispiel bereitgestellt. In dieser Topologie sind zwei Rechenzentren enthalten. In jedem Rechenzentrum sind vollständige Installationen von WebSphere eXtreme Scale (Client und Server) und reine Clientinstallationen auf den physischen Servern installiert. Die reinen Clientinstallationen befinden sich auf den Knoten, auf denen die Webanwendungen ausgeführt werden, die das Datengrid verwenden. Auf diesen Knoten werden keine Katalog- oder Container-Server ausgeführt, und deshalb ist die Serverinstallation nicht erforderlich. Eine Multimasterverbindung verbindet die beiden Katalogservicedomänen in der Konfiguration. Die Multimasterverbindun ermöglicht die Replikation zwischen den Shards in den Container-Servern in den verschiedenen Rechenzentren.

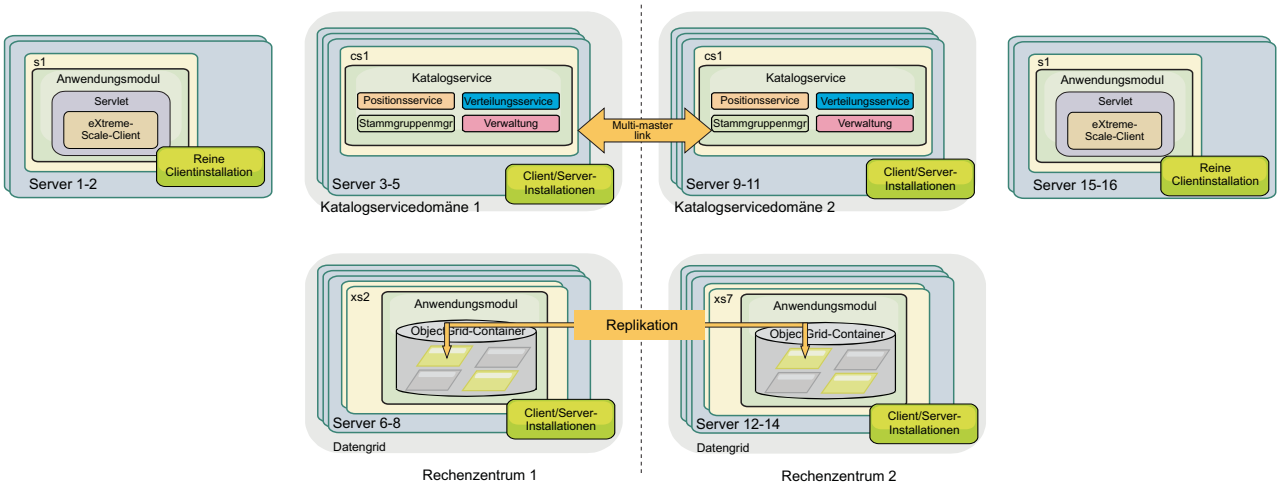


Abbildung 23. Eigenständige Topologie mit zwei Rechenzentren

Vorteile einer eigenständigen Topologie:

- Flexible Integrationsoptionen, die mit Frameworks und Bibliotheken anderer Anbieter integriert werden können
- Geringerer Speicherbedarf als bei einer Topologie mit WebSphere Application Server
- Weniger Lizenzvoraussetzungen als bei einer Topologie mit WebSphere Application Server
- Erweiterte JRE-Optionen (Java Runtime Environment)

Topologie mit WebSphere Application Server

Sie können auch eine Installation erstellen, die vollständig in einer Zelle von WebSphere Application Server ausgeführt wird. Die Clients, Katalogserver und Container-Server haben jeweils einen zugeordneten Cluster. Die Knoten, auf denen die Anwendung ausgeführt wird, haben eine reine Clientinstallation. Die anderen Knoten haben eine Client/Server-Installation.

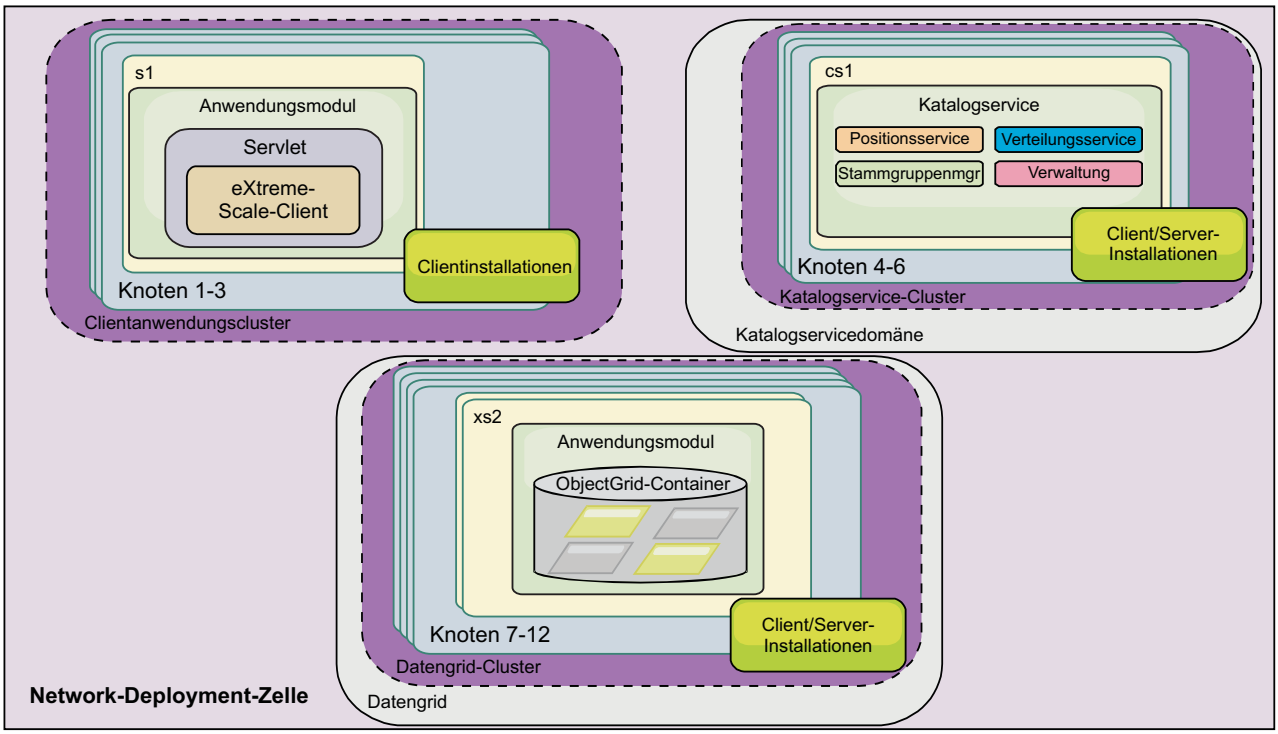


Abbildung 24. Beispiel für die Topologie mit WebSphere Application Server

Vorteile einer Topologie mit WebSphere Application Server.

- Zentrale und konsistente Verwaltung und Konfiguration
- Sicherheitsintegration
- Integration von Java-EE-Anwendungen
- PMI-Integration (Performance Monitoring Infrastructure)
- Integration mit den folgenden Komponenten von WebSphere Application Server: OpenJPA-L2-Cache, dynamischer Cache und HTTP-Sitzungspersistenz

Heterogene Topologie

Sie können eine heterogene Topologie einrichten, die WebSphere Application Server und eigenständige Server enthält. Im folgenden Beispiel werden die Clientanwendungen in der Zelle von WebSphere Application Server ausgeführt, während die Katalogserver und Container-Server im eigenständigen Modus ausgeführt werden.

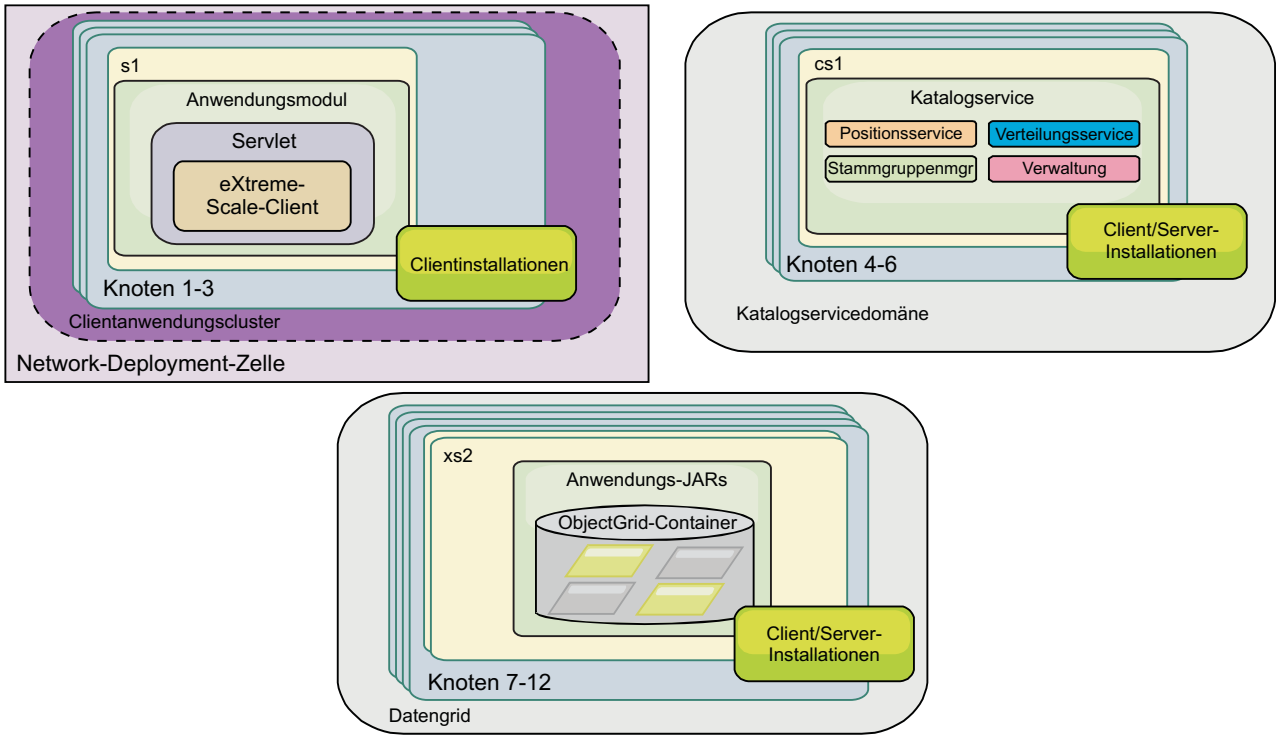


Abbildung 25. Beispiel für eine heterogene Topologie

Hardware- und Softwarevoraussetzungen

Dieser Abschnitt enthält eine Übersicht über die Hardware- und Betriebssystemvoraussetzungen. Sie müssen zwar keine bestimmte Version der Hardware oder des Betriebssystems für WebSphere eXtreme Scale verwenden, aber auf der Seite "Systems Requirements" der Produktunterstützungssite sind die formal unterstützten Hardware- und Softwareoptionen beschrieben. Sollte die Informationen im Information Center und auf der Seite "System Requirements" widersprüchlich sein, haben die Informationen auf der Website Vorrang. Die Informationen zu den Voraussetzungen im Information Center werden nur im Hinblick auf die Bedienerfreundlichkeit bereitgestellt.

Die offizielle Beschreibung der Hardware- und Softwarevoraussetzungen finden Sie auf der Webseite System Requirements.

Sie müssen eXtreme Scale nicht unter einer bestimmten Version des Betriebssystems installieren und implementieren. Jede Java-SE- (Java Platform, Standard Edition) und Java-EE-Installation (Java Platform, Enterprise Edition) erfordert andere Betriebssystemversionen oder -fixes.

Sie können das Produkt in Java-EE- und in Java-SE-Umgebungen installieren und implementieren. Außerdem können Sie die Clientkomponente direkt, ohne Integration mit WebSphere Application Server, mit JEE-Anwendungen bündeln. WebSphere eXtreme Scale unterstützt Java SE 5 und höher und WebSphere Application Server Version 6.1 und höher.

Hardwarevoraussetzungen

WebSphere eXtreme Scale setzt keine bestimmte Hardwareversion voraus. Die Hardwarevoraussetzungen richten sich nach der unterstützten Hardware für die Installation der Java Platform, Standard Edition, die Sie für die Ausführung von WebSphere eXtreme Scale verwenden. Wenn Sie eXtreme Scale mit WebSphere Application Server oder einer anderen Java-EE-Implementierung (Java Platform, Enterprise Edition) verwenden, sind die Hardwarevoraussetzungen dieser Plattformen für WebSphere eXtreme Scale ausreichend.

Betriebssystemvoraussetzungen

- **Ohne die Webkonsole**

  eXtreme Scale setzt keine bestimmte Betriebssystemversion voraus. Jede Java-SE- und jede Java-EE-Implementierung setzt verschiedene Betriebssystemversionen oder -Fixes für Probleme voraus, die während des Testens der Java-Implementierung erkannt werden. Die von diesen Implementierungen vorausgesetzten Versionen sind für eXtreme Scale ausreichend.
- **Mit der Webkonsole**

  Die folgenden Voraussetzungen gelten für jedes Betriebssystem, wenn die Konsole verwendet wird:
  - Linux: 32-Bit- oder 64-Bit-JVM
  - Linux PPC: nur 32-Bit-JVM
  - Windows: nur 32-Bit-JVM
  - AIX: nur 32-Bit-JVM

Web-Browser-Voraussetzungen

Die Webkonsole unterstützt die folgenden Web-Browser:

- Mozilla Firefox Version 3.5.x und höher
- Mozilla Firefox Version 3.6.x und höher
- Microsoft Internet Explorer Version 7 oder 8

Voraussetzungen in Bezug auf WebSphere Application Server

- WebSphere Application Server Version 6.1.0.39 oder höher
- WebSphere Application Server Version 7.0.0.19 oder höher
- WebSphere Application Server Version 8.0.0.1 oder höher

Weitere Informationen finden Sie auf der Webseite mit den empfohlenen Fixes für WebSphere Application Server.

Weitere Voraussetzungen für den Anwendungsserver

Andere Java-EE-Implementierungen können die Laufzeitumgebung von eXtreme Scale als lokale Instanz oder als Client für Server von eXtreme Scale verwenden. Zum Implementieren von Java SE müssen Sie Version 5 oder höher verwenden.

Hinweise zu Java SE

WebSphere eXtreme Scale setzt Java SE 5 oder höher voraus. Im Allgemeinen bieten neuere Versionen von Java SE eine bessere Funktionalität und Leistung.

Unterstützte Versionen

Sie können WebSphere eXtreme Scale mit Java SE 5 oder höher verwenden. Die Version, die Sie verwenden, muss vom JRE-Anbieter (Java Runtime Environment) vollständig unterstützt werden.

Eine vollständig unterstützte JRE wird im Rahmen der eigenständigen Installationen von WebSphere eXtreme Scale und WebSphere eXtreme Scale Client im Verzeichnis *WXS-Installationsstammverzeichnis*/java installiert und kann von Clients und Servern verwendet werden. Wenn Sie WebSphere eXtreme Scale in WebSphere Application Server installieren, können Sie die JRE verwenden, die in der Installation von WebSphere Application Server enthalten ist.

WebSphere eXtreme Scale nutzt die Funktionalität von Java Development Kit (JDK) 5 oder höher, sobald die diese Komponente verfügbar ist. Im Allgemeinen bieten neuere Versionen von Java Development Kit (JDK) und Java SE eine bessere Leistung und eine bessere Funktionalität.

Weitere Informationen finden Sie auf der Webseite Supported software.

Java-abhängige Features von WebSphere eXtreme Scale

Tabelle 4. Features, die Java SE 5 oder Java SE 6 voraussetzen.

WebSphere eXtreme Scale verwendet Funktionalität, die in Java SE 5 bzw. Java SE 6 eingeführt wurde, um die folgenden Produktfeatures bereitzustellen.

Feature	Unterstützt in Java SE 5 und höher	Unterstützt in Java SE 6 und höher
Annotationen der API "EntityManager" (optional: Sie können auch XML-Dateien verwenden)	X	X
Java Persistence API (JPA): JPA-Loader, JPA-Client-Loader und zeitbasierte JPA-Aktualisierungskomponente	X	X
Speicherbasierte Bereinigung (verwendet MemoryPoolMXBean)	X	X
Instrumentierungsagenten: • `wxssizeagent.jar`: Erhöht die Genauigkeit von Map-Metriken zur Bytebelegung. • `ogagent.jar`: Erhöht die Leistung von Entitäten mit Feldzugriff.	X	X
Webkonsole für die Überwachung		X

Hinweise zu Java EE

Bei der Vorbereitung der Integration von WebSphere eXtreme Scale in eine Java-EE-Umgebung müssen Sie bestimmte Punkte berücksichtigen, wie z. B. Versionen,

Konfigurationsoptionen, Voraussetzungen und Einschränkungen sowie Anwendungsimplementierung und -management.

eXtreme-Scale-Anwendungen in einer Java-EE-Umgebung ausführen

Eine Java-EE-Anwendung kann eine Verbindung zu einer fernen eXtreme-Scale-Anwendung herstellen. Außerdem unterstützt die Umgebung von WebSphere Application Server das Starten eines eXtreme-Scale-Servers beim Starten einer Anwendung im Anwendungsserver.

Wenn Sie eine XML-Datei zum Erstellen einer ObjectGrid-Instanz verwenden und die XML-Datei im Modul der EAR-Datei enthalten ist, greifen Sie mit der Methode getClass().getClassLoader().getResource("META-INF/objGrid.xml") auf die Datei zu, um ein URL-Objekt für die Erstellung einer ObjectGrid-Instanz abzurufen. Setzen Sie im Methodenaufruf den Namen der XML-Datei ein, die Sie verwenden.

Sie können Startup-Beans für eine Anwendung verwenden, um eine ObjectGrid-Instanz beim Starten einer Anwendung zu booten und um die Instanz beim Stoppen der Anwendung zu löschen. Eine Startup-Bean ist eine Stateless Session-Bean mit einer fernen com.ibm.websphere.startupservice.AppStartUpHome-Position und einer fernen com.ibm.websphere.startupservice.AppStartUp-Schnittstelle. Die ferne Schnittstelle hat zwei Methoden: die Methode start und die Methode stop. Verwenden Sie die Methode start, um die Instanz zu booten, und die Methode stop, um die Instanz zu löschen. Die Anwendung verwendet die Methode ObjectGridManager.getObjectGrid, um eine Referenz auf die Instanz zu verwalten. Weitere Einzelheiten finden Sie in den Informationen zum Zugriff auf ein ObjectGrid mit ObjectGridManager in der Veröffentlichung *Programmierung*.

Klassenladeprogramme verwenden

Wenn Anwendungsmodule, die unterschiedliche Klassenladeprogramme verwenden, eine einzige ObjectGrid-Instanz in einer Java-EE-Anwendung gemeinsam nutzen, müssen Sie sicherstellen, dass die Objekte, die in eXtreme Scale gespeichert werden, und die Plug-ins für das Produkt in einem gemeinsamen Loader der Anwendung enthalten sind.

Lebenszyklus von ObjectGrid-Instanzen in einem Servlet verwalten

Für die Verwaltung des Lebenszyklus einer ObjectGrid-Instanz in einem Servlet können Sie die Methode init verwenden, um die Instanz zu erstellen, und die Methode "destroy", um die Instanz zu entfernen. Wenn die Instanz zwischengespeichert ist, wird sie im Servlet-Code abgerufen und bearbeitet. Weitere Einzelheiten finden Sie in den Informationen zum Zugriff auf ein ObjectGrid mit der Schnittstelle ObjectGridManager in der Veröffentlichung *Programmierung*.

Verzeichniskonventionen

Die folgenden Verzeichniskonventionen werden in der Dokumentation verwendet, um auf spezielle Verzeichnisse zu verweisen, wie z.B. *WXS-Installationsstammverzeichnis* und *WXS-Ausgangsverzeichnis*. Sie greifen in verschiedenen Szenarien, wie z. B. während der Installation oder der Verwendung der Befehlszeilenprogramme, auf diese Verzeichnisse zu.

WXS-Installationsstammverzeichnis
Das Verzeichnis *WXS-Installationsstammverzeichnis* ist das Stammverzeichnis,

in dem die Produktdateien von WebSphere eXtreme Scale installiert sind. Das Verzeichnis *WXS-Installationsstammverzeichnis* kann das Verzeichnis sein, in dem das Testarchiv entpackt wurde, oder das Verzeichnis, in dem das Produkt WebSphere eXtreme Scale installiert ist.

- Beispiel für die entpackte Testversion:

  Beispiel: `/opt/IBM/WebSphere/eXtremeScale`
- Beispiel, wenn WebSphere eXtreme Scale in einem eigenständigen Verzeichnis installiert ist:

  Beispiel: `/opt/IBM/eXtremeScale`
- Beispiel, wenn WebSphere eXtreme Scale mit WebSphere Application Server integriert ist:

  Beispiel: `/opt/IBM/WebSphere/AppServer`

WXS-Ausgangsverzeichnis
Das Verzeichnis *WXS-Ausgangsverzeichnis* ist das Stammverzeichnis der Produktbibliotheken, Beispiele und Komponenten von WebSphere eXtreme Scale. Dieses Verzeichnis entspricht dem Verzeichnis *WXS-Installationsstammverzeichnis*, wenn die Testversion entpackt wurde. Bei eigenständigen Installationen ist das Verzeichnis *WXS-Ausgangsverzeichnis* das Unterverzeichnis `ObjectGrid` im Verzeichnis *WXS-Installationsstammverzeichnis*. Bei Installationen, die mit WebSphere Application Server integriert sind, ist dieses Verzeichnis das Verzeichnis `optionalLibraries/ObjectGrid` im Verzeichnis *WXS-Installationsstammverzeichnis*.

- Beispiel für die entpackte Testversion:

  Beispiel: `/opt/IBM/WebSphere/eXtremeScale`
- Beispiel, wenn WebSphere eXtreme Scale in einem eigenständigen Verzeichnis installiert ist:

  Beispiel: `/opt/IBM/eXtremeScale/ObjectGrid`
- Beispiel, wenn WebSphere eXtreme Scale mit WebSphere Application Server integriert ist:

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

WAS-Stammverzeichnis
Das Verzeichnis *WAS-Stammverzeichnis* ist das Stammverzeichnis einer Installation von WebSphere Application Server:

Beispiel: `/opt/IBM/WebSphere/AppServer`

Ausgangsverzeichnis_des_REST-Service
Das Verzeichnis *Ausgangsverzeichnis_des_REST-Service* ist das Verzeichnis, in dem die Bibliotheken und Beispiele des REST-Datenservice von WebSphere eXtreme Scale enthalten sind. Dieses Verzeichnis hat den Namen `restservice` und ist ein Unterverzeichnis im Verzeichnis *WXS-Ausgangsverzeichnis*.

- Beispiel für eigenständige Implementierungen:

  Beispiel: `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`
- Beispiel für integrierte Implementierungen mit WebSphere Application Server:

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

Tomcat-Stammverzeichnis
Das *Tomcat-Stammverzeichnis* ist das Stammverzeichnis der Apache-Tomcat-Installation.

Beispiel: `/opt/tomcat5.5`

WASCE-Stammverzeichnis
Das *WASCE-Stammverzeichnis* ist das Stammverzeichnis der Installation von WebSphere Application Server Community Edition.

Beispiel: `/opt/IBM/WebSphere/AppServerCE`

Java-Ausgangsverzeichnis
Das *Java-Ausgangsverzeichnis* ist das Stammverzeichnis der Installation von Java Runtime Environment (JRE).

Beispiel: `/opt/IBM/WebSphere/eXtremeScale/java`

Ausgangsverzeichnis_für_Beispiele
Das *Ausgangsverzeichnis_für_Beispiele* ist das Verzeichnis, in dem Sie die Beispieldateien entpacken, die für die Lernprogramme verwendet werden.

Beispiel: `/wxs-samples/`

DVD-Stammverzeichnis
Das Verzeichnis *DVD-Stammverzeichnis* ist das Stammverzeichnis der DVD, die das Produkt enthält.

Beispiel: `dvd_root/docs/`

Equinox-Stammverzeichnis
Das Verzeichnis *Equinox-Stammverzeichnis* ist das Stammverzeichnis der Eclipse-Equinox-OSGi-Framework-Installation.

Beispiel:`/opt/equinox`

Benutzerausgangsverzeichnis
Das Verzeichnis *Benutzerausgangsverzeichnis* ist die Position, an der Benutzerdateien gespeichert werden, wie z. B. Sicherheitsprofile.

Windows `c:\Documents and Settings\Benutzername`

UNIX `/home/Benutzername`

WebSphere eXtreme Scale mit dem Installationsassistenten installieren

Sie können den Installationsassistenten verwenden, um WebSphere eXtreme Scale für eigenständige Konfigurationen oder Konfigurationen von WebSphere Application Server zu installieren.

WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren

Sie können WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client in einer Umgebung installieren, in der WebSphere Application Server oder WebSphere Application Server Network Deployment installiert ist. Sie können die vorhandenen Features von WebSphere Application Server oder WebSphere Application Server Network Deployment verwenden, um die Anwendungen von eXtreme Scale zu erweitern.

Vorbereitende Schritte

- Installieren Sie WebSphere Application Server oder WebSphere Application Server Network Deployment. Weitere Informationen finden Sie unter Anwendungsserverumgebung installieren.
- Wenden Sie basierend auf der Version, die Sie installieren, Version 6.1 oder Version 7.0, das neueste Fixpack für WebSphere Application Server bzw. WebSphere

Application Server Network Deployment an, um den Änderungsstand zu aktualisieren. Weitere Informationen finden Sie im Information-Center-Artikel "Aktuelle Fixpacks für WebSphere Application Server".

- Vergewissern Sie sich, dass das Zielinstallationsverzeichnis keine vorhandene Installation von WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client enthält.
- Stoppen Sie alle Prozesse, die in Ihrer Umgebung mit WebSphere Application Server oder WebSphere Application Server Network Deployment aktiv sind. Weitere Informationen zu den Befehlen **stopManager**, **stopNode** und **stopServer** finden Sie unter Befehlszeilendienstprogramme.

  Vorsicht:
  Stellen Sie sicher, dass alle aktiven Prozesse gestoppt sind. Wenn die aktiven Prozesse nicht gestoppt sind, wird die Installation fortgesetzt, was zu unvorhersehbaren Ereignissen führen und die Installation auf einigen Plattformen in einem unbestimmten Zustand hinterlassen kann.
- Wenn Sie nur den Client installieren, können Sie die DVD für die Installation verwenden oder WebSphere eXtreme Scale Client für die jeweilige Plattform über den Downloadabschnitt auf der Unterstützungssite herunterladen.

Wichtig: Wenn Sie WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client installieren, müssen Sie dasselbe Verzeichnis verwenden, in dem Sie auch WebSphere Application Server installiert haben. Wenn Sie WebSphere Application Server beispielsweise in `C:\`*WAS-Stammverzeichnis* installiert haben, sollten Sie `C:`*WAS-Stammverzeichnis* auch als Zielverzeichnis für Ihre Installation von WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client auswählen.

Informationen zu diesem Vorgang

Integrieren Sie eXtreme Scale mit WebSphere Application Server oder WebSphere Application Server Network Deployment, um die Features von eXtreme Scale auf Ihre Java-EE-Anwendungen (Java Platform, Enterprise Edition) anzuwenden. Java-EE-Anwendungen enthalten Datengrids und greifen auf die Datengrids über eine Clientverbindung zu.

Vorgehensweise

1. Verwenden Sie den Assistenten, um die Installation durchzuführen.
   - Führen Sie das folgende Script aus, um den Assistenten für die vollständige Installation von WebSphere eXtreme Scale zu starten. Sie können nur den Client oder den Server und den Client installieren:
     - Linux UNIX *DVD-Stammverzeichnis*`/install`
     - Windows *DVD-Stammverzeichnis*`\install.bat`
   - Führen Sie das folgende Script aus, um den Assistenten für die Installation von WebSphere eXtreme Scale Client zu starten. Die Installationsdatei sind in der ZIP-Datei enthalten, die Sie über den Downloadabschnitt auf der Unterstützungssite herunterladen:
     - Linux UNIX `Stammverzeichnis/WXS_Client/install`
     - Windows `Stammverzeichnis\WXS_Client\install.bat`

Achtung: Wenn Sie einheitliche Namenskonventionen (UNC, Uniform Naming Convention) für die Identifizierung von Dateipfaden in Ihrem Installationsbefehl verwenden, werden die geplanten Komponenten nach der Ausführung des Befehls möglicherweise nicht alle installiert. Zur Vermeidung von Problemen ordnen Sie den Dateipfad einem Netzlaufwerk zu. Führen Sie den Befehl **install** für das zugeordnete Laufwerk aus. Durch die Verwendung eines zugeordneten wird sichergestellt, dass alle Komponenten installiert werden.

2. Folgen Sie den Eingabeaufforderungen im Assistenten.

   In der Anzeige mit den optionalen Features werden die Features aufgelistet, die Sie installieren können. Features können der Produktumgebung nach der Produktinstallation jedoch nicht einzeln hinzugefügt werden. Wenn Sie sich während der Erstinstallation des Produkts gegen die Installation eines Features entscheiden, müssen Sie das Produkt deinstallieren und erneut installieren, um das Feature hinzuzufügen.

   In der Anzeige "Profilerweiterung" werden die vorhandenen Profile aufgelistet, die Sie mit den Features von eXtreme Scale erweitern können. Wenn Sie vorhandene Profile auswählen, die bereits im Gebrauch sind, erscheint eine Warnanzeige. Zum Fortsetzen der Installation müssen Sie die in den Profilen konfigurierten Server stoppen oder auf **Zurück** klicken, um die Profile aus Ihrer Auswahl zu entfernen.

Ergebnisse

Windows Wenn Sie WebSphere eXtreme Scale Client unter Windows installieren, finden Sie in den Ergebnissen der Installation möglicherweise den folgenden Text:

```
Erfolg: Die Installation des folgenden Produkts war erfolgreich:
WebSphere eXtreme Scale Client. In einigen Konfigurationsschritten sind Fehler
aufgetreten. Weitere Informationen finden Sie in der folgenden Protokolldatei:
<WAS-Installationsstammverzeichnis>\logs\wxs_client\install\log.txt"
Sehen Sie sich das Installationsprotokoll (log.txt) und das Erweiterungsprotokoll
des Deployment Manager an.
```

Wenn Sie einen Fehler sehen, der sich auf die Datei `iscdeploy.sh` bezieht, können Sie diesen ignorieren. Dieser Fehler verursacht keine Probleme.

Nächste Schritte

- Wenn Sie WebSphere Application Server Version 6.1 oder Version 7.0 ausführen, können Sie das PMT-Plug-in (Profile Management Tool) oder den Befehl "**manageprofiles**" verwenden. Weitere Informationen finden Sie unter „Profile für WebSphere eXtreme Scale erstellen und erweitern” auf Seite 189.
- Überprüfen Sie die Installation. Weitere Informationen finden Sie unter „Installation überprüfen” auf Seite 212.
- Beginnen Sie mit der Installation von WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client. Weitere Informationen finden Sie unter „Erste Schritte nach der Installation” auf Seite 213.

Laufzeitdateien für eine integrierte Installation von WebSphere eXtreme Scale in WebSphere Application Server

JAR-Dateien (Java-Archiv) sind in der Installation enthalten. Sie können die enthaltenen JAR-Dateien und die Position, an denen sie installiert sind, anzeigen.

Tabelle 5. Laufzeitdateien für WebSphere eXtreme Scale. In der folgenden Tabelle sind die JAR-Dateien aufgelistet, die in der Installation enthalten sind. Die Installationsposition ist relativ zum Verzeichnis von *WXS-Ausgangsverzeichnis*, das Sie während der Installation auswählen.

Dateiname	Umgebung	Installationsposition	Beschreibung
wxsdynacache.jar	Client und Server	lib	Die Datei wxsdynacache.jar enthält die erforderlichen Klassen für den dynamischen Cache-Provider.
wsobjectgrid.jar	Lokal und Client	lib	Die Datei wsobjectgrid.jar enthält die lokale, die Client- und die Serverlaufzeitumgebung von eXtreme Scale.
ogagent.jar	Lokal, Client und Server	lib	Die Datei ogagent.jar enthält die Laufzeitklassen, die für die Ausführung des Java-Instrumentierungsagenten erforderlich sind, der mit der API "EntityManager" verwendet wird.
ogsip.jar	Server	lib	Die Datei ogsip.jar enthält die eXtreme-Scale-Laufzeitumgebung für das SIP-Sitzungsmanagement (Session Initiation Protocol), die mit WebSphere Application Server Version 6.1.x kompatibel ist.
sessionobjectgrid.jar	Client und Server	lib	Die Datei sessionobjectgrid.jar enthält die Laufzeitumgebung von eXtreme Scale für das HTTP-Sitzungsmanagement.
sessionobjectgridsip.jar	Server	lib	Die Datei sessionobjectgridsip.jar enthält die Laufzeitumgebung von eXtreme Scale für das SIP-Sitzungsmanagement, die mit WebSphere Application Server Version 7.x kompatibel ist.
wsogclient.jar	Lokal und Client	lib	Die Datei wsogclient.jar wird installiert, wenn Sie eine Umgebung verwenden, die WebSphere Application Server Version 6.0.2 und höher verwendet. Diese Datei enthält nur die lokale Laufzeitumgebung und die Clientlaufzeitumgebung.
wxssizeagent.jar	Lokal, Client und Server	lib	Die Datei wxssizeagent.jar wird verwendet, um genauere Größeninformationen zu den Cacheeinträgen bereitzustellen, wenn Java Runtime Environment (JRE) Version 1.5 oder höher verwendet wird.
oghibernate-cache.jar	Client und Server	optionalLibraries/ObjectGrid	Die Datei oghibernate-cache.jar enthält das L2-Cache-Plug-in von eXtreme Scale für JBoss Hibernate.
ogspring.jar	Lokal, Client und Server	optionalLibraries/ObjectGrid	Die Datei ogspring.jar lt enthält Unterstützungsklassen für die Integration des SpringSource-Spring-Frameworks.
xsadmin.jar	Dienstprogramm	optionalLibraries/ObjectGrid	Die Datei xsadmin.jar enthält das Verwaltungsdienstprogramm für die Beispiele von eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client und Server	optionalLibraries/ObjectGrid/endorsed	Diese Gruppe von Dateien enthält die ORB-Laufzeitumgebung, die für die Ausführung von Anwendungen in Java-SE-Prozessen verwendet wird.
wxshyperic.jar	Dienstprogramm	optionalLibraries/ObjectGrid/hyperic/lib	Das Erkennungs-Plug-in von WebSphere eXtreme Scale für den SpringSource-Hyperic-Überwachungsagenten
restservice.ear	Client	optionalLibraries/ObjectGrid/restservice/lib	Die Datei restservice.ear enthält das Unternehmensarchiv der REST-Datenserviceanwendung von eXtreme Scale für Umgebungen von WebSphere Application Server.
restservice.war	Client	optionalLibraries/ObjectGrid/restservice/lib	Die Datei restservice.war enthält das Webarchiv des REST-Datenservice von eXtreme Scale für Anwendungsserver eines anderen Anbieters.
splicerlistener.jar	Dienstprogramm	optionalLibraries/ObjectGrid/session/lib	Die Datei splicerlistener.jar enthält das Splicer-Dienstprogramm für den Filter von eXtreme Scale für den HTTP-Sitzungsmanager.
splicer.jar	Dienstprogramm	optionalLibraries/ObjectGrid/legacy/session/lib	Die Datei splicer.jar enthält das Splicer-Dienstprogramm der Version 7.0 für den HTTP-Sitzungsmanagerfilter von eXtreme Scale.

Tabelle 6. Laufzeitdateien für WebSphere eXtreme Scale Client. In der folgenden Tabelle sind die JAR-Dateien aufgelistet, die in der Installation enthalten sind. Die Installationsposition ist relativ zum Verzeichnis von *WXS-Ausgangsverzeichnis*, das Sie während der Installation auswählen.

Dateiname	Umgebung	Installationsposition	Beschreibung
wxsdynacache.jar	Client und Server	lib	Die Datei wxsdynacache.jar enthält die erforderlichen Klassen für den dynamischen Cache-Provider.
ogagent.jar	Lokal, Client und Server	lib	Die Datei ogagent.jar enthält die Laufzeitklassen, die für die Ausführung des Java-Instrumentierungsagenten erforderlich sind, der mit der API "EntityManager" verwendet wird.
ogsip.jar	Server	lib	Die Datei ogsip.jar enthält die eXtreme-Scale-Laufzeitumgebung für das SIP-Sitzungsmanagement (Session Initiation Protocol), die mit WebSphere Application Server Version 6.1.x kompatibel ist.
sessionobjectgrid.jar	Client und Server	lib	Die Datei sessionobjectgrid.jar enthält die Laufzeitumgebung von eXtreme Scale für das HTTP-Sitzungsmanagement.
sessionobjectgridsip.jar	Server	lib	Die Datei sessionobjectgridsip.jar enthält die Laufzeitumgebung von eXtreme Scale für das SIP-Sitzungsmanagement, die mit WebSphere Application Server Version 7.x kompatibel ist.
wsogclient.jar	Lokal und Client	lib	Die Datei wsogclient.jar wird installiert, wenn Sie eine Umgebung verwenden, die WebSphere Application Server Version 6.0.2 und höher verwendet. Diese Datei enthält nur die lokale Laufzeitumgebung und die Clientlaufzeitumgebung.
wxssizeagent.jar	Lokal, Client und Server	lib	Die Datei wxssizeagent.jar wird verwendet, um genauere Größeninformationen zu den Cacheeinträgen bereitzustellen, wenn Java Runtime Environment (JRE) Version 1.5 oder höher verwendet wird.
oghibernate-cache.jar	Client und Server	optionalLibraries/ObjectGrid	Die Datei oghibernate-cache.jar enthält das L2-Cache-Plug-in von eXtreme Scale für JBoss Hibernate.
ogspring.jar	Lokal, Client und Server	optionalLibraries/ObjectGrid	Die Datei ogspring.jar lt enthält Unterstützungsklassen für die Integration des SpringSource-Spring-Frameworks.
xsadmin.jar	Dienstprogramm	optionalLibraries/ObjectGrid	Die Datei xsadmin.jar enthält das Verwaltungsdienstprogramm für die Beispiele von eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Client und Server	optionalLibraries/ObjectGrid/endorsed	Diese Gruppe von Dateien enthält die ORB-Laufzeitumgebung, die für die Ausführung von Anwendungen in Java-SE-Prozessen verwendet wird.
wxshyperic.jar	Dienstprogramm	optionalLibraries/ObjectGrid/hyperic/lib	Das Erkennungs-Plug-in von WebSphere eXtreme Scale für den SpringSource-Hyperic-Überwachungsagenten
restservice.ear	Client	optionalLibraries/ObjectGrid/restservice/lib	Die Datei restservice.ear enthält das Unternehmensarchiv der REST-Datenserviceanwendung von eXtreme Scale für Umgebungen von WebSphere Application Server.
restservice.war	Client	optionalLibraries/ObjectGrid/restservice/lib	Die Datei restservice.war enthält das Webarchiv des REST-Datenservice von eXtreme Scale für Anwendungsserver eines anderen Anbieters.
splicerlistener.jar	Dienstprogramm	optionalLibraries/ObjectGrid/session/lib	Die Datei splicerlistener.jar enthält das Splicer-Dienstprogramm für den Filter von eXtreme Scale für den HTTP-Sitzungsmanager.
splicer.jar	Dienstprogramm	optionalLibraries/ObjectGrid/legacy/session/lib	Die Datei splicer.jar enthält das Splicer-Dienstprogramm der Version 7.0 für den HTTP-Sitzungsmanagerfilter von eXtreme Scale.

Installation-Factory-Plug-in zum Erstellen und Installieren angepasster Pakete verwenden

Verwenden Sie das IBM® Installation-Factory-Plug-in für WebSphere eXtreme Scale, um ein angepasstes Installationspaket (CIP, Customized Installation Package) oder

ein integriertes Installationspaket (IIP, Integrated Installation Package) zu erstellen. Ein angepasstes Installationspaket enthält ein Installationspaket für ein Einzelprodukt und verschiedene optionale Assets. Ein integriertes Installationspaket kombiniert ein oder mehrere Installationspakete zu einem einzigen Installations-Workflow, den Sie entwerfen können.

Vorbereitende Schritte

Bevor Sie angepasste Pakete für eXtreme Scale erstellen, müssen Sie die folgenden Produkte herunterladen:

- IBM Installation Factory for WebSphere Application Server
- IBM Installation-Factory-Plug-in für WebSphere eXtreme Scale

Informationen zu diesem Vorgang

Mit Installation Factory können Sie ein angepasstes Installationspaket erstellen, indem Sie eine einzelne Produktkomponente mit Wartungspaketen, Anpassungsscripts und anderen Dateien kombinieren. Wenn Sie ein integriertes Installationspaket erstellen, fassen Sie einzelne Komponenten oder Installationspakete zu einem einzelnen Installationspaket zusammen.

Build-Definitionsdatei:

Eine Build-Definitionsdatei ist ein XML-Dokument, in dem spezifiziert wird, wie ein angepasstes Installationspaket (CIP, Customized Installation Package) oder integriertes Installationspaket (IIP, Integrated Installation Package) erstellt und installiert wird. IBM Installation Factory for WebSphere eXtreme Scale liest die Paketdetails in der Build-Definitionsdatei, um ein angepasstes Installationspaket bzw. integriertes Installationspaket zu generieren.

Damit Sie ein angepasstes Installationspaket oder ein integriertes Installationspaket erstellen können, müssen Sie für jedes angepasste Paket eine Build-Definitionsdatei erstellen. In der Build-Definitionsdatei wird beschrieben, welche Produktkomponenten oder Installationspakete installiert werden sollen. Außerdem enthält diese Datei Angaben zur Position des angepassten Installationspakets bzw. integrierten Installationspakets, zu den einzuschließenden Wartungspaketen, zu den Installationsscripts und anderen ausgewählten Dateien, die in das Paket eingeschlossen werden sollen. Außerdem können Sie in der Build-Definitionsdatei für das integrierte Installationspaket die Reihenfolge festlegen, in der Installation Factory die Installationspakete installiert.

Der Assistent für Build-Definition führt Sie schrittweise durch die Erstellung einer Build-Definitionsdatei. Sie können mit dem Assistenten für Build-Definition auch eine vorhandene Build-Definitionsdatei ändern. In jeder Anzeige des Assistenten für Build-Definition werden Sie zur Eingabe von Informationen zu einem angepassten Paket aufgefordert, z. B. der Paketkennung, der Installationsposition für die Build-Definition und der Installationsposition für das angepasste Paket. Alle diese Informationen werden in der neuen Build-Definitionsdatei gespeichert bzw. in einer vorhandenen Build-Definitionsdatei geändert und gespeichert. Weitere Informationen finden Sie in den Abschnitten "Anzeigen des Assistenten für Build-Definition für angepasste Installationspakete" und "Anzeigen des Assistenten für Build-Definition für integrierte Installationspakete".

Wenn Sie nur die Build-Definitionsdatei erstellen möchten, können Sie das Befehlszeilenschnittstellentool ausführen, um das angepasste Paket außerhalb der grafi-

schen Benutzerschnittstelle zu generieren. Weitere Informationen finden Sie im Abschnitt „Angepasstes oder integriertes Installationspaket im unbeaufsichtigten Modus installieren” auf Seite 180.

Build-Definitionsdatei erstellen und ein angepasstes Installationspaket erstellen:

Das IBM Installation-Factory-Plug-in für WebSphere eXtreme Scale generiert ein angepasstes Installationspaket entsprechend den Details, die Sie in der Build-Definitionsdatei angeben. Die Build-Definition gibt das zu installierende Produktpaket, die Position des angepassten Installationspakets, die in die Installation einzuschließenden Wartungspakete, die Installationsscriptdateien und alle zusätzlichen Dateien an, die in das angepasste Installationspaket eingeschlossen werden sollen.

Informationen zu diesem Vorgang

Sie können den Assistenten für Build-Definition verwenden, um eine Build-Definitionsdatei zu erstellen und ein angepasstes Installationspaket zu generieren.

Vorgehensweise

1. Führen Sie das folgende Script im Verzeichnis *IF-Ausgangsverzeichnis*/bin aus, um Installation Factory zu starten:
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`

   Klicken Sie auf das Symbol für **Neue Build-Definition**.
2. Wählen Sie das Produkt aus, das in die Build-Definitionsdatei eingeschlossen werden soll, und klicken Sie anschließend auf **Fertig stellen**, um den Assistenten für Build-Definition zu starten.
3. Folgen Sie den Eingabeaufforderungen im Assistenten.

   Klicken Sie in der Anzeige "Installations- und Deinstallationsscripts" auf **Scripts hinzufügen...**, um alle angepassten Installationsscripts in die Tabelle einzutragen. Geben Sie die Position der Scriptdateien ein, und wählen Sie das Kontrollkästchen ab, um fortzufahren, wenn eine Fehlernachricht angezeigt wird. Die Operation wird standardmäßig gestoppt. Klicken Sie auf **OK**, um zur Anzeige zurückzukehren.

Ergebnisse

Sie haben die Build-Definitionsdatei erstellt und angepasst und das angepasste Installationspaket generiert, wenn Sie sich für die Arbeit im Modus "Verbunden" entschieden haben.

Wenn der Assistent für Build-Definition Ihnen die Option zum Generieren des angepassten Installationspakets aus der Build-Definitionsdatei nicht anbietet, können Sie das Paket trotzdem generieren, indem Sie das Script `ifcli.sh|bat` im Verzeichnis *IF-Ausgangsverzeichnis*/bin ausführen.

Nächste Schritte

Installieren Sie das angepasste Installationspaket. Weitere Informationen finden Sie unter „Angepasstes Installationspaket installieren”.

Angepasstes Installationspaket installieren:

Sie können den Produktinstallationsprozess vereinfachen, indem Sie ein angepasstes Installationspaket (CIP, Customized Installation Package) installieren. Ein angepasstes Installationspaket ist ein Installations-Image für ein einziges Produkt, das ein oder mehrere Pakete, Konfigurationsscripts und andere Dateien enthalten kann.

Vorbereitende Schritte

Damit Sie ein integriertes Installationspaket installieren können, müssen Sie zuerst eine Build-Definitionsdatei erstellen, in der Sie die in das integrierte Installationspaket einzuschließenden Optionen angeben. Weitere Informationen finden Sie im Abschnitt „Build-Definitionsdatei erstellen und ein angepasstes Installationspaket erstellen" auf Seite 173.

Informationen zu diesem Vorgang

Mit einem angepassten Installationspaket kann eine einzelne Produktkomponente mit Wartungspaketen, Anpassungsscripts und anderen Dateien kombiniert und installiert werden.

Vorgehensweise

1. Stoppen Sie alle Prozesse, die auf der Workstation ausgeführt werden, die Sie für die Installation vorbereiten. Zum Stoppen des Deployment Manager führen Sie das folgende Script aus:
   - Linux UNIX *Profilstammverzeichnis*/bin/stopManager.sh
   - Windows *Profilstammverzeichnis*\bin\stopManager.bat

   Zum Stoppen der Knoten führen Sie das folgende Script aus:
   - Linux UNIX *Profilstammverzeichnis*/bin/stopNode.sh
   - Windows *Profilstammverzeichnis*\bin\stopNode.bat
2. Führen Sie das folgende Script aus, um die Installation zu starten:
   - Linux UNIX *CIP-Ausgangsverzeichnis*/bin/install
   - Windows *CIP-Ausgangsverzeichnis*\bin\install.bat
3. Folgen Sie den Aufforderungen des Assistenten, um die Installation durchzuführen:

   In der Anzeige mit den optionalen Features werden die Features aufgelistet, die Sie installieren können. Features können der Produktumgebung nach der Produktinstallation jedoch nicht einzeln hinzugefügt werden. Wenn Sie sich während der Erstinstallation des Produkts gegen die Installation eines Features entscheiden, müssen Sie das Produkt deinstallieren und erneut installieren, um das Feature hinzuzufügen.

   In der Anzeige "Profilerweiterung" werden die vorhandenen Profile aufgelistet, die Sie mit den Features von eXtreme Scale erweitern können. Wenn Sie vorhandene Profile auswählen, die bereits im Gebrauch sind, erscheint eine Warnanzeige. Zum Fortsetzen der Installation müssen Sie die in den Profilen konfigurierten Server stoppen oder auf **Zurück** klicken, um die Profile aus Ihrer Auswahl zu entfernen.

Ergebnisse

Sie haben das angepasste Installationspaket ordnungsgemäß installiert.

Nächste Schritte

Wenn Sie WebSphere Application Server Version 6.1 oder Version 7.0 ausführen, können Sie das PMT-Plug-in (Profile Management Tool) oder den Befehl "**manageprofiles**" verwenden, um Profile zu erstellen und zu erweitern. Weitere Informationen finden Sie im Abschnitt „Profile für WebSphere eXtreme Scale erstellen und erweitern" auf Seite 189.

Wenn Sie Profile für eXtreme Scale während des Installationsprozesses erweitert haben, können Sie Anwendungen implementieren, einen Katalogservice starten und die Container in Ihrer Umgebung von WebSphere Application Server starten. Weitere Informationen hierzu finden Sie im Abschnitt „WebSphere eXtreme Scale mit WebSphere Application Server konfigurieren" auf Seite 263.

Ein angepasstes Installationspaket zum Anwenden von Wartungspaketen auf eine vorhandene Produktinstallation installieren:

Sie können Wartungspakete für eine vorhandene Produktinstallation anwenden, indem Sie ein angepasstes Installationspaket (CIP, Customized Installation Package) installieren. Die Anwendung von Wartungspaketen auf eine vorhandene Installation mit einem angepassten Installationspaket wird häufig als *Slip-Installation* bezeichnet.

Vorbereitende Schritte

Erstellen Sie eine Build-Definitionsdatei, um die in das angepasste Installationspaket einzuschließenden Optionen anzugeben. Weitere Informationen finden Sie im Abschnitt „Build-Definitionsdatei erstellen und ein angepasstes Installationspaket erstellen" auf Seite 173.

Informationen zu diesem Vorgang

Wenn Sie Wartungspakete mit einem angepassten Installationspaket anwenden, das ein Aktualisierungspaket und/oder einen Fixpack enthält, werden alle zuvor installierten APARs (Authorized Program Analysis Report) in dieser Installation vom Assistenten deinstalliert. Wenn das angepasste Installationspaket dieselbe Version hat wie das Produkt, werden die zuvor installierten APARs nur dann nicht deinstalliert, wenn sie im angepassten Installationspaket enthalten sind. Für eine erfolgreiche Anwendung von Wartungspaketen auf eine vorhandene Installation müssen Sie die installierten Features im angepassten Installationspaket anwenden.

Vorgehensweise

1. Stoppen Sie alle Prozesse, die auf der Workstation ausgeführt werden, die Sie für die Installation vorbereiten. Zum Stoppen des Deployment Manager führen Sie das folgende Script aus:
   - Linux UNIX *Profilstammverzeichnis*/bin/stopManager.sh
   - Windows *Profilstammverzeichnis*\bin\stopManager.bat

   Zum Stoppen der Knoten führen Sie das folgende Script aus:
   - Linux UNIX *Profilstammverzeichnis*\bin\stopNode.sh
   - Windows *Profilstammverzeichnis*\bin\stopNode.bat
2. Führen Sie das folgende Script aus, um die Installation zu starten:
   - Linux UNIX *CIP-Ausgangsverzeichnis*/bin/install

- Windows *CIP-Ausgangsverzeichnis*\bin\install.bat

3. Folgen Sie den Aufforderungen des Assistenten, um die Installation durchzuführen:

   Die Zusammenfassung in der Installationsvoranzeige listet die Produktversion und alle Features und vorläufigen Fixes auf, die angewendet werden sollen. Als Nächstes wendet der Assistent die Wartung erfolgreich an und aktualisiert die Features des Produkts.

Ergebnisse

Die Produktbibliotheksdateien werden in das Verzeichnis *WAS-Stammverzeichnis*/properties/version/nif/backup kopiert. Sie können IBM Update Installer verwenden, um die Aktualisierung zu deinstallieren und die Workstation wiederherzustellen. Weitere Informationen finden Sie im Abschnitt „CIP-Aktualisierungen aus einer vorhandenen Produktinstallation deinstallieren".

CIP-Aktualisierungen aus einer vorhandenen Produktinstallation deinstallieren:

Sie können CIP-Aktualisierungen (Customized Installation Package, angepasstes Installationspaket) aus einer vorhandenen Produktinstallation deinstallieren, ohne das gesamte Produkt deinstallieren zu müssen. Verwenden Sie IBM Update Installer Version 7.0.0.4, um CIP-Aktualisierungen zu deinstallieren. Diese Task wird auch als *Slip-Deinstallation* bezeichnet.

Vorbereitende Schritte

Es muss mindestens eine Kopie des Produkts auf dem System installiert sein.

Vorgehensweise

1. Laden Sie Version 7.0.0.4 von Update Installer von der folgenden FTP-Site herunter:

   ftp://ftp.software.ibm.com/software/websphere/cw/process_server/FEP/UPDI/7004
2. Installieren Sie Update Installer. Weitere Informationen finden Sie im Artikel "Update Installer für WebSphere Software" im Information Center von WebSphere Application Server.
3. Deinstallieren Sie alle Fixpacks, Refresh-Packs und vorläufigen Fixes, die Sie der Umgebung nach der Installation des angepassten Installationspakets hinzugefügt haben.
4. Deinstallieren Sie alle vorläufigen Fixes, die Sie in die Slip-Installation eingeschlossen haben. Diese Vorgehensweise ist dieselbe wie bei der Deinstallation eines einzelnen Fixpacks oder Refresh-Packs. Die Wartung, die im angepassten Installationspaket enthalten war, ist jetzt allerdings in einer einzelnen Operation enthalten.
5. Deinstallieren Sie das angepasste Installationspaket mit Update Installer. Die Wartungsstufen werden auf den Stand vor Aktualisierung zurückgesetzt, und das angepasste Installationspaket wird mit der CIP-ID gekennzeichnet, die dem Dateinamen als Präfix hinzugefügt wird. Das folgende Beispiel veranschaulicht, das ein angepasstes Installationspaket anders als andere reguläre Wartungspakete in der Anzeige zur Auswahl der Wartungspakete angezeigt wird:

   Angepasstes Installationspaket

   com.ibm.ws.cip.7000.wxs.primary.ext.pak

Ergebnisse

Die CIP-Aktualisierungen wurden ordnungsgemäß aus einer vorhandenen Produktinstallation entfernt.

Build-Definitionsdatei erstellen und integriertes Installationspaket generieren:

Das IBM Installation-Factory-Plug-in für WebSphere eXtreme Scale generiert ein integriertes Installationspaket, das auf den Eigenschaften basiert, die in der Build-Definitionsdatei bereitgestellt werden. Die Build-Definitionsdatei enthält Informationen, wie z. B. Informationen zu den in das integrierte Installationspaket einzuschließenden Paketen, zur Reihenfolge, in der Installation Factory die Pakete installiert, und zur Position des integrierten Installationspakets.

Informationen zu diesem Vorgang

Sie können den Assistenten für Build-Definition verwenden, um eine Build-Definitionsdatei zu erstellen und ein integriertes Installationspaket zu generieren.

Vorgehensweise

1. Führen Sie das folgende Script im Verzeichnis *IF-Ausgangsverzeichnis*/bin aus, um Installation Factory zu starten:
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`
2. Klicken Sie auf das Symbol **Neues integriertes Installationspaket erstellen**, um den Assistenten für Build-Definition zu starten.
3. Folgen Sie den Eingabeaufforderungen im Assistenten.
   a. Wählen Sie in der Anzeige "Integriertes Installationspaket erstellen" in der Liste ein unterstütztes Installationspaket aus, und klicken Sie auf **Installationsprogramm hinzufügen**, um das Installationspaket dem integrierten Installationspaket hinzuzufügen. Es erscheint eine Anzeige, in der der Paketname, die Paketkennung und die Paketeigenschaften angezeigt werden. Wenn Sie bestimmte Informationen zum ausgewählten Paket anzeigen möchten, klicken Sie auf **Informationen zum Installationspaket anzeigen**. Klicken Sie auf **Ändern**, um den Verzeichnispfad zum Installationspaket für jedes Betriebssystem einzugeben. Wenn Sie ein Installationspaket für WebSphere Extended Deployment hinzufügen, wählen Sie das Kontrollkästchen aus, das Ihnen die Möglichkeit gibt, dasselbe Paket für alle unterstützten Betriebssysteme zu verwenden. Klicken Sie auf **OK**, und kehren Sie in die Anzeige "Integriertes Installationspaket erstellen" zurück. Es wird standardmäßig ein Aufruf erstellt.
      - Wenn Sie den Verzeichnispfad zu einem Installationspaket ändern möchten, wählen Sie das Paket in der Liste "In diesem integrierten Installationspaket verwendete Installationspakete" aus, und klicken Sie auf **Ändern**.
      - Wenn Sie einen Aufruf ändern möchten, wählen Sie den Aufruf aus, und klicken Sie auf **Ändern**. Geben Sie die Standardinstallationsposition für den Aufruf auf jedem Betriebssystem an. Geben Sie die Position zur Antwortdatei an, wenn Sie eine unbeaufsichtigte Installation als Standardinstallationsmodus auswählen.
      - Klicken Sie auf **Aufruf hinzufügen**, um dem Installationspaket einen Aufrufbeitrag hinzuzufügen. Es erscheint eine Anzeige, in der Sie die Eigenschaften für den Aufruf eingeben können.

- Klicken Sie auf **Entfernen**, um Installationspakete oder Aufrufe zu entfernen.

4. Sehen Sie sich die Zusammenfassung der von Ihnen ausgewählten Optionen an, wählen Sie die Option **Build-Definitionsdatei speichern und integriertes Installationspaket generieren** aus, und klicken Sie anschließend auf **Fertig stellen**.

   Alternativ können Sie die Build-Definitionsdatei auch ohne Generierung des integrierten Installationspakets speichern. Mit dieser Option generieren Sie das integrierte Installationspaket außerhalb des Assistenten, indem Sie das Script `ifcli.bat | ifcli.sh` im Verzeichnis "*IF-Ausgangsverzeichnis*`/bin/`" ausführen.

Ergebnisse

Sie haben die Build-Definitionsdatei für ein integriertes Installationspaket erstellt und angepasst.

Nächste Schritte

Installieren Sie das integrierte Installationspaket.

Integriertes Installationspaket installieren:

Verwenden Sie das IBM Installation-Factory-Plug-in für WebSphere eXtreme Scale, um ein integriertes Installationspaket (IIP, Integrated Installation Package) zu installieren. Ein integriertes Installationspaket kombiniert ein oder mehrere Installationspakete zu einem einzigen Workflow, den Sie entwerfen können.

Vorbereitende Schritte

Damit Sie ein integriertes Installationspaket installieren können, müssen Sie zuerst eine Build-Definitionsdatei erstellen, in der Sie die in das integrierte Installationspaket einzuschließenden Optionen angeben. Weitere Informationen finden Sie im Abschnitt „Build-Definitionsdatei erstellen und integriertes Installationspaket generieren” auf Seite 177.

Informationen zu diesem Vorgang

Ein integriertes Installationspaket kann ein oder mehrere allgemein verfügbare Installationspakete, ein oder mehrere angepasste Installationspakete (CIP, Customized Installation Packages) und weitere optionale Dateien und Verzeichnisse enthalten. Indem Sie ein integriertes Installationspaket installieren, fassen Sie mehrere Installationspakete oder *Beiträge* zu einem einem einzelnen Paket zusammen und installieren die Beiträge in einer bestimmte Reihenfolge in einer End-to-End-Installation.

Vorgehensweise

1. Führen Sie das folgende Script aus, um den Assistenten zu starten:
   - Linux UNIX *IIP-Ausgangsverzeichnis*`/bin/install`
   - Windows *IIP-Ausgangsverzeichnis*`\bin\install.bat`
2. Klicken Sie in der Eingangsanzeige auf **Produktinfo**, um die Details zum integrierten Installationspaket anzuzeigen, wie z. B. die Paketkennung, die unterstützten Betriebssysteme und die eingeschlossenen Installationspakete.

   Optional: Wenn Sie die Installationsoptionen der einzelnen Pakete ändern möchten, klicken Sie auf **Ändern**.

Optional: Auf der Assistentenseite werden zwei Schaltflächen **Protokoll anzeigen** angezeigt. Wenn Sie die Protokolle der einzelnen Pakete anzeigen möchten, klicken Sie auf die Schaltfläche **Protokoll anzeigen**, die neben der Tabelle angezeigt wird, in der die Installationspakete aufgelistet sind. Klicken Sie auf die Schaltfläche **Protokoll anzeigen**, die neben den Statusinformationen angezeigt wird, um die allgemeinen Protokolldetails des integrierten Installationspakets anzuzeigen.

3. Wählen Sie die auszuführenden Installationspakete aus, und klicken Sie auf **Installieren**. Es wird eine nach Aufruf sortierte Liste aller Beiträge angezeigt, die im integrierten Installationspaket enthalten sind. Wenn ein Beitragsaufruf während der Installation nicht ausgeführt werden soll, wählen Sie das Kontrollkästchen neben dem Feld **Installationsname** ab.

Ergebnisse

Sie haben erfolgreich ein integriertes Installationspaket installiert.

Eine vorhandene Build-Definitionsdatei für ein integriertes Installationspaket ändern:

Sie können die Eigenschaften eines integrierten Installationspakets bearbeiten oder weitere Eigenschaften hinzufügen, um die Installation weiter anzupassen.

Informationen zu diesem Vorgang

Wenn Sie die Eigenschaften eines integrierten Installationspakets ändern möchten, ändern Sie die vorhandene Build-Definitionsdatei.

Vorgehensweise

1. Führen Sie das folgende Script im Verzeichnis *IF-Ausgangsverzeichnis*`/bin` aus, um Installation Factory zu starten:
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`
2. Klicken Sie auf das Symbol **Build-Definition öffnen**, und wählen Sie die zu änderende Build-Definitionsdatei aus.
3. Wählen Sie die Eigenschaften des integrierten Installationspakets aus, die Sie ändern möchten. Die folgende Liste enthält die möglichen Änderungen, die Sie vornehmen können:
   - Aktuelle Modusauswahl ändern. Im Modus "Verbunden" erstellen Sie eine Build-Definitionsdatei für die Verwendung auf Ihrer Workstation und generieren optional ein integriertes Installationspaket. Im Modus "Nicht verbunden" erstellen Sie die Build-Definitionsdatei für die Verwendung auf einer anderen Workstation.
   - Vom integrierten Installationspaket unterstützte Betriebssysteme hinzufügen oder entfernen.
   - Vorhandene Kennung und Version für das integrierte Installationspaket bearbeiten.
   - Zielposition für die Build-Definitionsdatei bearbeiten.
   - Zielposition für das integrierte Installationspaket bearbeiten.
   - Festlegen, ob ein Installationsassistent für das integrierte Installationspaket angezeigt wird. Im Installationsassistenten werden Informationen zum integrierten Installationspaket und die Installationsoptionen für die Ausführung des integrierten Installationspakets angezeigt.

- Im integrierten Installationspaket enthaltene Installationspaket entfernen, bearbeiten oder neue Installationspakete hinzufügen.

  Wichtig: Wenn Sie ein unterstütztes Betriebssystem hinzugefügt und die Eigenschaften des Installationspaket im integrierten Installationspaket nicht aktualisiert haben, empfangen Sie eine Warnung, in der Sie darauf hingewiesen werden, dass die ausgewählten Beiträge keine Installationspakete enthalten, die für alle Betriebssysteme angegeben sind, die vom integrierten Installationspaket unterstützt werden. Klicken Sie auf **Ja**, um fortzufahren, oder klicken Sie auf **Nein**, um das Installationspaket zu bearbeiten.

4. Sehen Sie sich die Zusammenfassung der von Ihnen ausgewählten Optionen an, wählen Sie die Option **Build-Definitionsdatei speichern und integriertes Installationspaket generieren** aus, und klicken Sie anschließend auf **Fertig stellen**.

Angepasstes oder integriertes Installationspaket im unbeaufsichtigten Modus installieren:

Sie können ein angepasstes Installationspaket (Customized Installation Package) oder integriertes Installationspaket (IIP, Integrated Installation Package) für ein Produkt im unbeaufsichtigten Modus installieren, indem Sie entweder eine vollständig qualifizierte Antwortdatei, die Sie speziell auf Ihre Anforderungen konfigurieren, oder Parameter verwenden, die Sie an die Befehlszeile übergeben.

Vorbereitende Schritte

Erstellen Sie die Build-Definitionsdatei für das angepasste Installationspaket bzw. das integrierte Installationspaket. Weitere Informationen finden Sie im Abschnitt „Build-Definitionsdatei erstellen und ein angepasstes Installationspaket erstellen" auf Seite 173.

Informationen zu diesem Vorgang

Bei einer unbeaufsichtigten Installation wird dasselbe Installationsprogramm verwendet, das auch bei der Installation über die grafische Benutzerschnittstelle verwendet wird. Anstatt jedoch eine Assistentenschnittstelle anzuzeigen, werden bei der unbeaufsichtigten Installation alle Antworten aus einer Datei, die Sie anpassen, oder aus Parametern gelesen, die Sie an die Befehlszeile übergeben. Wenn Sie ein integriertes Installationspaket im unbeaufsichtigten Modus installieren, können Sie einen Beitrag mit einer Kombination von Optionen, die Sie direkt in der Befehlszeile angeben, und Optionen, die Sie in einer Antwortdatei angeben, aufrufen. Alle Beitragsoptionen, die Sie an die Befehlszeile übergeben, bewirken jedoch, dass das Installationsprogramm des integrierten Installationspakets alle Optionen ignoriert, die in der Antwortdatei eines bestimmten Beitrags angegeben sind. Ausführliche Informationen finden Sie unter Integriertes Installationspaket im unbeaufsichtigten Modus installieren.

Anmerkung: Sie müssen den vollständig qualifizierten Namen der Antwortdatei angeben. Wenn Sie den relativen Pfad angeben, scheitert die Installation ohne Ausgabe einer entsprechenden Fehlernachricht.

Vorgehensweise

1. Optional: Wenn Sie sich für die Installation des angepassten Installationspakets bzw. integrierten Installationspakets über eine Antwortdatei entscheiden, passen Sie zuerst die Datei an.

a. Kopieren Sie die Antwortdatei `wxssetup.response.txt` von der Produkt-DVD auf Ihr Plattenlaufwerk.
b. Öffnen Sie die Antwortdatei mit einem Texteditor Ihrer Wahl, und bearbeiten Sie sie. Die Datei enthält Kommentare, die Sie beim Konfigurationsprozess unterstützen. Sie muss die folgenden Parameter enthalten:
   - die Lizenzvereinbarung,
   - die Position der Produktinstallation.

   Tipp: Das Installationsprogramm verwendet die Position, die Sie für Ihre Installation auswählen, um zu bestimmen, wo Ihre Instanz von WebSphere Application Server installiert ist. Wenn Sie die Installation auf einem Knoten mit mehreren Instanzen von WebSphere Application Server durchführen, müssen Sie die Position klar definieren.
c. Führen Sie das folgende Script aus, um die angepasste Antwortdatei zu starten.
   - Linux UNIX `install -options /absoluter_Pfad/Antwortdatei.txt -silent`
   - Windows `install.bat -options C:\Laufwerkspfad\Antwortdatei.txt -silent`

2. Optional: Wenn Sie sich für die Installation des angepassten bzw. integrierten Installationspakets durch Übergabe bestimmter Parameter in der Befehlszeile entscheiden, führen Sie das folgende Script aus, um die Installation zu starten:
   - Linux UNIX `install -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=Installationsposition`
   - Windows `install.bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=Installationsposition`

   Installationsposition steht für die Position der vorhandenen Installation von WebSphere Application Server.
3. Suchen Sie in den während der Installation erstellten Protokollen nach Fehlern oder Installationsfehlern.

Ergebnisse

Sie haben das angepasste Installationspaket bzw. das integrierte Installationspaket im unbeaufsichtigten Modus installiert.

Nächste Schritte

Wenn Sie WebSphere Application Server Version 6.1 oder Version 7.0 ausführen, können Sie das PMT-Plug-in (Profile Management Tool) oder den Befehl "**manageprofiles**" verwenden, um Profile zu erstellen und zu erweitern.

Wenn Sie Profile für eXtreme Scale während des Installationsprozesses erweitert haben, können Sie Anwendungen implementieren, einen Katalogservice starten und die Container in Ihrer Umgebung von WebSphere Application Server starten. Weitere Informationen hierzu finden Sie im Abschnitt „WebSphere eXtreme Scale mit WebSphere Application Server konfigurieren” auf Seite 263.

Datei wxssetup.response.txt:

Sie können eine vollständig qualifizierte Antwortdatei verwenden, um WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client unbeaufsichtigt zu installieren.

Vorsicht:
Fügen Sie keine abschließenden Schrägstriche wie / oder \ am Ende der Installationspositionspfade hinzu. Diese Pfade werden mit dem Attribut "installLocation" angegeben. Wenn Sie einen Schrägstrich am Ende der Installationsposition hinzufügen, kann die Installation fehlschlagen. Der folgende Pfad würde beispielsweise dazu führen, dass die Installation fehlschlägt:

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale/"
```

Der Pfad muss wie folgt angegeben werden:

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
```

Antwortdatei für eine vollständige Installation von WebSphere eXtreme Scale

```
################################################################################
#
# InstallShield-Optionsdatei für IBM WebSphere eXtreme Scale V7.1.1
#
# Name des Assistenten: Install
# Quelle des Assistenten: setup.jar
#
# Diese Datei kann verwendet werden, um den Assistenten "Install" mit den im
# Folgenden angegebenen Optionen zu konfigurieren, wenn der Assistent mit der
# Befehlszeilenoption "-options" ausgeführt wird. Die Dokumentation zu jeder
# Einstellung enthält Informationen zum Ändern des Einstellungswerts.
# Schließen Sie alle Werte in Anführungszeichen ein.
#
# Eine gängige Verwendung für eine Optionsdatei ist die Ausführung des
# Assistenten im unbeaufsichtigten Modus. Auf diese Weise kann der Autor
# der Optionsdatei Assistenteneinstellungen festlegen, ohne den Assistenten
# im grafischen Modus oder im Konsolmodus ausführen zu müssen. Wenn Sie
# diese Optionsdatei für die Ausführung im unbeaufsichtigten Modus verwenden
# möchten, verwenden Sie die folgenden Befehlszeilenargumente bei der
# Ausführung des Assistenten:
#
#    -options "D:\installImage\WXS\wxssetup.response" -silent
#
# Sie müssen den vollständig qualifizierten Namen der Antwortdatei angeben.
#
################################################################################

################################################################################
#
# Lizenz akzeptieren
#
# Gültige Werte:
#  true - Akzeptiert die Lizenz. Das Produkt wird installiert.
#  false - Die Lizenz wird nicht akzeptiert. Es findet keine Installation statt.
#
# Wenn keine Installation stattfindet, wird dies in einer temporären
# Protokolldatei im temporären Verzeichnis des Benutzers aufgezeichnet.
#
# Wenn Sie die Option silentInstallLicenseAcceptance in dieser Antwortdatei auf
# "true" setzen, geben Sie an, dass Sie die internationalen Nutzungsbedingungen
# für Programmpakete der IBM unter
# CD_ROOT\XD\wxs.primary.pak\repository\legal.xs\license.xs
# gelesen haben und diese akzeptieren. Falls Sie diesen Bedingungen nicht
# zustimmen, dürfen Sie diesen Wert nicht ändern und das Programm weder
# herunterladen, noch installieren, kopieren, aufrufen oder verwenden. Geben
# Sie das Programm und den Berechtigungsnachweis umgehend an den Verkäufer
# zurück, um sich den Kaufbetrag erstatten zu lassen.
#
```

```
-OPT silentInstallLicenseAcceptance="false"


###############################################################################
# Nicht blockierende Prüfung der Voraussetzungen
#
# Wenn Sie die nicht blockierende Prüfung der Voraussetzungen inaktivieren möchten,
# müssen Sie das Kommentarzeichen aus der folgenden Anweisungszeile entfernen.
# Mit dieser Option wird das Installationsprogramm angewiesen, die Installation
# fortzusetzen und die Warnungen zu protokollieren, selbst wenn die
# Prüfung der Voraussetzungen scheitert.
#
#-OPT disableNonBlockingPrereqChecking="true"


###############################################################################
#
# Installationspfad:
#
# Dies ist das Installationsverzeichnis für das Produkt. Geben Sie ein gültiges
# Verzeichnis für die Installation des Produkts an. Falls der Verzeichnisname
# Leerzeichen enthält, setzen Sie ihn wie im folgenden Windows-Beispiel in
# Anführungszeichen. Installationsverzeichnisse mit Leerzeichen werden nur unter
# Windows-Betriebssystemen unterstützt. Die maximale Pfadlänge für Windows
# ist 60 Zeichen.
#
# Nachfolgend sind die Standardinstallationspfade für alle unterstützten
# Betriebssysteme aufgelistet, wenn Sie die Installation als Root durchführen.
# In dieser Antwortdatei wird standardmäßig das Installationsverzeichnis für
# Windows verwendet. Falls Sie das Standardinstallationsverzeichnis für ein
# anderes Betriebssystem verwenden möchten, entfernen Sie das Kommentarzeichen
# ('#') vor dem entsprechenden Verzeichnis, und fügen Sie vor dem Eintrag für
# das Windows-Betriebssystem ein Kommentarzeichen ('#') hinzu.
#
# Der Installationspfad wird verwendet, um festzustellen, ob WebSphere eXtreme
# Scale als eigenständige Implementierung installiert oder mit einer vorhandenen
# Installation von WebSphere Application Server integriert werden muss.
#
# Wenn der angegebene Pfad eine vorhandene Installation von WebSphere Application
# Server oder WebSphere Network Deployment enthält, wird eXtreme Scale mit dem
# vorhandenen Produkt integriert. Ist der angegebene Pfad neu oder leer, wird
# WebSphere eXtreme Scale als eigenständige Implementierung installiert.
#
# Anmerkung: Wenn der angegebene Installationspfad eine frühere
# Installation von WebSphere eXtreme Scale, WebSphere eXtended
# Deployment DataGrid oder ObjectGrid enthält, schlägt die
# Installation fehl:
#
# Standardinstallationspfad für AIX:
#
#  -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Standardinstallationspfad für HP-UX, Solaris oder Linux:
#
#  -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Standardinstallationspfad für Windows:
#
-OPT installLocation="C:\Program Files\IBM\WebSphere\eXtremeScale"

#
# Wenn Sie die Installation unter einer Benutzer-ID ohne Root- (UNIX)
# bzw. Administratorrechte (Windows) durchführen, werden die folgenden
# Standardinstallationspfade empfohlen.
# Stellen Sie sicher, dass Sie Schreibzugriff auf den ausgewählten
# Installationspfad haben.
```

```
#
# Standardinstallationspfad für AIX:
#
#  -OPT installLocation="<Ausgangsverzeichnis_des_Benutzers>/IBM/WebSphere/eXtremeScale"
#
# Standardinstallationspfad für HP-UX, Solaris oder Linux:
#
#  -OPT installLocation="<Ausgangsverzeichnis_des_Benutzers>/IBM/WebSphere/eXtremeScale"
#
# Standardinstallationspfad für Windows:
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"


################################################################################
#  Installation von Zusatzfunktionen
#
# Geben Sie an, welche Zusatzfunktionen Sie installieren möchten, indem Sie jede
# gewünschte Funktion auf "true" setzen. Setzen Sie alle Zusatzfunktionen, die
# Sie nicht installieren möchten, auf "false".
#
# Die Optionen selectServer, selectClient, selectPF und selectXSStreamQuery sind
# nur gültig, wenn die Option installLocation zuvor eine Installation von
# WebSphere Application Server enthält. Die Optionen werden in einer
# eigenständigen Installation von WebSphere eXtreme Scale ignoriert.
#
# In der eigenständigen Installation von WebSphere eXtreme Scale werden der
# eXtreme-Scale-Server und der eXtreme-Scale-Client automatisch installiert.
# Die Zusatzeinrichtungen für die eigenständige Installation von
# eXtreme Scale sind selectXSConsoleOther und selectXSStreamQueryOther.

#
# Wenn diese Option ausgewählt ist, werden die Komponenten installiert, die
# erforderlich sind, um Server von WebSphere eXtreme Scale und den dynamischen
# Cacheserviceprovider von eXtreme Scale auszuführen. Wenn diese Option
# ausgewählt wird, muss auch WebSphere eXtreme Scale Client ausgewählt werden,
# indem das Kommentarzeichen aus der Zeile mit der entsprechenden Option
# entfernt und die Option auf "true" gesetzt wird.
# Andernfalls schlägt die unbeaufsichtigte Installation fehl.
#
-OPT selectServer="true"

#
# Wenn diese Option ausgewählt ist, werden die Komponenten installiert, die
# erforderlich sind, um die Clientanwendungen von WebSphere eXtreme Scale
# auszuführen. Wenn die Serveroption ausgewählt wurde, muss diese Option
# ebenfalls ausgewählt werden, indem das Kommentarzeichen entfernt und die
# Option auf "true" gesetzt wird. Andernfalls schlägt die unbeaufsichtigte
# Installation fehl.
#
-OPT selectClient="true"

#
# Wenn diese Option ausgewählt ist, werden die Komponenten installiert, die
# erforderlich sind, um die Konsole von WebSphere eXtreme Scale auszuführen.
# Wenn diese Option ausgewählt wird, muss der zuvor angegebene Installationspfad
# ein neues oder leeres Verzeichnis sein, da die Konsoloption nur für die
# eigenständige Implementierung von WebSphere eXtreme Scale gültig ist. Zum
# Installieren dieser Zusatzeinrichtung muss das Kommentarzeichen aus der
# folgenden Optionszeile entfernt und die Option auf "true" gesetzt werden.
#-OPT selectXSConsoleOther="false"

#
# Wenn die folgenden Optionen ausgewählt werden, werden veraltete Funktionen
# installiert.
#
# Diese Option wählt WebSphere Partition Facility zur Installation aus.
```

```
# Diese Funktion ist veraltet. Zum Installieren dieser Zusatzeinrichtung
# muss das Kommentarzeichen aus der folgenden Optionszeile entfernt und
# die Option auf "true" gesetzt werden.
#
#-OPT selectPF="false"

#
# Diese Option wählt WebSphere eXtreme Scale StreamQuery for WAS zur
# Installation aus. Diese Funktion ist veraltet. Zum Installieren
# dieser Zusatzeinrichtung muss das Kommentarzeichen aus der folgenden
# Optionszeile entfernt und die Option auf "true" gesetzt werden.
# Wenn diese Option ausgewählt wird, muss auch WebSphere eXtreme Scale Client
# ausgewählt werden, indem das Kommentarzeichen aus der entsprechenden
# Optionszeile entfernt und die Option auf "true" gesetzt wird.
# Andernfalls schlägt die unbeaufsichtigte Installation fehl.
#
#-OPT selectXSStreamQuery="false"

#
# Diese Option wählt WebSphere eXtreme Scale StreamQuery for J2SE zur
# Installation aus. Diese Funktion ist veraltet. Zum Installieren
# dieser Zusatzeinrichtung muss das Kommentarzeichen aus der folgenden
# Optionszeile entfernt und die Option auf "true" gesetzt werden.
# Wenn diese Option ausgewählt wird, muss auch WebSphere eXtreme Scale Client
# ausgewählt werden, indem das Kommentarzeichen aus der entsprechenden
# Optionszeile entfernt und die Option auf "true" gesetzt wird.
# Andernfalls schlägt die unbeaufsichtigte Installation fehl.
#
#-OPT selectXSStreamQueryOther="false"


###############################################################################
#  Profilliste für Erweiterung
#
# Geben Sie an, welches der vorhandenen Profile Sie erweitern möchten, oder
# setzen Sie diese Zeile auf Kommentar, wenn alle vorhandenen Profile, die
# während der Installation erkannt werden, erweitert werden sollen.
#
# Wenn Sie mehrere Profile angeben möchten, verwenden Sie zwischen den
# einzelnen Namen ein Komma, z. B. "AppSrv01,Dmgr01,Custom01". Die Liste
# darf keine Leerzeichen enthalten.
#
-OPT profileAugmentList=""


###############################################################################
# Tracesteuerung
#
# Das Format der Traceausgabe kann über die folgende Option gesteuert werden:
# -OPT traceFormat=ALL
#
# Die Formatoptionen sind 'text' und 'XML'. Standardmäßig werden beide Formate
# in zwei unterschiedlichen Tracedateien erzeugt.
#
# Wenn nur ein Format erforderlich ist, können Sie mit der Option traceFormat das
# gewünschte Format wie folgt angeben:
#
# Gültige Werte:
#
#  text - Die Zeilen in der Tracedatei haben zur besseren Lesbarkeit
#         ein reines Textformat.
#  XML  - Die Zeilen in der Tracedatei haben das Standard-XML-Format für
#         Java-Protokollierung, das mit jedem Text- oder XML-Editor und mit
#         dem Tool Chainsaw von Apache
#         (http://logging.apache.org/log4j/docs/chainsaw.html) angezeigt
#         werden kann.
#
```

```
# Wie viele Traceinformationen erfasst werden sollen, kann mit der
# folgenden Option gesteuert werden:
# -OPT traceLevel=INFO
#
# Gültige Werte:
#
# Stufe  Numer. Wert  Beschreibung
# -------  ---------  ---------------------------------------------------------
# OFF       0         Es wird keine Tracedatei erzeugt.
# SEVERE    1         Es werden nur schwerwiegende Fehler in der Tracedatei
#                     ausgegeben.
# WARNING   2         Der Tracedatei werden Nachrichten zu nicht schwerwiegenden
#                     Ausnahmen und Warnungen hinzugefügt.
# INFO      3         Der Tracedatei werden Informationsnachrichten hinzugefügt.
#                     (Dies ist die Standardtracestufe.)
# CONFIG    4         Der Tracedatei werden konfigurationsbezogene Nachrichten
#                     hinzugefügt.
# FINE      5         Es wird ein Trace für die Methodenaufrufe für allgemein
#                     zugängliche Methoden durchgeführt.
# FINER     6         Es wird ein Trace für die Methodenaufrufe für nicht
#                     allgemein zugängliche Methoden durchgeführt. Eine Ausnahme
#                     bilden Getter und Setter.
# FINEST    7         Es wird ein Trace für alle Methodenaufrufe durchgeführt.
#                     Der Trace für den Eintritt in die und den Austritt aus
#                     der Methode enthält die Parameter und den Rückgabewert.
```

Antwortdatei für die Installation von WebSphere eXtreme Scale Client

```
################################################################################
#
# InstallShield-Optionsdatei für IBM WebSphere eXtreme Scale Client V7.1.1
#
# Name des Assistenten: Install
# Quelle des Assistenten: setup.jar
#
# Diese Datei kann verwendet werden, um den Assistenten "Install" mit den im
# Folgenden angegebenen Optionen zu konfigurieren, wenn der Assistent mit der
# Befehlszeilenoption "-options" ausgeführt wird. Die Dokumentation zu jeder
# Einstellung enthält Informationen zum Ändern des Einstellungswerts.
# Schließen Sie alle Werte in Anführungszeichen ein.
#
# Eine gängige Verwendung für eine Optionsdatei ist die Ausführung des
# Assistenten im unbeaufsichtigten Modus. Auf diese Weise kann der Autor
# der Optionsdatei Assistenteneinstellungen festlegen, ohne den Assistenten
# im grafischen Modus oder im Konsolmodus ausführen zu müssen. Wenn Sie
# diese Optionsdatei für die Ausführung im unbeaufsichtigten Modus verwenden
# möchten, verwenden Sie die folgenden Befehlszeilenargumente bei der
# Ausführung des Assistenten:
#
#    -options "D:\installImage\WXS_Client\wxssetup.response" -silent
#
# Sie müssen den vollständig qualifizierten Namen der Antwortdatei angeben.
#
################################################################################

################################################################################
#
# Lizenz akzeptieren
#
# Gültige Werte:
#  true - Akzeptiert die Lizenz. Das Produkt wird installiert.
#  false - Die Lizenz wird nicht akzeptiert. Es findet keine Installation statt.
#
# Wenn keine Installation stattfindet, wird dies in einer temporären
# Protokolldatei im temporären Verzeichnis des Benutzers aufgezeichnet.
#
# Wenn Sie die Option silentInstallLicenseAcceptance in dieser Antwortdatei auf
```

```
# "true" setzen, geben Sie an, dass Sie die internationalen Nutzungsbedingungen
# für Programmpakete der IBM unter
# CD_ROOT\WXS_Cleint\wxs.client.primary.pak\repository\legal.xs.client\license.xs
# gelesen haben und diese akzeptieren. Falls Sie diesen Bedingungen nicht zustimmen,
# dürfen Sie diesen Wert nicht ändern und das Programm weder herunterladen,
# noch installieren, kopieren, aufrufen oder verwenden. Geben Sie das Programm
# und den Berechtigungsnachweis umgehend an den Verkäufer zurück, um sich den
# Kaufbetrag erstatten zu lassen.
#
-OPT silentInstallLicenseAcceptance="false"


################################################################################
# Nicht blockierende Prüfung der Voraussetzungen
#
# Wenn Sie die nicht blockierende Prüfung der Voraussetzungen inaktivieren möchten,
# müssen Sie das Kommentarzeichen aus der folgenden Anweisungszeile entfernen.
# Mit dieser Option wird das Installationsprogramm angewiesen, die Installation
# fortzusetzen und die Warnungen zu protokollieren, selbst wenn die
# Prüfung der Voraussetzungen scheitert.
#
#-OPT disableNonBlockingPrereqChecking="true"


################################################################################
#
# Installationspfad:
#
# Dies ist das Installationsverzeichnis für das Produkt. Geben Sie ein gültiges
# Verzeichnis für die Installation des Produkts an. Falls der Verzeichnisname
# Leerzeichen enthält, setzen Sie ihn wie im folgenden Windows-Beispiel in
# Anführungszeichen. Installationsverzeichnisse mit Leerzeichen werden nur unter
# Windows-Betriebssystemen unterstützt. Die maximale Pfadlänge für Windows
# ist 60 Zeichen.
#
# Nachfolgend sind die Standardinstallationspfade für alle unterstützten
# Betriebssysteme aufgelistet, wenn Sie die Installation als Root durchführen.
# In dieser Antwortdatei wird standardmäßig das Installationsverzeichnis für
# Windows verwendet. Falls Sie das Standardinstallationsverzeichnis für ein
# anderes Betriebssystem verwenden möchten, entfernen Sie das Kommentarzeichen
# ('#') vor dem entsprechenden Verzeichnis, und fügen Sie vor dem Eintrag für
# das Windows-Betriebssystem ein Kommentarzeichen ('#') hinzu.
#
# Der Installationspfad wird verwendet, um festzustellen, ob WebSphere eXtreme
# Scale als eigenständige Implementierung installiert oder mit einer vorhandenen
# Installation von WebSphere Application Server integriert werden muss.
#
# Wenn der angegebene Pfad eine vorhandene Installation von WebSphere Application
# Server oder WebSphere Network Deployment enthält, wird eXtreme Scale mit dem
# vorhandenen Produkt integriert. Ist der angegebene Pfad neu oder leer, wird
# WebSphere eXtreme Scale als eigenständige Implementierung installiert.
#
# Anmerkung: Wenn der angegebene Installationspfad eine frühere
# Installation von WebSphere eXtreme Scale, WebSphere eXtended
# Deployment DataGrid oder ObjectGrid enthält, schlägt die
# Installation fehl:
#
# Standardinstallationspfad für AIX:
#
#  -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Standardinstallationspfad für HP-UX, Solaris oder Linux:
#
#  -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Standardinstallationspfad für Windows:
```

```
#
-OPT installLocation="C:\Program Files\IBM\WebSphere\eXtremeScale"

#
# Wenn Sie die Installation unter einer Benutzer-ID ohne Root- (UNIX)
# bzw. Administratorrechte (Windows) durchführen, werden die folgenden
# Standardinstallationspfade empfohlen.
# Stellen Sie sicher, dass Sie Schreibzugriff auf den ausgewählten
# Installationspfad haben.
#
# Standardinstallationspfad für AIX:
#
#  -OPT installLocation="<Ausgangsverzeichnis_des_Benutzers>/IBM/WebSphere/eXtremeScale"
#
# Standardinstallationspfad für HP-UX, Solaris oder Linux:
#
#  -OPT installLocation="<Ausgangsverzeichnis_des_Benutzers>/IBM/WebSphere/eXtremeScale"
#
# Standardinstallationspfad für Windows:
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"


###############################################################################
#  Profilliste für Erweiterung
#
# Geben Sie an, welches der vorhandenen Profile Sie erweitern möchten, oder
# setzen Sie diese Zeile auf Kommentar, wenn alle vorhandenen Profile, die
# während der Installation erkannt werden, erweitert werden sollen.
#
# Wenn Sie mehrere Profile angeben möchten, verwenden Sie zwischen den
# einzelnen Namen ein Komma, z. B. "AppSrv01,Dmgr01,Custom01". Die Liste
# darf keine Leerzeichen enthalten.
#
-OPT profileAugmentList=""


###############################################################################
# Tracesteuerung
#
# Das Format der Traceausgabe kann über die folgende Option gesteuert werden:
# -OPT traceFormat=ALL
#
# Die Formatoptionen sind 'text' und 'XML'. Standardmäßig werden beide Formate
# in zwei unterschiedlichen Tracedateien erzeugt.
#
# Wenn nur ein Format erforderlich ist, können Sie mit der Option traceFormat das
# gewünschte Format wie folgt angeben:
#
# Gültige Werte:
#
#  text - Die Zeilen in der Tracedatei haben zur besseren Lesbarkeit
#         ein reines Textformat.
#  XML  - Die Zeilen in der Tracedatei haben das Standard-XML-Format für
#         Java-Protokollierung, das mit jedem Text- oder XML-Editor und mit
#         dem Tool Chainsaw von Apache
#         (http://logging.apache.org/log4j/docs/chainsaw.html) angezeigt
#         werden kann.
#
# Wie viele Traceinformationen erfasst werden sollen, kann mit der
# folgenden Option gesteuert werden:
# -OPT traceLevel=INFO
#
# Gültige Werte:
#
# Stufe  Numer. Wert  Beschreibung
# -------  ---------  ---------------------------------------------------------
```

```
# OFF       0         Es wird keine Tracedatei erzeugt.
# SEVERE    1         Es werden nur schwerwiegende Fehler in der Tracedatei
#                     ausgegeben.
# WARNING   2         Der Tracedatei werden Nachrichten zu nicht schwerwiegenden
#                     Ausnahmen und Warnungen hinzugefügt.
# INFO      3         Der Tracedatei werden Informationsnachrichten hinzugefügt.
#                     (Dies ist die Standardtracestufe.)
# CONFIG    4         Der Tracedatei werden konfigurationsbezogene Nachrichten
#                     hinzugefügt.
# FINE      5         Es wird ein Trace für die Methodenaufrufe für allgemein
#                     zugängliche Methoden durchgeführt.
# FINER     6         Es wird ein Trace für die Methodenaufrufe für nicht
#                     allgemein zugängliche Methoden durchgeführt. Eine Ausnahme
#                     bilden Getter und Setter.
# FINEST    7         Es wird ein Trace für alle Methodenaufrufe durchgeführt.
#                     Der Trace für den Eintritt in die und den Austritt aus
#                     der Methode enthält die Parameter und den Rückgabewert.
```

Profile für WebSphere eXtreme Scale erstellen und erweitern

Nachdem Sie das Produkt installiert haben, erstellen Sie eindeutige Typen von Profilen und erweitern vorhandene Profile für WebSphere eXtreme Scale.

Vorbereitende Schritte

Installieren Sie WebSphere eXtreme Scale. Weitere Informationen finden Sie im Abschnitt „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren” auf Seite 167.

Die Erweiterung von Profilen für WebSphere eXtreme Scale ist optional, aber in den folgenden Einsatzszenarien erforderlich:

- Automatisches Starten eines Katalogservice oder -containers in einem Prozess von WebSphere Application Server. Wenn die Serverprofile nicht erweitert werden, können Server nur über das Programm mit der API ServerFactory oder als gesonderte Prozesse mit Hilfe der **startOgServer**-Scripts gestartet werden.
- Verwendung von Performance Monitoring Infrastructure (PMI) für die Überwachung der eXtreme-Scale-Metriken.
- Anzeige der Version von WebSphere eXtreme Scale in der Administrationskonsole von WebSphere Application Server.

Informationen zu diesem Vorgang

Ausführung in WebSphere Application Server Version 6.1 oder Version 7.0

Wenn Ihre Umgebung WebSphere Application Server Version 6.1 oder Version 7.0 enthält, können Sie das PMT-Plug-in (Profile Management Tool) oder den Befehl "**manageprofiles**" verwenden, um Profile zu erstellen und zu erweitern.

Nächste Schritte

Abhängig davon, welche Aufgabe Sie ausführen möchten, starten Sie dafür die Konsole "Erste Schritte", um Unterstützung bei der Konfiguration und beim Test Ihrer Produktumgebung zu erhalten. Die Konsole "Erste Schritte" ist im Verzeichnis *WXS-Installationsstammverzeichnis*\firststeps\wxs\firststeps.bat enthalten. Sie können weitere Profile erstellen oder erweitern, indem Sie die vorherigen Tasks wiederholen.

Grafische Benutzerschnittstelle für die Erstellung von Profilen verwenden:

Verwenden Sie die grafische Benutzerschnittstelle (GUI), die mit dem PMT-Plug-in (Profile Management Tool) bereitgestellt wird, um Profile für WebSphere eXtreme Scale zu erstellen. Ein Profil besteht aus einer Gruppe von Dateien, die die Laufzeitumgebung definieren.

Vorbereitende Schritte

Im folgenden Szenario können Sie die grafische Benutzerschnittstelle nicht verwenden, um Profile zu erweitern:

- **64-Bit-Installationen von WebSphere Application Server**:

  Profile Management Tool ist nicht für 64-Bit-Installationen von WebSphere Application Server verfügbar. Verwenden Sie für diese Installationen das Script **manageprofiles** über die Befehlszeile.

Informationen zu diesem Vorgang

Damit Sie die Produktfeatures nutzen können, aktiviert das PMT-Plug-in (Profile Management Tool) die GUI, um Sie bei der Konfiguration von Profilen, z. B. Profile von WebSphere Application Server, Deployment-Manager-Profilen, Zellenprofilen oder angepassten Profilen, zu unterstützen. Sie können Profile während oder nach der Installation von WebSphere eXtreme Scale erweitern.

Vorgehensweise

Verwenden Sie die GUI von Profile Management Tool, um Profile zu erstellen. Wählen Sie eine der folgenden Optionen aus, um den Assistenten zu starten:

- Wählen Sie in der Konsole "Erste Schritte" die Option **Profile Management Tool** aus.
- Rufen Sie Profile Management Tool über das Menü **Start** auf.
- Führen Sie das Script "./pmt.sh|bat" im Verzeichnis *Installationsstammverzeichnis*/bin/ProfileManagement aus.

Nächste Schritte

Sie können weitere Profile erstellen oder vorhandene Profile erweitern. Zum erneuten Starten von Profile Management Tool führen Sie den Befehl **./pmt.sh|bat** im Verzeichnis *WAS-Stammverzeichnis*/bin/ProfileManagement aus, oder wählen Sie **Profile Management Tool** in der Konsole "Erste Schritte" aus.

Starten Sie einen Katalogservice, starten Sie Container, und konfigurieren Sie TCP-Ports in Ihrer Umgebung mit WebSphere Application Server. Weitere Informationen finden Sie unter „WebSphere eXtreme Scale mit WebSphere Application Server konfigurieren” auf Seite 263.

Grafische Benutzerschnittstelle für die Erweiterung von Profilen verwenden:

Nach der Installation des Produkts können Sie ein vorhandenes Profil erweitern, um es mit WebSphere eXtreme Scale kompatibel zu machen.

Informationen zu diesem Vorgang

Wenn Sie ein vorhandenes Profil erweitern, ändern Sie das Profil, indem Sie eine produktspezifische Erweiterungsschablone anwenden. Server von WebSphere eXtreme Scale werden beispielsweise nicht automatisch gestartet, wenn das Serverprofil nicht mit der Schablone "xs_augment" erweitert wurde.

- Erweitern Sie das Profil mit der Schablone "xs_augment", wenn Sie den eXtreme-Scale-Client oder den Client und den Server installiert haben.
- Erweitern Sie das Profil nur dann mit der Schablone "pf_augment", wenn Sie das Partitionierungsfeature installiert haben.
- Wenden Sie beide Schablonen an, wenn Ihre Umgebung den eXtreme-Scale-Client und das Partitionierungsfeature enthält.

Vorgehensweise

Verwenden Sie die GUI von Profile Management Tool, um Profile für eXtreme Scale zu erweitern. Wählen Sie eine der folgenden Optionen aus, um den Assistenten zu starten:

- Wählen Sie in der Konsole "Erste Schritte" die Option **Profile Management Tool** aus.
- Rufen Sie Profile Management Tool über das Menü **Start** auf.
- Führen Sie das Script "./pmt.sh|bat" im Verzeichnis *WAS-Stammverzeichnis*/bin/ProfileManagement aus.

Nächste Schritte

Sie können weitere Profile erweitern. Zum erneuten Starten von Profile Management Tool führen Sie den Befehl **./pmt.sh|bat** im Verzeichnis *WAS-Stammverzeichnis*/bin/ProfileManagement aus, oder wählen Sie **Profile Management Tool** in der Konsole "Erste Schritte" aus.

Starten Sie einen Katalogservice, starten Sie Container, und konfigurieren Sie TCP-Ports in Ihrer Umgebung mit WebSphere Application Server. Weitere Informationen hierzu finden Sie im Abschnitt „WebSphere eXtreme Scale mit WebSphere Application Server konfigurieren” auf Seite 263.

Befehl "manageprofiles":

Sie können das Dienstprogramm "**manageprofiles**" verwenden, um Profile mit der eXtreme-Scale-Schablone zu erstellen und vorhandene Anwendungsserverprofile mit den eXtreme-Scale-Erweiterungsschablonen zu erweitern bzw. deren Erweiterung aufzuheben. Zur Verwendung der Produktfeatures muss Ihre Umgebung mindestens ein Profil enthalten, das für das Produkt erweitert wurde.

- Bevor Sie Profile erstellen und erweitern können, müssen Sie eXtreme Scale installieren. Weitere Informationen finden Sie im Abschnitt „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren” auf Seite 167.

Zweck

Der Befehl "**manageprofiles**" erstellt die Laufzeitumgebung für einen Produktprozess in einem Satz von Dateien, einem so genannten Profil. Das Profil definiert die Laufzeitumgebung. Sie können mit dem Befehl "**manageprofiles**" die folgenden Aktionen ausführen:

- Deployment-Manager-Profil erstellen und erweitern
- Angepasstes Profil erstellen und erweitern
- Eigenständiges Anwendungsserverprofil erstellen und erweitern
- Zellenprofil erstellen und erweitern
- Erweiterung jedes Typs von Profil aufheben

Wenn Sie ein vorhandenes Profil erweitern, ändern Sie das Profil, indem Sie eine produktspezifische Erweiterungsschablone anwenden.

- Erweitern Sie das Profil mit der Schablone "xs_augment", wenn Sie den eXtreme-Scale-Client oder den Client und den Server installiert haben.
- Erweitern Sie das Profil mit der Schablone "pf_augment", wenn Sie nur das Partitionierungsfeature installiert haben.
- Wenden Sie beide Schablonen an, wenn Ihre Umgebung den eXtreme-Scale-Client und das Partitionierungsfeature enthält.

Position

Die Befehlsdatei befindet sich im Verzeichnis *Installationsstammverzeichnis*/bin.

Verwendung

Ausführliche Hilfe können Sie mit dem Parameter **-help** abrufen:

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/dmgr -help
```

In den folgenden Abschnitten wird jede Task, die Sie mit dem Befehl "**manageprofiles**" ausführen können, zusammen mit einer Liste der erforderlichen Parameter beschrieben. Einzelheiten zu den optionalen Parametern, die Sie für jede Task angeben können finden Sie im Artikel "Befehl **manageprofiles**" im Information Center von WebSphere Application Server.

Deployment-Manager-Profil erstellen

Sie können den Befehl "**manageprofiles**" verwenden, um ein Deployment-Manager-Profil zu erstellen. Der Deployment Manager verwaltet die Anwendungsserver, die in die Zelle eingebunden sind.

Parameter

-create
Erstellt ein Profil. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Dateipfad der Schablone an. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/dmgr
```

Schablonentyp steht für xs_augment oder pf_augment.

Beispiel

- Mit der Schablone "xs_augment":

  ```
  ./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/dmgr
  ```

- Mit der Schablone "pf_augment":

  ```
  ./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/dmgr
  ```

Angepasstes Profil erstellen

Sie können den Befehl "**manageprofiles**" verwenden, um ein angepasstes Profil zu erstellen. Ein angepasstes Profil ist ein leerer Knoten, den Sie über den Deployment Manager anpassen, indem Sie Anwendungsserver, Cluster oder andere Java-Prozesse aufnehmen.

Parameter

-create
Erstellt ein Profil. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Dateipfad der Schablone an. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/managed
```

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/managed
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/managed
```

Eigenständiges Anwendungsserverprofil erstellen

Sie können den Befehl "**manageprofiles**" verwenden, um ein eigenständiges Anwendungsserverprofil zu erstellen.

Parameter

-create
Erstellt ein Profil. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Dateipfad der Schablone an. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/default
```

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/default
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/default
```

Zellenprofil erstellen

Sie können den Befehl "**manageprofiles**" verwenden, um ein Zellenprofil zu erstellen, das sich aus einem Deployment Manager und einem Anwendungsserver zusammensetzt.

Parameter

Geben Sie die folgenden Parameter in der Deployment-Manager-Schablone an:

-create
Erstellt ein Profil. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Dateipfad der Schablone an. (Erforderlich)

Verwenden Sie das folgende Format:

`-templatePath` *`Installationsstammverzeichnis`*`/profileTemplates/`*`Schablonentyp`*`/cell/dmgr`

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Geben Sie die folgenden Parameter für die Anwendungsserverschablone an:

-create
Erstellt ein Profil. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Dateipfad der Schablone an. (Erforderlich)

Verwenden Sie das folgende Format:

`-templatePath` *`Installationsstammverzeichnis`*`/profileTemplates/`*`Schablonentyp`*`/cell/default`

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/cell/dmgr
-nodeProfilePath Installationsstammverzeichnis/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/cell/default
-dmgrProfilePath Installationsstammverzeichnis/profiles/Dmgr01 -portsFile
Installationsstammverzeichnis/profiles/Dmgr01/properties/portdef.props -nodePortsFile
Installationsstammverzeichnis/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/cell/dmgr
-nodeProfilePath Installationsstammverzeichnis/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/cell/default
-dmgrProfilePath Installationsstammverzeichnis/profiles/Dmgr01 -portsFile
Installationsstammverzeichnis/profiles/Dmgr01/properties/portdef.props -nodePortsFile
Installationsstammverzeichnis/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

Deployment-Manager-Profil erweitern

Sie können den Befehl "**manageprofiles**" verwenden, um ein Deployment-Manager-Profil zu erweitern.

Parameter

-augment
Erweitert ein vorhandenes Profil. (Erforderlich)

-profileName
Gibt den Namen des Profils an. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Pfad der Schablonendateien an, die sich im Installationsstammverzeichnis befinden. (Erforderlich)

Verwenden Sie das folgende Format:

`-templatePath` *`Installationsstammverzeichnis`*`/profileTemplates/`*`Schablonentyp`*`/dmgr`

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/dmgr
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/dmgr
```

Angepasstes Profil erweitern

Sie können den Befehl "**manageprofiles**" verwenden, um ein angepasstes Profil zu erweitern.

Parameter

-augment
Erweitert ein vorhandenes Profil. (Erforderlich)

-profileName
Gibt den Namen des Profils an. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Pfad der Schablonendateien an, die sich im Installationsstammverzeichnis befinden. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/managed
```

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/managed
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/managed
```

Eigenständiges Anwendungsserverprofil erweitern

Sie können den Befehl "**manageprofiles**" verwenden, um ein eigenständiges Anwendungsserverprofil zu erweitern.

Parameter

-augment
Erweitert ein vorhandenes Profil. (Erforderlich)

-profileName
Gibt den Namen des Profils an. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Pfad der Schablonendateien an, die sich im Installationsstammverzeichnis befinden. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/default
```

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/default
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/default
```

Zellenprofil erweitern

Sie können den Befehl "**manageprofiles**" verwenden, um ein Zellenprofil zu erweitern.

Parameter

Geben Sie die folgenden Parameter für das Deployment-Manager-Profil an:

-augment
Erweitert ein vorhandenes Profil. (Erforderlich)

-profileName
Gibt den Namen des Profils an. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Pfad der Schablonendateien an, die sich im Installationsstammverzeichnis befinden. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/cell/dmgr
```

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Geben Sie die folgenden Parameter für das Anwendungsserverprofil an:

-augment
Erweitert ein vorhandenes Profil. (Erforderlich)

-profileName
Gibt den Namen des Profils an. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Pfad der Schablonendateien an, die sich im Installationsstammverzeichnis befinden. (Erforderlich)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/cell/default
```

Schablonentyp steht für `xs_augment` oder `pf_augment`.

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath Installationsstammverzeichnis
    /profileTemplates/xs_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath Installationsstammverzeichnis
    /profileTemplates/xs_augment/cell/default
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath Installationsstammverzeichnis
    /profileTemplates/pf_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath Installationsstammverzeichnis
    /profileTemplates/pf_augment/cell/default
```

Erweiterung eines Profils aufheben

Wenn Sie die Erweiterung eines Profils aufheben möchten, geben Sie den Parameter **-ignoreStack** mit dem Parameter **-templatePath** und die erforderlichen Parameter **-unaugment** und **-profileName** an.

Parameter

-unaugment
Hebt die Erweiterung eines zuvor erweiterten Profils auf. (Erforderlich)

-profileName
Gibt den Namen des Profils an. Der Parameter wird standardmäßig verwendet, wenn keine Werte angegeben sind. (Erforderlich)

-templatePath *Schablonenpfad*
Gibt den Pfad der Schablonendateien an, die sich im Installationsstammverzeichnis befinden. (Optional)

Verwenden Sie das folgende Format:

```
-templatePath Installationsstammverzeichnis/profileTemplates/Schablonentyp/Profiltyp
```

Schablonentyp steht für `xs_augment` oder `pf_augment` und *Profiltyp* für einen der folgenden vier Profiltypen:

- `dmgr`: Deployment-Manager-Profil
- `managed`: Angepasstes Profil
- `default`: Eigenständiges Anwendungsserverprofil
- `cell`: Zellenprofil

-ignoreStack
Wird zusammen mit dem Parameter **-templatePath** verwendet, um die Erweiterung eines bestimmten erweiterten Profils aufzuheben. (Optional)

Beispiel

- Mit der Schablone "xs_augment":

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath Installationsstammverzeichnis/profileTemplates/xs_augment/Profiltyp
```

- Mit der Schablone "pf_augment":

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath Installationsstammverzeichnis/profileTemplates/pf_augment/Profiltyp
```

Profile ohne Root-Rechte:

Sie können einem Benutzer ohne Root-Rechte Berechtigungen für Dateien und Verzeichnisse erteilen, so dass dieser ein Profil für das Produkt erstellen kann. Der Benutzer ohne Root-Rechte kann auch ein Profil erweitern, das von einem Root-Benutzer oder von ihm selbst erstellt wurde.

In einer Umgebung von WebSphere Application Server können Benutzer ohne Root- bzw. Administratorrechte nur Profile in ihren eigenen Umgebungen erstellen und verwenden. Im PMT-Plug-in (Profile Management Tool) sind eindeutige Namen und Portwerte für Benutzer ohne Root-Rechte inaktiviert. Ein Benutzer ohne Root-Rechte muss die Standardfeldwerte für Profilnamen, Knotennamen, Zellennamen und Portzuordnungen in Profile Management Tool ändern. Daher sollte den Benutzern ohne Root-Rechte ein Wertebereich für jedes dieser Felder zugewiesen

werden. Sie können die Zuständigkeit für die Verwendung der richtigen Wertebereiche und die Wahrung der Integrität ihrer eigenen Definitionen an die Benutzer delegieren.

Der Begriff *Installationsverantwortlicher* bezieht sich sowohl auf Benutzer mit Root-Rechten als auch auf Benutzer ohne Root-Rechte. Als Installationsverantwortlicher können Sie Benutzern ohne Root-Rechte Berechtigungen zum Erstellen von Profilen und zum Erstellen ihrer eigenen Produktumgebungen erteilen. Beispielsweise könnte ein Benutzer ohne Root-Rechte eine Produktumgebung erstellen, um die Anwendungsimplementierung mit einem ihm bekannten Profil zu testen. Sie können die folgenden speziellen Tasks ausführen, um das Erstellen eines Profils ohne Root-Rechte zuzulassen:

- Ein Profil erstellen und einen Benutzer ohne Root-Rechte zum Eigner des Profilverzeichnisses machen, so dass dieser WebSphere Application Server für ein bestimmtes Profil starten kann.
- Einem Benutzer ohne Root-Rechte Schreibzugriff für die entsprechenden Dateien und Verzeichnisse erteilen, damit dieser das Profil erstellen kann. Wenn Sie diese Task ausführen, können Sie eine Gruppe von Benutzern erstellen, die zum Erstellen von Profilen berechtigt sind, oder einzelnen Benutzern die Berechtigung zum Erstellen von Profilen erteilen.
- Wartungspakete für das Produkt installieren, einschließlich der erforderlichen Services für vorhandene Profile, die Eigentum eines Benutzers ohne Root-Rechte sind. Als Installationsverantwortlicher sind Sie Eigner aller neuen Dateien, die vom Wartungspaket erstellt werden.

Weitere Informationen zum Erstellen von Profilen für Benutzer ohne Root-Rechte finden Sie unter Profile für Benutzer ohne Root-Rechte erstellen.

Als Installationsverantwortlicher können Sie die Berechtigung zum Erweitern von Profilen ebenfalls einem Benutzer ohne Root-Rechte erteilen. Beispielsweise kann ein Benutzer ohne Root-Rechte ein Profil erweitern, das von einem Installationsverantwortlichen oder von ihm selbst erstellt wurde. Befolgen Sie in WebSphere Application Server Network Deployment den Prozess zur Erweiterung von Profilen für Benutzer ohne Root-Rechte.

Wenn ein Benutzer ohne Root-Rechte jedoch ein Profil erweitert, das vom Installationsverantwortlichen erstellt wurde, muss der Benutzer ohne Root-Rechte die folgenden Dateien vor der Erweiterung nicht erstellen: Die folgenden Dateien werden während des Profilerstellungsprozesses erstellt:

- *WAS-Stammverzeichnis*`/logs/manageprofiles.xml`
- *WAS-Stammverzeichnis*`/properties/fsdb.xml`
- *WAS-Stammverzeichnis*`/properties/profileRegistry.xml`

Wenn ein Benutzer ohne Root-Rechte ein von ihm selbst erstelltes Profil erweitert, muss er die Berechtigungen für die Dokumente ändern, die sich in den Profilschablonen von eXtreme Scale befinden.

Achtung: Sie können ein Profil ohne Root-Rechte (Administratorrechte) auch für WebSphere eXtreme Scale in einer eigenständigen Umgebung verwenden, d. h. einer Umgebung außerhalb von WebSphere Application Server. Sie müssen den Eigner des ObjectGrid-Verzeichnisses in das Profil ohne Root-Rechte ändern. Anschließend können Sie sich mit diesem Profil ohne Root-Rechte anmelden und wie mit einem Profil mit Root-Rechten (Administratorrechten) mit eXtreme Scale arbeiten.

WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client standalone installieren

Sie können WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client standalone (eigenständig) in einer Umgebung installieren, in der WebSphere Application Server bzw. WebSphere Application Server Network Deployment nicht vorhanden ist.

Vorbereitende Schritte

- Stellen Sie sicher, dass das Zielinstallationsverzeichnis leer bzw. nicht vorhanden ist.

  Wichtig: Wenn eine frühere Version von WebSphere eXtreme Scale oder die Komponente "ObjectGrid" in dem Verzeichnis vorhanden ist, das Sie für die Installation von Version 7.1.1 angeben, wird das Produkt nicht installiert. Sie können beispielsweise bereits einen Ordner *WXS-Installationsstammverzeichnis*/`ObjectGrid` haben. In diesem Fall können Sie ein anderes Installationsverzeichnis angeben oder die Installation abbrechen. Anschließend deinstallieren Sie die frühere Installation und führen dann den Assistenten erneut aus.
- Eine IBM Runtime Environment wird im Rahmen der eigenständigen Installation im Ordner *WXS-Installationsstammverzeichnis*`/java` installiert.
- Reine Clientinstallation: Laden Sie WebSphere eXtreme Scale Client für die entsprechende Plattform von der Support site herunter.

Informationen zu diesem Vorgang

Wenn Sie das Produkt eigenständig installieren, installieren Sie Client und Server von WebSphere eXtreme Scale zusammen. Mit der Installation von WebSphere eXtreme Scale Client im eigenständigen Modus installieren Sie einen Client für den Zugriff auf die Daten in Ihren Datengrids. Server- und Clientprozesse greifen deshalb auf alle erforderlichen Ressourcen lokal zu. Sie können WebSphere eXtreme Scale auch mit Scripts und JAR-Dateien (Java-Archiv) in vorhandene J2SE-Anwendungen (Java Platform, Standard Edition) integrieren.

Achtung: Sie können ein Profil ohne Root-Rechte (Administratorrechte) auch für WebSphere eXtreme Scale in einer eigenständigen Umgebung verwenden. Wenn Sie ein Profil ohne Root-Rechte verwenden möchten, müssen Sie den Eigner des Verzeichnisses `ObjectGrid` in das Profil ohne Root-Rechte ändern. Anschließend können Sie sich mit diesem Profil ohne Root-Rechte anmelden und wie mit einem Profil mit Root-Rechten (Administratorrechten) mit eXtreme Scale arbeiten.

Vorgehensweise

1. Assistenten für die Installation des Servers und des Clients von der DVD verwenden.
   - Führen Sie das folgende Script aus, um den Assistenten für die vollständige Installation von WebSphere eXtreme Scale zu starten:
     - Linux UNIX *DVD-Stammverzeichnis*`/install`
     - Windows *DVD-Stammverzeichnis*`\install.bat`
   - Führen Sie das folgende Script aus, um den Assistenten für die Installation von WebSphere eXtreme Scale Client zu starten. Die Installationsdateien sind in der ZIP-Datei enthalten, die sie von der Unterstützungssite herunterladen:
     - Linux UNIX `Stammverzeichnis/WXS_Client/install`
     - Windows `Stammverzeichnis\WXS_Client\install.bat`

Achtung: Wenn Sie einheitliche Namenskonventionen (UNC, Uniform Naming Convention) für die Identifizierung von Dateipfaden in Ihrem Installationsbefehl verwenden, werden die geplanten Komponenten nach der Ausführung des Befehls möglicherweise nicht alle installiert. Zur Vermeidung von Problemen ordnen Sie den Dateipfad einem Netzlaufwerk zu. Führen Sie den Befehl **install** für das zugeordnete Laufwerk aus. Durch die Verwendung eines zugeordneten wird sichergestellt, dass alle Komponenten installiert werden.

2. Folgen Sie der Bedienerführung des Assistenten, und klicken Sie anschließend auf **Fertig stellen**.

   Einschränkung: In der Anzeige mit den optionalen Features werden die Features aufgelistet, die Sie installieren können. Features können der Produktumgebung nach der Produktinstallation jedoch nicht einzeln hinzugefügt werden. Wenn Sie sich während der Erstinstallation des Produkts gegen die Installation eines Features entscheiden, müssen Sie das Produkt deinstallieren und erneut installieren, um das Feature hinzuzufügen.

Ergebnisse

Windows Wenn Sie WebSphere eXtreme Scale Client unter Windows installieren, finden Sie in den Ergebnissen der Installation möglicherweise den folgenden Text:

```
Erfolg: Die Installation des folgenden Produkts war erfolgreich:
WebSphere eXtreme Scale Client. In einigen Konfigurationsschritten sind Fehler
aufgetreten. Weitere Informationen finden Sie in der folgenden Protokolldatei:
<WAS-Installationsstammverzeichnis>\logs\wxs_client\install\log.txt"
Sehen Sie sich das Installationsprotokoll (log.txt) und das Erweiterungsprotokoll
des Deployment Manager an.
```

Wenn Sie einen Fehler sehen, der sich auf die Datei `iscdeploy.sh` bezieht, können Sie diesen ignorieren. Dieser Fehler verursacht keine Probleme.

Nächste Schritte

- Überprüfen Sie die Installation. Weitere Informationen finden Sie unter „Installation überprüfen" auf Seite 212.
- Beginnen Sie mit der Installation von WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client. Weitere Informationen finden Sie unter „Erste Schritte nach der Installation" auf Seite 213.

Laufzeitdateien für eine eigenständige Installation von WebSphere eXtreme Scale

JAR-Dateien (Java Archive) sind in der Installation enthalten. Sie können die enthaltenen JAR-Dateien und die Position, an denen sie installiert sind, anzeigen.

Tabelle 7. Laufzeitdateien für eine vollständige Installation von WebSphere eXtreme Scale. WebSphere eXtreme Scale stützt sich auf ObjectGrid-Prozesse und zugehörige APIs. In der folgenden Tabelle sind die JAR-Dateien aufgelistet, die in der Installation enthalten sind. Die Installationsposition ist relativ zum Verzeichnis von *WXS-Ausgangsverzeichnis*, das Sie während der Installation auswählen.

Dateiname	Umgebung	Installationsposition	Beschreibung
`wxsdynacache.jar`	Client und Server	`dynacache/lib`	Die Datei `wxsdynacache.jar` enthält die erforderlichen Klassen für den dynamischen Cache-Provider. Die Datei wird automatisch in die Serverlaufzeitumgebung eingeschlossen, wenn Sie die bereitgestellten Scripts verwenden.
`wxshyperic.jar`	Dienstprogramm	`hyperic/lib`	Das Erkennungs-Plug-in von WebSphere eXtreme Scale für den SpringSource-Hyperic-Überwachungsagenten

Tabelle 7. Laufzeitdateien für eine vollständige Installation von WebSphere eXtreme Scale (Forts.). WebSphere eXtreme Scale stützt sich auf ObjectGrid-Prozesse und zugehörige APIs. In der folgenden Tabelle sind die JAR-Dateien aufgelistet, die in der Installation enthalten sind. Die Installationsposition ist relativ zum Verzeichnis von *WXS-Ausgangsverzeichnis*, das Sie während der Installation auswählen.

Dateiname	Umgebung	Installationsposition	Beschreibung
`objectgrid.jar`	Lokal, Client und Server	`lib`	Die Datei `objectgrid.jar` ist ein OSGi-Bundle, das von der Serverlaufzeitumgebung von Java SE 5.0 und höher verwendet wird. Die Datei wird automatisch in die Serverlaufzeitumgebung eingeschlossen, wenn Sie die bereitgestellten Scripts verwenden.
`ogagent.jar`	Lokal, Client und Server	`lib`	Die Datei `ogagent.jar` enthält die Laufzeitklassen, die für die Ausführung des Java-Instrumentierungsagenten erforderlich sind, der mit der API "EntityManager" verwendet wird.
`ogclient.jar`	Lokal und Client	`lib`	Die Datei `ogclient.jar` ist ein OSGi-Bundle, das nur die lokale und die Clientlaufzeitumgebung enthält. Sie können diese Datei mit Java SE 5.0 und höher verwenden.
`ogspring.jar`	Lokal, Client und Server	`lib`	Die Datei `ogspring.jar` lt enthält Unterstützungsklassen für die Integration des SpringSource-Spring-Frameworks.
`wsogclient.jar`	Lokal und Client	`lib`	Die Datei `wsogclient.jar` wird installiert, wenn Sie eine Umgebung verwenden, die WebSphere Application Server Version 6.0.2 und höher verwendet. Diese Datei enthält nur die lokale Laufzeitumgebung und die Clientlaufzeitumgebung.
`wxssizeagent.jar`	Lokal, Client und Server	`lib`	Die Datei `wxssizeagent.jar` wird verwendet, um genauere Größeninformationen zu den Cacheeinträgen bereitzustellen, wenn Java Runtime Environment (JRE) Version 1.5 oder höher verwendet wird.

`ibmcfw.jar`
`ibmorb.jar`
`ibmorbapi.jar`
	Client und Server	`lib/endorsed`	Diese Gruppe von Dateien enthält die ORB-Laufzeitumgebung, die für die Ausführung von Anwendungen in Java-SE-Prozessen verwendet wird.

`restservice.ear`	Client	`restservice/lib`	Die Datei `restservice.ear` enthält das Unternehmensarchiv der REST-Datenserviceanwendung von eXtreme Scale für Umgebungen von WebSphere Application Server.
`restservice.war`	Client	`restservice/lib`	Die Datei `restservice.war` enthält das Webarchiv des REST-Datenservice von eXtreme Scale für Anwendungsserver eines anderen Anbieters.
`xsadmin.jar`	Dienstprogramm	`samples`	Die Datei `xsadmin.jar` enthält das Verwaltungsdienstprogramm für die Beispiele von eXtreme Scale.
`sessionobjectgrid.jar`	Client und Server	`session/lib`	Die Datei `sessionobjectgrid.jar` enthält die Laufzeitumgebung von eXtreme Scale für das HTTP-Sitzungsmanagement.
`splicerlistener.jar`	Dienstprogramm	`session/lib`	Die Datei `splicerlistener.jar` enthält das Dienstprogramm "splicer" für den HTTP-Sitzungs-Listener von eXtreme Scale Version 7.1 und höher.
`xsgbean.jar`	Server	`wasce/lib`	Die Datei `xsgbean.jar` enthält die GBean für die Integration von Servern von eXtreme Scale in Anwendungsservern von WebSphere Application Server Community Edition.
`splicer.jar`	Dienstprogramm	`legacy/session/lib`	Das Dienstprogramm "splicer" für den HTTP-Sitzungsmanagerfilter von WebSphere eXtreme Scale Version 7.0.

Tabelle 8. Laufzeitdateien für WebSphere eXtreme Scale Client. WebSphere eXtreme Scale Client stützt sich auf ObjectGrid-Prozesse und zugehörige APIs. In der folgenden Tabelle sind die JAR-Dateien aufgelistet, die in der Installation enthalten sind. Die Installationsposition ist relativ zum Verzeichnis von *WXS-Ausgangsverzeichnis*, das Sie während der Installation auswählen.

Dateiname	Umgebung	Installationsposition	Beschreibung
`wxsdynacache.jar`	Client und Server	`dynacache/lib`	Die Datei `wxsdynacache.jar` enthält die erforderlichen Klassen für den dynamischen Cache-Provider. Die Datei wird automatisch in die Serverlaufzeitumgebung eingeschlossen, wenn Sie die bereitgestellten Scripts verwenden.
`wxshyperic.jar`	Dienstprogramm	`hyperic/lib`	Das Erkennungs-Plug-in von WebSphere eXtreme Scale für den SpringSource-Hyperic-Überwachungsagenten
`ogagent.jar`	Lokal, Client und Server	`lib`	Die Datei `ogagent.jar` enthält die Laufzeitklassen, die für die Ausführung des Java-Instrumentierungsagenten erforderlich sind, der mit der API "EntityManager" verwendet wird.

Tabelle 8. Laufzeitdateien für WebSphere eXtreme Scale Client (Forts.). WebSphere eXtreme Scale Client stützt sich auf ObjectGrid-Prozesse und zugehörige APIs. In der folgenden Tabelle sind die JAR-Dateien aufgelistet, die in der Installation enthalten sind. Die Installationsposition ist relativ zum Verzeichnis von *WXS-Ausgangsverzeichnis*, das Sie während der Installation auswählen.

Dateiname	Umgebung	Installationsposition	Beschreibung
`ogclient.jar`	Lokal und Client	`lib`	Die Datei `ogclient.jar` ist ein OSGi-Bundle, das nur die lokale und die Clientlaufzeitumgebung enthält. Sie können diese Datei mit Java SE 5 und höher verwenden.
`ogspring.jar`	Lokal, Client und Server	`lib`	Die Datei `ogspring.jar` lt enthält Unterstützungsklassen für die Integration des SpringSource-Spring-Frameworks.
`wsogclient.jar`	Lokal und Client	`lib`	Die Datei `wsogclient.jar` wird installiert, wenn Sie eine Umgebung verwenden, die WebSphere Application Server Version 6.0.2 und höher verwendet. Diese Datei enthält nur die lokale Laufzeitumgebung und die Clientlaufzeitumgebung.
`wxssizeagent.jar`	Lokal, Client und Server	`lib`	Die Datei `wxssizeagent.jar` wird verwendet, um genauere Größeninformationen zu den Cacheeinträgen bereitzustellen, wenn Java Runtime Environment (JRE) Version 1.5 oder höher verwendet wird.

`ibmcfw.jar`
`ibmorb.jar`
`ibmorbapi.jar`
	Client und Server	`lib/endorsed`	Diese Gruppe von Dateien enthält die ORB-Laufzeitumgebung, die für die Ausführung von Anwendungen in Java-SE-Prozessen verwendet wird.

`restservice.ear`	Client	`restservice/lib`	Die Datei `restservice.ear` enthält das Unternehmensarchiv der REST-Datenserviceanwendung von eXtreme Scale für Umgebungen von WebSphere Application Server.
`restservice.war`	Client	`restservice/lib`	Die Datei `restservice.war` enthält das Webarchiv des REST-Datenservice von eXtreme Scale für Anwendungsserver eines anderen Anbieters.
`xsadmin.jar`	Dienstprogramm	`samples`	Die Datei `xsadmin.jar` enthält das Verwaltungsdienstprogramm für die Beispiele von eXtreme Scale.
`sessionobjectgrid.jar`	Client und Server	`session/lib`	Die Datei `sessionobjectgrid.jar` enthält die Laufzeitumgebung von eXtreme Scale für das HTTP-Sitzungsmanagement.
`splicerlistener.jar`	Dienstprogramm	`session/lib`	Die Datei `splicerlistener.jar` enthält das Dienstprogramm "splicer" für den HTTP-Sitzungs-Listener von eXtreme Scale Version 7.1 und höher.
`splicer.jar`	Dienstprogramm	`legacy/session/lib`	Das Dienstprogramm "splicer" für den HTTP-Sitzungsmanagerfilter von WebSphere eXtreme Scale Version 7.0.

WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client im unbeaufsichtigten Modus installieren

Verwenden Sie eine vollständig qualifizierte Antwortdatei, die Sie speziell für Ihre Anforderungen konfigurieren, oder übergeben Sie Parameter in der Befehlszeile, um WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client unbeaufsichtigt zu installieren.

Vorbereitende Schritte

- Stoppen Sie alle Prozesse, die in Ihrer Umgebung mit WebSphere Application Server oder WebSphere Application Server Network Deployment aktiv sind. Weitere Informationen zu den Befehlen **stopManager**, **stopNode** und **stopServer** finden Sie unter Befehlszeilendienstprogramme.

  Vorsicht:
  Stellen Sie sicher, dass alle aktiven Prozesse gestoppt sind. Wenn die aktiven Prozesse nicht gestoppt sind, wird die Installation fortgesetzt, was zu unvorhersehbaren Ereignissen führen und die Installation auf einigen Plattformen in einem unbestimmten Zustand hinterlassen kann.
- Stellen Sie sicher, dass das Zielinstallationsverzeichnis leer bzw. nicht vorhanden ist.

Wichtig: Wenn eine frühere Version von WebSphere eXtreme Scale oder die Komponente "ObjectGrid" in dem Verzeichnis vorhanden ist, das Sie für die Installation von Version 7.1.1 angeben, wird das Produkt nicht installiert. Sie können beispielsweise bereits einen Ordner *WXS-Installationsstammverzeichnis*/`ObjectGrid` haben. In diesem Fall können Sie ein anderes Installationsverzeichnis angeben oder die Installation abbrechen. Anschließend deinstallieren Sie die frühere Installation und führen dann den Assistenten erneut aus.

Informationen zu diesem Vorgang

Bei einer unbeaufsichtigten Installation wird dasselbe Installationsprogramm verwendet, das auch bei der Installation über die grafische Benutzerschnittstelle verwendet wird. Bei der unbeaufsichtigten Installation wird jedoch keine Assistentenschnittstelle angezeigt. Stattdessen werden Ihre Antworten aus einer Datei gelesen, die Sie anpassen, bzw. aus Parametern, die die in der Befehlszeile übergeben. Sehen Sie sich das Beispiel für die „Datei `wxssetup.response.txt`" auf Seite 181 an, das eine Beschreibung jeder Option enthält.

Vorgehensweise

1. Optional: Wenn Sie sich für die Installation von WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client über eine Antwortdatei entscheiden, müssen Sie zuerst die Datei `wxssetup.response.txt` anpassen.

   Hinweis: Sie müssen den vollständig qualifizierten Namen der Antwortdatei angeben. Wenn Sie den relativen Pfad angeben, scheitert die Installation ohne Ausgabe einer entsprechenden Fehlernachricht.

   a. Erstellen Sie eine Kopie der anzupassenden Antwortdatei.

      Für die vollständige Installation von WebSphere eXtreme Scale kopieren Sie die Antwortdatei von der Produkt-DVD auf Ihr Plattenlaufwerk.

      Für WebSphere eXtreme Scale Client dekomprimieren Sie die ZIP-Datei von WebSphere eXtreme Scale Client auf Ihrem Festplattenlaufwerk. Suchen Sie anschließend die Antwortdatei.

   b. Öffnen Sie die Antwortdatei mit einem Texteditor Ihrer Wahl, und bearbeiten Sie sie. Die vorherige Beispielantwortdatei enthält Details zur Angabe der einzelnen Parameter. Sie müssen die folgenden Parameter angeben:

      - Lizenzvereinbarung
      - Installationsverzeichnis

      Tipp: Wenn Sie WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client in einer Umgebung von WebSphere Application Server installieren, verwendet das Installationsprogramm das Installationsverzeichnis, um zu bestimmen, wo die vorhandene Instanz von WebSphere Application Server installiert ist. Wenn Sie die Installation auf einem Knoten durchführen, der mehrere Instanzen von WebSphere Application Server enthält, müssen Sie Ihren Standort eindeutig definieren.

   c. Führen Sie das folgende Script aus, um die Installation zu starten.

      Für die vollständige Installation von WebSphere eXtreme Scale:

      ```
      ./install.sh|bat -options C:/Laufwerkspfad/Antwortdatei.txt -silent
      ```

      Für die Installation von WebSphere eXtreme Scale Client:

      ```
      ./WXS_Client/install.sh|bat -options C:/Laufwerkspfad/Antwortdatei.txt -silent
      ```

      Sie können die Antwortdatei auch verwenden, wenn Sie eine Installation über die grafische Benutzerschnittstelle durchführen. Mit der Antwortdatei können Sie bei einer Installation über die grafische Benutzerschnittstelle Pro-

bleme beheben, die bei der unbeaufsichtigten Installation verborgen bleiben. Wenn Sie die Datei `wxssetup.response` für Installationen über die grafische Benutzerschnittstelle und für unbeaufsichtigte Installationen angeben, müssen Sie den vollständig qualifizierten Pfad verwenden. Führen Sie das folgende Script aus, um die Installation über die grafische Benutzerschnittstelle mit der Antwortdatei durchzuführen:

- Linux UNIX `<Ausgangsverzeichnis_der_Installation>/install.sh -options <vollständig_qualifizierter_Installationspfad>/wxssetup.response`
- Windows `<Ausgangsverzeichnis_der_Installation>\install.exe -options c:\<vollständig_qualifizierter_Installationspfad>\wxssetup.response`

2. Optional: Wenn Sie sich für die Installation von eXtreme Scale durch Übergabe bestimmter Parameter in der Befehlszeile entscheiden, führen Sie das folgende Script aus, um die Installation zu starten:

   Für die vollständige Installation von WebSphere eXtreme Scale:

   `./install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=Installationsposition`

   Für die Installation von WebSphere eXtreme Scale Client:

   `./WXS_Client/install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=Installationsposition`

Antwortdatei für unbeaufsichtigte Installation

Geben Sie Parameter in der Befehlszeile an, um Ihre Produktinstallation anzupassen und zu konfigurieren.

Anmerkung: Sie müssen den vollständig qualifizierten Namen der Antwortdatei angeben. Wenn Sie den relativen Pfad angeben, scheitert die Installation ohne Ausgabe einer entsprechenden Fehlernachricht.

Parameter

Sie können die folgenden Parameter bei einer Installation des Produkts über die Befehlszeile oder eine Optionsdatei übergeben:

`-silent`
Unterdrückt die grafische Benutzerschnittstelle (GUI). Geben Sie den Parameter **`-options`** an, um anzuzeigen, dass das Installationsprogramm die Installation auf der Basis einer angepassten Optionsdatei durchführen soll. Wenn Sie den Parameter **`-options`** nicht angeben, werden die Standardwerte verwendet.

Beispielsyntax

`./install.sh|bat -silent -options Optionsdatei.txt`

`-options` *`Pfadname/Dateiname`*
Gibt eine Optionsdatei an, die das Installationsprogramm für die Durchführung einer unbeaufsichtigten Installation verwenden soll. Angaben in der Befehlszeile haben Vorrang.

Beispielsyntax

`./install.sh|bat -options c:/Pfadname/Optionsdatei.txt`

`-log # !file_name @`*`Ereignistyp`*
Generiert eine Installationsprotokolldatei, in der die folgenden Ereignistypen protokolliert werden:

- `err`
- `wrn`

- msg1
- msg2
- dbg
- ALL

Beispielsyntax

```
./install.sh|bat -log # !c:/temp/logfiles.txt @ALL
```

-is:log *Pfadname/Dateiname*
Erstellt eine Protokolldatei, die die Suchoperationen des Installationsprogramms in der JVM beim Versuch, die GUI zu starten, enthält. Die Protokolldatei wird nur erstellt, wenn Sie dieses angeben.

Beispielsyntax

```
./install.sh|bat -is:log c:/logs/javalog.txt
```

-is:javaconsole
Während des Installationsprozesses wird ein Konsolfenster angezeigt.

Beispielsyntax

```
./install.sh|bat -is:javaconsole
```

-is:silent
Unterdrückt das Java-Initialisierungsfenster, das beim Start des Installationsprogramms angezeigt wird.

Beispielsyntax

```
./install.sh|bat -is:silent
```

-is:tempdir *Pfadname*
Gibt das temporäre Verzeichnis an, das vom Installationsprogramm während der Installation verwendet wird.

Beispielsyntax

```
./install.sh|bat -is:tempdir c:/temp
```

REST-Datenservice installieren

In diesem Abschnitt wird beschrieben, wie der REST-Datenservice von WebSphere eXtreme Scale in einem Webserver installiert wird.

Vorbereitende Schritte

Softwarevoraussetzungen

Der REST-Datenservice von WebSphere eXtreme Scale ist eine Java-Webanwendung, die in jedem Anwendungsserver implementiert werden kann, der die Java-Servlet-Spezifikation Version 2.3 und eine Java Runtime Environment der Version Version 5 oder höher unterstützt.

Die folgende Software ist erforderlich:

- Java Standard Edition 5 oder höher
- Web-Servlet-Container Version 2.3 oder höher mit einer der folgenden Komponenten:
  - WebSphere Application Server Version 6.1.0.25 oder höher
  - WebSphere Application Server Version 7.0.0.5 oder höher
  - WebSphere Community Edition Version 2.1.1.3 oder höher

  - Apache Tomcat Version 5.5 oder höher
- WebSphere eXtreme Scale Version 7.1 oder höher ,einschließlich der Testversion.

Informationen zu diesem Vorgang

Der REST-Datenservice von WebSphere eXtreme Scale enthält eine einzige Datei `wxsrestservice.war`. Die Datei `wxsrestservice.war` enthält ein einziges Servlet, das als Gateway zwischen Ihren WCF-Data-Services-Clientanwendungen oder einem anderen HTTP-REST-Client und einem Datengrid agiert.

Der REST-Datenservice enthält ein Beispiel, das Ihnen ermöglicht, schnell ein Datengrid zu erstellen und mit diesem über einen eXtreme-Scale-Client oder den REST-Datenservice zu interagieren. Weitere Einzelheiten zur Verwendung des Beispiels finden Sie unter „REST-Datenservices konfigurieren" auf Seite 365.

Wenn WebSphere eXtreme Scale 7.1 installiert oder die Testversion von eXtreme Scale Version 7.1 extrahiert wird, sind die folgenden Verzeichnisse und Dateien enthalten:

- `Ausgangsverzeichnis_des_REST-Service/lib`

  Das Verzeichnis "lib" enthält die folgenden Dateien:

  - `wxsrestservice.ear` – Der Unternehmensanwendungsarchiv des REST-Datenservice für WebSphere Application Server und WebSphere Application Server CE.
  - `wxsrestservice.war` – Das Webmodul des REST-Datenservice für Apache Tomcat.

  Die Datei `wxsrestservice.ear` enthält die Datei `wxsrestservice.war`, und beide sind eng mit der Laufzeitumgebung von WebSphere WebSphere eXtreme Scale gekoppelt. Wenn ein Upgrade von WebSphere eXtreme Scale auf eine neue Version vorgenommen oder ein Fixpack angewendet wird, muss die Datei `wxsrestservice.war` bzw. `wxsrestservice.ear` manuell auf die in diesem Verzeichnis installiert Version aktualisiert werden.
- `Ausgangsverzeichnis_des_REST-Service/gettingstarted`

  Das Verzeichnis `gettingstarted` enthält ein einfaches Beispiel, das veranschaulicht, wie der REST-Datenservice von WebSphere eXtreme Scale mit einem Datengrid verwendet wird.

Vorgehensweise

Packen und implementieren Sie den REST-Datenservice.
Der REST-Datenservice ist als eigenständiges WAR-Modul entworfen. Zum Konfigurieren des REST-Datenservice müssen Sie die Konfiguration des REST-Datenservice und die optionalen Konfigurationsdateien von WebSphere eXtreme Scale in eine JAR-Datei oder in ein Verzeichnis packen. Dieses Anwendungspaket wird anschließend von der Laufzeitumgebung des Webcontainerservers referenziert. Die folgende Abbildung zeigt die vom REST-Datenservice von eXtreme Scale verwendeten Dateien.

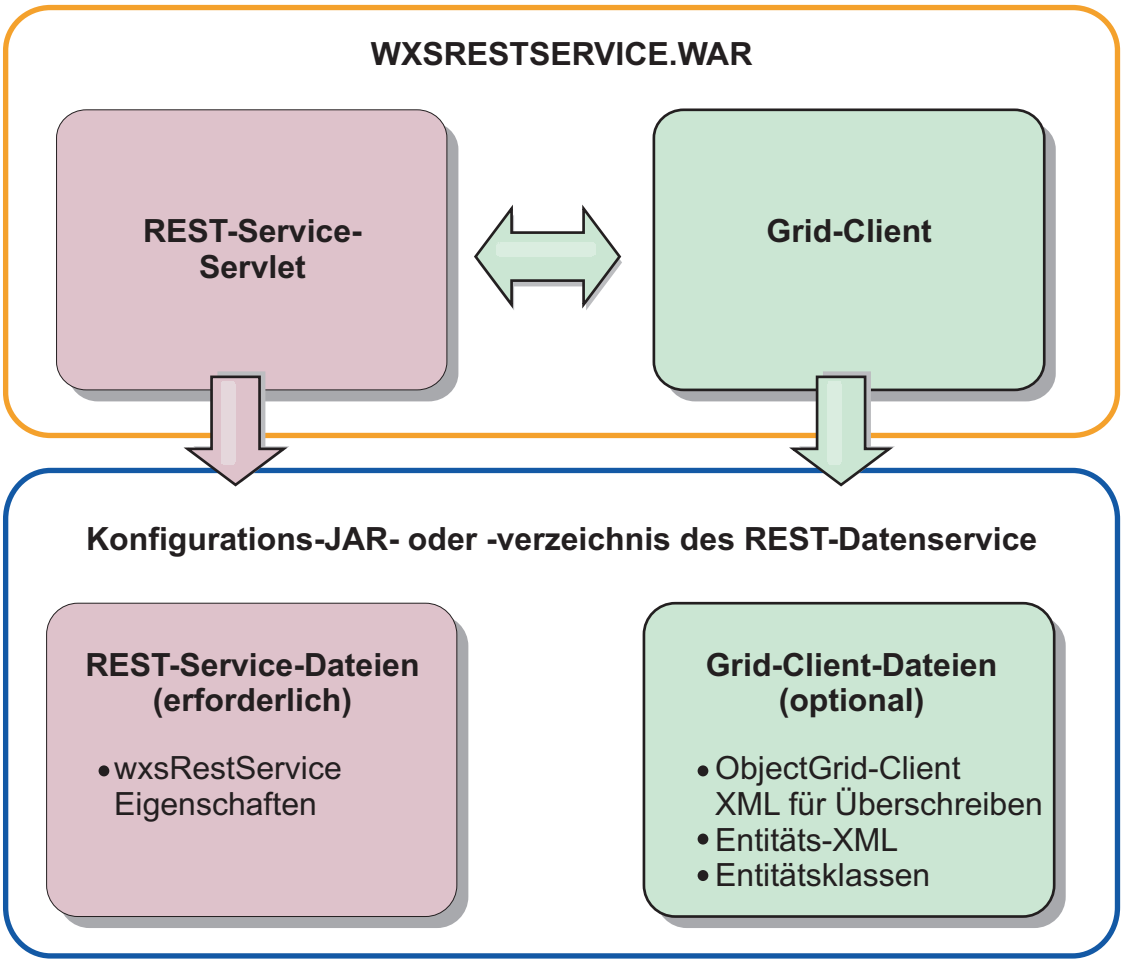


Abbildung 26. Dateien des REST-Datenservice von WebSphere eXtreme Scale

Die Konfigurations-JAR-Datei bzw. das Konfigurationsverzeichnis des REST-Service muss die folgenden Dateien enthalten:
`wxsRestService.properties`: Die Datei `wxsRestService.properties` enthält die Konfigurationsoptionen für den REST-Datenservice. Dazu gehören die Katalogserviceendpunkte, die Namen der bereitzustellenden ObjectGrids, Traceoptionen usw. Weitere Informationen finden Sie unter Eigenschaftendatei des REST-Datenservice.
Die folgenden ObjectGrid-Clientdateien sind optional:

- `META-INF/objectGridClient.xml`: Die XML-Datei für das Überschreiben von ObjectGrid-Clients wird verwendet, um eine Verbindung zum fernen Datengrid herzustellen. Diese Datei ist standardmäßig nicht erforderlich. Wenn diese Datei nicht vorhanden ist, verwendet der REST-Service die Serverkonfiguration, woraufhin der nahe Cahe inaktiviert wird.

  Der Name der Datei kann mit der Konfigurationseigenschaft "objectGridClientXML" des REST-Datenservice überschrieben werden. Wenn diese XML-Datei bereitgestellt wird, muss sie Folgendes enthalten:

  1. Alle ObjectGrids, die Sie dem REST-Service bereitstellen möchten
  2. Referenz auf die XML-Entitätsdeskriptordatei, die jeder ObjectGrid-Konfiguration zugeordnet ist

- `META-INF/`*`XML-Entitätsdeskriptordateien`*: XML-Entitätsdeskriptordateien sind nur erforderlich, wenn der Client die Entitätsdefinition des Clients überschreiben muss. die XML-Entitätsdeskriptordatei muss zusammen mit der XML-Deskriptordatei für das Überschreiben von ObjectGrid-Clients verwendet werden.

- **Entitätsklassen**: Zum Beschreiben der Entitätsmetadaten können Sie annotierte Entitätsklassen oder eine XML-Entitätsdeskriptordatei verwenden. DER REST-Service erfordert nur dann Entitätsklassen im Klassenpfad, wenn die eXtreme-Scale-Server mit Entitätsmetadatenklassen konfiguriert wurden und keine XML-Entitätsdeskriptordatei für das Überschreiben von Clients verwendet wird.

  Im Folgenden sehen Sie ein Beispiel mit der erforderlichen Mindestkonfigurationsdatei, in dem die Entitäten in der XML auf den Servern definiert sind:

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  Die Eigenschaftendatei enthält Folgendes:

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  Beispiel mit einer einzigen Entität, XML-Dateien für Überschreiben und Entitätsklassen:

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  Die Eigenschaftendatei enthält Folgendes:

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  ```
  com/acme/entities/Customer.class
  META-INF/objectGridClient.xml
  ```

  Die ObjectGrid-XML-Deskriptordatei des Clients enthält Folgendes:

  ```
  <objectGrid name="CustomerGrid" entityMetadataXMLFile="emd.xml"/>
  META-INF/emd.xml
  ```

  Die XML-Deskriptordatei für die Entitätsmetadaten enthält Folgendes:

  ```
  <entity class-name="com.acme.entities.Customer" name="Customer"/>
  ```

Eclipse-Equinox-OSGi-Framework mit Eclipse Gemini für Clients und Server installieren

Wenn Sie WebSphere eXtreme Scale im OSGi-Framework implementieren möchten, müssen Sie die Eclipse-Equinox-Umgebung einrichten.

Informationen zu diesem Vorgang

Diese Aufgabe erfordert, dass Sie das Blueprint-Framework herunterladen und installieren, das Ihnen ermöglicht, JavaBeans zu einem späteren Zeitpunkt zu konfigurieren und als Services bereitzustellen. Die Verwendung von Services ist wichtig, weil Sie Plug-ins als OSGi-Services bereitstellen können, damit diese von der Laufzeitumgebung von eXtreme Scale verwendet werden können. Das Produkt unterstützt zwei Blueprint-Container im Eclipse-Equinox-Basis-OSGi-Framework: Eclipse Gemini und Apache Aries. Verwenden Sie die folgende Prozedur, um den Eclipse-Gemini-Container zu konfigurieren.

Vorgehensweise

1. Laden Sie Eclipse Equinox SDK Version 3.6.1 oder höher von der Eclipse-Website herunter. Erstellen Sie ein Verzeichnis für das Equinox-Framework, z. B. `/opt/equinox`. In den folgenden Anweisungen wird das Verzeichnis `equinox_root` verwendet. Entpacken Sie die komprimierte Datei im Verzeichnis `equinox_root`.

2. Laden Sie die komprimierte Datei für gemini-blueprint incubation 1.0.0 von der Eclipse-Website herunter. Extrahieren Sie den Dateiinhalt in ein temporäres Verzeichnis, und kopieren Sie die folgenden extrahierten Dateien in das Verzeichnis `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
   ```

3. Laden Sie Spring Framework Version 3.0.5 von der folgenden Webseite mit dem Spring-Quellcode herunter: http://www.springsource.com/download/community. Entpacken Sie die Datei in einem temporären Verzeichnis, und kopieren Sie die folgenden extrahierten Dateien in das Verzeichnis `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. Laden Sie die JAR-Datei (Java-Archiv) von AOP Alliance von der Webseite SpringSource herunter. Kopieren Sie die Datei "com.springsource.org.aopalliance-1.0.0.jar" in das Verzeichnis `equinox_root/plugins`.
5. Laden Sie JAR-Datei von Apache Commons Logging 1.1.1 von der Webseite SpringSource herunter. Kopieren Sie die Datei `com.springsource.org.apache.commons.logging-1.1.1.jar` in das Verzeichnis `equinox_root/plugins`.
6. Laden Sie den Befehlszeilenclient "Luminis OSGi Configuration Admin" herunter. Verwenden Sie dieses Bundle für die Verwaltung von OSGi-Verwaltungskonfigurationen. Sie können die JAR-Datei von der folgenden Webseite herunterladen: https://opensource.luminis.net/wiki/display/SITE/OSGi+Configuration+Admin+command+line+client. Kopieren Sie die Datei `net.luminis.cmc-0.2.5.jar` in das Verzeichnis `equinox_root/plugins`.
7. Laden Sie das Bundle für Apache Felix File Installation Version 3.0.2 von der folgenden Webseite herunter: http://felix.apache.org/site/index.html. Kopieren Sie die Datei `org.apache.felix.fileinstall-3.0.2.jar` in das Verzeichnis `equinox_root/plugins`.
8. Erstellen Sie ein Konfigurationsverzeichnis im Verzeichnis `equinox_root/plugins`, z. B.:

   ```
   mkdir equinox_root/plugins/configuration
   ```

9. Erstellen Sie die folgende Datei `config.ini` im Verzeichnis `equinox_root/plugins/configuration`, und ersetzen Sie *equinox_root* durch den absoluten Pfad zu Ihrem Verzeichnis, das Sie anstelle von `equinox_root` verwenden, und entfernen Sie alle abschließenden Leerzeichen hinter dem Backslash in der jeder Zeile. Sie müssen am Ende der Datei eine Leerzeile einfügen, z. B.:

   ```
   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
```

Wenn Sie die Umgebung bereits eingerichtet haben, können Sie das Equinox-Plug-in-Repository bereinigen, indem Sie das folgende Verzeichnis entfernen: `equinox_root\plugins\configuration\org.eclipse.osgi`.

10. Führen Sie die folgenden Befehle aus, um die Equinox-Konsole zu starten.

    Wenn Sie eine andere Version von Equinox verwenden, ist der Name Ihrer JAR-Datei anders als der Name, der im folgenden Beispiel verwendet wird:

    ```
    java -jar plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
    ```

eXtreme-Scale-Bundles installieren

WebSphere eXtreme Scale enthält Bundles, die in einem Eclipse-Equinox-OSGi-Framework installiert werden können. Diese Bundles sind erforderlich, um eXtreme-Scale-Server zu starten oder um eXtreme-Scale-Clients in OSGi zu verwenden.

Vorbereitende Schritte

In dieser Aufgabe wird davon ausgegangen, dass die folgenden Produkte installiert wurden:

- Eclipse-Equinox-OSGi-Framework
- Eigenständiger eXtreme-Scale-Client oder -Server

Informationen zu diesem Vorgang

eXtreme Scale enthält zwei Bundles. In einem OSGi-Framework ist nur eines der folgenden Bundles erforderlich:

objectgrid.jar
: Das Server-Bundle ist die Datei `objectgrid.jar`, die mit der eigenständigen eXtreme-Scale-Serverinstallation installiert wird und die für die Ausführung von eXtreme-Scale-Servern erforderlich ist. Das Server-Bundle kann auch für die Ausführung von eXtreme-Scale-Clients oder lokalen speicherinternen Caches verwendet werden. Die Bundle-ID für die Datei `objectgrid.jar` ist "com.ibm.websphere.xs.server_`<Version>`", wobei Version das folgende Format hat: `<Version>.<Release>.<Modifikation>`. Das Server-Bundle für eXtreme Scale Version 7.1.1 ist beispielsweise com.ibm-.websphere.xs.server_7.1.1.

ogclient.jar
: Das Bundle `ogclient.jar` wird mit eigenständigen Installationen und Clientinstallationen von eXtreme Scale installiert und wird verwendet, um eXtreme-Scale-Clients oder lokale speicherinterne Caches auszuführen. eXtreme Scale Die Bundle-ID für die Datei `ogclient.jar` ist "com.ibm.websphere.xs.client_`<Version>`", wobei die Version das folgende Format hat: `<Version>_<Release>_<Modifikation`. Das Client-Bundle für eXtreme Scale Version 7.1.1 ist beispielsweise com.ibm.websphere.xs.client_7.1.1.

Weitere Informationen zum Entwickeln von eXtreme-Scale-Plug-ins finden Sie im Artikel System-APIs und Plug-ins.

Vorgehensweise

Client- oder Server-Bundle von eXtreme Scale über die OSGi-Konsole im Eclipse-Equinox-OSGi-Framework installieren:

1. Starten Sie das Eclipse-Equinox-Framework mit aktivierter Konsole, z. B.:
   ```
   Java-Ausgangsverzeichnis/bin/java -jar <equinox_root>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
2. Installieren Sie das Client- oder Server-Bundle von eXtreme Scale in der Equinox-Konsole:
   ```
   osgi> install file:///<Pfad_zum_Bundle>
   ```
3. Equinox zeigt die Bundle-ID für das neu installierte Bundle an:
   ```
   Bundle id is 25
   ```
4. Starten Sie das Bundle in der Equinox-Konsole, wobei <ID> für die Bundle-ID steht, die dem Bundle bei der Installation zugeordnet wurde:
   ```
   osgi> start <ID>
   ```
5. Rufen Sie den Servicestatus in der Equinox-Konsole ab, um sicherzustellen, dass das Bundle gestartet wurde, z. B.:
   ```
   osgi> ss
   ```

   Wenn das Bundle erfolgreich gestartet wurde, wird der Status ACTIVE für das Bundle angezeigt, z. B.:
   ```
   25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
   ```

Client- oder Server-Bundle von eXtreme Scale mit der Datei config.ini im Eclipse-Equonix-OSGi-Framework installieren:

6. Kopieren Sie das eXtreme-Scale-Client- oder Server-Bundle (objectgrid.jar oder ogclient.jar) aus dem Verzeichnis <WXS-Installationsstammverzeichnis>/ObjectGrid/lib in das Eclipse-Equinox-Plug-in-Verzeichnis, z. B.: <equinox_root>/plugins
7. Bearbeiten Sie die Eclipse-Equinox-Konfigurationsdatei config.ini, und fügen Sie das Bundle der Eigenschaft "osgi.bundles" hinzu, z. B.:
   ```
   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
   ```

   Wichtig: Vergewissern Sie sich, dass dem letzten Bundlenamen eine leere Zeile folgt. Jedes Bundle wird durch ein Komma abgetrennt.
8. Starten Sie das Eclipse-Equinox-Framework mit aktivierter Konsole, z. B.:
   ```
   Java-Ausgangsverzeichnis/bin/java -jar <equinox_root>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
9. Rufen Sie den Servicestatus in der Equinox-Konsole ab, um sicherzustellen, dass das Bundle gestartet wurde:
   ```
   osgi> ss
   ```

   Wenn das Bundle erfolgreich gestartet wurde, wird der Status ACTIVE für das Bundle angezeigt, z. B.:
   ```
   25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
   ```

Ergebnisse

Sie haben das Server- oder Client-Bundle von eXtreme Scale im Eclipse-Equinox-OSGi-Framework installiert und gestartet.

Installation überprüfen

Nach Abschluss des Installationsassistenten können Sie die Installation überprüfen, indem Sie mehrere Aspekte der Installation prüfen.

Vorgehensweise

- **Für eine eigenständige Installation oder eine Installation, die mit WebSphere Application Server integriert ist:**

  Verwenden Sie eine der folgenden Methoden, um sicherzustellen, dass die Installation erfolgreich durchgeführt wurde:

  – Führen Sie den Befehl zur Anzeige der Versionsinformationen für WebSphere eXtreme Scale aus:

    ```
    WAS-Stammverzeichnis/lib/> java -jar wsobjectgrid.jar version
    ```

    Die Ausgabe des Befehls enthält die Versionsnummer des Produkts und eine Buildnummer.

  – Suchen Sie in der Eigenschaftendatei nach der richtigen Versionsnummer.

    - Signaturdateien: Die Signaturdateien sind im Verzeichnis *WAS-Stammverzeichnis*/properties/version enthalten. Wenn ein Fixpack installiert wurde, sind in diesem Verzeichnis weitere Dateien fxtg enthalten. Im Folgenden sind verschiedene Beispiele für die Namen von Signaturdateien aufgeführt:

      ```
      WebSphere_eXtreme_Scale.7.1.1..swtag
      WebSphere_eXtreme_Scale.7.1.0.2.fxtag
      WebSphere_eXtreme_Scale.7.1.0.3.fxtag
      ```

    - Produktdatei von WebSphere eXtreme Scale:

      Die Produktdatei befindet sich im Verzeichnis *WAS-Stammverzeichnis*/properties/version. Suchen Sie die Datei WXS.product. Im Folgenden sehen Sie ein Beispiel für den Inhalt dieser Datei:

      ```
      <?xml version="1.0" encoding="UTF-8"?>
        <!DOCTYPE product SYSTEM "product.dtd">
        <product name="IBM WebSphere eXtreme Scale">
             <id>WXS</id>
             <version>7.1.1.0</version>
             <build-info
         date="8/5/11"
         level="a1132.68720"/>
        </product>
      ```

  – Vergewissern Sie sich, dass die Laufzeitdateien installiert sind. Die Listen der Laufzeitdateien für jeden Installationstyp sind in den folgenden Artikeln dokumentiert:

    - „Laufzeitdateien für eine eigenständige Installation von WebSphere eXtreme Scale" auf Seite 200
    - „Laufzeitdateien für eine integrierte Installation von WebSphere eXtreme Scale in WebSphere Application Server" auf Seite 169

- **Für eine Installation, die mit WebSphere Application Server integriert ist, stehen Ihnen die folgenden zusätzlichen Methoden zur Verfügung, mit denen Sie prüfen können, ob die Installation erfolgreich durchgeführt wurde:**

  – Ausführung des Befehls zur Ausgabe der Versionsinformationen für WebSphere Application Server:

    ```
    WAS-Stammverzeichnis/bin/> versionInfo.sh|.bat
    ```

Die Ausgabe enthält eine Liste der installierten Produkte, einschließlich Installationsverzeichnissen, installierter Produkte, Versionen, Buildstufen, Builddaten usw.

Tipp: Fügen Sie den Parameter **-maintenancePackages** hinzu, um weitere Details anzuzeigen.

```
WAS-Stammverzeichnis/bin/> versionInfo.sh|.bat -maintenancePackages
```

– Überprüfung der Eingangsanzeige für die Administrationskonsole von WebSphere Application Server. Rufen Sie `http://localhost:9060/ibm/console` auf. Melden Sie sich an der Konsole an. Die Version von WebSphere eXtreme Scale wird in der Eingangsanzeige angezeigt.
– Verwendung der Konsole "Erste Schritte" für die Erweiterung der Installation von WebSphere Application Server mit WebSphere eXtreme Scale:

```
WAS-Stammverzeichnis/firststeps/WXS> firststeps.sh|.bat
```

Weitere Informationen finden Sie unter „Profile für WebSphere eXtreme Scale erstellen und erweitern” auf Seite 189.

Nächste Schritte

Wenn Sie feststellen, dass die Installation nicht wie erwartet durchgeführt wurde, müssen Sie nach Fehlern in der Installation suchen. Weitere Informationen hierzu finden Sie unter „Fehlerbehebung bei der Installation” auf Seite 214.

Erste Schritte nach der Installation

Nach Abschluss und Prüfung der Installation können Sie mit der Verwendung von WebSphere eXtreme Scale für die Erstellung Ihres Datengrids beginnen.

Vorgehensweise

1. Aktualisieren Sie Ihre Installation, indem Sie Wartungspakete anwenden.

   Weitere Informationen: „eXtreme-Scale-Server aktualisieren” auf Seite 217.
2. Wenn Sie WebSphere eXtreme Scale zum ersten Mal verwenden, können Sie die Einführungsinformationen verwenden, um mehr über die Verwendung des Produkts zu erfahren.

   Weitere Informationen: Kapitel 1, „Einführung”, auf Seite 1
3. Konfigurieren Sie das Produkt. Erstellen Sie Eigenschaften- und XML-Dateien, um die Konfiguration für Datengrids, Server und Clients zu definieren. Außerdem können Sie die Cache- bzw. Datenbankintegration, REST-Datenservices oder OSGI-Plug-ins konfigurieren.

   Weitere Informationen: Kapitel 6, „Konfigurieren”, auf Seite 229
4. Entwickeln Sie eine Anwendung, die auf das Datengrid zugreift.

   Weitere Informationen: Anwendungen entwickeln
5. Starten und verwalten Sie Container- und Katalogserver mit Ihren Konfigurationsdateien und Ihrer Datengridanwendung.

   Weitere Informationen: Kapitel 7, „Verwalten”, auf Seite 403
6. Überwachen Sie die Leistung Ihrer Konfiguration mit den verschiedenen Überwachungstools.

   Weitere Informationen: Kapitel 8, „Überwachung”, auf Seite 449

Fehlerbehebung bei der Installation

Verwenden Sie diese Informationen, um Probleme in Ihrer Installation zu beheben.

Vorgehensweise

- **Problem:** Wenn Sie den Installationsbefehl über einen fernen Computer ausführen, z. B. `\\mymachine\downloads\`, wird die folgende Nachricht angezeigt: `CMD.EXE was started with the above path as the current directory. UNC paths are not supported. Defaulting to Windows directory.` Deshalb wird die Installation nicht ordnungsgemäß durchgeführt.

  Lösung: Ordnen Sie den fernen Computer einem Netzlaufwerk zu. Unter Windows können Sie beispielsweise mit der rechten Maustaste auf **Arbeitsplatz** klicken, **Netzlaufwerk verbinden** auswählen und den UNC-Pfad (Uniform Naming Conventions) zum fernen Computer einfügen. Anschließend können Sie das Installationsscript erfolgreich über das Netzlaufwerk ausführen, z. B.: `y:\mymachine\downloads\WXS\install.bat`.

- **Problem:** Die Installation wird nicht erfolgreich durchgeführt.

  Lösung: Überprüfen Sie die Protokolldateien, um festzustellen, wo die Installation fehlgeschlagen ist. Wenn die Installation scheitert, befinden sich die Protokolle im Verzeichnis *WXS-Installationsstammverzeichnis*`/logs/wxs`.

- **Problem:** Während der Installation tritt ein katastrophaler Fehler auf.

  Lösung: Überprüfen Sie die Protokolldateien, um festzustellen, wo die Installation fehlgeschlagen ist. Wenn die Installation nach teilweiser Ausführung scheitert, befinden sich die Protokolle im Allgemeinen im Verzeichnis *Benutzerstammverzeichnis*`/wxs_install_logs/`.

- Windows **Problem:** Wenn Sie WebSphere eXtreme Scale Client unter Windows installieren, finden Sie in den Ergebnissen der Installation möglicherweise den folgenden Text:

  ```
  Erfolg: Die Installation des folgenden Produkts war erfolgreich:
  WebSphere eXtreme Scale Client. In einigen Konfigurationsschritten sind Fehler
  aufgetreten. Weitere Informationen finden Sie in der folgenden Protokolldatei:
  <WAS-Installationsstammverzeichnis>\logs\wxs_client\install\log.txt"
  Sehen Sie sich das Installationsprotokoll (log.txt) und das Erweiterungsprotokoll
  des Deployment Manager an.
  ```

  Lösung: Wenn Sie einen Fehler sehen, der sich auf die Datei `iscdeploy.sh` bezieht, können Sie diesen ignorieren. Dieser Fehler verursacht keine Probleme.

WebSphere eXtreme Scale deinstallieren

Wenn Sie WebSphere eXtreme Scale aus Ihrer Umgebung entfernen möchten, können Sie dazu den Assistenten verwenden, oder Sie können das Produkt im unbeaufsichtigten Modus deinstallieren.

Vorbereitende Schritte

Achtung: Das Deinstallationsprogramm entfernt alle Binärdateien und alle Wartungspakete, wie z. B. Fixpacks und vorläufige Fixes, gleichzeitig.

Vorgehensweise

1. Stoppen Sie alle Prozesse, in denen eXtreme Scale ausgeführt wird.

   Vorsicht:
   Stellen Sie sicher, dass alle aktiven Prozesse gestoppt sind. Wenn die aktiven Prozesse nicht gestoppt sind, wird die Deinstallation fortgesetzt, was zu unvorhersehbaren Ereignissen führen und die Deinstallation auf einigen Plattformen in einem unbestimmten Zustand hinterlassen kann.

   - Wenn Sie eXtreme Scale im eigenständigen Modus installiert haben, verwenden Sie zum Stoppen von Prozessen die Informationen unter "Eigenständige Server stoppen".
   - Wenn Sie eXtreme Scale mit einer vorhandenen Installation von WebSphere Application Server installiert haben, lesen Sie die Informationen zu den Befehlszeilendienstprogrammen, in denen beschrieben wird, wie Prozesse von WebSphere Application Server gestoppt werden.
   - Wenn Sie die Webkonsole ausführen, verwenden Sie den Befehl `stopConsoleServer`, um den Server der Webkonsole zu stoppen. Das Script `stopConsoleServer` befindet sich im Verzeichnis *WXS-Installationsstammverzeichnis*`/ObjectGrid/bin`. Wenn Sie diesen Server vor der Deinstallation nicht stoppen, wird der Prozess während des Deinstallationsprozesses automatisch gestoppt.

2. Deinstallieren Sie das Produkt. Sie können die Deinstallation in einer grafischen Benutzerschnittstelle oder im unbeaufsichtigten Modus ausführen.

   Anmerkung: Wenn Sie die Antwortdatei `wxssetup.response` für unbeaufsichtigte oder GUI-Installationen bzw. -Deinstallationen verwenden, müssen Sie immer den vollständig qualifizierten Pfad angeben. Die Antwortdatei ist bei der GUI-Deinstallation optional.

   - **Gehen Sie zum Ausführen der Deinstallation über die grafische Benutzerschnittstelle (GUI) wie folgt vor:**
     - Linux UNIX *<Ausgangsverzeichnis_der_Installation>*`/uninstall_wxs/uninstall`
     - Windows *<Ausgangsverzeichnis_der_Installation>*`\uninstall_wxs\uninstall.exe`

     Wenn Sie die Deinstallation über die grafische Benutzerschnittstelle und die Datei `wxssetup.response` ausführen möchten, verwenden Sie einen der folgenden Befehle:

     - Linux UNIX

       <Ausgangsverzeichnis_der_Installation>`/uninstall_wxs/uninstall -options` *<erforderlicher_vollständiger_Installationspfad>*`/wxssetup.response`

     - Windows

       <Ausgangsverzeichnis_der_Installation>`\uninstall_wxs\uninstall.exe -options` *<erforderlicher_vollständiger_Installationspfad>*`\wxssetup.response`

   - **Gehen Sie zum Ausführen der Deinstallation im unbeaufsichtigten Modus unter Verwendung des Antwortdateiscripts `wxssetup.response` wie folgt vor:**
     - Linux UNIX

       <Ausgangsverzeichnis_der_Installation>`/uninstall_wxs/uninstall -options` *<erforderlicher_vollständiger_Installationspfad>*`/wxssetup.response -silent`

     - Windows

```
<Ausgangsverzeichnis_der_Installation>\uninstall_wxs\uninstall.exe -options
<erforderlicher_vollständiger_Installationspfad>\wxssetup.response -silent
```

Ergebnisse

Sie haben eXtreme Scale aus Ihrer Umgebung entfernt.

Kapitel 5. Upgrade und Migration von WebSphere eXtreme Scale durchführen

Sie können eine Migration von früheren Versionen auf Version 7.1.1 durchführen, und Sie können Wartungspakete auf Version 7.1.1 anwenden. Zur Vermeidung von Ausfällen müssen Sie die Aktualisierungen für die Server in Ihrer Konfiguration in einer bestimmten Reihenfolge installieren.

- Informationen zum Durchführen eines Upgrades einer Installation der Version 7.1.0.x finden Sie in den Abschnitten „eXtreme-Scale-Server aktualisieren" und „Update Installer zum Installieren von Wartungspaketen verwenden" auf Seite 221.
- Informationen zum Durchführen eines Upgrades einer Installation der Version 7.0.x finden Sie in den Abschnitten „eXtreme-Scale-Server aktualisieren" und „Migration auf WebSphere eXtreme Scale Version 7.1.1" auf Seite 220.

eXtreme-Scale-Server aktualisieren

Sie können ein Upgrade von WebSphere eXtreme Scale auf eine neue Version durchführen, indem Sie Wartungspakete anwenden oder eine neue Version installieren, ohne den Service zu unterbrechen.

Vorbereitende Schritte

Sie müssen die Binärdatei für das Release der übergeordneten Version bzw. das Wartungspaket haben, das Sie anwenden möchten. Die aktuellsten Informationen zu den verfügbaren Releases und Wartungspaketen können Sie aus dem IBM Unterstützungsportal für WebSphere eXtreme Scale abrufen.

Informationen zu diesem Vorgang

Zum Durchführen eines Upgrades ohne Serviceunterbrechung führen Sie zuerst das Upgrade Ihrer Katalogserver durch. Anschließend führen Sie das Upgrade der Container-Server und der Clients durch.

Vorgehensweise

1. Führen Sie das Upgrade der Katalogserviceschicht durch, indem Sie die folgenden Schritte für jeden Katalogserver im Datengrid wiederholen. Führen Sie das Upgrade der Katalogserviceschicht vor dem Upgrade der Container-Server oder Clients durch. Einzelne Katalogserver können mit Versionskompatibilität interagieren, d. h., Sie können das Upgrade nacheinander für jeden Katalogserver durchführen, ohne den Service zu unterbrechen.
   a. Überprüfen Sie, ob der Quorumstatus stabil ist. Führen Sie den folgenden Befehl aus:

      ```
      xsadmin -quorumStatus
      xscmd -c showQuorumStatus
      ```

      Dieses Ergebnis zeigt an, dass alle Katalogserver verbunden sind.
   b. Wenn Sie Multimasterreplikation zwischen zwei Katalogservicedomänen verwenden, trennen Sie die Verbindung zwischen den beiden Katalogservicedomänen, während Sie das Upgrade der Katalogserver durchführen.

      ```
      xsadmin –ch host -p 1099 -dismissLink Domänenname
      ```

7.1.1+

```
xscmd –c dismissLink –cep host:2809 -fd Domänenname
```

Sie müssen diesen Befehl nur in einer der Katalogservicedomänen ausführen, um die Verbindung zwischen zwei Katalogservicedomänen zu trennen.

c. Beenden Sie einen der Katalogserver. Sie können den Befehl **stopOgserver** oder den Befehl **xscmd -c teardown** verwenden oder den Anwendungsserver beenden, in dem der Katalogservice in WebSphere Application Server ausgeführt wird. Es gibt keine Vorgaben bezüglich der Reihenfolge, in der die Katalogserver gestoppt werden, aber der Zeitaufwand reduziert sich, wenn der primäre Katalogserver zuletzt beendet wird. Suchen Sie die Nachricht CWOBJ8106 in den Protokolldateien, um den primären Katalogserver zu ermitteln. Unter normalen Bedingungen wird das Quorum aufrecht erhalten, wenn ein Katalogserver beendet wird, aber es empfiehlt sich, den Quorumstatus nach der Beendigung eines Servers mit dem Befehl **xscmd -c showQuorumStatus** abzufragen.

   Wenn Sie den Befehl **xscmd -c teardown** verwenden, können Sie die Servernamen filtern. Mit dem Befehl **stopOgServer** muss ein extakter Servername oder eine Liste mit Namen von Servern angegeben werden, die gleichzeitig gestoppt werden sollen. Sie sollten den Beendigungsprozess für Servergruppen ausführen, anstatt den Stopp- oder Beendigungsprozess für viele Server parallel aufzurufen. Wenn die zu beendenden Server gruppiert werden, ist das Datengrid in der Lage, auf die Server zu reagieren, die beendet werden, indem es Shards im Datengrid verschiebt. Sie können einen der folgenden Befehle verwenden, um Ihre Server zu beenden:

   Sie können eine Liste der zu stoppenden Server mit dem Befehl **stopOgServer** oder **xscmd -c teardown** angeben.

   ```
   stopOgServer <Servername>[,<Servername>]
   ```

   ```
   xsadmin –teardown <Servername>[,<Servername>]
   ```

   7.1.1+

   ```
   xscmd –c teardown -sl <Servername>[,<Servername>]
   ```

   In den vorherigen Beispielen führen die Befehle **stopOgServer** und **xscmd -c teardown** dieselben Beendigungsaufgaben aus. Mit dem Befehl **xscmd -c teardown** können Sie die zu stoppenden Server jedoch filtern. Weitere Informationen zum Filtern der Server nach Zonen- oder Hostnamen finden Sie unter „Server mit dem Dienstprogramm **xscmd** normal stoppen" auf Seite 417. Der Befehl "teardown" filtert die übereinstimmenden Server heraus und fragt nach, ob die ausgewählten Server korrekt sind.

d. Aktualisierungen im Katalogserver installieren. Sie können den Katalogserver auf ein neues Hauptrelease des Produkts migrieren oder ein Wartungspaket anwenden. Weitere Informationen finden Sie in den folgenden Artikeln:

   - Migration von einer Installation der Version 7.0.x: „Migration auf WebSphere eXtreme Scale Version 7.1.1" auf Seite 220
   - Upgrade von einer Installation der Version 7.1.0.x: „Update Installer zum Installieren von Wartungspaketen verwenden" auf Seite 221

e. Starten Sie den Katalogserver erneut.

   Wenn Sie eine eigenständige Umgebung verwenden, lesen Sie den Abschnitt „Eigenständigen Katalogservice starten" auf Seite 403, um weitere Informationen zu erhalten. Wenn Sie eine Umgebung von WebSphere Application

Server verwenden, lesen Sie den Abschnitt „Server in einer Umgebung von WebSphere Application Server starten und stoppen" auf Seite 417, um weiter Informationen zu erhalten.

Der Katalogserver wird solange im Kompatibilitätsmodus ausgeführt, bis alle Katalogserver denselben Versionsstand haben. Der Kompatibilitätsmodus wird meistens für Migrationen von Hauptreleases verwendet, weil neue Funktionen in den nicht migrierten Servern nicht verfügbar sind. Wie lange Katalogserver im Kompatibilitätsmodus ausgeführt werden können, ist zwar nicht beschränkt, aber es empfiehlt sich, alle Katalogserver so schnell wie möglich auf denselben Versionsstand zu migrieren.

f. Wenden Sie Aktualisierungen auf die verbleibenden Katalogserver in Ihrer Konfiguration an.

2. Upgrade der Container-Server durchführen, indem die folgenden Schritte für jeden Container-Server im Datengrid ausgeführt werden. Sie können die Container-Server in beliebiger Reihenfolge aktualisieren. Sie sollten jedoch zuerst die Server und dann erst die Clients aktualisieren, wenn Sie im Upgrade neue Funktionen verwenden.

   a. Stoppen Sie die Container-Server, die Sie aktualisieren möchten. Sie können die Container-Server-Schicht gruppenweise mit dem Befehl **stopOgserver** oder mit dem Befehl **teardown** stoppen. Durch die Ausführung von Beendigungsoperationen (teardown) in Stapeln und die parallele Ausführung von Serverstartoperationen kann der Verteilungsmechanismus Shards in größeren Gruppen verschieben.

   ```
   xsadmin –teardown -fz DefaultZone
   ```

   7.1.1+

   ```
   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. Installieren Sie die Aktualisierungen in den Container-Servern. Sie können die Container-Server auf ein neues Hauptrelease des Produkts migrieren oder ein Wartungspaket anwenden. Weitere Informationen finden Sie in den folgenden Artikeln:

      - Migration von einer Installation der Version 7.0.x: „Migration auf WebSphere eXtreme Scale Version 7.1.1" auf Seite 220
      - Upgrade von einer Installation der Version 7.1.0.x: „Update Installer zum Installieren von Wartungspaketen verwenden" auf Seite 221

   c. Starten Sie Ihre Container-Server erneut.

   d. Führen Sie das Upgrade für alle verbleibenden Container-Server in Ihrer Konfiguration durch.

3. Bei Verwendung der Multimasterreplikation Verbindung zu den Katalogservicedomänen wiederherstellen. Verwenden Sie den Befehl **xscmd -c establishLink**, um die Verbindung zu den Katalogservicedomänen wiederherzustellen. 7.1.1+

```
xsadmin –ch host –p 1099 –establishLink dname fdHostA:2809,fdHostB:2809
xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
```

Nächste Schritte

Sie können diese Schritte auch verwenden, um die Installation auf eine ältere Version zurückzusetzen oder um Wartungspakete zu deinstallieren. Wenn Sie bei Verwendung der Multimasterreplikation die Installation jedoch auf Version 7.1.0 zurücksetzen, funktioniert die bidirektionale Replikation möglicherweise nicht ordnungsgemäß, wenn Sie die Verbindungen wiederherstellen. In einer solchen Situation starten Sie beide Katalogservicedomänen erneut und stellen die Verbindung zu den Katalogservicedomänen mit dem Befehl **establishLink** wieder her.

Migration auf WebSphere eXtreme Scale Version 7.1.1

Mit dem Installationsprogramm von WebSphere eXtreme Scale ist ein Upgrade oder eine Änderung einer vorherigen Installation nicht möglich. Sie müssen die vorherige Version deinstallieren, bevor Sie die neue Version installieren. Die Konfigurationsdateien müssen nicht migriert werden, weil sie abwärtskompatibel sind. Wenn Sie jedoch die mit dem Produkt gelieferten Scriptdateien geändert haben, müssen Sie diese Änderungen erneut auf die aktualisierten Scriptdateien anwenden.

Vorbereitende Schritte

Vergewissern Sie sich, dass Ihre Systeme die Mindestvoraussetzungen für die Produktversionen erfüllen, die Sie migrieren und installieren möchten. Weitere Informationen finden Sie unter „Hardware- und Softwarevoraussetzungen” auf Seite 50.

Informationen zu diesem Vorgang

Führen Sie alle geänderten Produktscriptdateien mit den neuen Produktscriptdateien im Verzeichnis `/bin` zusammen, um Ihre Änderungen beizubehalten.

Tipp: Wenn Sie die mit dem Produkt installierten Scriptdateien nicht geändert haben, müssen Sie die folgenden Migrationsschritte nicht ausführen. Stattdessen können Sie das Upgrade auf Version 7.1.1 durchführen, indem Sie die vorherige Version deinstallieren und die neue Version in demselben Verzeichnis installieren.

Vorgehensweise

1. Stoppen Sie alle Prozesse, die eXtreme Scale verwenden.
   - Lesen Sie den Abschnitt Eigenständige Server stoppen, um alle Prozesse zu stoppen, die in der eigenständigen Umgebung von eXtreme Scale ausgeführt werden.
   - Informationen zum Stoppen aller Prozesse, die in Ihrer Umgebung von WebSphere Application Server oder WebSphere Application Server Network Deployment ausgeführt werden, finden Sie unter "Befehlszeilendienstprogramme".
2. Speichern Sie alle geänderten Scripts aus Ihrem aktuellen Installationsverzeichnis in einem temporären Verzeichnis.
3. Deinstallieren Sie das Produkt.
4. Installieren Sie eXtreme Scale Version 7.1.1. Weitere Informationen finden Sie in „WebSphere eXtreme Scale mit dem Installationsassistenten installieren” auf Seite 167.

5. Fügen Sie Ihre Änderungen aus den Dateien im temporären Verzeichnis in die neuen Produktscriptdateien im Verzeichnis `/bin` ein.
6. Starten Sie alle Prozesse von eXtreme Scale, um mit der Verwendung des Produkts zu beginnen. Weitere Informationen finden Sie im Abschnitt Kapitel 7, „Verwalten", auf Seite 403.

Update Installer zum Installieren von Wartungspaketen verwenden

Verwenden Sie IBM Update Installer, um Ihre Umgebung von WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit verschiedenen Wartungspaketen wie vorläufigen Fixes, Fixpacks und Refresh-Packs zu aktualisieren.

Informationen zu diesem Vorgang

Verwenden Sie IBM Update Installer, um verschiedene Typen von Wartungspaketen für WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client zu installieren und anzuwenden. Da Update Installer in regelmäßigen Abständen aktualisiert wird, müssen Sie die aktuellste Version des Tools verwenden.

Vorgehensweise

1. Stoppen Sie alle Prozesse, die in Ihrer Umgebung aktiv sind.
   - Informationen zum Stoppen aller Prozesse, die in der eigenständigen Umgebung von eXtreme Scale ausgeführt werden, finden Sie unter „Eigenständige Server stoppen" auf Seite 414.
   - Informationen zum Stoppen aller Prozesse, die in der Umgebung von WebSphere Application Server ausgeführt werden, finden Sie unter Befehlszeilendienstprogramme.
2. Laden Sie die aktuelle Version von Update Installer herunter. Weitere Informationen finden Sie auf der Webseite "Recommended fixes".
3. Installieren Sie Update Installer. Weitere Informationen finden Sie unter Update Installer for WebSphere Software im Information Center von WebSphere Application Server.
4. Laden Sie die Wartungspakete, die Sie installieren möchten, in das Verzeichnis *UPDI-Stammverzeichnis*`/maintenance` herunter. Weitere Informationen finden Sie auf der Unterstützungssite.
5. Verwenden Sie Update Installer, um den vorläufigen Fix, das Fixpack oder das Refresh-Pack zu installieren. Sie können das Wartungspaket installieren, indem Sie die grafische Benutzerschnittstelle verwenden oder Update Installer im unbeaufsichtigten Modus ausführen.

   Führen Sie den folgenden Befehl im Verzeichnis *UPDI-Stammverzeichnis* aus, um die grafische Benutzerschnittstelle zu starten:

   - Linux UNIX `update.sh`
   - Windows `update.bat`

   Führen Sie den folgenden Befehl im Verzeichnis *UPDI-Stammverzeichnis* aus, um Update Installer im unbeaufsichtigten Modus auszuführen:

   - Linux UNIX `./update.sh -silent -options` *responsefile/Dateiname*
   - Windows `update.bat -silent -options` *responsefile\Dateiname*

   Wenn der Installationsprozess fehlschlägt, sehen Sie sich die temporäre Protokolldatei im Verzeichnis *UDPI-Stammverzeichnis*`/logs/update/tmp` an. Update Installer erstellt das Verzeichnis *Installationsstammverzeichnis*`/logs/update/`*Wartungspaket*`.install`, in dem sich die Installationsprotokolldateien befinden.

Tool xsadmin auf das Tool xscmd migrieren

In den früheren Releases war das Tool **xsadmin** ein Beispielbefehlszeilendienstprogramm für die Überwachung des Umgebungszustands. Das Tool **xscmd** wurde als offiziell unterstütztes Verwaltungs- und Überwachungsbefehlszeilentool eingeführt. Wenn Sie früher das Tool **xsadmin** verwendet haben, können Sie Ihre Befehle auf das neue Tool **xscmd** migrieren.

xsadmin und xscmd-Befehlsentsprechungen

Tabelle 9. Argumente für das Dienstprogramm **xsadmin** *und entsprechende* **xscmd**-*Befehle*. Einige **xscmd**-Befehle haben eine Kurz- und eine Langform. Die Befehle in Kurzform haben einen Strich (-) und die Befehle in Langform zwei Striche (--). Die beiden Formen sind untereinander auswechselbar.

Befehlszeilenargument von xsadmin	Entsprechender xscmd-Befehl	Parameter des Befehls xscmd
-bp	• **-cep** *Hostname*:*Listener-Port* • **--catalogEndpoint** *Hostname*:*Listener-Port*	Nicht zutreffend
-ch	• **-cep** *Hostname*:*Listener-Port* • **--catalogEndpoint** *Hostname*:*Listener-Port*	Nicht zutreffend
-clear	**-c clearGrid**	**-g, -ms, -v, -m, (-cep)**
-containers	• **-c listCoreGroups** • **-c listCoreGroupMembers -cg** *Stammgruppe*	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-continuous	• **-cnt** • **--continuous**	Nicht zutreffend
-coregroups	• **-c listCoreGroups** • **-c listCoreGroupMembers -cg** *Stammgruppe*	Nicht zutreffend
-dismissLink <*Katalogservicedomäne*>	**-c dismissLink**	• **-fd** <*fremde_Katalogservicedomäne*> • **--foreignCatalogServiceDomain** <*fremde_Katalogservicedomäne*>
-dmgr	Nicht zutreffend. Dieses Argument wird automatisch mit **xscmd** bestimmt.	Nicht zutreffend
-empties	Spezielles Argument für einen neuen Befehl	Nicht zutreffend
-establishLink <*fremde_Domäne*> <*Host1:Port1,Host2:Port2...*>	**-c establishLink**	• **-fd** <*fremde_Katalogservicedomäne*> **-fe** <*Host1:Port1,Host2:Port2...*> • **--foreignCatalogServiceDomain** <*fremde_Katalogservicedomäne*> **-foreignEndPoints** <*Host1:Port1,Host2:Port2...*>
-fc	• **-ct** • **--container**	Nicht zutreffend
-fh	• **-hf** • **--hostFilter**	Nicht zutreffend

Tabelle 9. Argumente für das Dienstprogramm **xsadmin** *und entsprechende* **xscmd**-*Befehle (Forts.).* Einige **xscmd**-Befehle haben eine Kurz- und eine Langform. Die Befehle in Kurzform haben einen Strich (-) und die Befehle in Langform zwei Striche (--). Die beiden Formen sind untereinander auswechselbar.

Befehlszeilenargument von **xsadmin**	Entsprechender **xscmd**-Befehl	Parameter des Befehls **xscmd**
-fm	• **-m** • **--map**	Nicht zutreffend
-fnp	• **-snp** • **--serversWithNoPrimaries**	Nicht zutreffend
-fp	• **-p** • **--partitionId**	Nicht zutreffend
-fs	• **-s** • **--server**	Nicht zutreffend
-fst	• **-st** <*Shard-Typ*> • **--shardType** <*Shard-Typ*> Shard-Werte: P=primäres Shard, A=asynchrones Replikat S=synchrones Replikat	Nicht zutreffend
-fz	• **-z** • **--zone**	Nicht zutreffend
-force	Spezielles Argument für einen neuen Befehl	
-g	• **-g** • **--objectGrid**	Nicht zutreffend
-getstatsspec	**-c getStatsSpec**	Nicht zutreffend
-getTraceSpec	**-c getTraceSpec**	Nicht zutreffend
-h	Sie können die Hilfe mit oder ohne bestimmten Befehlsnamen ausführen. • **-h** • **--help** • **-h** <*Befehlsname*> • **--help** <*Befehlsname*>	Nicht zutreffend
-hosts	**-c listHosts**	**-g, -ms, -st, -c, -s, -hf, -z**
-jmxUrl	• **-cep** *Hostname*:*Listener-Port* • **--catalogEndpoint** *Hostname*:*Listener-Port*	Nicht zutreffend
-l	**-c listObjectGridNames**	Nicht zutreffend
-m	• **-ms** • **--mapSet**	Nicht zutreffend
-mapsizes	**-c showMapSizes**	**-g, -ms, -cnt, -i, [-ct, -z, -s, -hf, sht [P,A,S], -p]**
-mbeanservers	**-c listAllJMXAddresses**	Nicht zutreffend
-overridequorum	**-c overrideQuorum**	Nicht zutreffend

Tabelle 9. Argumente für das Dienstprogramm **xsadmin** *und entsprechende* **xscmd**-*Befehle (Forts.).* Einige **xscmd**-Befehle haben eine Kurz- und eine Langform. Die Befehle in Kurzform haben einen Strich (-) und die Befehle in Langform zwei Striche (--). Die beiden Formen sind untereinander auswechselbar.

Befehlszeilenargument von xsadmin	Entsprechender xscmd-Befehl	Parameter des Befehls xscmd
-password	• **-pwd** • **--password**	Nicht zutreffend
-p	• **-cep** *Hostname*:*Listener-Port* • **--catalogEndpoint** *Hostname*:*Listener-Port*	Nicht zutreffend
-placementStatus	**-c placementServiceStatus**	**-g, -ms**
-primaries	**-c showPlacement -sf P**	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-profile	Aktuelle Sicherheitseinstellungen als Sicherheitsprofil speichern: • **-ssp** *Profilname* • **--saveSecProfile** *Profilname* Bestimmtes Sicherheitsprofil verwenden: • **-sp** *Profilname* • **--securityProfile** *Profilname*	
-quorumstatus	**-c showQuorumStatus**	Nicht zutreffend
-releaseShard *<Name_des_Container-Servers> <Objectgrid-Name> <MapSet-Name> <Partitionsname>*	**-c releaseShard**	**-c, -g, -ms, -p**
-reserved	• **-sf** *[R,U]* • **--shardFilter** *[R,U]* R=reserved, U=unassigned	Nicht zutreffend
-reserveShard *<Name_des_Container-Servers> <Objectgrid-Name> <MapSet-Name> <Partitionsname>*	**-c reserveShard**	**-c, -g, -ms, -p**
-resumeBalancing *<Objectgrid-Name> <MapSet-Name>*	**-c resumeBalancing**	**-g, -ms**
-revisions	**-c revisions**	**-s,-p,-g,-m**
-routetable	**-c routetable**	**-z, -hf,-p,-g,-ms**
-settracespec *<Tracezeichenfolge>*	**-c setTraceSpec**	**-spec** *<Tracezeichenfolge>*
-swapShardWithPrimary *<Name_des_Container-Servers> <Objectgrid-Name> <MapSet-Name> <Partitionsname>*	**-c swapShardWithPrimary**	**-c -g, -ms, -p**
-setstatsspec *<Statistikspezifikation>*	**-c setStatsSpec**	**-spec** *<Statistikspezifikation>*
-suspendBalancing *<Objectgrid-Name> <MapSet-Name>*	**-c suspendBalancing**	**-g, -ms**
-ssl	• **-ssl** • **--enableSSL**	Nicht zutreffend

Tabelle 9. Argumente für das Dienstprogramm **xsadmin** *und entsprechende* **xscmd**-*Befehle (Forts.).* Einige **xscmd**-Befehle haben eine Kurz- und eine Langform. Die Befehle in Kurzform haben einen Strich (-) und die Befehle in Langform zwei Striche (--). Die beiden Formen sind untereinander auswechselbar.

Befehlszeilenargument von xsadmin	Entsprechender xscmd-Befehl	Parameter des Befehls xscmd
`-teardown`	`-c teardown`	`-f,, -st, -snp, -c, -s, -p, -hf, -z, -g, -ms, -m`
`-triggerPlacement`	`-c triggerPlacement`	`-g, -ms`
`-trustPass`	• `-tsp` • `--trustStorePassword`	Nicht zutreffend
`-trustPath`	• `-ts` • `--trustStore`	Nicht zutreffend
`-trustType`	• `-tst` • `--trustStoreType`	Nicht zutreffend
`-unassigned`	`-c showPlacement -sf U`	`-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms`
`-username`	• `-user` • `--username`	Nicht zutreffend
`-v`	• `-v` • `--verbose`	Nicht zutreffend
`-xml`	`-c showPlacement`	Nicht zutreffend

Veraltete Eigenschaften und APIs

Die folgende Liste von Eigenschaften und APIs sind ab dem Release Version 7.1.1 veraltet. Verwenden Sie die empfohlene Migrationsaktion, um zu bestimmen, wie Sie Ihre Konfiguration aktualisieren.

7.1.1+ Veraltete Komponenten in Version 7.1.1

Tabelle 10. Veraltete Eigenschaften und APIs

Veraltet	Empfohlene Migrationsaktion
Klasse "com.ibm.websphere.objectgrid.plugins.builtins.TranPropListener" Diese Klasse wurde verwendet, um basierend auf dem ObjectGrid-Namen erfolgreiche ObjectGrid-Transaktionsfestschreibungsprozesse an andere WebSphere-Anwendungsserver weiterzuleiten, in denen dieselbe ObjectGrid-Instanz ausgeführt wird.	7.1.1+ Die Schnittstelle TranPropListener wurde durch die Schnittstelle JMSObjectGridEventListener ersetzt, die eine JMS-basierte Implementierung der Schnittstelle ObjectGridEventListener ist. Sie unterstützt die clientseitige Invalidierung von nahen Caches und die Peer-to-Peer-Replikation.
Klasse "com.ibm.websphere.objectgrid.plugins.OptimisticCallback" Diese Klasse wurde für die Unterstützung optimistischer Vergleichsoperationen für die Werte einer Map verwendet.	7.1.1+ Das Plug-in "OptimisticCallback" wurde druch die Schnittstelle "ValueDataSerializer.Versionable" ersetzt, die Sie implementieren können, wenn Sie das DataSerializer-Plug-in mit dem Kopiermodus "COPY_TO_BYTES" oder die Annotation "@Version" mit der API "EntityManager" verwenden. Weitere Informationen finden Sie in der API-Dokumetation.

Tabelle 10. Veraltete Eigenschaften und APIs (Forts.)

Veraltet	Empfohlene Migrationsaktion
Plug-in "com.ibm.websphere.objectgrid.plugins.NoVersioningOptimisticCallback" Dieses Plug-in wurde für optimistisches Sperren ohne Versionsprüfung verwendet. Mit diesem integrierten OptimisticCallback-Handler hat der Loader die Versionsprüfung durchgeführt, aber optimistisches Sperren wurde verwendet, um sicherzustellen, dass bei einer Leseoperation immer festgeschrieben Daten zurückgegeben werden.	7.1.1+ Die Schnittstelle NoVersioningOptimisticCallback erweitert die Schnittstelle OptimisticCallback. Verwenden Sie deshalb die pessimistische Sperrstrategie mit der Standardtransaktionsisolation READ_COMMITTED oder einer niedrigeren Isolationsstufe. Weitere Informationen finden Sie unter Leistung von Sperren optimieren.
Klasse "com.ibm.websphere.objectgrid.plugins.ObjectTransformer" Dieses Plug-in wurde verwendet, um Objekte zu serialisieren, zu deserialisieren und in den Cache zu kopieren.	7.1.1+ Die Schnittstelle ObjectTransformer wurde durch die DataSerializer-Plug-ins ersetzt, die Sie verwenden können, um beliebige Daten effizient in WebSphere eXtreme Scale speichern können, so dass vorhandene Produkt-APIs effizient mit Ihren Daten interagieren können.
Klasse "com.ibm.websphere.objectgrid.BackingMap.setMapEventListeners" Diese Methode wurde verwendet, um die liste der MapEventListener-Objekte festzulegen.	7.1.1+ Verwenden Sie die Methode addMapEventListener(EventListener) oder die Methode removeMapEventListener(EventListener), um Ereignislistener in einer BackingMap hinzuzufügen oder zu entfernen.
Methode "com.ibm.websphere.objectgrid.ObjectGrid.setEventListeners" Diese Methode wurde verwendet, um die aktuelle Liste der ObjectGridEventListener-Objekte zu überschreiben und durch die bereitgestellte Liste der ObjectGridEventListeners-Objekte zu ersetzen.	7.1.1+ Verwenden Sie die Methode addEventListener(EventListener) oder die Methode removeEventListener(EventListener), um Ereignislistener oder Lebenszykluslistener im Datengrid hinzuzufügen oder zu entfernen.

7.1.1+ Stabiliserte Features in Version 7.1.1

Wenn ein Feature als stabilisiert aufgelistet wird, plant IBM momentan nicht, dieses Featuer in einem der nachfolgenden Releases dieses Produkts als veraltet zu deklarieren oder zu entfernen. Künftige Investitionen konzentrieren sich jedoch auf die alternative Funktion. Benutzer müssen vorhandene Anwendungen und Scripts, die eine stabiliserte Funktion verwenden, nicht ändern, sollten aber die Verwendung der strategischen Alternative für neue Anwendungen in Erwägung ziehen.

Tabelle 11. Veraltete Eigenschaften und APIs

Stabilisiertes Feature	Empfohlene Migrationsaktion
xsadmin Das Dienstprogramm "xsadmin" wird als Beispiel bereitgestellt, das veranschaulicht, wie Sie angepasste Dienstprogramme für Ihre Implementierung erstellen können.	7.1.1+ Mit dem Dienstprogramm **xscmd** können Sie Verwaltungsaufgaben wie die folgenden in der Umgebung ausführen: Multimasterreplikationslinks konfigurieren, Quorum überschreiben und Gruppen von Servern mit dem Befehl "teardown" stoppen.

Veraltete Komponenten in Version 7.1

Tabelle 12. Veraltete Eigenschaften und APIs

Veraltet	Empfohlene Migrationsaktion
Zellen- und Servereigenschaft "catalog.services.cluster": Diese angepasste Eigenschaft wurde verwendet, um eine Gruppe von Katalogservern in der Konfiguration von WebSphere Application Server zu definieren.	Diese angepasste Eigenschaft ist ab dem Release Version 7.1 veraltet. Erstellen Sie über die Administrationskonsole von WebSphere Application Server eine Katalogservicedomäne. Damit wird dieselbe Konfiguration wie mit der angepassten Eigenschaft erstellt. Weitere Informationen finden Sie unter .
MBean und Schnittstelle "CoreGroupServicesMBean"	Diese MBean ist ab dem Release Version 7.1 veraltet. Verwenden Sie stattdessen CatalogServiceManagementMBean.

Tabelle 12. Veraltete Eigenschaften und APIs (Forts.)

Veraltet	Empfohlene Migrationsaktion
MBean-Operation "ServerMBean.updateTraceSpec()"	Diese Operation ist ab dem Release Version 7.1 veraltet. Verwenden Sie stattdessen das Attribut "TraceSpec" in DynamicServerMBean.
CoreGroupServicesMBean MBean	Diese MBean ist ab dem Release Version 7.1 veraltet. Verwenden Sie stattdessen die MBean "CatalogServiceManagementMbean".
Ausnahme "ServiceUnavailableException"	Diese Ausnahme ist ab dem Release Version 7.1 veraltet. Verwenden Sie stattdessen die Ausnahme "TargetNotAvailableException".
Partitionierungsfeature (WPF): Das Partitionierungsfeature ist eine Gruppe von Programmierungs-APIs, die Java-EE-Anwendungen die Unterstützung symmetrischer Cluster ermöglicht.	Die Funktionalität von WPF kann alternativ in WebSphere eXtreme Scale realisiert werden.
StreamQuery: Eine fortlaufende Abfrage unvollständiger Daten, die in ObjectGrid-Maps gespeichert sind.	Ohne
Konfiguration statischer Grids: Eine statische, clusterbasierte Topologie, die die XML-Datei für die Clusterimplementierung verwendet.	Ersetzt durch die verbesserte dynamische Implementierungstopologie für die Verwaltung großer Datengrids.
Veraltete Systemeigenschaften: Systemeigenschaften für die Angabe der Server- und Clienteigenschaftendateien sind veraltet.	Sie können diese Argumente zwar noch verwenden, sollten Ihre Systemeigenschaften aber auf die neuen Werte umstellen. **`-Dcom.ibm.websphere.objectgrid.CatalogServerProperties`** Die Eigenschaft ist ab WebSphere eXtreme Scale Version 7.0 veraltet. Verwenden Sie stattdessen die Eigenschaft **`-Dobjectgrid.server.props`**. **`-Dcom.ibm.websphere.objectgrid.ClientProperties`** Die Eigenschaft ist ab WebSphere eXtreme Scale Version 7.0 veraltet. Verwenden Sie stattdessen die Eigenschaft **`-Dobjectgrid.client.props`**. **`-Dobjectgrid.security.server.prop`** Die Eigenschaft ist ab WebSphere eXtreme Scale Version 6.1.0.3 veraltet. Verwenden Sie stattdessen die Eigenschaft **`-Dobjectgrid.server.prop`**. **`-serverSecurityFile`** Dieses Argument ist ab WebSphere eXtreme Scale Version 6.1.0.3 veraltet. Diese Option wird an das Script `startOgServer` übergeben. Verwenden Sie stattdessen das Argument **`-serverProps`**.

Kapitel 6. Konfigurieren

Sie können WebSphere eXtreme Scale für die Ausführung in einer eigenständigen Umgebung konfigurieren, oder Sie können eXtreme Scale für die Ausführung mit WebSphere Application Server oder WebSphere Application Server Network Deployment konfigurieren. Damit eine eXtreme-Scale-Implementierung Konfigurationsänderungen auf der Serverseite des Datengrids berücksichtigt, müssen Sie Prozesse erneut starten, damit diese Änderungen wirksam werden. Die Änderungen werden nicht dynamisch angewendet. Obwohl Sie auf der Clientseite die Konfigurationseinstellungen für eine vorhandene Clientinstanz nicht ändern können, können Sie jedoch unter Verwendung einer XML-Datei oder über das Programm einen neuen Client mit den erforderlichen Einstellungen erstellen. Wenn Sie einen Client erstellen, können Sie die Standardeinstellungen überschreiben, die aus der aktuellen Serverkonfiguration stammen.

Konfigurationsmethoden

Sie können die meisten Aspekte des Produkts mit XML- und Eigenschaftendateien konfigurieren. Sie können auch Programmiermethoden, einschließlich Anwendungs- und Systemprogrammierschnittstellen, Plug-ins und Managed Beans, verwenden.

Informationen zu diesem Vorgang

Verwenden Sie die folgenden Dateien, um eine Basiskonfiguration zu erstellen:

Servereigenschaftendatei
Verwenden Sie die Servereigenschaftendadtei, um Einstellungen für Katalog- und Container-Server zu definieren, wie z. B. Trace, Protokollierung, Sicherheit, Ports usw. Sie können eine Servereigenschaftendatei an das Script **startOgServer** übergeben, die Datei in Ihrem Klassenpfad speichern oder die Datei mit Systemeigenschaften definieren.

Clienteigenschaftendatei
Verwenden Sie die Clienteigenschaftendatei, um Eigenschaften in Ihren Clients festzulegen, einschließlich Ports und Sicherheitseinstellungen. Sie können die zu verwendende Clienteigenschaftendatei mit einer Systemeigenschaft angeben, indem Sie die Datei im Klassenpfad speichern oder die Methode ClientClusterContext.getClientProperties verwenden.

ObjectGrid-XML-Deskriptordatei
Die ObjectGrid-XML-Deskriptordatei beschreibt das Datengrid und die Map-Konfiguration. Geben Sie die zu verwendende Datei mit dem Script **startOgServer** für eigenständige Konfigurationen an, oder fügen Sie für Konfigurationen mit WebSphere Application Server die Datei dem Anwendungsmodul hinzu.

XML-Deskriptordatei für Implementierungsrichtlinie
Die XML-Implementierungsrichtliniendatei steuert die Verteilung von Shards und Daten auf die verschiedenden Container-Server in der Konfiguration. Geben Sie die zu verwendende Datei mit dem Script **startOgServer** für eigenständige Konfigurationen an, oder fügen Sie für Konfigurationen mit WebSphere Application Server die Datei dem Anwendungsmodul hinzu.

Datengrids konfigurieren

Verwenden Sie eine ObjectGrid-XML-Deskriptordatei, um Datengrids, BackingMaps, Plug-ins usw. zu konfigurieren. Verwenden Sie zum Konfigurieren von WebSphere eXtreme Scale eine ObjectGrid-XML-Deskriptordatei und die API "ObjectGrid". Für eine verteilte Topologie benötigen Sie eine ObjectGrid-XML-Deskriptordatei und eine XML-Datei für die Implementierungsrichtlinie.

Lokale Implementierungen konfigurieren

Eine lokale, speicherinterne Konfiguration von eXtreme Scale kann über eine ObjectGrid-XML-Deskriptordatei oder über APIs erstellt werden.

Informationen zu diesem Vorgang

Zum Erstellen einer lokalen Implementierung erstellen Sie eine ObjectGrid-XML-Deskriptordatei und übergeben die Datei dann an die createObjectGrid-Methoden in der Schnittstelle ObjectGridManager.

Alternativ können Sie die gesamte Implementierung auch über das Programm mit der Schnittstelle ObjectGridManager erstellen.

Vorgehensweise

1. ObjectGrid-XML-Deskriptordatei erstellen.

   Die folgende Datei `companyGrid.xml` ist ein Beispiel für eine ObjectGrid-Deskriptor-XML. Die ersten Zeilen der Datei enthalten den erforderlichen Header für jede ObjectGrid-XML-Datei. Die Datei definiert eine ObjectGrid-Instanz mit dem Namen "CompanyGrid" und mehrere BackingMaps mit den Namen "Customer," "Item," "OrderLine" und "Order."

   Datei "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. Übergeben Sie die XML-Datei an eine der createObjectGrid-Methoden in der Schnittstelle "ObjectGridManager.

   Das folgende Codemuster validiert die Datei `companyGrid.xml` anhand des XML-Schemas und erstellt die ObjectGrid-Instanz mit dem Namen "CompanyGrid." Die neu erstellte ObjectGrid-Instanz wird nicht zwischengespeichert.

   ```
   ObjectGridManager objectGridManager = ObjectGridManagerFactory.getObjectGridManager();
   ObjectGrid companyGrid = objectGridManager.createObjectGrid("CompanyGrid",
    new URL("file:etc/test/companyGrid.xml"), true, false);
   ```

Nächste Schritte

Weitere Informationen zum Definieren aller Maps über das Programm mit den createObjectGrid-Methoden in der Schnittstelle ObjectGridManager finden Sie un-

ter ObjectGrid-Instanzen mit der Schnittstelle ObjectGridManager erstellen.

Bereinigungsprogramme (Evictor) über XML-Konfiguration aktivieren

Anstelle der Verwendung der Schnittstelle "BackingMap" für die programmgesteuerte Definition der BackingMap-Attribute für den TTL-Evictor können Sie jede BackingMap-Instanz über eine XML-Datei konfigurieren. Der folgende Code veranschaulicht, wie Sie diese Attribute für drei verschiedene BackingMap-Maps definieren:

Vorbereitende Schritte

Bevor Sie beginnen, legen Sie den Typ des zu verwendenden Bereinigungsprogramms (Evictor) fest:

- **Auf der Lebensdauer basierendes Bereinigungsprogramm (Standard):** Das Standardbereinigungsprogramm verwendet eine Bereinigungsrichtlinie für jede BackingMap-Instanz, die auf der Lebensdauer (TTL, Time-to-Live) basiert.
- **Plug-in-Bereinigungsprogramm:** Plug-in-Bereinigungsprogramme verwenden gewöhnlich eine Bereinigungsrichtlinie, die auf der Anzahl der Einträge und nicht auf der Zeit basiert.

Die meisten Einstellungen des Evictors müssen vor der Initialisierung des ObjectGrids gesetzt werden.

Vorgehensweise

- Zum Definieren des TTL-Standardbereinigungsprogramms fügen Sie der ObjectGrid-XML-Deskriptordatei das Attribut "ttlEvictorType" hinzu.

  Das folgende Beispiel zeigt, dass die BackingMap-Instanz "map1" den TTL-Evictor-Typ NONE verwendet. Die BackingMap-Instanz "map2" verwendet den Evictortyp LAST_ACCESS_TIME oder LAST_UPDATE_TIME. Geben Sie nur eine dieser Einstellungen an. Die BackingMap-Instanz "map2" hat einen TTL-Wert von 1800 Sekunden bzw. 30 Minuten. Die BackingMap-Instanz "map3" ist mit dem TTL-Evictoryp CREATION_TIME und einem TTL-Wert von 1200 Sekunden bzw. 20 Minuten definiert.

```
<?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>
```

Abbildung 27. TimeToLive-Evictor mit XML aktivieren

- Verwenden Sie das folgende Beispiel, um einen Plug-in-Evictor zu definieren.

```
<?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>
```

Abbildung 28. Plug-in-Evictor mit XML definieren

Sperrstrategie konfigurieren

Sie können eine optimistische Strategie, eine pessimistische Strategie oder eine Strategie ohne Sperren für jede BackingMap in der WebSphere eXtreme Scale-Konfiguration definieren.

Informationen zu diesem Vorgang

Jede BackingMap-Instanz kann für die Verwendung einer der folgenden Sperrstrategien konfiguriert werden:

1. Optimistischer Sperrmodus
2. Pessimistischer Sperrmodus
3. Ohne

Die Standardsperrstrategie ist OPTIMISTIC (optimistisch). Verwenden Sie optimistisches Sperren, wenn die Daten nur selten geändert werden. Sperren werden nur für kurze Dauer gehalten, während die Daten aus dem Cache gelesen und in die Transaktion kopiert werden. Wenn der Transaktionscache mit dem Hauptcache synchronisiert wird, werden alle zwischengespeicherten Objekte, die aktualisiert wurden, mit der Originalversion verglichen. Wenn die Prüfung negativ ausfällt, wird eine Rollback-Operation für die Transaktion durchgeführt, und es wird eine Ausnahme des Typs "OptimisticCollisionException" ausgegeben.

Bei der Sperrstrategie PESSIMISTIC (pessimistisch) werden Sperren für Cacheeinträge angefordert. Diese Strategie sollte verwendet werden, wenn die Daten häufig geändert werden. Jedesmal, wenn ein Cacheeintrag gelesen wird, wird eine Sperre angefordert und so lange gehalten, bis die Transaktion abgeschlossen wird. Die Dauer einiger Sperren kann über die verfügbaren Isolationsstufen für die Sitzung optimiert werden.

Wenn keine Sperren erforderlich sind, weil die Daten nie oder nur in ruhigen Phasen aktualisiert werden, können Sie die Sperren inaktivieren, indem Sie die Sperr-

strategie NONE (Ohne) konfigurieren. Diese Strategie ist sehr schnell, weil kein Sperrenmanager erforderlich ist. Die Sperrstrategie NONE eignet sich ideal für Suchtabellen und schreibgeschützte Maps.

Weitere Einzelheiten zu Sperrstrategien finden Sie in Sperrstrategienden Informationen zu Sperrstrategien in der Veröffentlichung *Produktübersicht*.

Vorgehensweise

- **Optimistische Sperrstrategie konfigurieren**
  - Methode setLockStrategy über das Programm verwenden:

```
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 );
```

  - Attribut "lockStrategy" in ObjectGrid-XML-Deskriptordatei verwenden:

```
<?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>
```

- **Pessimistische Sperrstrategie konfigurieren**
  - Methode setLockStrategy über das Programm verwenden:

```
Pessimistische Sperrstrategie über das Programm angeben
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 );
```

  - Attribut "lockStrategy" in der ObjectGrid-XML-Deskriptordatei verwenden:

```
Pessimistische Sperrstrategie mit XML angeben
<?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>
```

- **Strategie ohne Sperren konfigurieren**
  - Methode setLockStrategy über das Programm verwenden:

```
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 );
```

– Attribut "lockStrategy" in ObjectGrid-XML-Deskriptordatei verwenden:

```
<?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>
```

Nächste Schritte

Um zu vermeiden, dass eine Ausnahme des Typs "java.lang.IllegalStateException" ausgelöst wird, müssen Sie die Methode "setLockStrategy" vor den Methoden "initialize" und "getSession" für die ObjectGrid-Instanz aufrufen.

Peer-to-Peer-Replikation mit JMS konfigurieren

Der JMS-basierte (Java Message Service) Peer-to-Peer-Replikationsmechanismus wird in der verteilten und in der lokalen Umgebung von WebSphere eXtreme Scale verwendet. JMS ist ein Kern-zu-Kern-Replikationsprozess und lässt die Übertragung von Datenaktualisierungen zwischen lokalen ObjectGrids und verteilten ObjectGrids zu. Mit diesem Mechanismus können Sie beispielsweise Datenaktualisierungen aus einem verteilten eXtreme-Scale-Datengrid in ein lokales eXtreme-Scale-Grid oder aus einem anderen Grid in einer anderen Systemdomäne verschieben.

Vorbereitende Schritte

Der JMS-basierte Peer-to-Peer-Replikationsmechanismus basiert auf den integrierten JMS-basierten Schnittstellen "ObjectGridEventListener" und "com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener". Ausführliche Informationen zum Aktivieren des Peer-to-Peer-Replikationsmechanismus finden Sie im Abschnitt „JMS-Ereignis-Listener” auf Seite 238.

Weitere Informationen finden Sie im Abschnitt „Mechanismus für Clientinvalidierung aktivieren” auf Seite 303.

Im Folgenden sehen Sie ein XML-Konfigurationsbeispiel für die Aktivierung eines Peer-to-Peer-Replikationsmechanismus in einer eXtreme-Scale-Konfiguration:

Konfiguration der Peer-to-Peer-Replikation - XML-Beispiel

```
<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>
```

Änderungen an Peer-JVMs verteilen

Die Objekte "LogSequence" und "LogElement" verteilen Änderungen zwischen Peer-JVMs und kommunizieren Änderungen, die in einer eXtreme-Scale-Transaktion stattfinden über ein ObjectGridEventListener-Plug-in.

Weitere Informationen zur Verwendung von Java Message Service (JMS) für die Verteilung von Transaktionsänderungen finden Sie im Abschnitt Transaktionen verteilen.

Eine Voraussetzung ist, dass die ObjectGrid-Instanz vom ObjectGridManager zwischengespeichert wird. Weitere Einzelheiten finden Sie in den Informationen zu den createObjectGrid-Methoden. Die boolesche Eigenschaft "cacheInstance" muss auf "true" gesetzt werden.

Dieser Mechanismus muss nicht implementiert werden. Sie können einen integrierten Mechanismus für die Peer-to-Peer-Replikation verwenden, um diese Funktion zu nutzen. Weitere Informationen finden Sie unter „Peer-to-Peer-Replikation mit JMS konfigurieren” auf Seite 234.

Eine Anwendung kann diese Objekte verwenden, um Änderungen, die in einem ObjectGrid vorgenommen werden, problemlos über einen Nachrichtentransport an die Peer-ObjectGrids in fernen Java Virtual Machines zu veröffentlichen und die Änderungen anschließend in dieser JVM anzuwenden. Die Klasse "LogSequenceTransformer" ist für die Aktivierung dieser Unterstützung kritisch. In diesem Abschnitt wird beschrieben, wie ein Listener mit einem JMS-Messaging-System für die Weitergabe der Nachrichten geschrieben wird. Zu diesem Zweck unterstützt eXtreme Scale die Übertragung von LogSequence-Objekten, die sich aus der Festschreibung einer eXtreme-Scale-Transaktion ergeben, über ein von IBM bereitgestelltes Plug-in an die Cluster-Member von WebSphere Application Server. Diese Funktion ist standardmäßig nicht aktiviert, kann aber konfiguriert werden. Wenn der Konsument oder Erzeuger jedoch kein WebSphere Application Server ist, kann die Verwendung eines externen JMS-Messaging-Systems erforderlich sein.

Mechanismus implementieren

Die Klasse "LogSequenceTransformer" und die APIs "ObjectGridEventListener", "LogSequence" und "LogElement" lassen die Verwendung jedes zuverlässigen Publish/Subscribe-Mechanismus für die Verteilung der Änderungen und die Filterung der zu verteilenden Maps zu. Die Snippets in diesem Abschnitt veranschaulichen, wie diese APIs mit JMS verwendet werden können, um ein Peer-to-Peer-ObjectGrid zu erstellen, das von Anwendungen gemeinsam genutzt wird, die sich auf verschiedenen Plattformen befinden, die einen gemeinsamen Nachrichtentransport verwenden.

Plug-in initialisieren

Das ObjectGrid ruft im Rahmen des ObjectGridEventListener-Schnittstellenvertrags die Methode "initialize" des Plug-ins auf, wenn das ObjectGrid gestartet wird. Die Methode "initialize" muss seine JMS-Ressourcen, einschließlich Verbindungen, Sit-

zungen und Veröffentlichungskomponenten (so genannten Publishern), anfordern und den Thread für den JMS-Listener starten.

Die folgenden Beispiele zeigen die Methode "initialize":

Beispiel für die Methode 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();

            // Die Verbindung muss gestartet werden, um Nachrichten zu empfangen.
            connection.start();

            // Die JMS_Sitzung ist nicht transaktionsorientiert (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);
            // Listener-Thread starten.
            listenerRunning = true;
            listenerThread = new Thread(this);
            listenerThread.start();
        } catch (Throwable e) {
            throw new ObjectGridRuntimeException("Cannot initialize", e);
        }
    }
```

Der Code zum Starten des Threads verwendet einen Java-SE-Thread (Java 2 Platform, Standard Edition). Wenn Sie einen Server von WebSphere Application Server Version 6.x oder WebSphere Application Server Version 5.x ausführen, verwenden Sie die API für asynchrone Beans, um diesen Dämon-Thread zu starten. Sie können auch die allgemeinen APIs verwenden. Im Folgenden sehen Sie ein Beispiel für ein Ersatz-Snippet, dass diese Aktion mit einem Arbeitsmanager veranschaulicht:

```
// Listener-Thread starten.
listenerRunning = true;
workManager.startWork(this, true);
```

Das Plug-in muss die Schnittstelle "Work" an Stelle der Schnittstelle "Runnable" implementieren. Außerdem müssen Sie eine Methode "release" hinzufügen, um die Variable "listenerRunning" auf "false" zu setzen. Das Plug-in muss mit einer WorkManager-Instanz in seinem Konstruktor bzw. bei Verwendung eines IoC-Containers (Inversion of Control) durch Injektion bereitgestellt werden.

Änderungen übertragen

Im Folgenden sehen Sie eine Beispielmethode "transactionEnd" für die Veröffentlichung der lokalen Änderungen, die in einem ObjectGrid vorgenommen werden. In diesem Beispiel wird JMS verwendet, aber Sie können jeden Nachrichtentransport verwenden, der zuverlässiges Publish/Subscribe-Messaging unterstützt.

Beispiel für die Methode transactionEnd

```
     // Diese Methode wird synchronisiert, um sicherzustellen,
    // dass die Nachrichten in der Reihenfolge veröffentlicht werden, in die
    // Transaktionen festgeschrieben werden. Falls die Veröffentlichung der Nachrichten
    // parallel gestartet wird, könnten die Empfänger die Map beschädigen,
    // da Löschanforderungen vor Einfügeanforderungen usw. ankommen könnten.
    public synchronized void transactionEnd(String txid, boolean isWriteThroughEnabled, boolean committed,
            Collection changes) {
        try {
            // Muss Write-through und festgeschrieben sein.
            if (isWriteThroughEnabled && committed) {
                // Folgen in eine Bytefeldgruppe (byte []) schreiben.
                ByteArrayOutputStream bos = new ByteArrayOutputStream();
                ObjectOutputStream oos = new ObjectOutputStream(bos);
                if (publishMaps.isEmpty()) {
                    // Gesamte Sammlung serialisieren
                    LogSequenceTransformer.serialize(changes, oos, this, mode);
                } else {
                    // LogSequence-Objekte auf der Basis des publishMaps-Inhalts filtern.
                    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);
                }
                // Objektnachricht für die Änderungen erstellen.
                oos.flush();
                ObjectMessage om = jmsSession.createObjectMessage(bos.toByteArray());
                // Eigenschaften festlegen.
                om.setStringProperty(PROP_TX, txid);
                om.setStringProperty(PROP_GRIDNAME, myGrid.getName());
                // Übertragen.
                publisher.publish(om);
            }
        } catch (Throwable e) {
            throw new ObjectGridRuntimeException("Cannot push changes", e);
        }
    }
```

In dieser Methode werden verschiedene Instanzvariablen verwendet:

- Variable `jmsSession`: Eine JMS-Sitzung, die zum Veröffentlichen von Nachrichten verwendet wird. Sie wird bei der Initialisierung des Plug-ins erstellt.
- Variable `mode`: Der Verteilungsmodus.
- Variable `publishMaps`: Eine Gruppe, die die Namen der einzelnen Maps mit zu veröffentlichenden Änderungen enthält. Wenn die Variable leer ist, werden alle Maps veröffentlicht.
- Variable `publisher`: Ein TopicPublisher-Objekt, das während der Ausführung der Methode "initialize" des Plug-ins ausgeführt wird.

Aktualisierungsnachricht empfangen und anwenden

Im Folgenden sehen Sie eine Beispielmethode "run". Diese Methode wird in einer Schleife ausgeführt, bis die Anwendung die Schleife stoppt. In jeder Schleifeniteration wird versucht, eine JMS-Nachricht zu empfangen und auf das ObjectGrid anzuwenden.

Beispiel für die Methode run für JMS-Nachrichten

```
private synchronized boolean isListenerRunning() {
        return listenerRunning;
    }

    public void run() {
        try {
            System.out.println("Listener starting");
            // JMS-Sitzung für den Empfang der Nachrichten abrufen.
            // Nicht transaktionsorientiert.
```

```
        TopicSession myTopicSession;
        myTopicSession = connection.createTopicSession(false, javax.jms.
    Session.AUTO_ACKNOWLEDGE);


        // Subskribenten für das Topic abrufen. True gibt an, dass keine Nachrichten
        // empfangen werden, die über Publisher in dieser Verbindung übertragen
        // wurden. Sonst werden eigene Aktualisierungen empfangen.
        TopicSubscriber subscriber = myTopicSession.createSubscriber(topic,
    null, true);
                    System.out.println("Listener started");
        while (isListenerRunning()) {
            ObjectMessage om = (ObjectMessage) subscriber.receive(2000);
            if (om != null) {
                // Sitzung verwenden, die bei der Initialisierung übergeben wurde.
                // Sehr wichtig, dass Write-through (Durchschreiben) hier nicht
                // verwendet wird.
                mySession.beginNoWriteThrough();
                byte[] raw = (byte[]) om.getObject();
                ByteArrayInputStream bis = new ByteArrayInputStream(raw);
                ObjectInputStream ois = new ObjectInputStream(bis);
                // LogSequence-Objekte dekomprimieren.
                Collection collection = LogSequenceTransformer.inflate(ois,
        myGrid);
                                    Iterator iter = collection.iterator();
                  while (iter.hasNext()) {
                    // Änderungen jeder Map entsprechend dem bei der
                    // Serialisierung des LogSequence-Objekts verwendeten
                    // Modus verarbeiten.
                                          LogSequence seq = (LogSequence) iter.next();
                    mySession.processLogSequence(seq);
                }
                mySession.commit();
            } // Wenn eine Nachricht vorhanden ist
        } // while loop
        // Verbindung stoppen.
        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");
  }
```

JMS-Ereignis-Listener

Der JMSObjectGridEventListener unterstützt einen Mechanismus für die Inaktivierung des clientseitigen nahen Caches und einen Mechanismus für die Peer-to-Peer-Replikation. Er ist eine JMS-Implementierung (Java Message Service) der Schnittstelle "ObjectGridEventListener".

Der Mechanismus für die Clientinaktivierung kann in einer verteilten eXtreme-Scale-Umgebung verwendet werden, um sicherzustellen, dass die Daten im clientnahen Cache mit Servern oder anderen Clients synchronisiert werden. Ohne diese Funktion könnte der clientnahe Cache veraltete Daten enthalten. Aber selbst mit diesem JMS-basierten Mechanismus für Clientinaktivierung müssen Sie wegen der Verzögerung für die Laufzeitumgebung beim Veröffentlichen von Aktualisierungen das Zeitfenster für die Aktualisierung eines clientnahen Caches berücksichtigen.

Der Mechanismus für die Peer-to-Peer-Replikation kann in verteilten und lokalen eXtreme-Scale-Umgebungen verwendet werden. Er ist ein Kern-zu-Kern-Replikationsprozess und lässt die Übertragung von Datenaktualisierungen zwischen lokalen ObjectGrids und verteilten ObjectGrids zu. Mit diesem Mechanismus können Sie beispielsweise Datenaktualisierungen aus einem verteilten Grid in ein lokales ObjectGrid oder aus einem anderen Grid in einer anderen Systemdomäne verschieben.

Der JMSObjectGridEventListener erfordert, dass der Benutzer JMS- und JNDI-Informationen (Java Naming and Directory Interface) konfiguriert, damit die erforderlichen JMS-Ressourcen abgerufen werden. Außerdem müssen replikationsbezogene Eigenschaften ordnungsgemäß gesetzt werden. In einer JEE-Umgebung muss JNDI in Web- und EJB-Containern (Enterprise JavaBean) verfügbar sein. In diesem Fall ist die JNDI-Eigenschaft optional, sofern Sie keine externen JMS-Ressourcen abrufen möchten.

Dieser Ereignis-Listener hat Eigenschaften, die Sie über XML- oder programmgesteuerte Ansätze konfigurieren und nur für die Clientinaktivierung und/oder nur für die Peer-to-Peer-Replikation verwenden können. Die meisten Eigenschaften sind für die Anpassung des Verhaltens zum Erzielen der erforderlichen Funktionalität optional.

.

Weitere Informationen finden Sie in den Informationen zur API "JMSObjectGridEventListener".

JMSObjectGridEventListener-Plug-in erweitern

Das JMSObjectGridEventListener-Plug-in ermöglicht Peer-ObjectGrid-Instanzen, Aktualisierungen zu empfangen, wenn die Daten im Grid geändert oder entfernt wurden. Außerdem ermöglicht es die Benachrichtigung von Clients, wenn Einträge aktualisiert oder einem eXtreme-Scale-Grid entfernt werden. In diesem Abschnitt wird beschrieben, wie das JMSObjectGridEventListener-Plug-in erweitert werden kann, damit Anwendungen beim Empfang einer JMS-Nachricht benachrichtigt werden können. Dies ist äußerst hilfreich, wenn die Einstellung CLIENT_SERVER_MODEL für die Clientinaktivierung verwendet wird.

Bei der Ausführung in der Empfängerrolle wird die überschriebene Methode "JMSObjectGridEventListener.onMessage" automatisch von der eXtreme-Scale-Laufzeitumgebung aufgerufen, wenn die JMSObjectGridEventListener-Instanz JMS-Nachrichtenaktualisierungen vom Grid empfängt. Diese Nachrichten schließen eine Sammlung von LogSequence-Objekten ein. Die LogSequence-Objekte werden an die Methode "onMessage" übergeben, und die Anwendung verwendet das LogSequence-Objekt, um die Cacheeinträge zu identifizieren, die eingefügt, gelöscht, aktualisiert oder ungültig gemacht wurden.

Zur Verwendung des onMessage-Erweiterungspunkts führen Anwendungen die folgenden Schritte aus:

1. Erstellen Sie eine neue Klasse, die die Klasse "JMSObjectGridEventListener" erweitert und die Methode "onMessage" überschreibt.
2. Konfigurieren Sie den erweiterten JMSObjectGridEventListener auf dieselbe Weise wie den ObjectGridEventListener für ObjectGrid.

Die erweiterte Klasse "JMSObjectGridEventListener" ist eine Unterklasse der Klasse "JMSObjectGridEventListener" und kann nur zwei Methoden überschreiben: initialize (optional) und onMessage. Wenn eine Unterklasse der Klasse "JMSObjectGridEventListener" ObjectGrid-Artefakte wie ObjectGrid oder Session in der Methode "onMessage" verwenden muss, kann sie diese Artefakte in der Methode "initialize" abrufen und als Instanzvariablen zwischenspeichern. In der Methode "onMessage" können zwischengespeicherte ObjectGrid-Artefakte verwendet werden, um eine übergebene LogSequence-Sammlun zu verarbeiten.

Anmerkung: Die überschriebene Methode "initialize" muss die Methode "super.initialize" aufrufen, um den übergeordneten JMSObjectGridEventListener ordnungsgemäß zu initialisieren.

Im Folgenden sehen Sie ein Beispiel für eine erweiterte JMSObjectGridEventListener-Klasse:

```
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;

    /**
     * Dieses Grid ist dem Listener zugeordnet.
     */
     ObjectGrid grid;

    /**
     * Die Sitzung, die dem Listener zugeordnet ist
     */
     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) {
  // Anmerkung: Wenn Sie ein ObjectGrid-Artefakt verwenden müssen, muss diese Klasse
  // ObjectGrid von der übergebenen Session-Instanz und die ObjectMap von der
  // Session-instanz für jede transaktionsorientierte ObjectGrid-Map-Operation abrufen.

  super.initialize(session);  // Die Methode initialize der Superklasse muss aufgerufen werden
  this.session = session; // Session-Instanz zwischenspeichern, falls sie
  //   zum Durchführen der Map-Operation benötigt wird
  this.grid = session.getObjectGrid(); // ObjectGrid abrufen, falls
  //   ObjectGrid-Informationen abgerufen werden müssen

  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 + "]";
}
}
```

Konfiguration

Die erweiterte JMSObjectGridEventListener-Klasse muss für den Mechanismus für die Clientinaktivierung und für den Mechanismus für die Peer-to-Peer-Replikation gleich konfiguriert werden. Im Folgenden sehen Sie ein Beispiel für die XML-Konfiguration:

```
<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>
```

Anmerkung: Der Klassenname der Bean "ObjectGridEventListener" wird mit der erweiterten JMSObjectGridEventListener-Klasse mit denselben Eigenschaften wie die generische JMSObjectGridEventListener-Klasse konfiguriert.

Implementierungsrichtlinien konfigurieren

Verwenden Sie die XML-Deskriptordatei für die Implementierungsrichtlinie und die XML-ObjectGrid-Deskriptordatei, um eine verteilte Topologie zu verwalten. Die Implementierungsrichtlinie wird als XML-Datei codiert, die dem Container-Server bereitgestellt wird. Die Implementierungsrichtlinie enthält Informationen zu Maps, MapSets, Partitionen, Replikaten usw. Außerdem steuert sie das Verhalten für die Shard-Verteilung.

Verteilte Implementierungen konfigurieren

Verwenden Sie die XML-Deskriptordatei für die Implementierungsrichtlinie und die ObjectGrid-XML-Deskriptordatei, um Ihre Topologie zu verwalten.

Die Implementierungsrichtlinie wird als XML-Datei codiert, die dem Container-Server von eXtreme Scale bereitgestellt wird. Die XML-Datei enthält die folgenden Informationen:

- die Maps, die zu den einzelnen MapSets gehören,
- die Anzahl der Partitionen,
- die Anzahl synchroner und asynchroner Replikate.

Die Implementierungsrichtlinie steuert auch die folgenden Verteilungsverhalten:

- die Mindestanzahl aktiver Container-Server, bevor die Verteilung stattfindet,
- die automatische Verteilung verloren gegangener Shards,
- die Verteilung jedes Shards einer einzelnen Partition an eine jeweils andere Maschine.

Endpunktinformationen werden in der dynamischen Umgebung nicht vorkonfiguriert. Es sind keine Servernamen oder Informationen zur physischen Topologie in der Implementierungsrichtlinie enthalten. Alle Shards in einem Datengrid werden automatisch vom Katalogservice an die Container-Server verteilt. Der Katalogservice verwendet die in der Implementierungsrichtlinie definierten Vorgaben, um die Verteilung der Shards automatisch zu verwalten. Mit dieser automatischen Shard-Verteilung lassen sich große Datengrids ohne großen Aufwand konfigurieren. Sie können der Umgebung bei Bedarf auch Server hinzufügen.

Einschränkung: In einer Umgebung mit WebSphere Application Server werden Stammgruppen mit mehr als 50 Membern nicht unterstützt.

Während des Starts wird eine XML-Implementierungsrichtliniendatei an einen Container-Server übergeben. Eine Implementierungsrichtlinie muss zusammen mit einer ObjectGrid-XML-Datei verwendet werden. Die Implementierungsrichtlinie ist zum Starten eines Container-Servers zwar nicht erforderlich, aber empfehlenswert. Die Implementierungsrichtlinie muss mit der verwendeten ObjectGrid-XML-Datei kompatibel sein. Für jedes Element "objectgridDeployment" in der Implementierungsrichtlinie müssen Sie ein entsprechendes Element "objectGrid" in die Object-

Grid-XML-Datei einfügen. Die Maps im objectgridDeployment-Element müssen mit den backingMap-Elementen in der ObjectGrid-XML konsistent sein. Jedes backingMap-Element darf nur in einem einzigen mapSet-Element referenziert werden.

Im folgenden Beispiel soll die Datei `companyGridDpReplication.xml` mit der entsprechenden Datei `companyGrid.xml` gepaart werden:

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>
```

Die Datei `companyGridDpReplication.xml` enthält ein mapSet-Element , das in 11 Partitionen eingeteilt ist. Jede Partition muss genau ein synchrones Replikat haben. Die Anzahl der synchronen Replikate wird über die Attribute "minSyncReplicas" und "maxSyncReplicas" festgelegt. Da das Attribut "minSyncReplicas" auf 1 gesetzt ist, muss jede Partition im mapSet-Element mindestens ein verfügbares synchrones Replikat für die Verarbeitung von Schreiboperationen haben. Da das Attribut "maxSyncReplicas" auf 1 gesetzt ist, darf jede Partition maximal ein einziges synchrones Replikat haben. Die Partitionen in diesem mapSet-Element haben keine asynchronen Replikate.

Das Attribut "numInitialContainers" weist den Katalogservice an, die Verteilung zu verzögern, bis vier Container-Server für die Unterstützung dieser ObjectGrid-Instanz verfügbar sind. Das Attribut "numInitialContainers" wird ignoriert, sobald die angegebene Anzahl an Container-Servern erreicht ist.

7.1.1+ Sie können auch die Eigenschaft **placementDeferralInterval** und den Befehl **xscmd -c suspendBalancing** verwenden, um die Verteilung der Shards auf die Container-Server zu verzögern.

Die Datei `companyGridDpReplication.xml` ist nur ein Basisbeispiel. Eine Implementierungsrichtlinie bietet Ihnen jedoch die vollständige Kontrolle über Ihre Umgebung.

Verteilte Topologie

Verteilte kohärente Caches bieten eine höhere Leistung, Verfügbarkeit und Skalierbarkeit, die Sie konfigurieren können.

WebSphere eXtreme Scale führt eine automatische gleichmäßige Verteilung der Server durch. Sie können weitere Server hinzufügen, ohne WebSphere eXtreme Scale erneut starten zu müssen. Das Hinzufügen zusätzlicher Server ohne Neustart von eXtreme Scale ermöglicht Ihnen die Verwendung einfacher Implementierungen und Implementierungen in Terabytegröße, in denen Tausende von Servern benötigt werden.

Diese Implementierungstopologie ist flexibel. Mit dem Katalogservice können Sie Server hinzufügen und entfernen, um Ressourcen besser zu nutzen, ohne den gesamten Cache entfernen zu müssen. Sie können die Befehle **startOgServer** und **stopOgServer** verwenden, um Container-Server zu starten und zu stoppen. Beide Befehle erfordern die Angabe der Option **-catalogServiceEndPoints**. Alle Clients der verteilten Topologie kommunizieren mit dem Katalogservice über Internet Interoperability Object Protocol (IIOP). Alle Clients verwenden die Schnittstelle "ObjectGrid", um mit Servern zu kommunizieren.

Mit den dynamischen Konfigurationsfunktionen von WebSphere eXtreme Scale können dem System Ressourcen ohne großen Aufwand hinzugefügt werden. Container enthalten die Daten, und der Katalogservice ermöglicht Clients die Kommunikation mit dem Container-Server-Grid. Der Katalogservice leitet Anforderungen weiter, reserviert Speicherplatz in den Host-Container-Servern und verwaltet den Status und die Verfügbarkeit des Gesamtsystems. Clients stellen eine Verbindung zu einem Katalogservice her, rufen eine Beschreibung der Container-Server-Topologie ab und kommunizieren dann direkt mit jedem Server. Wenn sich die Servertopologie ändert, weil neue Server hinzugefügt werden oder weil Server ausfallen, leitet der Katalogservice Clientanforderungen automatisch an den entsprechenden Server weiter, der die Daten enthält.

Ein Katalogservice existiert gewöhnlich in einem eigenen Grid von Java Virtual Machines. Ein einziger Katalogserver kann mehrere Server verwalten. Sie können einen Container-Server in einer JVM eigenständig starten oder den Container-Server in eine beliebige JVM mit anderen Container-Servern für verschiedene Server laden. Ein Client kann in jeder JVM ausgeführt werden und mit einem oder mehreren Servern kommunizieren. Ein Client kann auch in derselben JVM wie ein Container-Server ausgeführt werden.

Sie können eine Implementierungsrichtlinie auch über das Programm erstellen, wenn Sie einen Container-Server in einen vorhandenen Java-Prozess oder in eine vorhandene Anwendung integrieren. Weitere Informationen finden Sie in der Dokumentation zur API DeploymentPolicy.

Shard-Verteilung mit Zonen steuern

Verwenden Sie Ihre Implementierungsrichtlinie, um Zonen zu definieren. Mit Zonen können Sie die Shard-Verteilung in WebSphere eXtreme Scale steuern. Zonen sind ein logisches, benutzerdefiniertes Konzept für die Darstellung logischer Gruppierungen physischer Server.

Zonen für die Verteilung von Replikaten konfigurieren

Die Zonenunterstützung ermöglicht fortgeschrittene Konfigurationen für die Verteilung von Replikaten auf Rechenzentren. Mit dieser Funktionalität können Grid mit Tausenden von Partitionen ohne großen Aufwand mit einer Handvoll optionaler

Verteilungsregeln verwaltet werden. Ein Rechenzentrum kann auf verschiedene Stockwerke eines Gebäudes, verschiedene Gebäude oder selbst verschiedene Städte verteilt sein. Die Unterscheidungsmerkmale werden über Zonenregeln konfiguriert.

Flexibilität von Zonen

Sie können Shards auf Zonen verteilen. Diese Funktion gibt Ihnen mehr Kontrolle darüber, wie eXtreme Scale Shards in einem Grid verteilt. Java Virtual Machines, die einen eXtreme-Scale-Server enthalten, können mit einer Zonen-ID gekennzeichnet werden. Die Implementierungsdatei kann jetzt eine oder mehrere Zonenregeln enthalten, und diese Zonenregeln werden einem Shard-Typ zugeordnet. Der folgende Abschnitt enthält eine Übersicht über die Zonennutzung. Weitere Einzelheiten finden Sie in den Informationen zur Steuerung der Shard-Verteilung mit Zonen in der Veröffentlichung *Verwaltung*.

Verteilungszonen steuern, wie eXtreme Scale primäre Shards und Replikate zuordnet, um erweiterte Topologie zu konfigurieren.

Eine Java Virtual Machine kann mehrere Container, aber nur einen einzigen Server haben. Ein Container kann mehrere Shards aus einem einzigen ObjectGrid enthalten.

Diese Funktionalität ist hilfreich, um sicherzustellen, dass Replikate und primäre Shards für eine höhere Verfügbarkeit auf unterschiedliche Positionen oder Zonen verteilt werden. Normalerweise verteilt eXtreme Scale ein primäres Shard und ein Replikat-Shard nicht an Java Virtual Machines mit derselben IP-Adresse. Diese einfache Regel verhindert gewöhnlich, dass zwei eXtreme-Scale-Server auf demselben physischen Computer platziert werden. Möglicherweise benötigen Sie jedoch einen flexibleren Mechanismus. Sie verwenden beispielsweise zwei Blade-Gehäuse und möchten, das die primären Shards *einheitenübergreifend* auf beide Gehäuse verteilt werden und dass das Replikat jedes primären Shards nicht in demselben Gehäuse platziert wird wie das zugehörige primäre Shard.

Einheitenübergreifend verteilen bedeutet, dass die primären Shards in jeweils einer Zone und die zugehörigen Replikate in der jeweils anderen Zone platziert werden. Das primäre Shard 0 wird beispielsweise in Zone A und das synchrone Replikat 0 in Zone B platziert. Das primäre Shard 1 wird in Zone B und das synchrone Replikat 1 wird in Zone A platziert.

Der Gehäusename ist in diesem Fall der Zonenname. Alternativ können Sie Zonen nach den Stockwerken in einem Gebäude benennen und Zonen verwenden, um sicherzustellen, dass primäre Shards und Replikate derselben Daten auf unterschiedlichen Stockwerken gespeichert werden. Gebäude und Rechenzentren können ebenfalls verwendet werden. Es wurden Tests mit Zonen in Rechenzentren durchgeführt, um sicherzustellen, dass die Daten ordnungsgemäß zwischen den Rechenzentren repliziert werden. Wenn Sie den HTTP-Sitzungsmanager für eXtreme Scale verwenden, können Sie ebenfalls Zonen verwenden. Mit diesem Feature können Sie eine einzelne Webanwendung auf drei Rechenzentren verteilen und sicherstellen, dass HTTP-Sitzungen für Benutzer in den Rechenzentren repliziert werden, so dass die Sitzungen wiederhergestellt werden können, falls ein komplettes Rechenzentrum ausfällt.

WebSphere eXtreme Scale kennt den Bedarf, ein großes Grid über mehrere Datenzentren hinweg zu verwalten. Das Produkt kann sicherstellen, dass Sicherungen und primäre Shards für dieselbe Partition bei Bedarf auf unterschiedliche Rechenzentren verteilt werden. Es kann alle primären Shards im Rechenzentrum 1 platzie-

ren und alle Replikate im Rechenzentrum 2, oder es kann die primären Shards und Replikate im Umlaufverfahren auf beide Datenzentren verteilen. Die Regeln sind so flexibel, dass zahlreiche Szenarien möglich sind. eXtreme Scale kann auch Tausende von Servern verwalten. Diese Funktionalität, kombiniert mit der vollständig automatischen Verteilung unter Berücksichtigung von Rechenzentren macht solch große Grids aus Verwaltungssicht kosteneffizient. Administratoren können ohne großen Aufwand und effizient festlegen, was sie möchten.

Als Administrator verwenden Sie Verteilungszonen, um zu steuern, wo primäre Shards und Replikat-Shards platziert werden. Auf diese Weise können fortgeschrittene Topologien mit hoher Leistung und hoher Verfügbarkeit konfiguriert werden. Wie bereits erwähnt, können Sie eine Zone für jede logische Gruppierung von eXtreme-Scale-Prozessen definieren. Diese Zonen können physischen Workstationstandorten wie Rechenzentren, Stockwerken eines Rechenzentrums oder Blade-Gehäusen entsprechen. Sie können Daten einheitenübergreifend auf Zonen verteilen, was Ihnen eine höhere Verfügbarkeit bietet, oder Sie können die primären Shards und Replikate auf verschiedene Zonen aufteilen, wenn ein fehlertoleranter Modus erforderlich ist.

eXtreme-Scale-Server einer Zone zuordnen, die nicht WebSphere Extended Deployment verwendet

Wenn eXtreme Scale mit Java Standard Edition oder einem Anwendungsserver verwendet wird, der nicht auf WebSphere Extended Deployment Version 6.1 basiert, kann eine JVM, die ein Shard-Container ist, mit den folgenden Verfahren einer Zone zugeordnet werden.

Anwendungen, die das Script "startOgServer" verwenden

Das Script "startOgServer" wird verwendet, um eine eXtreme-Scale-Anwendung zu starten, wenn diese nicht in einen vorhandenen Server integriert ist. Mit dem Parameter **-zone** wird die Zone angegeben, die für alle Container im Server verwendet werden soll.

Zone beim Starten eines Containers über APIs angeben

Knoten von WebSphere Extended Deployment Zonen zuordnen

Wenn Sie eXtreme Scale mit Java EE-Anwendungen von WebSphere Extended Deployment verwenden, können Sie die Knotengruppen von WebSphere Extended Deployment nutzen, um Server auf bestimmte Zonen zu verteilen.

In eXtreme Scale kann eine JVM nur zu einer einzigen Zone gehören. WebSphere lässt jedoch die Zugehörigkeit eines Knotens zu mehreren Knotengruppen zu. Sie können die Funktionalität von eXtreme-Scale-Zonen verwenden, wenn Sie sicherstellen, dass jeder Ihrer Knoten nur zu einer einzigen Zonenknotengruppe gehört.

Mit der folgenden Syntax können Sie Ihre Knotengruppe benennen, um sie als Zone zu deklarieren: `Replikationszone<eindeutiges_Suffix>`. Server, die auf einem Knoten ausgeführt werden, der zu einer solchen Knotengruppe gehört, werden in die Zone eingeschlossen, die über den Namen der Knotengruppe angegeben wird. Im Folgenden finden Sie eine Beschreibung einer Beispieltopologie.

Zuerst konfigurieren Sie vier Knoten, Knoten1, Knoten2, Knoten2 und Knoten4, mit jeweils zwei Servern. Anschließend erstellen Sie eine Knotengruppe mit dem Namen "ReplikationszoneA" und eine Knotengruppe mit dem Namen "Replikati-

onszoneB". Danach fügen Sie Knoten1 und Knoten2 der ReplikationszoneA und Knoten3 und Knoten4 der ReplikationszoneB hinzu.

Wenn die Server auf Knoten1 und Knoten2 gestartet werden, werden sie der ReplikationszoneA zugeordnet. Knoten3 und Knoten4 werden beim Starten der ReplikationszoneB zugeordnet.

Eine Grid-Member-JVM überprüft die Zonenzugehörigkeit nur beim Start. Das Hinzufügen einer neuen Knotengruppe und das Ändern der Zugehörigkeit haben nur Auswirkungen auf neu gestartete bzw. erneut gestartete JVMs.

Zonenregeln

Eine eXtreme-Scale-Partition hat ein einziges primäres Shard und kein oder mehrere Replikat-Shards. In diesem Beispiel wird die folgende Namenskonvention für diese Shards verwendet: `P` ist das primäre Shard, `S` ist ein synchrones Replikat und `A` ein asynchrones Replikat. Eine Zonenregeln besteht aus drei Komponenten:

- Regelname,
- Zonenliste,
- Attribut inclusive oder exclusive.

Der Zonenname für einen Container kann gemäß der Beschreibung in der Dokumentation für „Integrierte Server-API” auf Seite 421 angegeben werden. Eine Zonenregel gibt die gültige Gruppe von Zonen an, an die ein Shard verteilt werden kann. Das Attribut "inclusive" zeigt an, dass nach der Verteilung eines Shards an eine Zone aus der Liste alle anderen Shards ebenfalls an diese Zone verteilt werden. Die Einstellung "exclusive" zeigt an, dass jedes Shard für eine Partition an eine jeweils andere Zone aus der Zonenliste verteilt wird. Die Verwendung der Einstellung "exclusive" bedeutet beispielsweise, dass die Zonenliste bei drei Shards (primäres Shard und zwei synchrone Replikate) drei Zonen enthalten muss.

Jedem Shard kann eine einzige Zonenregel zugeordnet werden. Eine Zonenregel kann von zwei Shards gemeinsam verwendet werden. Wenn eine Regel gemeinsam genutzt wird, gilt das Attribut "inclusive" bzw. "exclusive" für die Shards aller Typen, die eine Regel gemeinsam nutzen.

Beispiele

Es folgen diverse Beispiele, die verschiedene Szenarien und die entsprechende Konfiguration für die Implementierung des jeweiligen Szenarios veranschaulichen.

Primäre Shards und Replikate einheitenübergreifend auf Zonen verteilen

Sie haben drei Blade-Gehäuse und möchten, dass die primären Shards auf alle drei Gehäuse verteilt werden und dass jeweils ein einziges Replikat auf einem jeweils anderen Gehäuse als das zugehörige primäre Shard platziert wird. Definieren Sie jedes Gehäuse als Zone mit den Gehäusenamen `ALPHA`, `BETA` und `GAMMA`. Im Folgenden sehen Sie die zugehörige Implementierungs-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="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>
```

Diese Implementierungs-XML enthält ein Grid mit dem Namen "library" mit einer einzigen Map mit dem Namen "book". Es werden vier Partitionen mit einem einzigen synchronen Replikat verwendet. Die Klausel für die Zonenmetadaten zeigt die Definition einer einzigen Zonenregel und die Zuordnung von Zonenregeln zu Shards. Die primären und synchronen Shards werden der Zonenregel "stripeZone" zugeordnet. Die Zonenregel umfasst alle drei Zonen und verwendet eine exklusive Verteilung. Diese Regel bedeutet Folgendes: Wenn das primäre Shard für die Partition 0 in Zone `ALPHA` platziert wird, dann wird das Replikat für die Partition 0 in Zone `BETA` oder `GAMMA` platziert. Die primären Shards für andere Partitionen werden in anderen Zonen platziert und die zugehörigen Replikate dann in einer jeweils anderen als das entsprechende primäre Shard.

Asynchrones Replikat in einer anderen Zone als das primäre Shard und das synchrone Replikat

In diesem Beispiel gibt es zwei Gebäude, die über eine Verbindung mit hoher Latenzzeit miteinander verbunden sind. In allen Szenarien möchten Sie eine hohe Verfügbarkeit, um einen Datenverlust zu vermeiden. Der Einfluss der synchronen Replikation zwischen den Gebäuden auf die Leistung drängt Sie jedoch zu einem Kompromiss. Sie möchten ein primäres Shard mit einem synchronen Replikat in einem Gebäude und ein asynchrones Replikat in einem anderen Gebäude. Normalerweise sind Ausfälle auf JVM-Abstürze oder Computerausfälle und nicht auf Probleme mit der Größe zurückzuführen. Mit dieser Topologie können Sie normale Ausfälle ohne Datenverlust bewältigen. Der Verlust eines Gebäudes tritt so selten auf, dass in diesem Fall ein gewisser Datenverlust akzeptabel ist. Sie können zwei Zonen erstellen, eine für jedes Gebäude. Im Folgenden sehen Sie die zugehörige XML-Implementierungsdatei:

```
<?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>
```

Das primäre Shard und das synchrone Replikat-Shard verwenden beide die Zonenregel "primaySync" mit der Einstellung "false" für das Attribut "exclusive". Wenn das primäre Shard oder das synchrone Replikat in einer Zone platziert wird, wird deshalb das jeweils andere Shard in derselben Zone platziert. Das asynchrone Replikat verwendet eine zweite Zonenregel mit denselben Zonen wie die Zonenregel "primarySync", verwendet aber die Einstellung "true" für das Attribut **exclusive-**

Placement. Dieses Attribut gibt an, dass ein Shard nicht zusammen mit einem anderen Shard aus derselben Partition in einer Zone platziert werden kann. Deshalb wird das asynchrone Replikat nicht in derselben Zone platziert wie das primäre Shard bzw. das synchrone Replikat.

Alle primären Shards an eine Zone und alle Replikate an eine andere Zone verteilen

In diesem Szenario befinden sich alle primären Shards in einer bestimmten Zone und alle Replikate in einer anderen Zone. Es gibt ein primäres Shard und ein einziges asynchrones Replikat. Alle Replikate befinden sich in Zone A und alle primären Shards in 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>
```

In diesem Beispiel sehen Sie zwei Regeln, eine für die primären Shards (P) und eine andere für die Replikate (A).

Zonen über Weitverkehrsnetze (WANs)

Sie können eine einzige Instanz von eXtreme Scale über mehrere Gebäude oder Rechenzentren hinweg mit langsameren Netzverbindungen untereinander implementieren. Langsamere Netzverbindungen führen zu einer geringeren Bandbreite und zu einer höheren Latenzzeit. Das Risiko von Netzpartitionen ist in diesem Modus aufgrund der Netzüberlastung und anderen Faktoren ebenfalls erhöht. eXtreme Scale nähert sich diesen ungeeigneten Umgebungsbedingungen auf die folgenden Arten.

Begrenzter Austausch von Überwachungssignalen zwischen Zonen

Java Virtual Machines, die in Stammgruppen zusammengefasst sind, tauschen Überwachungssignale miteinander aus. Wenn der Katalogservice die Java Virtual Machines in Gruppen organisiert, können sich diese Gruppen nicht über mehrere Zonen erstrecken. Ein führendes Member dieser Gruppe überträgt die Zugehörigkeitsdaten mit Push an den Katalogservice. Der Katalogservice prüft alle gemeldeten Fehler, bevor er Maßnahmen ergreift. Dazu versucht er, eine Verbindung zu den fehlerverdächtigen Java Virtual Machines herzustellen. Wenn der Katalogservice eine falsche Fehlererkennung feststellt, ergreift er keine Maßnahmen, da die Stammgruppenpartition in kurzer Zeit wiederhergestellt ist.

Außerdem sendet der Katalogservice in regelmäßigen Abständen Überwachungssignale an das führende Member jeder Stammgruppe, um Fälle von Stammgruppenisolation zu behandeln.

Routing an bevorzugte Zonen

Mithilfe des Routings an bevorzugte Zonen können Sie definieren, wie WebSphere eXtreme Scale Transaktionen an Zonen weiterleitet.

Sie können steuern, wohin die Shards eines Datengrids verteilt werden. Weitere Informationen zu einigen Basisszenarien und zur Konfiguration der entsprechenden Implementierungsrichtlinie finden Sie unter „Zonen für die Verteilung von Replikaten konfigurieren" auf Seite 244.

Das Routing an bevorzugte Zonen gibt Clients von WebSphere eXtreme Scale die Möglichkeit, Vorgaben für eine bestimmte Zonen festzulegen. Auf diese Weise werden Clienttransaktionen zunächst an bevorzugte Zonen weitergeleitet, bevor versucht wird, sie an andere Zonen weiterzuleiten.

Voraussetzungen für das Routing an bevorzugte Zonen

Stellen Sie vor der Verwendung des Routings an bevorzugte Zonen sicher, dass die Anwendung die Voraussetzungen Ihres Szenarios zu erfüllen.

Für die Verwendung des Routings an bevorzugte Zonen ist eine containerbezogene Partitionsverteilung erforderlich. Diese Verteilungsstrategie eignet sich gut für Anwendungen, die Sitzungsdaten im ObjectGrid speichern. Die Standardverteilungsstrategie für WebSphere eXtreme Scale ist `fixed-partition`. Bei der Festschreibung einer Transaktion wird ein Hash-basierter Algorithmus für die Schlüssel verwendet, um zu bestimmen, welche Partition das Schlüssel/Wert-Paar der Map aufnimmt, wenn die Verteilung mit festen Partitionen verwendet wird.

Bei der containerbezogenen Verteilung werden Ihre Daten über das SessionHandle-Objekt einer zufälligen Partition zugeordnet, wenn die Transaktion festgeschrieben wird. Sie müssen das SessionHandle-Objekt rekonstruieren können, um Ihre Daten aus dem Datengrid abrufen zu können.

Wenn Sie Zonen verwenden, haben Sie mehr Kontrolle darüber, wohin die primären Shards und Replikat-Shards in der Domäne verteilt werden. Die Verwendung mehrerer Zonen in der Implementierung ist vorteilhaft, wenn Ihre Daten an mehreren physischen Standorten gespeichert sind. Die geographische Trennung von primären Shards und Replikaten ist eine Methode sicherzustellen, dass der katastrophale Verlust eines Rechenzentrums keine Auswirkung auf die Verfügbarkeit der Daten hat.

Wenn die Daten auf mehrere Zonen verteilt sind, ist es wahrscheinlich, dass auch die Clients in der Topologie verteilt werden. Das Routing von Clients an ihre lokalen Zonen oder Rechenzentren hat den offensichtlichen Vorteil einer kürzeren Netzlatenzzeit. Leiten Sie Clients, sofern möglich, an lokale Zonen oder Rechenzentren weiter.

Topologie für das Routing an bevorzugte Zonen konfigurieren

Stellen Sie sich das folgende Szenario vor: Sie haben zwei Rechenzentren: Chicago und London. Zur Minimierung der Clientantwortzeiten möchten Sie, dass Clients Daten aus ihrem lokalen Rechenzentrum lesen und ihre Daten auch dorthin schreiben.

Primäre Shards müssen in jedem Rechenzentrum verteilt werden, so dass Transaktionen lokal von jeder Position aus geschrieben werden können. Clients müssen die Zonen kennen, um Anforderungen an die lokale Zone weiterzuleiten.

Bei der containerbezogenen Verteilung werden neue primäre Shards auf jeden gestarteten Container-Server gefunden. Replikate werden entsprechend den Zonen- und Verteilungsregeln in der Implementierungsrichtlinie verteilt. Standardmäßig wird ein Replikat einer anderen Zone als das primäre Shard zugeteilt. Sehen Sie sich die folgende Implementierungsrichtlinie für dieses Szenario an:

```
<?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>
```

Jeder Container, der mit der Implementierungsrichtlinie gestartet wird, erhält drei neue primäre Shards. Jedes primäre Shard hat ein asynchrones Replikat. Starten Sie jeden Container mit dem entsprechenden Zonennamen. Verwenden Sie den Parameter **-zone**, wenn Sie Ihre Container mit dem Script **startOgServer** starten.

Für einen Container-Server in 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
```

Wenn Ihre Container in WebSphere Application Server ausgeführt werden, müssen Sie eine Knotengruppe erstellen und dem Namen dieser Knotengruppe das Präfix `ReplicationZone` voranstellen. Auf den Knoten in diesen Knotengruppen ausgeführte Knoten werden an die entsprechende Zone verteilt. Server, die auf einem Knoten in Chicago ausgeführt werden, könnten beispielsweise in einer Knotengruppe mit dem Namen `ReplicationZoneChicago` enthalten sein.

Weitere Informationen finden Sie unter „Zonen für die Verteilung von Replikaten konfigurieren” auf Seite 244.

Primäre Shards für die Zone "Chicago" haben Replikate in der Zone "London". Primäre Shards für die Zone "London" haben Replikate in der Zone "Chicago".

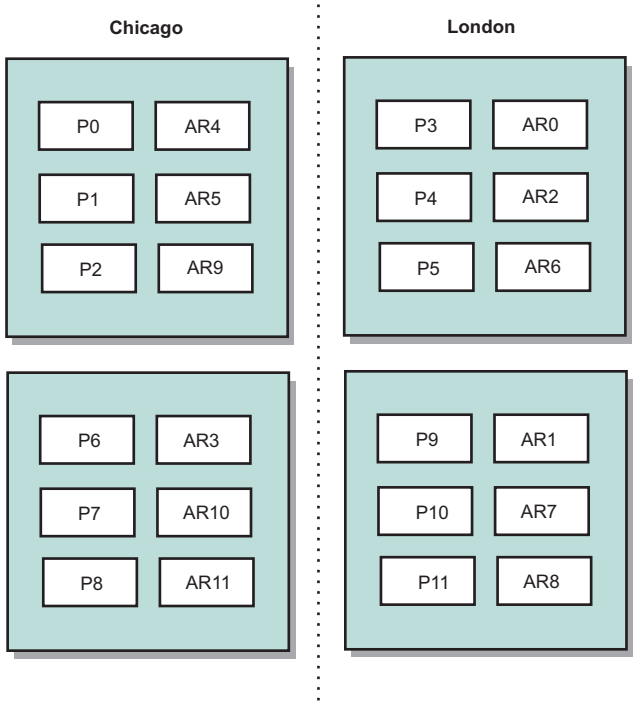


Abbildung 29. Primäre Shards und Replikate in Zonen

Legen Sie die bevorzugten Zonen für die Clients fest. Stellen Sie Ihrer Client-JVM eine Clienteigenschaftendatei bereit. Erstellen Sie eine Datei mit dem Namen `objectGridClient.properties`, und stellen Sie sicher, dass diese Datei im Klassenpfad enthalten ist.

Schließen Sie die Eigenschaft **`preferZones`** in die Datei ein. Setzen Sie den Eigenschaftswert auf die entsprechende Zone. Clients in Chicago müssen den folgenden Wert in der Datei `objectGridClient.properties` haben:

```
preferZones=Chicago
```

Die Eigenschaftendatei für London-Clients müssen den folgenden Wert haben:

```
preferZones=London
```

Diese Eigenschaft weist jeden einzelnen Client an, Transaktionen an seine lokale Zone weiterzuleiten, sofern dies möglich ist. Die Topologie repliziert die Daten, die in ein primäres Shard in der lokalen Zone eingefügt werden, asynchron in der fremden Zone.

Schnittstelle SessionHandle für das Routing an die lokale Zone verwenden

Die containerbezogene Verteilungsstrategie verwendet keinen Hash-basierten Algorithmus, um die Position der Schlüssel/Wert-Paare im Datengrid zu bestimmen. Sie müssen SessionHandle-Objekte verwenden, um sicherzustellen, dass Transaktionen an die richtige Position weitergeleitet werden, wenn Sie diese Verteilungsstrategie verwenden. Beim Festschreiben einer Transaktion wird ein SessionHandle-Objekt an die Sitzung gebunden, sofern noch keines definiert ist. Das SessionHandle-Objekt kann auch vor dem Festschreiben der Transaktion mit der Methode Session.getSessionHandle an die Sitzung gebunden werden. Das folgende

Code-Snippet zeigt, wie ein SessionHandle-Objekt vor der Festschreibung der Transaktion an das Session-Objekt gebunden wird:

```
Session ogSession = objectGrid.getSession();

// SessionHandle-Objekt binden
SessionHandle sessionHandle = ogSession.getSessionHandle();

ogSession.begin();
ObjectMap map = ogSession.getMap("planet");
map.insert("planet1", "mercury");

// Transaktion wird an die vom SessionHandle-Objekt angegebene Partition weitergeleitet
ogSession.commit();
```

Angenommen, der vorherige Code wird in einem Client im Rechenzentrum Chicago ausgeführt. Das Attribut **preferZones** wird für diesen Client auf `Chicago` gesetzt. Deshalb leitet Ihre Implementierung Transaktionen an eine der primären Partitionen in der Zone "Chicago" weiter: Partition 0, 1, 2, 6, 7 oder 8.

Das SessionHandle-Objekt stellt einen Pfad zurück zu der Partition dar, in der diese festgeschriebenen Daten gespeichert sind. Das SessionHandle-Objekt muss wiederverwendet oder wiederhergestellt und im Session-Objekt gesetzt werden, um zu der Partition mit den festgeschriebenen Daten zurückzugelangen.

```
ogSession.setSessionHandle(sessionHandle);
ogSession.begin();

// Zurückgegebener Wert ist "mercury "
String value = map.get("planet1");
ogSession.commit();
```

Die Transaktion in diesem Code verwendet das SessionHandle-Objekt, das während der Einfügetransaktion erstellt wurde, wieder. Anschließend leitet die get-Transaktion Anforderungen an die Partition weiter, die die eingefügten Daten enthält. Ohne das SessionHandle-Objekt kann die Transaktion die eingefügten Daten nicht abrufen.

Auswirkung von Container- und Zonenfehlern auf das Routing an bevorzugte Zonen

Im Allgemeinen leitet ein Client mit definierter Eigenschaft **preferZones** alle Transaktionen an die angegebenen Zonen weiter. Der Verlust eines Container führt jedoch zur Hochstufung eines Replikat-Shards in ein primäres Shard. Ein Client, der Anforderungen zuvor an Partitionen in der lokalen Zone weitergeleitet hat, muss zuvor eingegebene Daten aus der fernen Zone abrufen.

Stellen Sie sich das folgende Szenario vor: Ein Container in der Zone "Chicago" geht verloren. Dieser Container enthält die primären Shards für die Partitionen 0, 1 und 2. Die neuen primären Shards für diese Partitionen werden dann an die Zone "London" verteilt, weil die Zone "London" die Replikate für diese Partitionen enthält.

Jeder Chicago-Client, der ein SessionHandle-Objekt verwendet, das auf eine der übernommenen Partitionen verweist, leitet Anforderungen jetzt an die Zone "London" um. Chicago-Clients, die neue SessionHandle-Objekte verwenden, leiten Anforderungen an Chicago-basierte primäre Shards weiter.

Wenn die gesamte Zone "Chicago" verloren geht, werden alle Replikate in der Zone "London" zu primären Shards. In diesem Szenario leiten alle Chicago-Clients ihre Transaktionen an die Zone "London" weiter.

Zonen für Container-Server definieren

Zonen sind Verbünde von Container-Servern. Ein Container-Server kann nur zu einer einzigen Zone gehören. Ein Container-Server wird bei Start einer Zone zugeordnet.

Informationen zu diesem Vorgang

Sie müssen Ihre Zonen planen, bevor Sie Ihre Container-Server starten, weil Container-Server ihre Zonenzugehörigkeit beim Start definieren. Wenn Sie die Zonenzugehörigkeit eines Container-Servers ändern möchten müssen Sie den Server mit den neuen Zoneninformationen erneut starten.

Vorgehensweise

- **Zonen für eigenständige Container-Server definieren.**
  1. Verwenden Sie den Parameter **-zone** des Scripts **startOgServer**, um die Zone für alle Container im gestarteten Server anzugeben. Weitere Informationen zum Starten von Servern finden Sie unter „Script **startOgServer**" auf Seite 409.
  2. Sie können Zonennamen auch angeben, wenn Sie Container-Server mit der integrierten Server-API über das Programm starten. Weitere Informationen finden Sie im Abschnitt „Integrierte Server-API zum Starten und Stoppen von Servern verwenden" auf Seite 418.
- **Zonen für Container-Server definieren, die in WebSphere Application Server ausgeführt werden.**

  Sie können Knotengruppen verwenden, um Container-Server bestimmten Zonen zuzuordnen. Verwenden Sie die folgende Syntax, um die Knotengruppe zu benennen, die Sie einer Zone zuordnen möchten: `Replikationszone<ID>`. Wenn Sie Zonen in der Implementierungsrichtlinie definieren, müssen Sie die Zonen genauso benennen wie die Knotengruppen. Der Knotengruppenname und der Zonenname in der XML-Deskriptordatei der Implementierungsrichtlinie müssen identisch sein.

  Wichtig: WebSphere Application Server verhindert die Zuordnung von Knoten zu mehreren Knotengruppen Knotengruppen nicht. Da Container-Server nur zu einer einzigen Zone gehören können, müssen Sie sicherstellen, dass Ihre Knoten zu exakt einer ReplicationZone-Knotengruppe gehören.

  Teilen Sie Ihre vier Knoten beispielsweise in zwei Zonen ein, A und B.

  1. Konfigurieren Sie vier Knoten: node1, node2, node3 und node4 mit jeweils zwei Servern.
  2. Erstellen Sie eine Knotengruppe mit dem Namen "ReplicationZoneA" und eine Knotengruppe mit dem Namen "ReplicationZoneB".
  3. Fügen Sie die Knoten "node1" und "node2" der Knotengruppe "ReplicationZoneA" und die Knoten "node3" und "node4" der Knotengruppe "ReplicationZoneB" hinzu.
  4. Definieren Sie ReplicationZoneA und ReplicationZoneB in der XML-Deskriptordatei Ihrer Implementierungsrichtlinie. Ein Beispiel finden Sie im Artikel „Beispiel: Zonen in einer Umgebung von WebSphere Application Server" auf Seite 257.
  5. Wenn die Server auf den Knoten "node1" und "node2" gestartet werden, werden diese ReplicationZoneA bzw. Zone A in der Konfiguration von WebSphere eXtreme Scale zugeordnet. Die Server auf den Knoten "node3" und "node4" werden ReplicationZoneB bzw. Zone B in der Konfiguration von WebSphere eXtreme Scale zugeordnet.

Beispiel: Zonendefinitionen in der XML-Implementierungsrichtliniendeskriptordatei

Sie können Zonen und Zonenregeln mit der XML-Implementierungsrichtliniendeskriptordatei definieren.

Beispiel: Primäre Shards und Replikat-Shards in verschiedenen Zonen

In diesem Beispiel werden primäre Shards einer Zone zugeordnet und Replikat-Shards einer anderen Zone, wobei es nur ein einziges asynchrones Replikat gibt. Alle primären Shards werden in der Zone DC1 gestartet. Replikat-Shards werden in der Zone DC2 gestartet.

```
<?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>
```

Es ist ein asynchrones Replikat im Element "mapSet" von ms1 definiert. Deshalb sind zwei Shards für jede Partition vorhanden: ein primäres Shard und ein asynchrones Replikat. Im Element "zoneMetadata" ist ein Element "shardMapping" für jedes Shard definiert: P für das primäre Shards und DC1 für das asynchrone Replikat. Das Attribut "primaryRule" definiert die für die primären Shards festgelegte Zone (Zone DC1), und diese Regel muss für die Verteilung primärer Shards verwendet werden. Asynchrone Replikate werden an die Zone DC2 verteilt.

Wenn die Zone DC2 jedoch verloren geht, sind die Replikat-Shards nicht mehr verfügbar. Der Verlust oder Ausfall eines Container-Servers in der Zone DC1 kann zu einem Datenverlust führen, obwohl ein Replikat angegeben wurde.

Zur Berücksichtigung dieser Möglichkeit können Sie entweder eine Zone oder ein Replikat hinzufügen, wie in den folgenden Abschnitten beschrieben wird.

Beispiel: Zone hinzufügen und Shards einheitenübergreifended speichern

Der folgende Code konfiguriert eine neue 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>
```

Es wurden insgesamt drei Zonen in diesem Code definiert: A, B und C. Anstelle separater Zonenregeln für primäre Shards und Replikat-Shards wird eine gemeinsame Zonenregel mit dem Namen "stripeRule" definiert. Diese Regel enthält alle Zonen und das Attribut "exclusivePlacement", das auf `true` gesetzt ist. Die eXtreme-Scale-Verteilungsrichtlinie stellt sicher, dass primäre Shards und Replikat-Shards in separaten Zonen enthalten sind. Diese einheitenübergreifende Verteilung bewirkt, dass primäre Shards und Replikat-Shards auf beide Zonen verteilt werden, um dieser Richtlinie zu entsprechen. Durch das Hinzufügen eine dritten Zone C wird sichergestellt, dass der Verlust einer Zone nicht zu einem Datenverlust führt und weiterhin primäre Shards und Replikat-Shards für jede Partititon verfügbar sind. Ein Zonenausfall führt entweder zum Verlust des primären Shards oder zum Verlust des Replikat-Shards, aber nicht zum Verlust beider Shards. Jedes Shard, das verloren geht, wird über das verbleibende Shard in eine verbleidenden Zone ersetzt, indem es in die andere verbleibende Zone gestellt wird.

Beispiel: Replikat hinzufügen und mehrere Rechenzentren definieren

Das klassische Szenario mit zwei Rechenzentren hat Hochgeschwindigkeitsnetze mit geringen Latenzzeiten in jedem Rechenzentrum, aber hohen Latenzzeiten zwischen den Rechenzentren. Synchrone Replikate werden in jedem Rechenzentrum verwendet, in dem die geringen Latenzzeiten die Auswirkungen der Replikation auf die Antwortzeiten minimiert. Asynchrone Replikation wird zwischen Rechenzentren verwendet, so dass das Netz mit den hohen Latenzzeiten keine Auswirkungen auf die Antwortzeit hat.

```
<?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>
```

Das primäre Shard und das synchrone Replikat-Shard verwenden beide die Zonenregel "primarySync" mit der Einstellung `false` für das Attribut "exclusivePlacement" Da das Attribut "exclusivePlacement" auf "false" gesetzt ist, wird eine Konfiguration erstellt, in der die primären Shards und die synchronen Replikat-Shards jeder Partition auf dieselbe Zone verteilt werden. Das asynchrone Replikat-Shard verwendet eine zweite Zonenregel mit nahezu denselben Zonen wie die Zonenregel "primarySync". Das asynchrone Replikat-Shard verwendet jedoch das Attribut "exclusivePlacement" mit dem Wert `true`. Wenn das Attribut "exclusivePlacement" auf "true" gesetzt ist, kann ein Shard nicht zusammen mit einem anderen Shard aus derselben Partition in derselben Zone platziert werden. Deshalb wird das asynchrone Replikat-Shard nicht in derselben Zone platziert wie das primäre Shard bzw. das synchrone Replikat-Shard. Es gibt drei Shards pro Partition in diesem MapSet: ein primäres Shard mit eínem synchronen und einem asynchronen Replikat-Shard. Deshalb gibt es drei Elemente "shardMapping", eines für jedes Shard.

Wenn eine Zone verloren geht, gehen alle asynchronen Replikate verloren und werden nicht neu generiert, weil sie keine separate Zone haben. Wenn das primäre Shard und das Replikat-Shard verloren gehen, wird das verbleibende asynchrone Replikat-Shard in ein primäres Shard hochgestuft, und es wird ein neues synchrones Replikat-Shard in der Zone erstellt. Die primären Shards und die Replikat-Shards werden einheitenübergreifend auf jede Zone verteilt.

Bei der exklusiven Verteilung hat jedes Shard eine eigene Zone: Sie müssen genügend Zonen für alle Shards haben, die Sie in jeweils eigenen Zonen platzieren möchten. Wenn eine Regel nur eine einzige Zone enthält, kann nur ein einziges Shard in der Zone platziert werden. Bei zwei Zonen können bis zu zwei Shards in der Zone platziert werden.

Beispiel: Zonen in einer Umgebung von WebSphere Application Server

Der folgende Code konfiguriert eine neue 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>
```

Für dieses Beispiel wurden drei Knotengruppen in der Umgebung von WebSphere Application Server definiert: ReplicationZoneA, ReplicationZoneB und ReplicationZoneC. Der Knotengruppenname und der Zonenname in der XML-Deskriptordatei der Implementierungsrichtlinie müssen identisch sein und den Text `ReplicationZone<identifier>` enthalten. Diese Datei definiert eine ähnliche Konfiguration wie im Beispiel mit der einheitenübergreifenden Speicherung von Shards,

zeigt aber die erforderliche Benennung in einer Konfiguration von WebSphere Application Server.

Zoneninformationen mit dem Dienstprogramm `xscmd` anzeigen

Sie können das Beispieldienstprogramm **xscmd** verwenden, um Informationen zur aktuellen Zonenimplementierung, einschließlich Daten zur Shard-Verteilung, anzuzeigen.

Vorbereitende Schritte

- Implementieren Sie ein verteiltes Datengrid mit mehreren Rechenzentren. Weitere Informationen finden Sie unter „Routing an bevorzugte Zonen" auf Seite 250.

Informationen zu diesem Vorgang

Sie können Informationen zu Ihrer Konfiguration in Bezug auf Zoneneinstellungen bestimmen, indem Sie das mit dem Produkt bereitgestellte Dienstprogramm **xscmd** verwenden.

Vorgehensweise

Dienstprogramm **xscmd** verwenden, um Informationen zu den Daten-Shards zu bestimmen. Führen Sie den folgenden Befehl aus:

```
xscmd -c showPlacement -z Zonenname
```

Beispiel

Sie können auch ein einfacheres Szenario ausführen, indem Sie das Einführungsbeispiel verwenden: *WXS-Installationsstammverzeichnis*`/ObjectGrid/gettingstarted`. Weitere Informationen finden Sie unter „Lernprogramm: Einführung in WebSphere eXtreme Scale" auf Seite 1.

1. Starten Sie einen Katalogserver:

   ```
   runcat.bat
   ```

2. Bestimmen Sie die erforderliche Anzahl an Replikaten, Zonenregeln, Container und andere Einstellungen mit dem folgenden Befehl: `startOgServer.bat serverA0 -objectgridFile xml\objectgrid.xml -deploymentPolicyFile xml\deployment.xml -zone zoneA`

3. Sie können Containerprozesse stoppen, um einen Ausfall im Datengrid zu simulieren: `stopOgServer.bat serverA0,serverA1,serverB0 -catalogServiceEndPoints localhost:2809`.

   Wenn der Server, der das letzte Shard einer Partition enthält, gestoppt wird, ordnet eXtreme Scale ein neues primäres Shard zu. Sie können prüfen, ob ein Datenverlust eingetreten ist:

   - Das Script **runclient** fügt Elemente in Ihr Datengrid ein und liest Elemente im Datengrid.
   - Der Befehl **xscmd -c showMapSizes** zeigt die Anzahl der Elemente im Datengrid an.

4. Aktive Container-Server mit dem folgenden Befehl anzeigen:

   ```
   xscmd -c showPlacement -z Zonenname
   ```

Katalog- und Container-Server konfigurieren

WebSphere eXtreme Scale hat zwei Typen von Servern: Katalogserver und Container-Server. Katalogserver steuern die Verteilung von Shards und erkennen und überwachen die Container-Server. Mehrere Katalogserver bilden den Katalogservice. Ein Container-Server ist eine Java Virtual Machine (JVM), die die Anwendungsdaten für das Datengrid speichert.

Informationen zu diesem Vorgang

Katalog- und Container-Server können in Prozessen von WebSphere Application Server als eigenständige Java-SE-Prozesse oder durch Integration der Server in Java-SE-Anwendungen gestartet werden. Wie Sie die Katalog- und Container-Server konfigurieren, richtet sich nach Ihrer Topologie.

Katalogserver

- **Eigenständige Katalogserver:**

  Konfigurieren Sie eigenständige Katalogserver mit einer Servereigenschaftendatei. Steuern Sie den Lebenszyklus eines Katalogservers mit den Scripts **startOgServer** und **stopOgServer** oder durch Verwendung der integrierten Server-API.
- **Katalogserver, die in WebSphere Application Server gestartet werden:**

  Konfigurieren Sie Katalogserver, die in WebSphere Application Server ausgeführt werden, mit der Administrationskonsole von WebSphere Application Server, mit Verwaltungsaufgaben und der Servereigenschaftendatei. Der Lebenszyklus des Servers wird durch den Prozesslebenszyklus in WebSphere Application Server gesteuert. Wenn Prozesse in WebSphere Application Server gestartet oder gestoppt werden, werden auch die Katalogserver, die in diesen Prozessen ausgeführt werden, gestartet bzw. gestoppt.

Container-Server

- **Eigenständige Container-Server**:

  Konfigurieren Sie eigenständige Container-Server mit einer Servereigenschaftendatei und einer XML-Implementierungsrichtliniendatei. Steuern Sie den Lebenszyklus eines Container-Servers mit den Scripts **startOgServer** und **stopOgServer** oder durch Verwendung der integrierten Server-API.
- **Container-Server, die in WebSphere Application Server gestartet werden:**

  Konfigurieren Sie Container-Server in WebSphere Application Server mit einer Servereigenschaftendatei und einer XML-Implementierungsrichtliniendatei, die in ein Java-EE-Anwendungsmodul integriert sind. Der Lebenszyklus der Container-Server wird von der Anwendung gesteuert. Container-Server werden mit der Anwendung gestartet und gestoppt.

Verwenden Sie die folgenden Artikel, um Ihre Katalog- und Container-Server zu konfigurieren:

Bewährte Verfahren: Clustering des Katalogservice mit Katalogservicedomänen

Wenn Sie den Katalogservice verwenden, sind mindestens zwei Katalogserver erforderlich, um einen Single Point of Failure zu vermeiden. Je nach Anzahl der Knoten in Ihrer Umgebung können Sie verschiedene Konfigurationen erstellen, um sicherzustellen, dass immer mindestens zwei Katalogserver aktiv sind.

Anzahl der Katalogserver

Das bewährte Verfahren zur Vermeidung eines Single Point of Failure für Ihre Katalogservicedomäne ist das Starten von mindestens drei Katalogservern auf drei verschiedenen Knoten.

Wenn Sie nur zwei Knoten verwenden, konfigurieren Sie zwei Katalogserver auf jedem der beiden Knoten für insgesamt vier Katalogserverprozesse. Die Erstellung dieser Konfiguration stellt sicher, dass in dem Fall, dass nur ein einziger Knoten gestartet wird, die erforderlichen beiden Katalogserver aktiv sind. Sie müssen mindestens zwei Katalogserver gleichzeitig starten. Wenn Katalogserver gestartet werden, suchen sie nach anderen Katalogservern in der Konfiguration und werden erst dann erfolgreich gestartet, wenn mindestens ein weiterer Katalogserver gefunden wird.

Beispiel: Vier Katalogserver auf zwei Knoten in einer eigenständigen Umgebung starten

Das folgende Script startet die Katalogserver "cs0" und "cs1" auf dem Knoten "host1" und die Katalogserver "cs2" und "cs3" auf dem Knoten "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
```

Hinweis: Sie müssen die Option **-listenerPort** verwenden, weil die Katalogserver, die auf einem Knoten ausgeführt werden, jeweils eine eindeutige Portnummer erfordern.

Beispiel: Mehrere Katalogserver in einer Umgebung von WebSphere Application Server starten

Katalogserver werden in einer Umgebung von WebSphere Application Server automatisch gestartet. Sie können mehrere zu startende Katalogserver definieren, indem Sie eine Katalogservicedomäne erstellen. Nachdem Sie mehrere Endpunkte in der Katalogservicedomäne angegeben haben, starten Sie die eingeschlossenen Anwendungsserver erneut, damit die Katalogserver parallel gestartet werden.

- **WebSphere Application Server Network Deployment**: Sie können mehrere vorhandene Anwendungsserver aus der Zelle als Member für Ihre Katalogservicedomäne auswählen.
- **WebSphere Application Server** (Basis): Sie können den Katalogservice auf mehreren eigenständigen Knoten starten. Wenn Sie mit Profile Management Tool mehrere Profile in demselben Installationsimage definieren, können Sie eine Reihe eigenständiger Knoten erstellen, denen jeweils ein eindeutiger Port zugeordnet wird. Definieren Sie in jedem Anwendungsserver die Katalogservicedomäne. Sie können alle weiteren Anwendungsserver angeben, indem Sie der Konfiguration ferne Server hinzufügen. Nachdem Sie diese Konfiguration in allen eigen-

ständigen Servern erstellt haben, können Sie die Basisanwendungsserver parallel starten, indem Sie das Script **startServer** ausführen oder einen Windows-Dienst verwenden, um die Server zu starten.

Einstellung für das Intervall der Überwachungssignale für Failover-Erkennung optimieren

Sie können das Intervall, in dem das System nach ausgefallenen Servern sucht, mit der Einstellung für das Intervall der Überwachungssignale konfigurieren.

Informationen zu diesem Vorgang

Die Konfiguration des Failovers variiert je nach Typ der verwendeten Umgebung. Wenn Sie eine eigenständige Umgebung verwenden, können Sie das Failover über die Befehlszeile konfigurieren. Wenn Sie eine Umgebung mit WebSphere Application Server Network Deployment verwenden, müssen Sie das Failover über die Administrationskonsole von WebSphere Application Server Network Deployment konfigurieren.

Vorgehensweise

- Failover für eigenständige Umgebungen konfigurieren

  Sie können die Intervalle der Überwachungssignale über die Befehlszeile mit dem Parameter **-heartbeat** in der Scriptdatei **startOgServer** konfigurieren. Setzen Sie den Parameter auf einen der folgenden Werte:

  Tabelle 13. Intervall der Überwachungssignale

Wert	Aktion	Beschreibung
0	Typisch (Standard-einstellung)	Failover werden gewöhnlich innerhalb von 30 Sekunden erkannt.
-1	Aggressiv	Failover werden gewöhnlich innerhalb von 5 Sekunden erkannt.
1	Gelockert	Failover werden gewöhnlich innerhalb von 180 Sekunden erkannt.

  Ein aggressives Intervall der Überwachungssignale kann hilfreich sein, wenn die Prozesse und das Netz stabil sind. Wenn das Netz oder die Prozesse nicht optimal konfiguriert sind, können Überwachungssignale verpasst werden, was zu einer falschen Fehlererkennung führen kann.

- Failover für Umgebungen mit WebSphere Application Server konfigurieren

  Sie können WebSphere Application Server Network Deployment Version 6.0.2 und höher so konfigurieren, dass ein schnelles Failover von WebSphere eXtreme Scale unterstützt wird. Die Standard-Failover-Zeit für permanente Fehler sind 200 Sekunden. Ein permanenter Fehler ist ein physischer Computer- oder Serverabsturz, das Ziehen des Netzkabels oder ein Betriebssystemfehler. Bei Fehlern aufgrund von Prozessabstürzen oder temporären Fehlern findet das Failover gewöhnlich in weniger als einer Sekunde statt. Die Fehlererkennung für temporäre Fehler findet statt, wenn die Netz-Sockets des inaktiven Prozesses für den Server, in dem der Prozess ausgeführt wird, automatisch vom Betriebssystem geschlossen werden.

  Überwachungssignalkonfiguration für Stammgruppen

  Wenn WebSphere eXtreme Scale in einem Prozess von WebSphere Application Server ausgeführt wird, werden die Failover-Merkmale aus den Stammgruppeneinstellungen des Anwendungsservers übernommen. In den folgenden Abschnit-

ten wird beschrieben, wie Sie die Überwachungssignaleinstellungen der Stammgruppe für verschiedene Versionen von WebSphere Application Server Network Deployment konfigurieren:

– **Stammgruppeneinstellungen für WebSphere Application Server Network Deployment Version 6.x und 7.x aktualisieren**

  Geben Sie das Intervall der Überwachungssignale in WebSphere Application Server Version 6.0 bis Version 6.1.0.12 in Sekunden und ab Version 6.1.0.13 in Millisekunden an. Außerdem müssen Sie die Anzahl verpasster Überwachungssignale angeben. Dieser Wert gibt an, wie viele Überwachungssignale verpasst werden können, bevor eine Peer-JVM als ausgefallen betrachtet wird. Die Erkennungszeit für permanente Fehler entspricht in etwa dem Produkt aus Intervall der Überwachungssignale und Anzahl verpasster Überwachungssignale.

  Diese Eigenschaften werden mit Hilfe von angepassten Eigenschaften in der Stammgruppe über die WebSphere-Administrationskonsole angegeben. Einzelheiten zur Konfiguration finden Sie im Abschnitt Angepasste Eigenschaften der Stammgruppe. Diese Eigenschaften müssen für alle Stammgruppen angegeben werden, die von Anwendungen verwendet werden:

  - Das Intervall der Überwachungssignale wird mit der angepassten Eigenschaft IBM_CS_FD_PERIOD_SEC (in Sekunden) bzw. der angepassten Eigenschaft IBM_CS_FD_PERIOD_MILLIS (in Millisekunden) (erfordert Version 6.1.0.13 oder höher) angegeben.
  - Die Anzahl verpasster Überwachungssignale wird mit der angepassten Eigenschaft IBM_CS_FD_CONSECUTIVE_MISSED angegeben.

  Der Standardwert für die Eigenschaft IBM_CS_FD_PERIOD_SEC ist 20, und der Standardwert für die Eigenschaft IBM_CS_FD_CONSECUTIVE_MISSED ist 10. Wenn Sie die Eigenschaft IBM_CS_FD_PERIOD_MILLIS angeben, überschreibt diese jede definierte angepasste Eigenschaft IBM_CS_FD_PERIOD_SEC. Die Werte dieser Eigenschaften sind positive ganze Zahlen.

  Verwenden Sie die folgenden Einstellungen, um eine Erkennungszeit von 1500 ms für Server der WebSphere Application Server Network Deployment Version 6.x zu erzielen:

  - Setzen Sie `IBM_CS_FD_PERIOD_MILLIS = 750` (WebSphere Application Server Network Deployment Version 6.1.0.13 und höher).
  - Setzen Sie `IBM_CS_FD_CONSECUTIVE_MISSED = 2`.

– **Stammgruppeneinstellungen für WebSphere Application Server Network Deployment Version 7.0 aktualisieren**

  WebSphere Application Server Network Deployment Version 7.0 stellt zwei Stammgruppeneinstellungen bereit, die angepasst werden können, um die Failover-Erkennungszeit zu erhöhen oder zu verringern:

  - **Übertragungsintervall für Überwachungssignale.** Der Standardwert sind 30.000 Millisekunden.
  - **Überwachungssignalzeitlimit.** Der Standardwert sind 180.000 Millisekunden.

  Weitere Einzelheiten zum Ändern dieser Einstellungen finden Sie im Information Center von WebSphere Application Server Network Deployment unter "Einstellungen für die Erkennung und Fehlererkennung".

  Verwenden Sie die folgenden Einstellungen, um eine Fehlererkennungszeit von 1500 ms für Server der WebSphere Application Server Network Deployment Version 7 zu erzielen:

  - Setzen Sie das Übertragungsintervall für Überwachungssignale auf 750 Millisekunden.

- Setzen Sie das Überwachungssignalzeitlimit auf 1500 Millisekunden.

Nächste Schritte

Wenn Sie diese Einstellungen ändern, um kürzere Failover-Zeiten anzugeben, müssen verschiedene Probleme bei der Systemoptimierung beachtet werden. Java ist keine Echtzeitumgebung. Es ist möglich, dass Threads verzögert werden, wenn die JVM lange Garbage-Collection-Zeiten verzeichnet. Threads können auch verzögert werden, wenn die Maschine, auf der die JVM ausgeführt wird, unter hoher Last steht (durch die JVM selbst oder durch andere Prozesse, die auf der Maschine ausgeführt werden). Wenn Threads verzögert werden, werden Überwachungssignale möglicherweise nicht rechtzeitig gesendet. Im schlimmsten Fall werden die durch die erforderliche Failover-Zeit verzögert. Wenn Threads verzögert werden, treten falsche Fehlererkennungen auf. Das System muss optimiert und dimensioniert werden, um sicherzustellen, dass falsche Fehlererkennungen in der Produktionsumgebung nicht auftreten. Dies kann am Zuverlässigsten durch angemessene Lasttests sichergestellt werden.

Anmerkung: Die aktuelle Version von eXtreme Scale unterstützt WebSphere Real Time.

WebSphere eXtreme Scale mit WebSphere Application Server konfigurieren

Sie können Katalogservice- und Container-Serverprozesse in WebSphere Application Server ausführen. Der Prozess zum Konfigurieren dieser Server unterscheidet sich von dem in einer eigenständigen Konfiguration. Der Katalogservice kann automatisch in Servern oder im Deployment Manager von WebSphere Application Server gestartet werden. Der Containerprozess wird gestartet, wenn eine eXtreme-Scale-Anwendung in der Umgebung von WebSphere Application Server implementiert und gestartet wird.

Informationen zu diesem Vorgang

Achtung: Fassen Sie Ihre Container-Server in einer Produktionsumgebung nicht durch Kollokation mit Katalogservern zusammen. Führen Sie den Katalogservice in mehreren Node-Agent-Prozessen oder in einem Anwendungsserver ohne eine eXtreme-Scale-Anwendung aus.

Katalogservice in WebSphere Application Server konfigurieren

Katalogserviceprozesse können in WebSphere Application Server ausgeführt werden. Der Serverlebenszyklus in WebSphere Application Server bestimmt, wann der Katalogservice gestartet und gestoppt wird.

Vorgehensweise

1. Wählen Sie einen oder mehrere Prozesse von WebSphere Application Server aus, die Sie mit dem Profil von WebSphere eXtreme Scale erweitern möchten. Weitere Informationen finden Sie im Abschnitt „Profile für WebSphere eXtreme Scale erstellen und erweitern" auf Seite 189. Wenn Sie möchten, dass der Katalogservice automatisch im Deployment Manager in WebSphere Application Server Network Deployment gestartet wird, installieren Sie WebSphere eXtreme Scale auf dem Deployment-Manager-Knoten, und erweitern Sie das Deployment-Manager-Profil.
2. Servereigenschaftendatei für die Prozesse von WebSphere Application Server konfigurieren und dem Klassenpfad für den Knoten hinzufügen. Weitere Informationen finden Sie unter Servereigenschaftendatei.

3. Konfigurieren Sie eine Katalogservicedomäne. Die Katalogservicedomäne ist eine Gruppe von Katalogservern in Ihrer Umgebung. Weitere Informationen finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen".
4. Prozesse von WebSphere Application Server starten, in denen die Katalogserver ausgeführt werden. Weitere Informationen finden Sie im Artikel „Server in einer Umgebung von WebSphere Application Server starten und stoppen" auf Seite 417.

Katalogservicedomänen in WebSphere Application Server erstellen:

Katalogservicedomänen definieren eine Gruppe von Katalogservern, die die Verteilung von Shards verwalten und die Vitalität (ordnungsgemäßer Betrieb) der Container-Server in Ihrem Datengrid überwachen.

Vorbereitende Schritte

- WebSphere eXtreme Scale in WebSphere Application Server installieren. Weitere Informationen finden Sie unter „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren" auf Seite 167.

Informationen zu diesem Vorgang

Durch die Erstellung einer Katalogservicedomäne definieren Sie eine hoch verfügbare Sammlung von Katalogservern.

Diese Katalogserver können in WebSphere Application Server in einer einzigen Zelle oder in einer Stammgruppe ausgeführt werden. Die Katalogservicedomäne kann auch eine ferne Gruppe von Servern definieren, die in verschiedenen Java-SE-Prozessoren oder in anderen Zellen von WebSphere Application Server ausgeführt werden.

Für Katalogserver, die in vorhandenen Servern in der Zelle ausgeführt werden: Wenn Sie eine Katalogservicedomäne definieren, die Katalogserver in den Anwendungsservern innerhalb der Zelle platziert, werden die Stammgruppenmechanismen von WebSphere Application Server verwendet. Der Katalogservice wird automatisch in den Anwendungsservern in der Zelle gestartet. Deshalb können die Member einer Katalogservicedomäne nicht die Grenzen einer Stammgruppe überschreiten, und deshalb kann sich eine Katalogservicedomäne nicht über mehrere Zellen erstrecken. Container-Server und -Clients von WebSphere eXtreme Scale können jedoch über mehrere Zellen verteilt sein, wenn eine Verbindung zu einem Katalogserver über Zellengrenzen hinweg hergestellt wird, wie z. B. zu einer eigenständigen Katalogservicedomäne oder einer Katalogservicedomäne, die in eine andere Zelle eingebettet ist.

Für ferne Katalogserver: Sie können Container und Clients von WebSphere eXtreme Scale mit einer Katalogservicedomäne verbinden, die in einer anderen Zelle von WebSphere Application Server oder als eigenständiger Prozess ausgeführt wird. Da über Fernzugriff konfigurierte Katalogserver in der Zelle nicht automatisch gestartet werden, müssen Sie diese manuell starten. Wenn Sie eine ferne Katalogservicedomäne konfigurieren, muss der Domänenname mit dem Domänennamen übereinstimmen, den Sie beim Starten der fernen Katalogserver angegeben haben. Standardmäßig wird als Katalogservicedomänenname für eigenständige Katalogserver `DefaultDomain` verwendet. Geben Sie einen Katalogservicedomänennamen mit dem Parameter **`-domain`** des Befehls **`startOgServer`**, mit einer Servereigenschaftendatei oder mit der integrierten Server-API an. Sie müssen jeden fernen Katalogserverprozess in der fernen Domäne mit demselben Domänennamen star-

ten. Weitere Informationen zum Starten von Katalogservern finden Sie unter „Eigenständigen Katalogservice starten” auf Seite 403.

Achtung: Fassen Sie die Katalogservices nicht durch Kollokation mit Container-Servern von WebSphere eXtreme Scale in einer Produktionsumgebung zusammen. Schließen Sie den Katalogservice in mehrere Node-Agent-Prozess oder in einen Anwendungsserver ein, der keine Anwendung von WebSphere eXtreme Scale enthält.

Vorgehensweise

1. Erstellen Sie die Katalogservicedomäne.
   a. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen** > **Neu**.
   b. Definieren Sie einen Namen, einen Standardwert und JMX-Authentifizierungsnachweise für Ihre Katalogservicedomäne. Wenn Sie ferne Endpunkte für die Katalogservicedomäne konfigurieren, muss der Name der Katalogservicedomäne mit dem Namen der Katalogservicedomäne übereinstimmen, den Sie beim Starten der Katalogserver angeben.
   c. Fügen Sie Katalogserviceendpunkte hinzu. Sie können vorhandene Anwendungsserver auswählen oder ferne Server hinzufügen, in einen ein Katalogservice ausgeführt wird.
2. Testen Sie die Verbindung zu den Katalogservern in Ihrer Katalogservicedomäne. Für vorhandene Anwendungsserver werden Katalogserver gestartet, wenn der zugehörige Anwendungsserver gestartet wird. Für ferne Anwendungsserver müssen Sie die Server manuell mit dem Befehl **startOgServer** oder mit der integrierten Server-API starten.
   a. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen**.
   b. Wählen Sie die Katalogservicedomäne aus, die Sie testen möchten, und klicken Sie anschließend auf **Verbindung testen**. Wenn Sie auf diese Schaltfläche klicken, werden alle definierten Endpunkte der Katalogservicedomäne nacheinander abgefragt. Wenn ein Endpunkt verfügbar ist, wird eine Nachricht zurückgegeben, in der gemeldet wird, dass die Verbindung zur Katalogservicedomäne erfolgreich hergestellt werden konnte.

Verwaltungs-Tasks für Katalogservicedomäne:

Sie können die Scripting-Sprachen Jacl und Jython verwenden, um Katalogservicedomänen in Ihrer Konfiguration von WebSphere Application Server zu verwalten.

Voraussetzungen

WebSphere eXtreme Scale Client muss in der Umgebung von WebSphere Application Server installiert sein.

Alle Verwaltungs-Tasks auflisten

Führen Sie den folgenden Befehl mit wsadmin aus, um eine Liste aller Verwaltungs-Tasks für Katalogservicedomänen abzurufen.

```
wsadmin>$AdminTask help XSDomainManagement
```

Befehle

Die Verwaltungs-Tasks für Katalogservicedomänen umfassen die folgenden Befehle:

- „createXSDomain"
- „deleteXSDomain" auf Seite 269
- „getDefaultXSDomain" auf Seite 269
- „listXSDomains" auf Seite 270
- „modifyXSDomain" auf Seite 270
- „testXSDomainConnection" auf Seite 276
- „testXSServerConnection" auf Seite 276

createXSDomain

Der Befehl "**createXSDomain**" registriert eine neue Katalogservicedomäne.

Tabelle 14. Argumente für den Befehl "createXSDomain"

Argument	Beschreibung
-name (erforderlich)	Gibt den Namen der Katalogservicedomäne an, die Sie erstellen möchten.
-default	Gibt an, ob die Katalogservicedomäne die Standarddomäne für die Zelle ist. Der Standardwert ist `true`. (Boolescher Wert: `true` oder `false`)
-properties	Gibt angepasste Eigenschaften für die Katalogservicedomäne.

Tabelle 15. Argumente für den Schritt "defineDomainServers"

Argument	Beschreibung
Name_des_Endpunkts	Gibt den Namen des Katalogserviceendpunkts an. • **Für vorhandene Anwendungsserver:** Der Name des Endpunkts muss das folgende Format haben: *Zellenname**Knotenname**Servername*. • **Für ferne Server:** Gibt den Hostnamen des fernen Servers an. Sie können denselben Namen für mehrere Endpunkte verwenden, aber die Client-Port-Werte müssen für jeden Endpunkt eindeutig sein.
angepasste_Eigenschaften	Gibt angepasste Eigenschaften für den Endpunkt der Katalogservicedomäne an. Wenn Sie keine angepassten Eigenschaften haben, verwenden Sie für dieses Argument ein Paar doppelter Anführungszeichen ("").

Tabelle 15. Argumente für den Schritt "defineDomainServers" (Forts.)

Argument	Beschreibung
Endpunktports	Gibt die Portnummern für den Endpunkt der Katalogservicedomäne an. Die Ports müssen in der folgenden Reihenfolge angegeben werden: *<Client-Port>*,*<Listener-Port>*. **Client-Port** Gibt den Port an, der für die Kommunikation zwischen den Katalogservern in der Katalogservicedomäne verwendet wird. Dieser Wert ist nur für Katalogserver erforderlich, die in Prozessen von WebSphere Application Server ausgeführt werden und kann auf jeden Port gesetzt werden, der noch nicht verwendet wird. **Listener-Port** Gibt den Port an, der für die Kommunikation mit Clients verwendet wird. Dieser Wert ist für ferne Endpunkte erforderlich und muss mit dem Wert übereinstimmen, der beim Start des Katalogservice verwendet wurde. Der Listener-Port wird von Clients und Containern für die Kommunikation mit dem Katalogservice verwendet. **Für ferne Endpunkte von WebSphere eXtreme Scale:** Definiert den ORB-Listener-Port (Object Request Broker) für Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. Für Endpunkte von WebSphere Application Server ist der Listener-Port-Wert optional, weil der Wert aus der BOOTSTRAP_ADDRESS-Portkonfiguration entnommen wird.

Tabelle 16. Argumente für den Schritt "configureClientSecurity"

Argument	Beschreibung
-securityEnabled	Gibt an, dass die Clientsicherheit für den Katalogserver aktiviert werden soll. Die Servereigenschaftendatei, die dem ausgewählten Katalogserver zugeordnet ist, muss eine entsprechende Einstellung **securityEnabled** in der Servereigenschaftendatei haben. Wenn diese Einstellungen nicht übereinstimmen, tritt eine Ausnahme ein. (Boolescher Wert: `true` oder `false`)

Tabelle 16. Argumente für den Schritt "configureClientSecurity" (Forts.)

Argument	Beschreibung
-credentialAuthentication (optional)	Gibt an, ob die Authentifizierung mit Berechtigungsnachweisen erzwungen oder unterstützt wird. **Never** Es wird keine Clientzertifikatsauthentifizierung erzwungen. **Required** Die Authentifizierung mit Berechtigungsnachweisen wird immer erzwungen. Wenn der Server keine Authentifizierung mit Berechtigungsnachweisen unterstützt, kann der Client keine Verbindung zum Server herstellen. **Supported** (Standardeinstellung) Die Authentifizierung mit Berechtigungsnachweisen wird nur erzwungen, wenn der Client und der Server diese Art der Authentifizierung unterstützen.
-authenticationRetryCount (optional)	Gibt an, wie oft die Authentifizierung wiederholt wird, wenn der Berechtigungsnachweis abgelaufen ist. Wenn die Authentifizierung nicht wiederholt werden soll, setzen Sie die Einstellung auf den Wert `0`. Der Standardwert ist `0`.
-credentialGeneratorClass	Gibt die Implementierungsklasse com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator an, damit der Client die Sicherheitstoken aus dem Thread abruft.
-credentialGeneratorProps	Gibt die Eigenschaften für die Implementierungsklasse CredentialGenerator an. Die Eigenschaften werden mit der Methode setProperties(String) an das Objekt gesendet. Der Eigenschaftswert für den Berechtigungsnachweisgenerator wird nur verwendet, wenn ein Wert im Feld **Generatorklasse für Berechtigungsnachweise** angegeben ist.

Rückgabewert:

Beispielsyntax für den Stapelmodus

Im Stapelmodus ist eine korrekte Formatierung des Befehlseintrags erforderlich. Sie sollten den interaktiven Modus verwenden, wenn Sie sicherstellen möchten, dass die von Ihnen eingegebenen Werte ordnungsgemäß verarbeitet werden. Wenn Sie den Stapelmodus verwenden, müssen Sie die Argumente für den Schritt **-defineDomainServers** mit einem bestimmten Bereich von Eigenschaften definieren. Dieser Eigenschaftenbereich muss im folgenden Format angegeben werden: *Name_des_Endpunkts angepasste_Eigenschaften Endpunktports*. Der Wert für *Endpunktports* ist eine Liste von Ports, die Sie in der folgenden Reihenfolge angeben müssen: *<Client-Port>,<Listener-Port>*.

- Erstellen Sie mit Jacl eine Katalogservicedomäne der fernen Endpunkte:

```
$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"}}}
```

- Erstellen Sie mit Jython (Zeichenfolge) eine Katalogservicedomäne der fernen Endpunkte:

```
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"] ]')
```

- Erstellen Sie mit Jacl eine Katalogservicedomäne vorhandener Anwendungsserverendpunkte:

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
 {{cellName/nodeName/serverName "" 1109}}}
```

Beispielsyntax für den Dialogmodus:

- Mit Jacl:

```
$AdminTask createXSDomain {-interactive}
```

- Mit Jython (Zeichenfolge):

```
AdminTask.createXSDomain ('[-interactive]')
```

deleteXSDomain

Der Befehl "**deleteXSDomain**" löscht eine Katalogservicedomäne.

Erforderliche Parameter:

-name
Gibt den Namen der zu löschenden Katalogservicedomäne an.

Rückgabewert:

Beispielsyntax für den Stapelmodus

- Mit Jacl:

```
$AdminTask deleteXSDomain {-name TestDomain }
```

- Mit Jython (Zeichenfolge):

```
AdminTask.deleteXSDomain('[-name TestDomain ]')
```

Beispielsyntax für den Dialogmodus:

- Mit Jacl:

```
$AdminTask deleteXSDomain {-interactive}
```

- Mit Jython (Zeichenfolge):

```
AdminTask.deleteXSDomain ('[-interactive]')
```

getDefaultXSDomain

Der Befehl "**getDefaultXSDomain**" gibt die Standardkatalogservicedomäne für die Zelle zurück.

Erforderliche Parameter: Ohne

Rückgabewert: Der Name der Standardkatalogservicedomäne.

Beispielsyntax für den Stapelmodus

- Mit Jacl:
  ```
  $AdminTask getDefaultXSDomain
  ```
- Mit Jython (Zeichenfolge):
  ```
  AdminTask.getDefaultXSDomain
  ```

Beispielsyntax für den Dialogmodus:
- Mit Jacl:
  ```
  $AdminTask getDefaultXSDomain {-interactive}
  ```
- Mit Jython (Zeichenfolge):
  ```
  AdminTask.getDefaultXSDomain ('[-interactive]')
  ```

listXSDomains

Der Befehl "**listXSDomains**" gibt eine Liste vorhandener Katalogservicedomänen zurück.

Erforderliche Parameter: Ohne

Rückgabewert: Eine Liste aller Katalogservicedomänen in der Zelle.

Beispielsyntax für den Stapelmodus
- Mit Jacl:
  ```
  $AdminTask listXSDomains
  ```
- Mit Jython (Zeichenfolge):
  ```
  AdminTask.listXSDomains
  ```

Beispielsyntax für den Dialogmodus:
- Mit Jacl:
  ```
  $AdminTask listXSDomains {-interactive}
  ```
- Mit Jython (Zeichenfolge):
  ```
  AdminTask.listXSDomains ('[-interactive]')
  ```

modifyXSDomain

Der Befehl "**modifyXSDomain**" ändert eine vorhandene Katalogservicedomäne.

Im Stapelmodus ist eine korrekte Formatierung des Befehlseintrags erforderlich. Sie sollten den interaktiven Modus verwenden, wenn Sie sicherstellen möchten, dass die von Ihnen eingegebenen Werte ordnungsgemäß verarbeitet werden. Wenn Sie den Stapelmodus verwenden, müssen Sie die Argumente für die Schritte **-modifyEndpoints**, **-addEndpoints** und **-removeEndpoints** mit einem bestimmten Bereich von Eigenschaften definieren. Dieser Eigenschaftenbereich muss im folgenden Format angegeben werden: *Name_des_Endpunkts Hostname angepasste_Eigenschaften Endpunktports*. Der Wert für *Endpunktports* ist eine Liste von Ports, die Sie in der folgenden Reihenfolge angeben müssen: *<Client-Port>,<Listener-Port>*.

Tabelle 17. Argumente für den Befehl "modifyXSDomain"

Argument	Beschreibung
-name (erforderlich)	Gibt den Namen der Katalogservicedomäne an, die Sie bearbeiten möchten.

Tabelle 17. Argumente für den Befehl "modifyXSDomain" (Forts.)

Argument	Beschreibung
-default	Wenn dieses Argument auf `true` gesetzt wird, ist die ausgewählte Katalogservicedomäne die Standarddomäne für die Zelle. (Boolean)
-properties	Gibt angepasste Eigenschaften für die Katalogservicedomäne.

Tabelle 18. Argumente für den Schritt "modifyEndpoints"

Argument	Beschreibung
Name_des_Endpunkts	Gibt den Namen des Katalogserviceendpunkts an. • **Für vorhandene Anwendungsserver:** Der Name des Endpunkts muss das folgende Format haben: *Zellenname**Knotenname*\ *Servername*. • **Für ferne Server:** Gibt den Hostnamen des fernen Servers an. Sie können denselben Namen für mehrere Endpunkte verwenden, aber die Listener-Port-Werte müssen für jeden Endpunkt eindeutig sein.

Tabelle 18. Argumente für den Schritt "modifyEndpoints" (Forts.)

Argument	Beschreibung
Endpunktports	Gibt die Portnummern für den Endpunkt der Katalogservicedomäne an. Die Endpunkte müssen in der folgenden Reihenfolge angegeben werden: *<Client-Port>*,*<Listener-Port>*. **Client-Port** Gibt den Port an, der für die Kommunikation zwischen den Katalogservern in der Katalogservicedomäne verwendet wird. Dieser Wert ist nur für Katalogserver erforderlich, die in Prozessen von WebSphere Application Server ausgeführt werden und kann auf jeden Port gesetzt werden, der noch nicht verwendet wird. **Listener-Port** Gibt den Port an, der für die Kommunikation mit Clients verwendet wird. Dieser Wert ist für ferne Endpunkte erforderlich und muss mit dem Wert übereinstimmen, der beim Start des Katalogservice verwendet wurde. Der Listener-Port wird von Clients und Containern für die Kommunikation mit dem Katalogservice verwendet. **Für ferne Endpunkte von WebSphere eXtreme Scale:** Definiert den ORB-Listener-Port (Object Request Broker) für Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. Für Endpunkte von WebSphere Application Server ist die Angabe des Listener-Port-Werts optional, weil der Wert aus der BOOTSTRAP_ADDRESS-Portkonfiguration entnommen wird.

Tabelle 19. Argumente für den Schritt "addEndpoints"

Argument	Beschreibung
Name_des_Endpunkts	Gibt den Namen des Katalogserviceendpunkts an. • **Für vorhandene Anwendungsserver:** Der Name des Endpunkts muss das folgende Format haben: *Zellenname**Knotenname**Servername*. • **Für ferne Server:** Gibt den Hostnamen des fernen Servers an. Sie können denselben Namen für mehrere Endpunkte verwenden, aber die Listener-Port-Werte müssen für jeden Endpunkt eindeutig sein.
angepasste_Eigenschaften	Gibt angepasste Eigenschaften für den Endpunkt der Katalogservicedomäne an. Wenn Sie keine angepassten Eigenschaften haben, verwenden Sie für dieses Argument ein Paar doppelter Anführungszeichen ("").

Tabelle 19. Argumente für den Schritt "addEndpoints" (Forts.)

Argument	Beschreibung
Endpunktports	Gibt die Portnummern für den Endpunkt der Katalogservicedomäne an.Die Endpunkte müssen in der folgenden Reihenfolge angegeben werden: *<Client-Port>*,*<Listener-Port>*. **Client-Port** Gibt den Port an, der für die Kommunikation zwischen den Katalogservern in der Katalogservicedomäne verwendet wird. Dieser Wert ist nur für Katalogserver erforderlich, die in Prozessen von WebSphere Application Server ausgeführt werden und kann auf jeden Port gesetzt werden, der noch nicht verwendet wird. **Listener-Port** Gibt den Port an, der für die Kommunikation mit Clients verwendet wird. Dieser Wert ist für ferne Endpunkte erforderlich und muss mit dem Wert übereinstimmen, der beim Start des Katalogservice verwendet wurde. Der Listener-Port wird von Clients und Containern für die Kommunikation mit dem Katalogservice verwendet. **Für ferne Endpunkte von WebSphere eXtreme Scale:** Definiert den ORB-Listener-Port (Object Request Broker) für Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. Für Endpunkte von WebSphere Application Server ist die Angabe des Listener-Port-Werts optional, weil der Wert aus der BOOTSTRAP_ADDRESS-Portkonfiguration entnommen wird.

Tabelle 20. Argumente für den Schritt "removeEndpoints"

Argument	Beschreibung
Name_des_Endpunkts	Gibt den Namen des zu löschenden Katalogserviceendpunkts an.

Tabelle 21. Argumente für den Schritt "configureClientSecurity"

Argument	Beschreibung
-securityEnabled	Gibt an, dass die Clientsicherheit für den Katalogserver aktiviert werden soll. Die Servereigenschaftendatei, die dem ausgewählten Katalogserver zugeordnet ist, muss eine entsprechende Einstellung **securityEnabled** in der Servereigenschaftendatei haben. Wenn diese Einstellungen nicht übereinstimmen, tritt eine Ausnahme ein. (Boolescher Wert: `true` oder `false`)
-credentialAuthentication (optional)	Gibt an, ob die Authentifizierung mit Berechtigungsnachweisen erzwungen oder unterstützt wird. **Nie** Es wird keine Clientzertifikatsauthentifizierung erzwungen. **Erforderlich** Die Authentifizierung mit Berechtigungsnachweisen wird immer erzwungen. Wenn der Server keine Authentifizierung mit Berechtigungsnachweisen unterstützt, kann der Client keine Verbindung zum Server herstellen. **Unterstützt** (Standardeinstellung) Die Authentifizierung mit Berechtigungsnachweisen wird nur erzwungen, wenn der Client und der Server diese Art der Authentifizierung unterstützen.
-authenticationRetryCount (optional)	Gibt an, wie oft die Authentifizierung wiederholt wird, wenn der Berechtigungsnachweis abgelaufen ist. Wenn die Authentifizierung nicht wiederholt werden soll, setzen Sie die Einstellung auf den Wert `0`. Der Standardwert ist `0`.
-credentialGeneratorClass	Gibt die Implementierungsklasse com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator an, damit der Client die Sicherheitstoken aus dem Thread abruft.
-credentialGeneratorProps	Gibt die Eigenschaften für die Implementierungsklasse CredentialGenerator an. Die Eigenschaften werden mit der Methode setProperties(String) an das Objekt gesendet. Der Eigenschaftswert für den Berechtigungsnachweisgenerator wird nur verwendet, wenn ein Wert im Feld **Generatorklasse für Berechtigungsnachweise** angegeben ist.

Rückgabewert:

Beispielsyntax für den Stapelmodus

- Mit Jacl:

```
$AdminTask modifyXSDomain {-name TestDomain -default true -modifyEndpoints
{{xhost1.ibm.com "" ,2809}} -addEndpoints {{xhost2.ibm.com "" ,2809}}}
-removeEndpoints {{xhost3.ibm.com}}}
```

- Mit Jython (Zeichenfolge):

```
AdminTask.modifyXSDomain('[-name TestDomain
-default false -modifyEndpoints [[xhost1.ibm.com "" ,2809]]
-addEndpoints [[xhost3.ibm.com "" ,2809]]
-removeEndpoints [[xhost2.ibm.com]]]')
```

- Clientsicherheitsspezifikation bei der Ausführung des Änderungsbefehls (modify) verwenden:

```
$AdminTask modifyXSDomain {-name myDomain -default false
-configureClientSecurity {-securityEnabled true -
Supported -authenticationRetryCount 1 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
  -credentialGeneratorProps "manager manager1"}}
```

Beispielsyntax für den Dialogmodus:

- Mit Jacl:

```
$AdminTask modifyXSDomain {-interactive}
```

- Mit Jython (Zeichenfolge):

```
AdminTask.modifyXSDomain ('[-interactive]')
```

testXSDomainConnection

Der Befehl "**testXSDomainConnection**" testet die Verbindung zu einer Katalogservicedomäne.

Erforderliche Parameter:

-name
Gibt den Namen der Katalogservicedomäne an, zu der die Verbindung getestet werden soll.

Optionale Parameter

-timeout
Gibt an, wie lange maximal (in Sekunden) auf die Verbindung gewartet wird.

Rückgabewert: Wenn eine Verbindung hergestellt werden kann, gibt der Befehl `true` zurück. Andernfalls werden Informationen zum Verbindungsfehler zurückgegeben.

Beispielsyntax für den Stapelmodus

- Mit Jacl:

```
$Admintask testXSDomainConnection
```

- Mit Jython (Zeichenfolge):

```
AdminTask.testXSDomainConnection
```

Beispielsyntax für den Dialogmodus:

- Mit Jacl:

```
$AdminTask testXSDomainConnection {-interactive}
```

- Mit Jython (Zeichenfolge):

```
AdminTask.testXSDomainConnection ('[-interactive]')
```

testXSServerConnection

Der Befehl "**testXSServerConnection**" testet die Verbindung zu einem Katalogserver. Dieser Befehl funktioniert für eigenständige Server und für Server, die zu einer Katalogservicedomäne gehören.

Erforderliche Parameter:

host
Gibt den Host an, auf dem sich der Katalogserver befindet.

listenerPort
Gibt den Listener-Port für den Katalogserver an.

Optionale Parameter

timeout
Gibt an, wie lange maximal (in Sekunden) auf eine Verbindung zum Katalogserver gewartet wird.

domain
Gibt den Namen einer Katalogservicedomäne an. Wenn Sie einen Wert für diesen Parameter definieren, werden die Clientsicherheitseigenschaften für die angegebene Katalogservicedomäne zum Testen der Verbindung verwendet. Andernfalls wird eine Suche durchgeführt, um die Katalogservicedomäne für den angegebenen Host und den angegebenen Listener-Port zu ermitteln. Wenn eine Katalogservicedomäne gefunden wird, werden die für die Katalogservicedomäne definierten Clientsicherheitseigenschaften zum Testen des Servers verwendet. Wird keine Katalogservicedomäne gefunden, werden während des Tests keine Clientsicherheitseigenschaften verwendet.

Rückgabewert:

Beispielsyntax für den Stapelmodus

- Mit Jacl:
```
$Admintask testXSServerConnection {-host xhost1.ibm.com -listenerPort 2809}
```
- Mit Jython (Zeichenfolge):
```
AdminTask.testXSServerConnection('[-host xshost3.ibm.com -listenerPort 2809]')
```

Beispielsyntax für den Dialogmodus:

- Mit Jacl:
```
$AdminTask testXSServerConnection {-interactive}
```
- Mit Jython (Zeichenfolge):
```
AdminTask.testXSServerConnection ('[-interactive]')
```

Katalogservicedomänen:

Verwenden Sie diese Seite, um Katalogservicedomänen zu verwalten. Katalogservicedomänen definieren eine Gruppe von Katalogservern, die die Verteilung von Shards verwalten und die Vitalität (ordnungsgemäßer Betrieb) der Container-Server in Ihrem Datengrid überwachen.

Klicken Sie zum Anzeigen dieser Seite der Administrationskonsole auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen**. Klicken Sie zum Erstellen einer neuen Katalogservicedomäne auf **Neu**. Zum Löschen einer Katalogservicedomäne wählen Sie die Katalogservicedomäne aus, die Sie entfernen möchten, und klicken Sie dann auf **Löschen**.

Verbindung testen:

Wenn Sie auf die Schaltfläche **Verbindung testen** klicken, werden alle definierten Endpunkte der Katalogservicedomäne nacheinander abgefragt. Wenn ein Endpunkt

verfügbar ist, wird eine Nachricht zurückgegeben, in der gemeldet wird, dass die Verbindung zur Katalogservicedomäne erfolgreich hergestellt werden konnte. Sie können diese Schaltfläche verwenden, um zu überprüfen, ob die Verbindungs- und Sicherheitsinformationen ordnungsgemäß konfiguriert wurden.

Standard festlegen:

Definiert die Katalogservicedomäne, die als Standarddomäne verwendet wird. Wählen Sie eine Katalogservicedomäne als Standarddomäne aus, und klicken Sie auf **Standard festlegen**. Es kann nur eine einzige Katalogservicedomäne als Standarddomäne ausgewählt werden.

Name:

Gibt den Namen für die Katalogservicedomäne an.

Standardwert:

Gibt an, welche Katalogservicedomäne in der Liste die Standarddomäne ist. Die Standardkatalogservicedomäne ist mit dem folgenden Symbol markiert: 

Einstellungen der Katalogservicedomäne:

Verwenden Sie diese Seite, um die Einstellungen für eine bestimmte Katalogservicedomäne zu verwalten. Katalogservicedomänen definieren eine Gruppe von Katalogservern, die die Verteilung von Shards verwalten und die Vitalität (ordnungsgemäßer Betrieb) der Container-Server in Ihrem Datengrid überwachen. Sie können eine Katalogservicedomäne definieren, die sich in derselben Zelle wie Ihr Deployment Manager befindet. Sie können ferne Katalogservicedomänen auch definieren, wenn Ihre Konfiguration von WebSphere eXtreme Scale in einer anderen Zelle enthalten ist oder wenn sich Ihr Datengrid aus Java-SE-Prozessen zusammensetzt.

Klicken Sie zum Anzeigen dieser Seite der Administrationskonsole auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen** > ***Name_der_Katalogservicedomäne***.

Verbindung testen:

Wenn Sie auf die Schaltfläche **Verbindung testen** klicken, werden alle definierten Endpunkte der Katalogservicedomäne nacheinander abgefragt. Wenn ein Endpunkt verfügbar ist, wird eine Nachricht zurückgegeben, in der gemeldet wird, dass die Verbindung zur Katalogservicedomäne erfolgreich hergestellt werden konnte. Sie können diese Schaltfläche verwenden, um zu überprüfen, ob die Verbindungs- und Sicherheitsinformationen ordnungsgemäß konfiguriert wurden.

Name:

Gibt den Namen der Katalogservicedomäne an.

Diese Katalogservicedomäne als Standarddomäne aktivieren, sofern keine andere Katalogservicedomäne explizit angegeben ist:

Wenn Sie dieses Kontrollkästchen auswählen, wird die ausgewählte Katalogservicedomäne zur Standardkatalogservicedomäne für die Zelle. Jedes Serverprofil in der Zelle, das mit dem Profil von WebSphere eXtreme Scale erweitert wird, gehört zur ausgewählten Katalogservicedomäne.

Bei WebSphere eXtreme Scale stellen alle eXtreme-Scale-Container, die in Java-EE-Anwendungsmodule eingebettet sind, eine Verbindung zur Standarddomäne her. Clients können mit der API ServerFactory.getServerProperties().getCatalogServiceBootstrap() eine Verbindung zur Standarddomäne herstellen, um die Katalogserviceendpunkte abzurufen, die beim Aufruf der API ObjectGridManager.connect() zu verwenden sind.

Wenn Sie die Standarddomäne ändern, sodass sie auf eine andere Gruppe von Katalogservern verweist, verweisen alle Container und Clients auf die neue Domäne, nachdem sie erneut gestartet wurden.

Katalogserver:

Gibt eine Liste mit Katalogservern an, die zu dieser Katalogservicedomäne gehören.

Klicken Sie auf **Neu**, um der Liste einen Katalogserver hinzuzufügen. Dieser Katalogserver muss bereits in der eXtreme-Scale-Konfiguration vorhanden sein. Es ist auch möglich, einen Server in der Liste zu bearbeiten oder zu löschen, indem Sie den Endpunkt auswählen und dann auf **Bearbeiten** oder **Löschen** klicken. Definieren Sie die folgenden Eigenschaften für jeden Katalogserverendpunkt:

Katalogserverendpunkt
Gibt den Namen des vorhandenen Anwendungsservers oder fernen Servers an, in dem der Katalogservice ausgeführt wird. Eine Katalogservicedomäne kann keine Mischung vorhandener Anwendungsserver und ferner Serverendpunkte enthalten.

- **Vorhandener Anwendungsserver**: Gibt den Pfad eines Anwendungsservers, Node Agent oder Deployment Manager in der Zelle an. Ein Katalogservice wird automatisch im ausgewählten Server gestartet. Treffen Sie Ihre Auswahl in der Liste der vorhandenen Anwendungsserver. Alle Anwendungsserver, die Sie in der Katalogservicedomäne definieren, müssen in derselben Stammgruppe enthalten sein.
- **Ferner Server**: Gibt den Hostnamen des fernen Katalogservers an.

  Für ferne Endpunkte von WebSphere eXtremeScale: Gibt den Hostnamen des fernen Katalogserverprozesses an. Sie müssen die fernen Server mit dem Script **`startOgServer`** oder mit der integrierten Server-API starten.

Client-Port
Gibt den Port an, der für die Kommunikation zwischen den Katalogservern in der Katalogservicedomäne verwendet wird. Dieser Wert ist für Katalogserver erforderlich, die in Prozessen von WebSphere Application Server ausgeführt werden. Sie können den Wert auf einen beliebigen Port setzen, der nicht von anderen Prozessen verwendet wird.

Listener-Port
Gibt den Port an, der für die Kommunikation mit Clients verwendet wird. Dieser Wert ist für ferne Endpunkte erforderlich und muss mit dem Wert übereinstimmen, der beim Start des Katalogservice verwendet wurde. Der Listener-Port wird von Clients und Containern für die Kommunikation mit dem Katalogservice verwendet.

Für ferne Endpunkte von WebSphere eXtreme Scale: Definiert den ORB-Listener-Port (Object Request Broker) für Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. Für Endpunkte

von WebSphere Application Server wird der Listener-Port aus der BOOTSTRAP_ADDRESS-Portkonfiguration übernommen.

Status

Tabelle 22. Status des Katalogserverendpunkts

Symbol	Begriffsbestimmung
	Unbekannt
	Gestartet
	Gestoppt

Clientsicherheitseigenschaften:

Verwenden Sie diese Seite, um die Clientsicherheit für eine Katalogservicedomäne zu konfigurieren. Diese Einstellungen gelten für alle Server in der Katalogservicedomäne. Sie können diese Eigenschaften überschreiben, indem Sie eine Datei `splicer.properties` mit der angepassten Eigenschaft "com.ibm.websphere.xs.sessionFilterProps" angeben oder die EAR-Datei der Anwendung verbinden.

Klicken Sie zum Anzeigen dieser Seite der Administrationskonsole auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen** > ***Name_der_Katalogservicedomäne*** > **Sicherheitseigenschaften des Clients**.

Clientsicherheit aktivieren:

Gibt an, dass die Clientsicherheit für den Katalogserver aktiviert werden soll. Die Servereigenschaftendatei, die dem ausgewählten Katalogserver zugeordnet ist, muss eine entsprechende Einstellung **`securityEnabled`** in der Servereigenschaftendatei haben. Wenn diese Einstellungen nicht übereinstimmen, tritt eine Ausnahme ein.

Berechtigungsnachweisauthentifizierung:

Gibt an, ob die Authentifizierung mit Berechtigungsnachweisen erzwungen oder unterstützt wird.

Nie

Es wird keine Berechtigungsnachweisauthentifizierung erzwungen.

Erforderlich

Die Authentifizierung mit Berechtigungsnachweisen wird immer erzwungen. Wenn der Server keine Authentifizierung mit Berechtigungsnachweisen unterstützt, kann der Client keine Verbindung zum Server herstellen.

Unterstützt

Die Berechtigungsnachweisauthentifizierung wird nur erzwungen, wenn der Client und der Server diese Art der Authentifizierung unterstützen.

Wiederholungszähler für Authentifizierung:

Gibt an, wie oft die Authentifizierung wiederholt wird, wenn der Berechtigungsnachweis abgelaufen ist.

Wenn die Authentifizierung nicht wiederholt werden soll, setzen Sie die Einstellung auf den Wert `0`.

Generatorklasse für Berechtigungsnachweise:

Gibt die Implementierungsklasse com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator an, damit der Client den Berechtigungsnachweis aus dem CredentialGenerator-Objekt abruft.

Sie können eine von zwei vordefinierten Berechtigungsnachweisgeneratorklassen auswählen, oder Sie können einen angepassten Berechtigungsnachweisgenerator angeben. Wenn Sie einen angepassten Berechtigungsnachweisgenerator verwenden, müssen Sie den Namen der Berechtigungsnachweisgeneratorklasse angeben.

- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- Angepasster Berechtigungsnachweisgenerator

Subjekttyp:

Gibt an, ob Sie den J2EE-Subjekttyp "caller" oder "runAs" verwenden. Sie müssen diesen Wert angeben, wenn Sie den Berechtigungsnachweisgenerator "WSTokenCredentialGenerator" auswählen.

- **runAs**: Das Subjekt enthält den Principal der J2EE-RunAs-Identität und den J2EE-RunAs-Berechtigungsnachweis.
- **caller**: Das Subjekt enthält den Principal des J2EE-Callers und den Berechtigungsnachweis des J2EE-Callers.

Benutzer-ID:

Geben Sie eine Benutzer-ID an, wenn Sie die Berechtigungsnachweisgeneratorimplementierung UserPasswordCredentialGenerator verwenden.

Kennwort:

Geben Sie ein Kennwort an, wenn Sie die Berechtigungsnachweisgeneratorimplementierung UserPasswordCredentialGenerator verwenden.

Eigenschaften des Berechtigungsnachweisgenerators:

Gibt die Eigenschaften für eine angepasste CredentialGenerator-Implementierungsklasse an. Die Eigenschaften werden mit der Methode setProperties(String) im Objekt gesetzt. Der Eigenschaftswert für den Berechtigungsnachweisgenerator wird nur verwendet, wenn ein Wert im Feld **Generatorklasse für Berechtigungsnachweise** angegeben ist.

Angepasste Eigenschaften der Katalogservicedomäne:

Sie können die Konfiguration der Katalogservicedomäne durch die Definition angepasster Eigenschaften weiter bearbeiten.

Zum Anzeigen dieser Seite der Administrationskonsole klicken Sie auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen** > **Angepasste Eigenschaften**. Wenn Sie eine neue angepasste Eigenschaft erstellen möchten, klicken Sie auf **Neu**.

Name:

Gibt den Namen der angepassten Eigenschaft für die Katalogservicedomäne an.

Wert:

Gibt einen Wert für die angepasste Eigenschaft für die Katalogservicedomäne an.

Container-Server in WebSphere Application Server konfigurieren

Konfigurieren Sie Container-Server in WebSphere Application Server mithilfe einer Servereigenschaftendatei und einer XML-Implementierungsrichtliniendatei, die in ein Java-EE-Anwendungsmodul integriert ist. Container-Server werden gestoppt und gestartet, wenn die Anwendung gestoppt und gestartet wird.

Vorbereitende Schritte

Konfigurieren Sie eine Katalogservicedomäne. Weitere Informationen finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen" auf Seite 264.

Informationen zu diesem Vorgang

Zum Erstellen von Container-Servern in WebSphere Application Server müssen Sie die XML-Konfigurationsdateien von WebSphere eXtreme Scale integrieren, um die Container-Server im Anwendungsmodul zu erstellen.

Vorgehensweise

1. Identifizieren Sie die Anwendungsserver, in denen Sie die Java-EE-Anwendung implementieren möchten, die die Container-Server-Definitionen von WebSphere eXtreme Scale enthält. Vergewissern Sie sich, dass die Zielanwendungsserverprofile mit dem Profil von WebSphere eXtreme Scale erweitert wurden. Vermeiden Sie in einer Produktionsumgebung die Kollokation von Servern, die Sie für Container-Server verwenden, mit Katalogservern. Weitere Informationen finden Sie unter „Profile für WebSphere eXtreme Scale erstellen und erweitern" auf Seite 189.
2. Konfigurieren Sie eine Servereigenschaftendatei, und fügen Sie diese dem Klassenpfad für jeden Zielanwendungsserverknoten hinzu. Weitere Informationen finden Sie unter Servereigenschaftendatei.
3. Fügen Sie die ObjectGrid-XML-Deskriptordatei und die XML-Implementierungsrichtliniendatei dem Anwendungsmodul hinzu. Weitere Informationen finden Sie unter „Anwendungen von WebSphere Application Server für den automatischen Start von Container-Servern konfigurieren".

Anwendungen von WebSphere Application Server für den automatischen Start von Container-Servern konfigurieren:

Container-Server in einer Umgebung mit WebSphere Application Server werden automatisch gestartet, wenn ein Modul gestartet wird, in dem die XML-Dateien von eXtreme Scale enthalten sind.

Vorbereitende Schritte

WebSphere Application Server und WebSphere eXtreme Scale müssen installiert und in der Lage sein, auf die Administrationskonsole von WebSphere Application Server zuzugreifen.

Informationen zu diesem Vorgang

Java-EE-Anwendungen haben komplexe Klassenladerregeln, die das Laden von Klassen erheblich komplizieren, wenn ein Datengrid mit Shards in einem Java-EE-Server verwendet wird. Eine Java-EE-Anwendung ist gewöhnlich eine einzige EAR-Datei. Die EAR-Datei enthält ein oder mehrere EJB- oder WAR-Module.

WebSphere eXtreme Scale überwacht den Start jedes Moduls und sucht nach XML-Dateien von eXtreme Scale. Wenn der Katalogservice erkennt, dass ein Modul mithilfe von XML-Dateien gestartet wird, wird der Anwendungsserver als Container-Server-JVM (Java Virtual Machine) registriert. Durch die Registrierung der Container-Server beim Katalogservice kann dieselbe Anwendung in mehreren Datengrids implementiert und trotzdem vom Katalogservice als einzelnes Datengrid behandelt werden. Der Katalogservice kümmert sich nicht um Zellen, Grids oder dynamische Grids. Ein einziges Datengrid kann, sofern erforderlich, mehrere Zellen umspannen.

Vorgehensweise

1. Packen Sie in Ihre EAR-Datei Module, die die XML-Dateien von eXtreme Scale im Ordner `META-INF` enthalten. WebSphere eXtreme Scale erkennt das Vorhandensein der Dateien `objectGrid.xml` und `objectGridDeployment.xml` im Ordner `META-INF` von EJB- und WEB-Modulen, wenn diese gestartet werden. Wenn nur eine Datei `objectGrid.xml` gefunden wird, wird davon ausgegangen, dass die JVM ein Client ist. Sind beide Dateien vorhanden. wird davon ausgegangen, dass diese JVM als Container für das Datengrid agiert, das in der Datei `objectGridDeployment.xml` definiert ist.

   Sie müssen die richtigen Namen für diese XML-Dateien verwenden. Bei den Dateinamen muss die Groß-/Kleinschreibung beachtet werden. Wenn die Dateien nicht vorhanden sind, wird der Container nicht gestartet. Sie können in der Datei `systemout.log` nach Nachrichten suchen, die darauf hinweisen, dass Shards verteilt werden. Ein EJB-Modul oder WAR-Modul, das eXtreme Scale verwendet, muss XML-Dateien von eXtreme Scale in seinem Verzeichnis `META-INF` enthalten.

   Die XML-Dateien von eXtreme Scale enthalten Folgendes:

   - ObjectGrid-XML-Deskriptordatei mit dem Namen `objectGrid.xml`. Weitere Informationen finden Sie unter ObjectGrid-XML-Deskriptordatei.
   - XML-Implementierungsdeskriptordatei mit dem Namen `objectGridDeployment.xml`. Weitere Informationen finden Sie unter XML-Deskriptordatei für Implementierungsrichtlinie.
   - (Optional) XML-Deskriptordatei mit Entitätsmetadaten, wenn Entitäten verwendet werden. Der Name der Datei `entity.xml` muss mit dem Namen übereinstimmen, der in der Datei `objectGrid.xml` angegeben ist. Weitere Informationen finden Sie unter XML-Deskriptordatei für Entitätsmetadaten.

   Die Laufzeitumgebung erkennt diese Dateien und stellt dann eine Verbindung zum Katalogservice her, um ihn darüber zu informieren, dass ein weiterer Container verfügbar ist, der Shards für diese Instanz von eXtreme Scale aufnehmen kann.

Tipp: Wenn Ihre Anwendung Entitäten hat und Sie einen Container-Server verwenden möchten, setzen Sie **minSyncReplicas** in der XML-Implementierungsdeskriptordatei auf den Wert `0`. Andernfalls können Sie eine der folgenden Nachrichten in der Datei `SystemOut.log` sehen, weil keine Verteilung stattfinden kann, bis ein anderer Server der minSyncReplica-Richtlinie entspricht:

```
CWPRJ1005E: Fehler beim Auflösen der Entitätsassoziation. Entity=entity_name,
association=association_name.

CWOBJ3013E: Das EntityMetadata-Repository ist nicht verfügbar. Beim Versuch, die Entität
zu registrieren, wurde das zulässige Zeitlimit erreicht: Entitätsname
```

2. Implementieren und starten Sie Ihre Anwendung.

   Der Container wird automatisch gestartet, wenn das Modul gestartet wird. Der Katalogservice beginnt sobald wie möglich mit der Verteilung der primären und Replikat-Shards der Partition. Diese Verteilung findet unverzüglich statt, sofern Sie keine Verzögerung der Verteilung in der Umgebung konfigurieren. Weitere Informationen finden Sie unter „Verteilung steuern” auf Seite 435.

Nächste Schritte

Anwendungen, die sich in derselben Zelle wie die Container befinden, können eine Verbindung zu diesen Datengrids über die Methode "ObjectGridManager.connect-(null, null) herstellen und anschließend die Methode "getObjectGrid(ccc, "object grid name")" aufrufen. Die Methoden "connect" und "getObjectGrid" können blockiert werden, bis die Container die Shards verteilt haben, aber diese Blockierung tritt nur auf, wenn das Datengrid gestartet wird.

ClassLoaders

Alle Plug-ins oder Objekte, die in eXtreme Scale gespeichert sind, werden in ein bestimmtes Klassenladeprogramm geladen. Zwei EJB-Module in derselben EAR-Datei können diese Objekte enthalten. Die Objekte sind identisch, werden aber von verschiedenen Klassenladeprogrammen geladen. Wenn Anwendung A ein Person-Objekt in einer Map speichert, die eine lokale Map des Servers ist, empfängt Anwendung B eine Ausnahme des Typs `ClassCastException`, wenn sie versucht, das Objekt zu lesen. Diese Ausnahme tritt ein, weil Anwendung B das Person-Objekt in einem anderen Klassenladeprogramm geladen hat.

Eine Methode zur Behebung dieses Problems ist die Verwendung eines Stammmoduls, das die erforderlichen Plug-ins und Objekte enthält, die in eXtreme Scale gespeichert werden. Jedes Modul, das eXtreme Scale verwendet, muss dieses Modul für seine Klassen referenzieren. Eine andere Lösung ist die Verwaltung dieser gemeinsam genutzten Objekte in einer Dienstprogramm-JAR-Datei, die in einem Klassenladeprogramm enthalten ist, das von Modulen und Anwendungen gemeinsam genutzt wird. Die Objekte können auch in WebSphere-Klassen oder in ein Verzeichnis `lib/ext` gestellt werden, aber diese Verteilung macht die Implementierung komplexer.

EJB-Module in einer EAR-Datei nutzen gewöhnlich dasselbe Klassenladeprogramm und sind von diesem Problem nicht betroffen. Jedes WAR-Modul hat ein eigenes Klassenladeprogramm und ist von diesem Problem betroffen.

Nur Verbindung zu einem Datengrid-Client herstellen:

Wenn die Eigenschaft **catalog.services.cluster** in den angepassten Zellen-, Knoten- oder Servereigenschaften definiert ist, kann jedes Modul in der EAR-Datei die Methode ObjectGridManager.connect (ServerFactory.getServerProperties().getCata-

logServiceBootstrap(), null, null) aufrufen, um einen ClientClusterContext abzurufen. Das Modul kann auch die Methode ObjectGridManager.getObjectGrid(ccc, "grid name") aufrufen, um eine Referenz auf das Datengrid anzufordern. Wenn Anwendungsobjekte in Maps gespeichert werden, müssen Sie sicherstellen, dass diese Objekte in einem gemeinsamen Klassenladeprogramm vorhanden sind.

Java-Clients und Clients außerhalb der Zelle können eine Verbindung zum Bootstrap-IIOP-port des Katalogservice herstellen. In WebSphere Application Server wird der Katalogservice standardmäßig im Deployment Manager ausgeführt. Der Client kann anschließend einen ClientClusterContext und das benannte Datengrid anfordern.

EntityManager

Mit dem Entitätsmanager werden die Tupel in den Maps und nicht in Anwendungsobjekten gespeichert, was zu weniger Problemen beim Laden von Klassen führt. Plug-ins können jedoch ein Problem sein. Beachten Sie auch, das immer eine ObjectGrid-XML-Deskriptordatei für Clientkorrekturwerte erforderlich ist, wenn eine Verbindung zu einem Datengrid hergestellt wird, in der Entitäten definiert sind: ObjectGridManager.connect("host:port[,host:port], null, objectGridOverride) oder ObjectGridManager.connect(null, objectGridOverride).

IBM eXtremeMemory und IBM eXtremeIO konfigurieren

Wenn Sie eXtremeMemory konfigurieren, können Sie Objekte im nativen Speicher anstelle des Java-Heapspeichers speichern. Die Konfiguration von eXtremeMemory aktiviert eXtremeIO, einen neuen Transportmechanismus. Durch die Auslagerung von Objekten aus dem Java-Heapspeicher können Sie Garbage-Collection-Pausen vermeiden und damit eine konstantere Leistung und vorhersehbare Antwortzeiten erzielen.

Vorbereitende Schritte

- Linux eXtremeIO und eXtremeMemory werden nur auf x86-64-Bit-Linux-Systemen unterstützt, die ein 64-Bit-SDK verwenden.
- Sie müssen MapSets verwenden, in denen alle Maps mit dem Kopiermodus `COPY_TO_BYTES` oder `COPY_TO_BYTES_RAW` konfiguriert sind. Wenn es Maps im MapSet gibt, die keinen dieser Kopiermodi verwenden, werden Objekte im Java-Heapspeicher gespeichert, und der Object Request Broker (ORB) wird verwendet.
- Sie können eXtremeIO und eXtremeMemory in den folgenden Konfigurationsszenarien nicht verwenden:
  - Sie verwenden Container-Server, die in einer Umgebung von WebSphere Application Server ausgeführt werden.
  - Sie verwenden angepasste Evictor-Plug-ins.
  - Sie verwenden zusammengesetzte Indizes.
  - Sie verwenden integrierte Write-behind-Loader.
  - Sie verwenden die Schnittstelle ReplicationMapListener, um eine Implementierung eines Ereignislisteners für clientseitige Maps zu erstellen, die sich im Replikationsmodus befinden.

Informationen zu diesem Vorgang

Die JVM stützt sich bei der Erfassung, Verkleinerung und Vergrößerung des Prozessspeichers auf Nutzungsheuristik. Der Garbage-Collector führt diese Operatio-

nen aus. Durchd ie Garbage-Collection entstehen jedoch Kosten. Die Kosten für die Garbage-Collection nehmen mit der Größe des Heapspeichers und der Anzahl der Objekte im Datengrid zu. Die JVM stellt verschiedene heuristische Verfahren für verschiedene Anwendungsfälle und Ziele bereit: optimaler Durchsatz, optimale Pausenzeiten, auf dem Objektalter basierende Garbage-Collection, gleichmäßig Verteilung und Echtzeit. Kein heuristisches Verfahren ist perfekt. Ein einziges heuristisches Verfahren kann nicht für alle Konfigurationen geeignet sein.

WebSphere eXtreme Scale verwendet Datencaching mit verteilten Maps, die Einträge mit einem bekannten Lebenszyklus haben. Dieser Lebenszyklus enthält die folgenden Operationen: GET, INSERT, DELETE und UPDATE. Durch die Verwendung dieser bekannten Map-Lebenszyklen können eXtremeMemory und eXtremeIO den Speicher effizienter nutzen als die JVM-Nutzungsheuristik.

Die folgende Abbildung veranschaulicht, wie mit eXtremeMemory konsistentere relative Antwortzeiten in der Umgebung erzielt werden. Im Bereich der hohen Perzentile sind die relativen Antwortzeiten bei Anforderungen, die eXtremeMemory verwenden, wesentlich kürzer. In der Abbbildung sind die 95-100 Perzentile dargestellt.

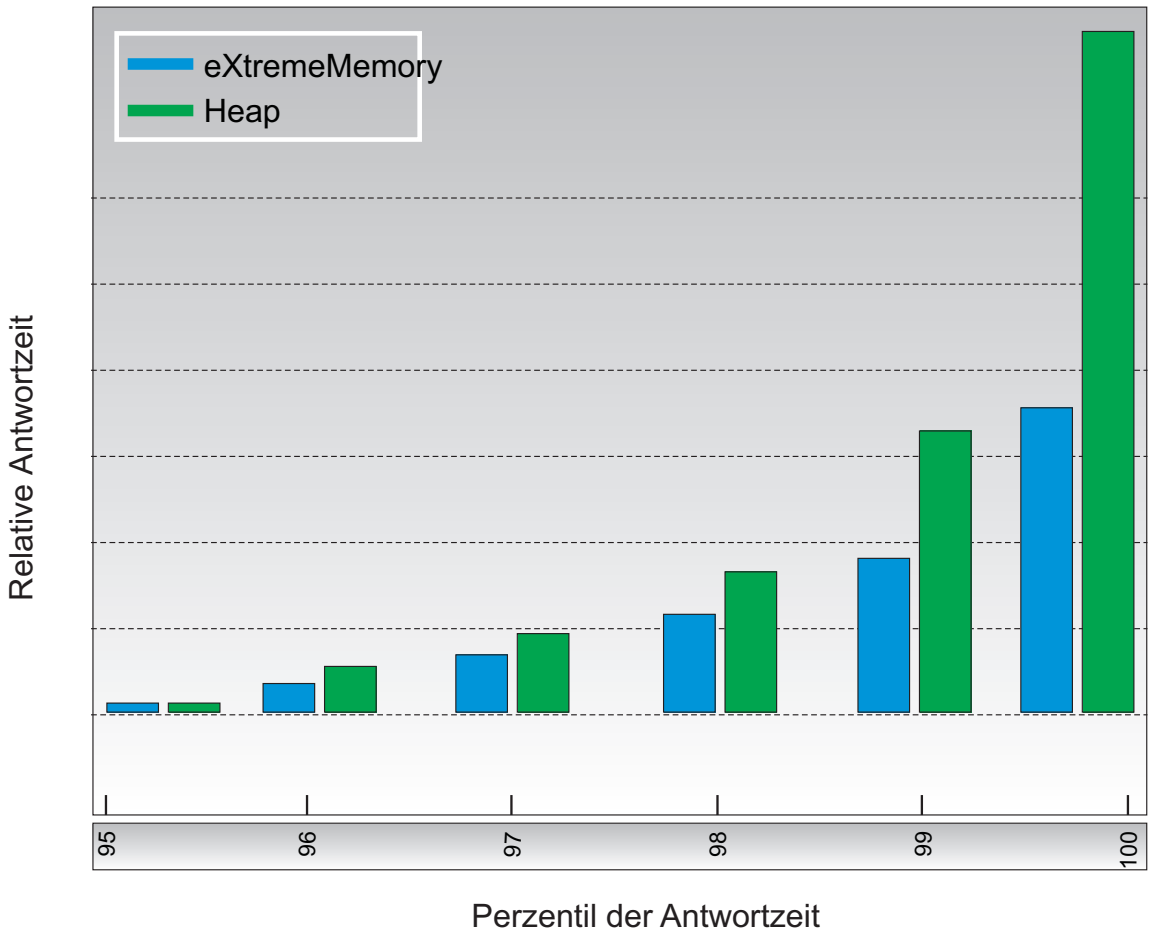


Abbildung 30. Antwortzeiten von eXtremeMemory und Heapspeicher im Vergleich

Wenn Sie eXtremeMemory verwenden, wird eXtremeIO für die Kommunikation zwischen Container-Servern verwendet. Objekte werden im Container-Server in Bytes serialisiert. Zum Aktivieren von eXtremeIO und eXtremeMemory setzen Sie die erforderlichen Servereigenschaften in allen Container-Servern im Datengrid und starten dann die Server erneut.

Vorgehensweise

1. Optional: Richtigen Wert für die Eigenschaft **maxXMSize** ermitteln.
   a. Bestimmen Sie die Größe pro Eintrag in Ihrer vorhandenen Konfiguration. Führen Sie den Befehl **xscmd -c showMapSizes** aus, um diese Größe zu bestimmen.
   b. Berechnen Sie den Wert für **maxXMSize**. Um die maximale Gesamtgröße der Einträge (*maximale_Gesamtgröße*) zu erhalten, multiplizieren Sie die *Größe_pro_Eintrag* mit *maximale_Eintragsanzahl*. Verwenden Sie nicht mehr als 60 % von **maxXMSize**, um die Verarbeitung der Metadaten zu berücksichtigen. Multiplizieren Sie *maximale_Gesamtgröße* mit 1,65, um den Wert von **maxXMSize** zu erhalten.
2. Aktualisieren Sie die Servereigenschaftendatei für jeden Container-Server in der Konfiguration, um den neuen Transport zu aktivieren. Die folgenden Servereigenschaften aktivieren den neuen Transport:

   Erforderliche Eigenschaften

   7.1.1+ **enableXM**
   Wenn Sie diese Eigenschaft auf `true` setzen, wird IBM eXtremeMemory auf dem Server aktiviert und der Server für die Verwendung von IBM eXtremeIO für synchrone und asynchrone Replikation konfiguriert. Cacheeinträge werden im nativen Speicher und nicht im Java-Heapspeicher gespeichert. Alle Container-Server im Datengrid müssen denselben Wert für die Eigenschaft **enableXM** verwenden.

   Standardwert: `false`

   Empfohlene Eigenschaften

   7.1.1+ **maxXMSize**
   Letzt die maximale Speicherkapazität in Megabyte fest, die dem Server für den eXtremeMemory-Speicher zur Verfügung steht.

   Standardeinstellung: 25 % des Gesamtspeichers auf dem System

   Optionale Eigenschaften

   7.1.1+ **maxXIONetworkThreads**
   Legt die maximale Anzahl an Threads fest, die dem Thread-Pool des eXtremeIO-Transportnetzes zugeordnet werden.

   Standardeinstellung: `50`

   7.1.1+ **minXIONetworkThreads**
   Legt die Mindestanzahl an Threads fest, die dem Thread-Pool des eXtremeIO-Transportnetzes zugeordnet werden.

   Standardeinstellung: `50`

   7.1.1+ **maxXIOWorkerThreads**
   Legt die maximale Anzahl an Threads fest, die dem Thread-Pool für die Verarbeitung von eXtremeIO-Transportanforderungen zugeordnet werden.

   Standardeinstellung: `128`

7.1.1+ **minXIOWorkerThreads**
Legt die Mindestanzahl an Threads fest, die dem Thread-Pool für die Verarbeitung von eXtremeIO-Transportanforderungen zugeordnet werden.

Standardeinstellung: 128

7.1.1+ **xioChannel.xioContainerTCPNonSecure.Port**
Gibt die Nummer des nicht sicheren Listener-Ports von eXtremeIO auf dem Server an. Wenn Sie keinen Wert festlegen, wird ein ephemerer Port verwendet. Diese Eigenschaft wird nur verwendet, wenn die Eigenschaft **transportType** auf `TCP/IP` gesetzt ist.

7.1.1+ **xioChannel.xioContainerTCPSecure.Port**
Gibt die SSL-Portnmmer von eXtremeIO auf dem Server an. Diese Eigenschaft wird nur verwendet, wenn die Eigenschaft **transportType** auf `SSL-Supported` oder `SSL-Required` gesetzt ist.

3. Starten Sie die Container-Server erneut, damit der neue Transportmechanismus verwendet wird. Weitere Informationen finden Sie unter „Eigenständige Server starten und stoppen" auf Seite 403 und „Server in einer Umgebung von WebSphere Application Server starten und stoppen" auf Seite 417.

Topologien mit mehreren Rechenzentren konfigurieren

Bei der asynchronen Multimasterreplikation verbinden Sie eine Gruppe von Katalogservicedomänen miteinander. Die verbundenen Katalogservicedomänen werden anschließend durch Replikation über die Verbindungen synchronisiert. Sie können die Verbindungen mithilfe von Eigenschaftendateien, zur Laufzeit mit JMX-Programmen (Java Management Extensions) oder mit Befehlszeilendienstprogrammen definieren. Die Gruppe aktueller Verbindungen für eine Domäne wird im Katalogservice gespeichert. Sie können Verbindungen hinzufügen und entfernen, ohne die Katalogservicedomäne, die das Datengrid hostet, erneut starten zu müssen.

Vorbereitende Schritte

- Weitere Informationen zu Multimasterreplikationstopologien und Designhinweise finden Sie unter „Topologien mit mehreren Rechenzentren planen" auf Seite 36. Sie können Verbindungen zwischen Katalogservicedomänen mithilfe der Servereigenschaftendatei konfigurieren, um die Topologie während des Serverstarts zu bilden. Verbindungen können auch zur Laufzeit konfiguriert werden.
- Wenn Sie Ladeprogramme (Loader) in Ihrer Multimasterreplikationstopologie verwenden, müssen Sie planen, wie Sie die Datengenauigkeit zwischen den Rechenzentren verwalten möchten. Die Methoden, die Sie verwenden können, variieren je nach verwendeter Topologie. Weitere Informationen finden Sie im Abschnitt „Hinweise zu Ladeprogrammen in einer Multimastertopologie" auf Seite 41.

Vorgehensweise

- Für Bootstrapzwecke Verbindungen in der Servereigenschaftendatei für den Katalogserver jeder Katalogservicedomäne in der Topologie definieren.

  Weitere Informationen zum Definieren dieser Datei für den Katalogserver finden Sie unter Servereigenschaftendatei.

  Wichtig: Bei der Eingabe von Eigenschaftsnamen muss die Groß-/Kleinschreibung beachtet werden.

Name der lokalen Domäne:

Geben Sie den Namen der Katalogservicedomäne für den aktuellen Katalogserver an, z. B. A:

```
domainName=A
```

Eine optionale Liste mit Namen fremder Domänen:

Geben Sie die Namen der Katalogservicedomänen an, zu denen Sie in der Multimasterreplikationstopologie eine Verbindung herstellen möchten, z. B. B:

```
foreignDomains=B
```

Eine optionale Liste mit Endpunkten für die fremden Domänen.

Gibt die Verbindungsinformationen für die Katalogserver der fremden Domänen an, z. B. Domäne B:

```
B.endPoints=hostB1:2809, hostB2:2809
```

Wenn eine fremde Domäne mehrere Katalogserver hat, geben Sie sie alle an.

- Verwenden Sie das Dienstprogramm **xscmd** oder JMX-Programmierung, um Verbindungen zur Laufzeit hinzuzufügen oder zu entfernen.

  Die Verbindungen für eine Domäne werden im Katalogservice im replizierten Speicher verwaltet. Diese Gruppe von Verbindungen kann jederzeit vom Administrator geändert werden, ohne dass ein Neustart dieser Domäne oder einer anderen Domäne erforderlich ist. Das Dienstprogramm **xscmd** enthält mehrere Optionen für die Bearbeitung von Verbindungen.

  Das Dienstprogramm **xscmd** stellt eine Verbindung zu einem Katalogservice und damit zu einer einzigen Katalogservicedomäne her. Deshalb kann das Dienstprogramm **xscmd** verwendet werden, um Verbindungen zwischen der Domäne, zu dem es eine Verbindung herstellt, und jeder anderen Domäne zu erstellen und zu entfernen.

  Verwenden Sie die Befehlszeile, um eine Verbindung zu erstellen. Beispiel:

  ```
  xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
  ```

  Der Befehl richtet eine neue Verbindung zwischen der lokalen Domäne und der fremden Domäne `dname` ein. Der Katalogservice `dname` wird unter `fdHostA:2809` und `fdHostB:2809` ausgeführt. Die lokale Katalogservicedomäne hat die Host/Port-Kombination `host:2809` für den Listener des Katalogservice. Geben Sie alle Katalogserviceendpunkte der fremden Domäne an, so dass eine Verbindung mit Fehlertoleranz zur Domäne möglich ist. Von der Verwendung eines einzigen `host:port`-Paars für den Katalogservice der fremden Katalogservicedomäne wird abgeraten.

  Sie können jede lokale Katalog-Service-JVM mit dem Dienstprogramm **xscmd** und der Option **-cep** angeben. Wenn der Katalogserver in einem Deployment Manager von WebSphere Application Server ausgeführt wird, ist der Port gewöhnlich 9809.

  Die für die fremde Domäne angegebenen Ports sind keine JMX-Ports. Es sind die gewöhnlichen Ports, die für Clients von eXtreme Scale verwendet werden.

  Nach dem Absetzen des Befehls zum Hinzufügen einer neuen Verbindung weist der Katalogservice alle Container unter seiner Verwaltung an, mit der Replikation in der fremden Domäne zu beginnen. Eine Verbindung ist nicht auf beiden Seiten erforderlich. Es muss nur auf einer Seite eine Verbindung erstellt werden.

  Verwenden Sie die Befehlszeile, um eine Verbindung zu entfernen. Beispiel:

  ```
  xscmd –c dismissLink -cep host:2809 -fd dname
  ```

Der Befehl stellt eine Verbindung zum Katalogservice für eine Domäne her und weist diesen an, die Replikation in einer bestimmten Domäne zu stoppen. Eine Verbindung muss nur auf einer Seite entfernt werden.

Verbindungen zwischen zwei Katalogservicedomänen

Angenommen, Sie möchten ein Setup mit zwei Katalogservicedomänen, Domäne A und Domäne B, konfigurieren.

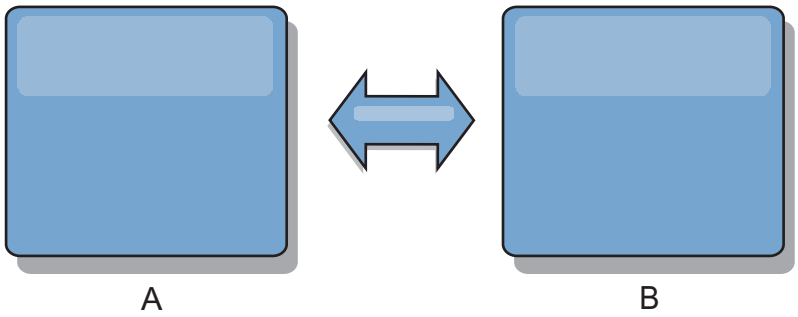


Abbildung 31. Verbindung zwischen Katalogservicedomänen

Im Folgenden sehen Sie die Servereigenschaftendatei für den Katalogserver in Domäne A:

```
domainName=A
foreignDomains=B
B.endPoints=hostB1:2809, hostB2:2809
```

Im Folgenden sehen Sie die Servereigenschaftendatei für den Katalogserver in Domäne B. Beachten Sie die Ähnlichkeit der beiden Eigenschaftendateien.

```
domainName=B
foreignDomains=A
A.endPoints=hostA1:2809,hostA2:2809
```

Nach dem Start der beiden Domänen werden alle Datengrids mit den folgenden Merkmalen zwischen den Domänen repliziert.

- Das Datengrid hat einen privaten Katalogservice mit einem eindeutigen Domänennamen.
- Das Datengrid hat denselben Gridnamen wie andere Grids in der Domäne.
- Das Datengrid hat dieselbe Anzahl an Partitionen wie andere Datengrids in der Domäne.
- Das Datengrid hat den Typ FIXED_PARTITION (Datengrids des Typs PER-_CONTAINER können nicht repliziert werden).
- Das Datengrid hat dieselbe Anzahl an Partitionen (kann, muss aber nicht dieselbe Anzahl und dieselben Typen von Replikaten haben).
- Das Datengrid hat dieselben Replikationsdatentypen wie andere Datengrids in der Domäne.
- Das Datengrid hat dieselben MapSet-Namen, Map-Namen und dynamischen Map-Schablonen wie andere Datengrids in der Domäne.

Die Replikationsrichtlinie einer Katalogservicedomäne wird ignoriert.

Das vorherige Beispiel zeigt, wie jede Domäne mit einer Verbindung zur anderen Domäne konfiguriert wird, aber es muss nur in eine einzige Richtung eine Verbin-

dung definiert werden. Diese Tatsache ist insbesondere in Hub- und Peripherietopologien hilfreich, weil die Konfiguration sehr viel einfacher ist. Die Hub-Eigenschaftendatei muss nicht aktualisiert werden, wenn Peripheriegeräte hinzugefügt werden, und jede Peripheriegerätedatei muss nur Hub-Informationen enthalten. Auch in einer Ringtopologie muss jede Domäne nur eine Verbindung zur vorherigen Domäne und zur nächsten Domäne im Ring haben.

Beispiel: Hub- und Peripherietopologie

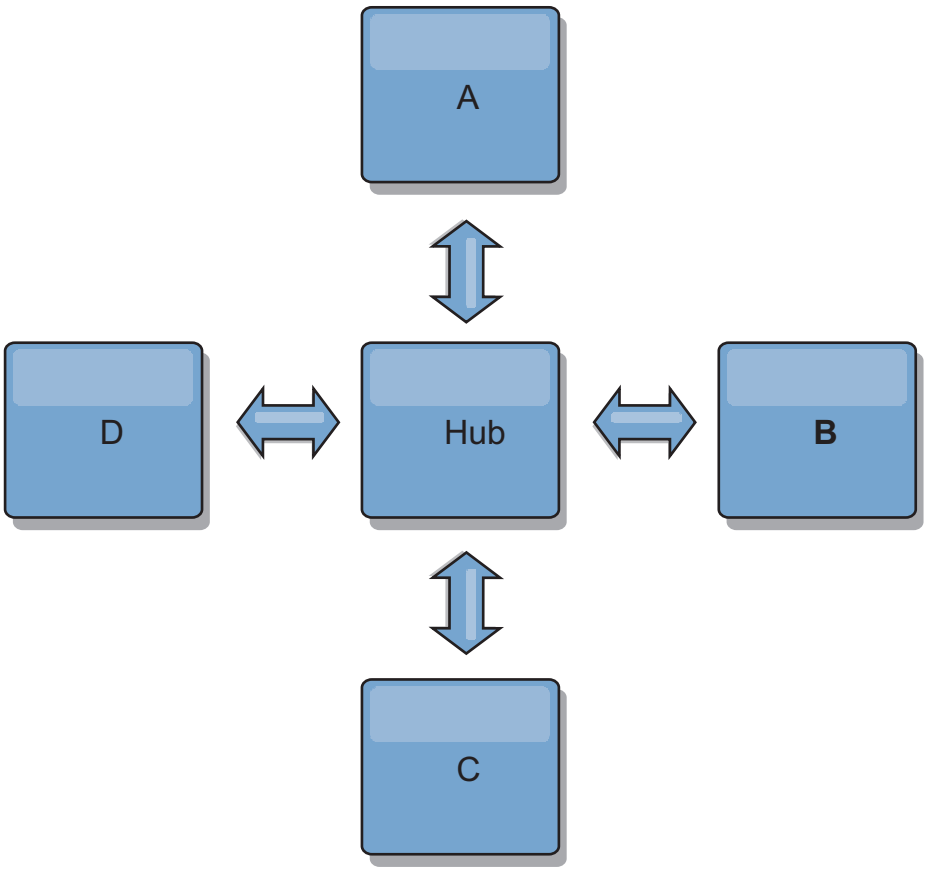


Abbildung 32. Hub- und Peripherietopologie

Der Hubs und die vier und Peripheriedomänen (Domänen A, B, C und D) haben Servereigenschaftendateien wie die folgenden:

```
domainName=Hub
```

Peripheriedomäne A hat die folgenden Servereigenschaften:

```
domainName=A
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Peripheriedomäne B hat die folgenden Servereigenschaften:

```
domainName=B
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Peripheriedomäne C hat die folgenden Servereigenschaften:

```
domainName=C
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Peripheriedomäne D hat die folgenden Eigenschaften:

```
domainName=D
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Nächste Schritte

Sie können einen angepassten Kollisionsarbiter für die Behebung von Kollisionen zwischen den Katalogservicedomänen angeben. Weitere Informationen finden Sie unter Angepasste Arbiter für Replikation mehrerer Master entwickeln.

Ports konfigurieren

WebSphere eXtreme Scale ist ein verteilter Cache, der das Öffnen von Ports für die Kommunikation mit dem ORB (Object Request Broker) und dem TCP-Stack (Transmission Control Protocol) zwischen Java Virtual Machine (JVM) und anderen Servern voraussetzt.

Ports im eigenständigen Modus konfigurieren

Sie können die erforderlichen Ports für Server und Clients in einer eXtreme-scale-Implementierung mit Befehlszeilenparametern, mit Eigenschaftendateien oder über das Programm konfigurieren. Die meisten Beispiele in den folgenden Abschnitten beschreiben Befehlszeilenparameter für das Script **startOgServer**. Entsprechende Konfigurationsoptionen können auch in Eigenschaftendateien, über die integrierte Server-API oder die Client-API definiert werden.

Vorgehensweise

1. Katalogserviceendpunkte starten.

   WebSphere eXtreme Scale verwendet IIOP für die Kommunikation zwischen Java Virtual Machines. Die Katalogservice-JVMs sind die einzigen Prozesse, die die explizite Konfiguration von Ports für die IIOP-Services und Gruppenservices erfordern. Alle anderen Prozesse ordnen Ports dynamisch zu.

   Der Client-Port und der Peer-Port werden für die Kommunikation zwischen Katalogservices in einer Katalogservicedomäne verwendet. Verwenden Sie die folgende Befehlszeilenoption, um den Client-Port und den Peer-Port anzugeben:

   -catalogServiceEndPoints <Servername:Hostname:Client-Port:Peer-Port>
   Verweist im Container auf den ORB-Host (Object Request Broker) und ORB-Port im Katalogservice. Die Attribute sind wie folgt definiert:

   Servername
   Gibt einen Namen an, der den Prozess identifiziert, den Sie starten.

   Hostname
   Gibt den Hostnamen des Computers an, auf dem der Server gestartet wird.

   Client-Port
   Gibt den Port an, der für die Kommunikation mit dem Peerkatalogservice verwendet wird.

Peer-Port

Dieser Wert entspricht dem HA-Manager-Port. Er gibt den Port an, der für die Kommunikation mit dem Peerkatalogservice verwendet wird.

Im folgenden Beispiel wird der Katalogserver "cs1" gestartet, der sich in derselben Katalogservicedomäne wie die Server "cs2" und "cs3" befindet:

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

Die Katalogserviceendpunkte können auch mit der Servereigenschaft catalogServiceEndPoints definiert werden. Der ORB-Listener-Port (Object Request Broker) wird für die Kommunikation zwischen Katalogservices in einer Katalogservicedomäne und für die Kommunikation zwischen Katalogservices und Container-Servern und Clients verwendet. Verwenden Sie die folgenden Befehlszeilenoptionen, um den Listener-Port und den Listener-Host anzugeben:

-listenerHost <Hostname>

Gibt den Hostnamen an, an den der Object Request Broker (ORB) für die Kommunikation mit Internet Inter-ORB Protocol (IIOP) gebunden wird. Der Wert muss ein vollständig qualifizierter Domänenname oder eine IP-Adresse sein. Wenn Ihre Konfiguration mehrere Netzkarten enthält, definieren Sie den Listener-Host und Listener-Port, so dass der Object Request Broker in der JVM die IP-Adresse für die Bindung kennt. Wenn Sie die zu verwendende IP-Adresse nicht kennen, können Symptome wie Überschreitungen des Verbindungszeitlimits, ungewöhnliche API-Fehler und Clients, die zu blockieren scheinen, auftreten.**Standardeinstellung:** `localhost`

-listenerPort <Port>

Gibt die Portnummer an, an die der Object Request Broker (ORB) gebunden wird. Diese Einstellung konfiguriert Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. In WebSphere Application Server wird der Listener-Port von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.**Standardeinstellung:** `2809`

Der Listener-Port und der Listener-Host können auch mit den Servereigenschaften listenerHost und listenerPort angegeben werden.

Der JMX-Service-Port wird für die Kommunikation über JMX-Clients verwendet. Verwenden Sie die folgende Befehlszeilenoption, um den JMX-Service-Port anzugeben:

-JMXServicePort <Port>

Gibt die Nummer des Ports an, über den der MBean-Server mit Java Management Extensions (JMX) kommuniziert. Sie müssen für jede JVM in Ihrer Konfiguration eine andere Portnummer verwenden. Wenn Sie JMX/RMI verwenden, geben Sie die Option **JMXServicePort** und die Portnummer explizit an, selbst wenn Sie den Standardportwert verwenden möchten. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.

Standardeinstellung: `1099`

Der JMX-Service-Port kann auch mit der Servereigenschaft JMXServicePort angegeben werden. Wenn die Sicherheit aktiviert ist, ist auch ein SSL-Port (Secure Socket Layer) erforderlich. Verwenden Sie die folgende Befehlszeilenoption, um den SSL-Port anzugeben:

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<SSL-Port>
```

```
./startOgServer.sh cs1 -listenerHost hostA -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

Abbildung 33. Beispiel für die Verwendung der Befehlszeile. Starten Sie den ersten Katalogserver auf hostA. Im Folgenden sehen Sie einen Beispielbefehl:

Starten Sie den zweiten Katalogserver auf hostB.Im Folgenden sehen Sie einen Beispielbefehl:

```
./startOgServer.sh cs2 -listenerHost hostB -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

2. Container-Server-Endpunkte starten.

   Der folgende Befehl startet eine Container-JVM für den Beispielkatalogservice:

   ```
   ./startOgServer.sh c0 -catalogServiceEndPoints hostA:2809,hostB:2809
   ```

   Die Java Virtual Machines des Container-Servers verwendet zwei Ports. Der HA-Manager-Port wird für die interne Kommunikation zwischen Peer-Container-Servern und Katalogservern verwendet. Der Listener-Port wird für die IIOP-Kommunikation zwischen Peer-Container-Servern, Katalogservern und Clients verwendet. Der Listener-Port wird verwendet, um den ORB an einen bestimmten Netzadapter zu binden. Wenn Sie keine Ports angeben, werden beide Ports dynamisch ausgewählt. Möchten Sie die Ports jedoch explizit konfigurieren, wie z. B. in einer Firewallumgebung, können Sie den ORB-Port mithilfe von Befehlszeilenoptionen angeben. Verwenden Sie die folgenden Befehlszeilenoptionen, um den Listener-Port und den Listener-Host anzugeben:

   ```
   -listenerHost <Hostname>
   -listenerPort <Port>
   ```

   Der Listener-Port und der Listener-Host können auch mit den Servereigenschaften "listenerHost" und "listenerPort " angegeben werden.

   Verwenden Sie die folgende Befehlszeilenoption, um den HA-Manager-Port anzugeben:

   -haManagerPort <Port>
   Synonym mit Peer-Port. Gibt die Nummer des vom High Availability Manager verwendeten Ports an. Wenn Sie diese Eigenschaft nicht definieren, generiert der Katalogservice automatisch einen verfügbaren Port. Diese Eigenschaft gilt für den Container-Server und den Katalogservice. (Nur erforderlich für Umgebungen von WebSphere Application Server.)

   Der HA-Manager-Port kann auch mit der Servereigenschaft HAManagerPort angegeben werden

   Wenn die Sicherheit aktiviert ist, ist auch ein SSL-Port (Secure Socket Layer) erforderlich. Verwenden Sie die folgende Befehlszeilenoption, um den SSL-Port anzugeben:

   ```
   -jvmArgs -Dcom.ibm.CSI.SSLPort=<SSL-Port>
   ```

3. Clientendpunkte starten.

   Clients müssen nur die Listenerendpunkte des Katalogservice kennen. Clients rufen Endpunkte für Container-Server-JVMs, die die JVMs sind, die die Daten enthalten, automatisch vom Katalogservice ab. Um eine Verbindung zu dem Katalogservice aus dem vorherigen Beispiel herzustellen, muss der Client die folgende Liste von `Host:Port`-Paaren an die API "connect" übergeben:

   ```
   hostA:2809,hostB:2809
   ```

   Der Client kann auch Callbacks von Container-Servern empfangen, wenn die API "DataGrid" verwendet wird. Diese Callbacks kommunizieren über IIOP mit dem ORB-Listener-Port. Wenn Sie festlegen möchten, dass der Port und der

Netzadapter Callbacks empfangen, definieren Sie die Eigenschaften **listenerHost** und **listenerPort** in der Clienteigenschaftendatei.

Wenn die Sicherheit aktiviert ist, ist auch ein SSL-Port (Secure Socket Layer) erforderlich. Verwenden Sie die folgende Systemeigenschaft beim Starten des Clientprozesses, um den SSL-Port anzugeben:

```
-Dcom.ibm.CSI.SSLPort=<SSL-Port>
```

Ports in einer Umgebung mit WebSphere Application Server konfigurieren

Katalogservices, Container-Server und Clients von WebSphere eXtreme Scale nutzen bei der Ausführung in Prozessen von WebSphere Application Server Ports und Services, die bereits für den Prozess definiert sind.

Informationen zu diesem Vorgang

In den folgenden Abschnitten finden Sie Einzelheiten zur Verwendung von Ports in Ihrer Implementierung.

1. Katalogserviceendpunkte

   Die Katalogservices von WebSphere eXtreme Scale werden in einem Prozess von WebSphere Application Server ausgeführt und über die Administrationskonsole oder mithilfe von Verwaltungsaufgaben konfiguriert. Alle Ports mit Ausnahme des Client-Ports, der explizit konfiguriert wird, werden vom Prozess übernommen. Einzelheiten zu den Ports, die vom Katalogservice verwendet werden, finden Sie unter „Netzports planen” auf Seite 65. Einzelheiten zum Konfigurieren einer Katalogservicedomäne finden Sie unter Katalogservice mit hoher Verfügbarkeit.

2. Container-Server-Endpunkte

   Container-Server von WebSphere eXtreme Scale werden in Java-EE-Modulen gehostet. Die Container-Server verwenden die Ports, die für den Anwendungsserverprozess definiert wurden. Einzelheiten zu den vom Container-Service verwendeten Ports finden Sie unter „Netzports planen” auf Seite 65. Einzelheiten zum Starten eines Containers in einem Java-EE-Modul, wie z. B. einem EJB- (Enterprise JavaBeans™) oder Webmodul, finden Sie unter „Anwendungen von WebSphere Application Server für den automatischen Start von Container-Servern konfigurieren” auf Seite 282.

3. Clientendpunkte

   Clients von WebSphere eXtreme Scale werden in Java-EE- oder EJB-Modulen gehostet.

   Clients stellen mit der API "ObjectGridManager.connect()" über das Programm eine Verbindung zur Katalogservicedomäne her. Wenn eine Verbindung zu einer Katalogservicedomäne in derselben Zelle hergestellt wird, sucht die Clientverbindung mit dem folgenden API-Aufruf in ObjectGridManager automatisch die Standardkatalogservicedomäne:

   ```
   connect(securityProps, overRideObjectGridXML)
   ```

   Wenn die Standardkatalogservicedomäne fern (außerhalb der Zelle) gehostet wird, müssen die Katalogserviceendpunkte mit der folgenden Methode in der API "ObjectGridManager" angegeben werden:

   ```
   connect(catalogServerAddresses, securityProps, overRideObjectGridXml)
   ```

   Wenn die Standardkatalogservicedomäne in der Zelle definiert ist, kann die API "CatalogServerProperties" verwendet werden, um die Katalogserveradressen ab-

zurufen. Zum Abrufen der konfigurierten Endpunkte der Katalogservicedomäne kann auch die Verwaltungsaufgabe "XSDomainManagement" verwendet werden.

Server mit mehreren Netzkarten

Sie können eXtreme-Scale-Prozesse in einem Server ausführen, der mehrere Netzkarten besitzt.

Wenn ein Server oder Client in eine Server mit mehreren Netzkarten ausgeführt wird müssen Sie den Netzport und den Hostnamen in Ihrer eXtreme-Scale-Konfiguration angeben, um eine Bindung an eine bestimmte Karte herzustellen. Wenn diese Konfiguration nicht angegeben wird, wählt die eXtreme-Scale-Laufzeitumgebung automatisch eine Karte aus, was zu Verbindungsfehlern oder einer geringeren Leistung führen kann.

Für Katalog- und Container-Server müssen Sie den Listener-Host und den Listener-Port mit einer der folgenden Methoden festlegen:

- Servereigenschaften
- Befehlszeilenparameter im Script startOgServer.sh | bat.

Für Clients können Sie die Befehlszeile nicht verwenden. Vielmehr müssen Sie Clienteigenschaften verwenden.

Transporte konfigurieren

Transporte ermöglichen den Austausch von Objekten und Daten zwischen verschiedenen Serverprozessen in Ihrer Konfiguration.

Informationen zu diesem Vorgang

Der Haupttransportmechanismus ist der Object Request Broker (ORB). Dieser Mechanismus speichert Cacheeinträge im Java-Heapspeicher.

7.1.1+ Die Verwendung des ORB als Transportmechanismus ist in den folgenden Konfigurationsszenarien erforderlich:

- Sie verwenden ein anderes System als ein x86-64-Bit-Linux-System.
- Sie verwenden Container-Server, die in einer Umgebung von WebSphere Application Server ausgeführt werden.
- Sie verwenden Evictor-Plug-ins (Bereinigungsprogramme) oder zusammengesetzte Indizes.

7.1.1+ Wenn Sie eXtremeMemory verwendet wird ein neuer Transport mit dem Namen eXtremeIO verwendet. Mit eXtremeMemory werden Cacheeinträge im nativen Speicher gespeichert. Der native Speicher wird keiner Garbage-Collection unterzogen was zu einer konstanteren Leistung und vorhersehbaren Antwortzeiten führt. Objekte werden im Container-Server in Bytes serialisiert. Weitere Informationen finden Sie im Abschnitt „IBM eXtremeMemory und IBM eXtremeIO konfigurieren” auf Seite 285.

Object Request Broker konfigurieren

Der ORB (Object Request Broker) wird von WebSphere eXtreme Scale für die Kommunikation über einen TCP-Stack verwendet. Mit der Datei `orb.properties` können Sie die vom Object Request Broker (ORB) verwendeten Eigenschaften übergeben, um das Transportverhalten des Datengrids zu ändern. Es ist keine Aktion

erforderlich, um den von WebSphere eXtreme Scale oder WebSphere Application Server für Ihre Server von WebSphere eXtreme Scale bereitgestellten ORB zu verwenden.

Object Request Broker in einer Umgebung von WebSphere Application Server konfigurieren

Sie können WebSphere eXtreme Scale mit Anwendungen verwenden, die den Object Request Broker (ORB) direkt in Umgebungen von WebSphere Application Server oder WebSphere Application Server Network Deployment verwenden.

Vorgehensweise

1. Benennen Sie Ihre Anwendungsserver entsprechend.

   Server desselben Namens in der Umgebung von WebSphere Application Server sind nicht zulässig, wenn der Object Request Broker (ORB) für die Kommunikation verwendet wird. Sie können diese Einschränkung aufheben, indem Sie die Systemeigenschaft **-Dcom.ibm.websphere.orb.uniqueServerName=true** für die Prozesse angeben, die denselben Namen haben, z. B., wenn beispielsweise Server mit dem Namen `server1` auf jedem Knoten als Katalogservicedomäne verwendet werden oder wenn mehrere Node Agents verwendet werden, um eine Katalogservicedomäne zu bilden.
2. Optimieren Sie die ORB-Eigenschaften in der Konfiguration von WebSphere Application Server.

   Weitere Informationen zu den Eigenschaften, die Sie optimieren können, finden Sie unter „ORB-Eigenschaften” auf Seite 500. Je nach Eigenschaft können Sie die Einstellung in der Administrationskonsole oder in der Datei *WAS-Stammverzeichnis*`properties/orb.properties` ändern.
3. Wenn Sie mehrere Netzschnittstellenkarten verwenden, müssen Sie den Wert von ORB_LISTENER_ADDRESS in der Anzeige "Ports" in der Administrationskonsole von WebSphere Application Server definieren. Wiederholen Sie diesen Schritt für jeden Anwendungsserver in der Konfiguration.
   a. Für einen Anwendungsserver klicken Sie auf **Server** > **Anwendungsserver** > ***Servername***. Klicken Sie unter "Kommunikation" auf **Ports**. Daraufhin wird die Anzeige "Ports" für den angegeben Server angezeigt.
   b. Klicken Sie auf **Details**, und ändern Sie den Wert von ORB_LISTENER_ADDRESS.
   c. Geben Sie die IP-Adresse im Feld **Host** ein. Dieser Wert muss eine private Adresse für eine Umgebung mit mehreren Netzschnittstellen sein.

      Anmerkung: DNS-Hostnamen als Wert für ORB_LISTENER_ADDRESS nicht unterstützt.
   d. Geben Sie die Portnummer im Feld **Port** an. Die Portnummer gibt den für den Service konfigurierten Port an, an dem Clientanforderungen akzeptiert werden. Der Portwert wird zusammen mit dem Hostnamen verwendet.

Nächste Schritte

7.1.1+ Sie können das Tool **wxsLogAnalyzer** verwenden, um die ORB-Einstellungen in Ihrer Umgebung zu überprüfen. Weitere Informationen finden Sie unter „Protokoll- und Tracedaten analysieren” auf Seite 548.

Object Request Broker mit eigenständigen eXtreme-Scale-Prozessen konfigurieren

Sie können WebSphere eXtreme Scale mit Anwendungen verwenden, die den Object Request Broker (ORB) direkt in Umgebungen ohne WebSphere Application Server oder WebSphere Application Server Network Deployment verwenden.

Vorbereitende Schritte

Wenn Sie den ORB in demselben Prozess wie eXtreme Scale verwenden, müssen Sie bei der Ausführung von Anwendungen oder anderen Komponenten und Frameworks, die nicht mit eXtreme Scale bereitgestellt werden, möglicherweise zusätzliche Tasks ausführen, um sicherzustellen, dass eXtreme Scale ordnungsgemäß in Ihrer Umgebung ausgeführt wird.

Informationen zu diesem Vorgang

Fügen Sie der Datei `orb.properties` die Eigenschaft **`ObjectGridInitializer`** hinzu, um die Verwendung des ORB in Ihrer Umgebung zu initialisieren. Verwenden Sie den ORB, um die Kommunikation zwischen eXtreme-Scale-Prozessen und anderen Prozessen in Ihrer Umgebung zu aktivieren.

Vorgehensweise

1. Die eigenständige Installation enthält keine Datei `orb.properties`. Sie müssen eine Datei `orb.properties` im Verzeichnis `java/jre/lib` speichern. Beschreibungen der Eigenschaften und Einstellungen finden Sie unter „ORB-Eigenschaften” auf Seite 500.
2. Geben Sie in der Datei `orb.properties` die folgende Zeile ein, und speichern Sie Ihre Änderunegn.

   ```
   org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer
   ```

Ergebnisse

eXtreme Scale initialisiert den ORB ordnungsgemäß und koexistiert mit anderen Anwendungen, für die der ORB aktiviert ist.

Wenn Sie eine angepasste Version des ORB mit eXtreme Scale verwenden möchten, lesen Sie den Abschnitt „Angepassten Object Request Broker konfigurieren”.

Nächste Schritte

7.1.1+ Sie können das Tool **`xsLogAnalyzer`** verwenden, um die ORB-Einstellungen in Ihrer Umgebung zu überprüfen. Weitere Informationen finden Sie unter „Protokoll- und Tracedaten analysieren” auf Seite 548.

Angepassten Object Request Broker konfigurieren

WebSphere eXtreme Scale verwenden den Object Request Broker (ORB), um die Kommunikation zwischen Prozessen zu ermöglichen. Es ist keine Aktion erforderlich, um den von WebSphere eXtreme Scale oder WebSphere Application Server bereitgestellten Object Request Broker (ORB) für Ihre eXtreme-Scale-Server zu verwenden. Der Aufwand für die Verwendung derselben ORBs für Ihre eXtreme-Scale-Clients zu verwenden, ist gering. Wenn Sie stattdessen einen angepassten ORB verwenden müssen, ist der mit IBM SDK bereitgestellte ORB eine gute Wahl, obwohl Sie den ORB konfigurieren müssen. ORBs anderer Anbieter können ebenfalls mit gewissen Konfigurationsanpassungen verwendet werden.

Vorbereitende Schritte

Legen Sie fest, ob Sie den mit WebSphere eXtreme Scale oder WebSphere Application Server bereitgestellten ORB, den mit IBM SDK bereitgestellten ORB oder einen ORB eines anderen Anbieters verwenden möchten.

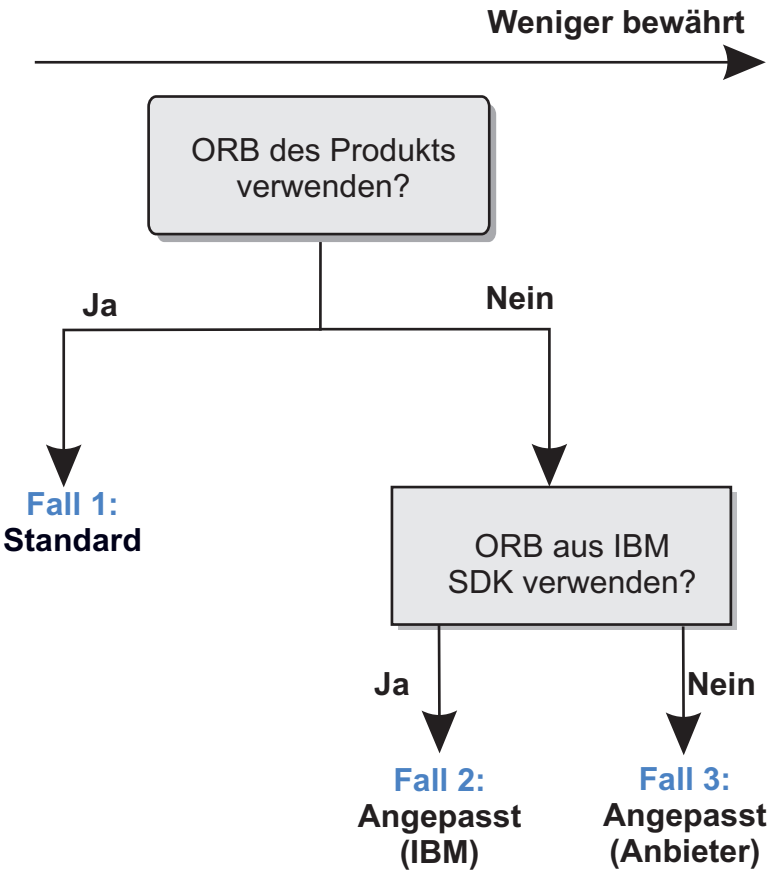


Abbildung 34. ORB auswählen

Sie können getrennte Entscheidungen für die eXtreme-Scale-Serverprozesse und die eXtreme-Scale-Clientprozesse verwenden. Obwohl eXtreme Scale Developer-Kits von den meisten Anbietern unterstützt, wird empfohlen, den ORB, der mit eXtreme Scale bereitgestellt wird, für Server- und Clientprozesse zu verwenden.eXtreme Scale bietet keine Unterstützung für den ORB, der mit dem Sun Microsystems Java Development Kit (JDK) bereitgestellt wird.

Informationen zu diesem Vorgang

Machen Sie sich mit der Konfiguration vertraut, die für die Verwendung des gewünschten ORB erforderlich ist.

Fall 1: Standard-ORB

- Für Ihre eXtreme-Scale-Serverprozesse ist zur Verwendung des mit WebSphere eXtreme Scale oder WebSphere Application Server bereitgestellten ORB keine Konfiguration erforderlich.
- Für Ihre eXtreme-Scale-Clientprozesse ist zur Verwendung des mit WebSphere eXtreme Scale oder WebSphere Application Server bereitgestellten ORB eine minimale Klassenpfadkonfiguration erforderlich.

Fall 2: Angepasster ORB (IBM)
Anweisungen zum Konfigurieren Ihrer eXtreme-Scale-Clientprozesse für die Verwendung des mit IBM SDK bereitgestellten ORB finden Sie in den Anweisungen weiter hinten in diesem Abschnitt. Sie können den IBM ORB verwenden, wenn Sie IBM SDK oder ein anderes Entwicklungskit verwenden. Sie können IBM SDK Version 5 oder höher verwenden.

Fall 3: Angepasster ORB (eines anderen Anbieters)
Die Verwendung eines ORB eines anderen Anbieters für Ihre eXtreme-Scale-Clientprozesse ist die Option, die am wenigsten erprobte Option. Stellen Sie sicher, dass alle Probleme, die bei der Verwendung von ORBs unabhängiger Softwareanbieter auftreten, mit dem IBM ORB und einer kompatiblen JRE reproduzierbar sind, bevor Sie Unterstützung anfordern.

Der mit Sun Microsystems Java Development Kit (JDK) bereitgestellte ORB wird nicht unterstützt.

Vorgehensweise

- Konfigurieren Sie Ihre Clientprozesse für die Verwendung eines der Standard-ORBs (**Fall 1)**. Verwenden Sie das folgende JVM-Argument: :

```
-jvmArgs –Djava.endorsed.dirs=Standard-ORB-Verzeichnis${pathSeparator}JRE_HOME/lib/endorsed
```

  Das Standard-ORB-Verzeichnis ist *WXS-Ausgangverzeichnis*`/lib/endorsed`. Möglicherweise müssen auch die folgenden Eigenschaften in der Datei `orb.properties` aktualisiert werden:

  ```
  org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
  org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton
  ```

- Client- oder Serverprozesse für die Verwendung von IBM SDK Version 5 konfigurieren **(Fall 2)**.
  1. Kopieren Sie die ORB-JAR-Dateien in ein leeres Verzeichnis oder das Verzeichnis *Verzeichnis_des_angepassten_ORB*.
     - `ibmorb.jar`
     - `ibmorbapi.jar`
  2. Geben Sie das Verzeichnis *Verzeichnis_des_angepassten_ORB* als endorsed-Verzeichnis in den Scripts an, die den Java-Befehl starten.

     Tipp: Wenn Ihre Java-Befehle bereits ein endorsed-Verzeichnis angeben, kann das Verzeichnis *Verzeichnis_des_angepassten_ORB* alternativ in das vorhandene endorsed-Verzeichnis gestellt werden. Wenn Sie das Verzeichnis *Verzeichnis_des_angepassten_ORB* in das vorhandene endorsed-Verzeichnis stellen, ist eine Aktualisierung der Scripts nicht erforderlich. Wenn Sie sich trotzdem für die Aktualisierung entscheiden, müssen Sie dem vorhandenen Argument `–Djava.endorsed.dirs=` das *Verzeichnis_des_angepassten_ORB* voranstellen, anstatt das vorhandene Argument vollständig zu ersetzen.

     - Aktualisieren Sie Scripts für eine eigenständige eXtreme-Scale-Umgebung.

       Ändern Sie den Pfad für die Variable *OBJECTGRID_ENDORSED_DIRS* in der Datei `setupCmdLine.bat|sh` so, dass sie auf das Verzeichnis *Verzeichnis_des_angepassten_ORB* zeigt. Speichern Sie Ihre Änderungen.
     - Aktualisieren Sie Scripts, wenn eXtreme Scale in eine eXtreme-Scale-Umgebung integriert ist.

       Fügen Sie die folgende Systemeigenschaft und die folgenden Parameter dem Script `startOgServer` hinzu:

       ```
       -jvmArgs –Djava.endorsed.dirs=Verzeichnis_des_angepassten_ORB
       ```

– Aktualisieren Sie angepasste Scripts, die Sie verwenden, um einen Clientanwendungsprozess oder Serverprozess zu starten.
  `-Djava.endorsed.dirs=`*`Verzeichnis_des_angepassten_ORB`*

Clients konfigurieren

Sie können WebSphere eXtreme Scale für die Ausführung in einer eigenständigen Umgebung oder für die Ausführung in einer Umgebung it WebSphere Application Server konfigurieren. Damit eine eXtreme-Scale-Implementierung Konfigurationsänderungen auf Server-Grid-Seite berücksichtigt, müssen Sie Prozesse erneut starten, damit diese Änderungen wirksam werden. Die Änderungen werden nicht dynamisch angewendet. Obwohl Sie auf der Clientseite die Konfigurationseinstellungen für eine vorhandene Clientinstanz nicht ändern können, können Sie jedoch unter Verwendung einer XML-Datei oder über das Programm eine neue Clientinstanz mit den erforderlichen Einstellungen erstellen. Wenn Sie einen Client erstellen, können Sie die Standardeinstellungen überschreiben, die aus der aktuellen Serverkonfiguration stammen.

Sie können einen Client von eXtreme Scale mit den folgenden Methoden konfigurieren, die jeweils mit einer XML-Datei zum Überschreiben des Clients oder über das Programm ausgeführt werden können.

- XML-Konfiguration
- Programmgesteuerte Konfiguration
- Konfiguration des Spring-Frameworks
- Nahen Cache inaktivieren

Sie können die folgenden Plug-ins in einem Client überschreiben:

- **ObjectGrid-Plug-ins**
  - TransactionCallback-Plug-in
  - ObjectGridEventListener-Plug-in
- **BackingMap-Plug-ins**
  - Evictor-Plug-in
  - MapEventListener-Plug-in
  - Attribut "numberOfBuckets"
  - Attribut "ttlEvictorType"
  - Attribut "timeToLive"

Clients mit XML-Konfiguration konfigurieren

Sie können eine ObjectGrid-XML-Konfigurationsdatei verwenden, um Einstellungen auf Clientseite zu ändern.

Informationen zu diesem Vorgang

Zum Ändern der Einstellungen in einem Client von WebSphere eXtreme Scale müssen Sie eine ObjectGrid-XML-Datei erstellen, die in ihrer Struktur der Datei gleicht, die für den Container-Server verwendet wurde.

Sie können die folgenden Einstellungen im Client überschreiben:

1. Erstellen Sie eine clientspezifische ObjectGrid-Instanz.
2. Kopieren Sie die ObjectGrid-XML-Datei, die zum Öffnen des Servers verwendet wurde.

3. Bearbeiten Sie die neue Datei, um Einstellungen auf der Clientseite anzupassen.
   - Zum Festlegen oder Aktualisieren von Attributen im Client geben Sie einen neuen Wert an oder ändern den vorhandenen Wert.
   - Zum Entfernen eines Plug-ins aus dem Client verwenden Sie eine leere Zeichenfolge als Wert für das Attribut "className".
   - Wenn Sie ein vorhandenes Plug-in ändern möchten, geben Sie einen neuen Wert für das Attribut "className" an.
   - Sie können auch Plug-ins hinzufügen, die für das Überschreiben eines Clients unterstützt werden: TRANSACTION_CALLBACK, OBJECTGRID_EVENT_LISTENER, EVICTOR, MAP_EVENT_LISTENER.
4. Erstellen Sie einen Client mit der neu erstellen XML-Datei für das Überschreiben des Clients:

Vorgehensweise

1. Erstellen Sie eine ObjectGrid-XML-Konfigurationsdatei für den Client, die in ihrer Struktur der Datei für den Container-Server gleicht.

   Angenommen, die folgende XML-Datei wurde mit einer XML-Deskriptordatei für die Implementierungsrichtlinie kombiniert und zum Starten eines Container-Servers verwendet.

   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" numberOfBuckets="1049"
                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>
```

   In einem Container-Server verhält sich die ObjectGrid-Instanz mit dem Namen "CompanyGrid" gemäß Definition in der Datei `companyGridServerSide.xml`. Standardmäßig hat der CompanyGrid-Client dieselben Einstellungen wie das im Server ausgeführte CompanyGrid.

   Die folgende ObjectGrid-XML-Datei kann verwendet werden, um einige Attribute und Plug-ins im CompanyGrid-Client anzugeben:

   `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" numberOfBuckets="1429"
                pluginCollectionRef="customerPlugins" />
            <backingMap name="Item" />
            <backingMap name="OrderLine" numberOfBuckets="701"
                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>
```

Im Folgenden sehen Sie eine Zusammenfassung der definierten Korrekturwerte:

- Der TransactionCallback im Client ist "com.company.MyClientTxCallback" und nicht wie auf der Serverseite "com.company.MyTxCallback".
- Der Client hat kein ObjectGridEventListener-Plug-in, weil der className-Wert die leere Zeichenfolge ist.
- Der Client setzt "numberOfBuckets" für die BackingMap "Customer" auf 1429, behält sein Evictor-Plug-in und entfernt das MapEventListener-Plug-in.
- Die Attribute "numberOfBuckets" und "timeToLive" der BackingMap "OrderLine" haben sich geändert.
- Obwohl ein anderes lockStrategy-Attribut angegeben ist, hat dieses keine Wirkung, weil das Attribut "lockStrategy" für das Überschreiben eines Clients nicht unterstützt wird.

2. Erstellen Sie den Client mit der XML-Datei.

   Zum Erstellen eines CompanyGrid-Clients über die Datei `companyGridClientSide.xml` übergeben Sie die ObjectGrid-XML-Datei als URL an eine der connect-Methoden in der Schnittstelle "ObjectGridManager".

```
ObjectGridManager ogManager =
 ObjectGridManagerFactory.ObjectGridManager();
ClientClusterContext clientClusterContext =
 ogManager.connect("MyServer1.company.com:2809", null, new URL(
                "file:xml/companyGridClientSide.xml"));
```

Mechanismus für Clientinvalidierung aktivieren

In einer verteilten Umgebung von WebSphere eXtreme Scale gibt es auf der Clientseite standardmäßig einen nahen Cache, wenn die optimistische Sperrstrategie verwendet wird oder Sperren inaktiviert sind. Der nahe Cache enthält seine eigenen lokalen zwischengespeicherten Daten. Wenn ein Client von eXtreme Scale eine Aktualisierung festschreibt, wird diese Aktualisierung an den nahen Cache des Clients und an den Server gesendet. Andere Clients von eXtreme Scale erhalten die Aktualisierungsinformationen jedoch nicht und können daraufhin Daten haben, die nicht auf dem neuesten Stand sind.

Naher Cache

Anwendungen müssen sich dieses Problems potenziell veralteter Daten in Clients von eXtreme Scale bewusst sein. Sie können die integrierte JMS-basierte 8Java Message Service) ObjectGridEventListener-Klasse "com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener" verwenden, um den Mechanismus für die Invalidierung von Clients in einer verteilten Umgebung von eXtreme Scale zu aktivieren.

Der Mechanismus für Clientinvalidierung ist die Lösung für das Problem veralteter Daten im nahen Cache des Clients in einer verteilten eXtreme-Scale-Umgebung. Diese Mechanismus stellt sicher, dass der nahe Cache des Clients mit Servern oder anderen Clients synchronisiert wird. Aber selbst mit diesem JMS-basierten Mechanismus für Clientinvalidierung wird der nahe Cache des Clients nicht sofort aktualisiert. Es tritt eine Verzögerung auf, wenn die Laufzeitumgebung von eXtreme Scale Aktualisierungen veröffentlicht.

Es sind zwei Modelle für den Mechanismus für Clientinvalidierung in einer verteilten eXtreme-Scale-Umgebung verfügbar:

- Client/Server-Modell: In diesem Modell haben alle Serverprozesse die Rolle "Publisher" (Bereitsteller), die alle Transaktionsänderungen an der vorgesehenen JMS-Destination veröffentlicht. Alle Clientprozesse haben die Rolle "Receiver" (Empfänger) und empfangen alle Transaktionsänderungen von der vorgesehenen JMS-Destination.
- Modell mit dem Client in zwei Rollen: In diesem Modell haben alle Serverprozesse nichts mit der JMS-Destination zu tun. Alle Clientprozesse übernehmen die Rollen "Publisher" (Veröffentlichung) und "Receiver" (Empfang) für die JMS-Destinations. Transaktionsänderungen, die auf Clientseite stattfinden, werden an der JMS-Destination veröffentlicht, und alle Clients empfangen diese Transaktionsänderungen.

Weitere Informationen finden Sie im Abschnitt „JMS-Ereignis-Listener” auf Seite 238.

Client/Server-Modell

In einem Client/Server-Modell haben die Server die Rolle "JMS-Publisher", und der Client hat die Rolle "JMS-Receiver".

XML-Beispiel für Client/Server-Modell

```
<?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>
```

Modell mit dem Client in zwei Rollen

In diesem Modell übernimmt jeder Client sowohl die Rolle "JMS-Publisher" als auch die Rolle "JMS-Receiver". Der Client veröffentlicht alle festgeschriebenen Transaktionsänderungen an der vorgesehenen JMS-Destination und empfängt alle festgeschriebenen Transaktionsänderungen von anderen Clients. Der Server selbst hat in diesem Modell nichts mit JMS zu tun.

XML-Beispiel mit Zweirollenmodell

```
<?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>
```

Zeitlimit für Anforderungswiederholung konfigurieren

Bei zuverlässigen Maps können Sie ein Wiederholungszeitlimit für Transaktionsanforderungen an WebSphere eXtreme Scale übergeben.

Informationen zu diesem Vorgang

Sie können das Zeitlimit in der Clienteigenschaftendatei oder in einer Sitzung konfigurieren. Der Sitzungswert überschreibt immer die Einstellung in der Clienteigenschaftendatei. Wenn ein Wert größer als null definiert wird, wird die Anforderung so lange wiederholt, bis das Zeitlimit abläuft oder ein permanenter Fehler auftritt. Ein permanenter Fehler ist beispielsweise eine Ausnahme des Typs `DuplicateKeyException`. Der Wert null steht für die Moduseinstellung "fail-fast", d. h., eXtreme Scale führt nach keiner Transaktion eine Wiederholung durch.

Zur Laufzeit wird das Transaktionszeitlimit zusammen mit dem Wiederholungszeitlimit verwendet, um sicherzustellen, dass das Wiederholungszeitlimit nicht höher ist als das Transaktionszeitlimit.

Es gibt zwei Typen von Transaktionen: Transaktionen mit automatischer Festschreibung und Transaktionen, die explizite begin- und commit-Methoden. Die gültigen Ausnahmen für Wiederholungen sind bei diesen beiden Typen von Transaktionen unterschiedlich:

- Transaktionen, die in einer Sitzung aufgerufen werden, werden bei CORBA-Ausnahmen des Typs "SystemException" und eXtreme-Scale-Ausnahmen des Typs "TargetNotAvailable" wiederholt.
- Transaktionen mit automatischer Festschreibung werden bei CORBA-Ausnahmen des Typs "SystemException" und eXtreme-Scale-Ausnahmen bezüglich der Verfügbarkeit wiederholt. Zu diesen Ausnahmen gehören die Ausnahmen "ReplicationVotedToRollbackTransactionException", "TargetNotAvailable" und "AvailabilityException".

Anwendungs- oder andere permanente Fehler werden sofort zurückgegeben, und die Transaktion wird vom Client nicht wiederholt. Zu diesen permanenten Fehlern gehören Ausnahmen des Typs "DuplicateKeyException" und "KeyNotFoundException". Verwenden Sie die Einstellung "fail-fast", damit alle Ausnahmen ohne Wiederholung der Transaktion zurückgegeben werden.

Ausnahmen mit Wiederholung auf Clientseite

- ReplicationVotedToRollbackTransactionException (nur bei automatischer Festschreibung)
- TargetNotAvailable
- org.omg.CORBA.SystemException

- AvailabilityException (nur bei automatischer Festschreibung)
- LockTimeoutException (nur bei automatischer Festschreibung)
- UnavailableServiceException (nur bei automatischer Festschreibung)

Permanente Ausnahmen ohne Transaktionswiederholung:

- DuplicateKeyException
- KeyNotFoundException
- LoaderException
- TransactionAffinityException
- LockDeadlockException
- OptimisticCollisionException

Vorgehensweise

- Zeitlimit in einer Clienteigenschaftendatei definieren.

  Zum Festlegen des requestRetryTimeout-Werts in einem Client müssen Sie in der Clienteigenschaftendatei die Eigenschaft "requestRetryTimeout" hinzufügen oder ändern. Die Clienteigenschaftendatei ist standardmäßig die Datei `objectGridClient.properties`. Die Eigenschaft "requestRetryTimeout" wird in Millisekunden festgelegt. Setzen Sie die Eigenschaft auf einen Wert größer als null, damit eine Anforderung bei Ausnahmen, für die eine Wiederholung verfügbar ist, wiederholt wird. Setzen Sie die Eigenschaft auf `0`, damit Anforderungen bei Ausnahmen ohne Wiederholung fehlschlagen. Zur Verwendung des Standardverhaltens entfernen Sie die Eigenschaft, oder setzen Sie sie auf den Wert `-1`. Im Folgenden sehen Sie ein Beispiel für den Wert in der Datei `objectGridClient.properties`:

  ```
  requestRetryTimeout = 30000
  ```

  Der requestRetryTimeout-Wert wird in Millisekunden angegeben. Wenn der Wert aus dem vorherigen Beispiel in einer ObjectGrid-Instanz verwendet wird, ist der requestRetryTimeout-Wert 30 Sekunden.

- Zeitlimit über das Programm definieren.

  Wenn Sie die Clienteigenschaften über das Programm festlegen möchten, müssen Sie zuerst eine Clienteigenschaftendatei an einer für Ihre Anwendung angemessenen Position erstellen. Im folgenden Beispiel verweist die Clienteigenschaftendatei auf das Snippet `objectGridClient.properties` im vorherigen Abschnitt. Nach der Herstellung der Verbindung zur ObjectGridManager-Instanz definieren Sie die Clienteigenschaften wie beschrieben. Die ObjectGrid-Instanz, die Sie erhalten, hat dann die Clienteigenschaften, die Sie in der Datei definiert haben. Wenn Sie die Clienteigenschaftendatei ändern, müssen Sie jedes Mal explizit eine neue ObjectGrid-Instanz anfordern.

  ```
  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);
  ```

- Überschreibungsdatei bei einer Sitzungsfestschreibung definieren.

  Wenn Sie das Zeitlimit für Anforderungswiederholung in einer Sitzung festlegen oder die Clienteigenschaft "requestRetryTimeout" überschreiben möchten, rufen Sie die Methode "setRequestRetryTimeout(long)" in der Schnittstelle "Session" auf.

```
Session sessionA = objectGrid.getSession();
sessionA.setRequestRetryTimeout(30000);
ObjectMap mapA = sessionA.getMap("payroll");
String key = "key:" + j;
mapA.insert(key, "valueA");
```

Diese Sitzung verwendet jetzt einen requestRetryTimeout-Wert von 30000 ms bzw. 30 Sekunden, unabhängig von dem in der Clienteigenschaftendatei festgelegten Wert. Weitere Informationen zur Schnittstelle "Session" finden Sie unter Session-Objekte für den Zugriff auf Daten im Grid verwenden.

Cacheintegration konfigurieren

WebSphere eXtreme Scale kann in andere Caching-Produkte integriert werden. Sie können auch den dynamischen Cache-Provider von WebSphere eXtreme Scale verwenden, um WebSphere eXtreme Scale als Plug-in in der dynamischen Cachekomponente von WebSphere Application Server zu verwenden. Eine andere Erweiterung von WebSphere Application Server ist der HTTP-Sitzungsmanager von WebSphere eXtreme Scale, der als Unterstützung für die Zwischenspeicherung von HTTP-Sitzungen eingesetzt werden kann.

HTTP-Sitzungsmanager konfigurieren

Der HTTP-Sitzungsmanager stellt Sitzungsreplikationsfunktionen für eine zugeordnete Anwendung bereit. Der Sitzungsmanager arbeitet mit dem Webcontainer zusammen, um die Lebenszyklen von HTTP-Sitzungen zu erstellen und zu verwalten, die der Anwendung zugeordnet sind.

HTTP-Sitzungsmanager mit WebSphere Application Server konfigurieren

WebSphere Application Server stellt zwar Funktionen für die Sitzungsverwaltung bereit, aber die Leistung nimmt mit zunehmender Anzahl an Anforderungen ab. Im Lieferumfang von WebSphere eXtreme Scale ist eine Sitzungsmanagementimplementierung enthalten, die Sitzungsreplikation, hohe Verfügbarkeit, eine bessere Skalierbarkeit und stabilere Konfigurationsoptionen bietet.

Vorbereitende Schritte

- WebSphere eXtreme Scale muss in der Zelle mit WebSphere Application Server oder WebSphere Application Server Network Deployment installiert werden, damit der Sitzungsmanager von eXtreme Scale verwendet wird. Weitere Informationen finden Sie im Abschnitt „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren” auf Seite 167.
- Die globale Sicherheit muss in der Administrationskonsole von WebSphere Application Server aktiviert werden, wenn in den Katalogservern in Ihrer Katalogservicedomäne Secure Sockets Layer (SSL) aktiviert ist oder wenn Sie SSL für eine Katalogservicedomäne mit SSL-Unterstützung verwenden möchten. Sie legen die SSL-Anforderung für einen Katalogserver fest, indem Sie das Attribut "transportType" in der Servereigenschaftendatei auf `SSL-Required` setzen. Weitere Informationen zum Konfigurieren der globalen Sicherheit finden Sie unter Globale Sicherheitseinstellungen konfigurieren.

Informationen zu diesem Vorgang

Der HTTP-Sitzungsmanager von WebSphere eXtreme Scale unterstützt integrierte und ferne Server für das Caching.

- **Szenario mit integriertem Sitzungsmanager**

Im Szenario mit dem integrierten Sitzungsmanager werden die WebSphere eXtreme Scale-Server in denselben Prozessen wie die Servlets ausgeführt. Der Sitzungsmanager kann direkt mit der lokalen ObjectGrid-Instanz kommunizieren, wodurch teure Verzögerungen bei der Netzübertragung vermieden werden.

Wenn Sie WebSphere Application Server verwenden, speichern Sie die bereitgestellten Dateien *WXS-Ausgangsverzeichnis*/session/samples/objectGrid.xml und *WXS-Ausgangsverzeichnis*/session/samples/objectGridDeployment.xml in den Verzeichnissen META-INF Ihrer WAR-Dateien (Webarchiv). eXtreme Scale erkennt diese Dateien beim Anwendungsstart automatisch und startet automatisch die eXtreme-Scale-Container in demselben Prozess wie den Sitzungsmanager.

Sie können die Datei objectGridDeployment.xml ändern, abhängig davon, ob Sie synchrone oder asynchrone Replikation verwenden möchten und wie viele Replikate konfiguriert werden sollen.

- **Szenario mit fernen Servern**

  Im Szenario mit fernen Servern werden die Container-Server in anderen Prozessen als die Servlets ausgeführt. Der Sitzungsmanager kommuniziert mit einem fernen Container-Server. Wenn Sie einen fernen, über ein Netz verbundenen Container-Server verwenden möchten, muss der Sitzungsmanager mit den Hostnamen und Portnummern der Katalogservicedomäne konfiguriert werden. Der Sitzungsmanager verwendet anschließend eine eXtreme-Scale-Clientverbindung, um mit dem Katalogserver und den Container-Servern zu kommunizieren.

  Wenn die Container-Server in unabhängigen, eigenständigen Prozessen gestartet werden, starten Sie die eXtreme-Scale-Container mit den Dateien objectGridStandAlone.xml und objectGridDeploymentStandAlone.xml, die im Verzeichnis "samples" des Sitzungsmanagers bereitgestellt werden.

Vorgehensweise

1. Fügen Sie Ihre Anwendung so zusammen, dass sie den Sitzungsmanager verwenden kann. Wenn Sie den Sitzungsmanager verwenden möchten, müssen Sie die entsprechenden Filterdeklarationen den Webimplementierungsdeskriptoren für die Anwendung hinzufügen. Außerdem werden die Konfigurationsparameter des Sitzungsmanagers in Form von Initialisierungsparametern für den Servlet-Kontext in den Implementierungsdeskriptoren an den Sitzungsmanager übergeben. Es gibt mehrere Methoden, mit denen Sie diese Informationen in Ihre Anwendung einführen können:

   - **Automatisches Verbinden mit WebSphere Application Server**

     Sie können die Verwendung des HTTP-Sitzungsmanager von WebSphere eXtreme Scale beim Installieren Ihrer Anwendung konfigurieren. Die Anwendungs- bzw. Serverkonfiguration kann auch geändert werden, damit der HTTP-Sitzungsmanager von WebSphere eXtreme Scale verwendet wird. Weitere Informationen finden Sie unter „Anwendungen für die HTTP-Sitzungsverwaltung in WebSphere Application Server” auf Seite 312.

   - **Automatisches Verinden der Anwendung mit angepassten Eigenschaften**

     Sie müssen Ihre Anwendungen nicht manuell zusammenfügen, wenn die Anwendung in WebSphere Application Server oder WebSphere Application Server Network Deployment ausgeführt wird.

     Fügen Sie einer Zelle oder einem Server eine angepasste Eigenschaft hinzu, um die Datei splicer.properties für alle Webanwendungen in diesem Geltungsbereich zu definieren. Gehen Sie zum Konfigurieren der angepassten Eigenschaft wie folgt vor:

     a. Navigieren Sie in der Administrationskonsole von WebSphere Application Server zum dem Pfad, in dem Sie die angepasste Eigenschaft definieren möchten, um die Position der Datei splicer.properties anzugeben.

– Wenn Sie die angepasste Eigenschaft für alle Anwendungen oder nur für eine bestimmte Anwendung konfigurieren möchten, klicken Sie auf **Systemverwaltung** > **Zelle** > **Angepasste Eigenschaften**.
– Wenn Sie die angepasste Eigenschaft für alle Anwendungen in einem bestimmten Anwendungsserver konfigurieren möchten, klicken Sie auf **Anwendungsserver** > **<*Servername*>** > **Verwaltung** > **Angepasste Eigenschaften**. Der Eigenschaftsname ist `com.ibm.websphere.xs.sessionFilterProps` und der Eigenschaftswert ist die erforderliche Position der Datei `splicer.properties` für Ihre Anwendungen. Ein Beispielpfad für die Position der Datei ist `/opt/splicer.properties`.

b. Fügen Sie die angepasste Eigenschaft `com.ibm.websphere.xs.sessionFilterProps` hinzu. Diese angepasste Eigenschaft gibt die Position der zu bearbeitenden Datei `splicer.properties` an. Die Datei ist im Deployment Manager enthalten. Wenn Sie die Datei `splicer.properties` für eine bestimmte Anwendung mit einer angepassten Eigenschaft auf Zellenebene konfigurieren möchten, geben Sie als Namen der angepassten Eigenschaft `<Anwendungsname>,com.ibm.websphere.xs.sessionFilterProps` an, wobei *Anwendungsname* für den Namen der Anwendung steht, auf die Sie die angepasste Eigenschaft anwenden möchten.

Wichtig: Stellen Sie sicher, dass die aktualisierte Datei `splicer.properties` auf allen Knoten, die einen Anwendungsserver enthalten, in dem die Anwendungen gehostet werden, die für die Sitzungsreplikation verbunden werden, in demselben Pfad enthalten ist.

Die einzigen gültigen Geltungsbereiche sind "Zelle", "Server" und "Anwendung" und nur verfügbar, wenn mit einem Deployment Manager gearbeitet wird. Wenn Sie einen anderen Geltungsbereich benötigen, müssen Sie Ihre Webanwendungen manuell verbinden.

Hinweis: Die Option für automatisches Verbinden funktioniert nur, wenn alle Knoten, auf denen die Anwendung ausgeführt wird, die Datei `splicer.properties` in demselben Pfad enthalten. Für heterogene Umgebungen, die Windows- und UNIX-Knoten enthalten, ist diese Option nicht möglich, und Sie müssen die Anwendung manuell zusammenfügen.

- **Anwendung mit dem Script `addObjectGridFilter` verbinden**

  Verwenden Sie ein Befehlszeilenscript, das mit eXtreme Scale bereitgestellt wird, um eine Anwendung mit Filterdeklarationen und Konfigurationen in Form von Initialisierungsparametern für den Servlet-Kontext zusammenzufügen. Bei einer Implementierung mit WebSphere Application Server befindet sich dieses Script unter `<was_home>/optionalLibraries/ObjectGrid/session/bin/addObjectGridFilter.bat/sh`. Bei einer eigenständigen Implementierung befindet sich das Script unter `WXS_HOME/ObjectGrid/session/bin/addObjectGridFilter.sh/bat`. Das Script **`oddObjectGridFilter`** akzeptiert zwei Parameter:

  – Anwendung - Der absolute Pfad zu der zu verbindenden EAR-Datei.
  – Absoluter Pfad zur Datei "splicer.properties", die verschiedene Konfigurationseigenschaften enthält.

  Das Syntaxformat dieses Scripts ist wie folgt:

  Windows

  ```
  addObjectGridFilter.bat [EAR-Datei] [splicer-properties-Datei]
  ```

UNIX

`addObjectGridFilter.sh [EAR-Datei] [splicer.properties-Datei]`

UNIX **Beispiel mit eXtreme Scale in WebSphere Application Server unter UNIX:**

a. `cd WXS-Ausgangsverzeichnis/optionalLibraries/ObjectGrid/session/bin`

b. `addObjectGridFilter.sh /tmp/mySessionTest.ear WAS-Stammverzeichnis/optionalLibraries/ObjectGrid/session/samples/splicer.properties`

UNIX **Beispiel mit eXtreme Scale in einem eigenständigen Verzeichnis unter UNIX:**

a. `cd WAS-Stammverzeichnis/session/bin`

b. `addObjectGridFilter.sh /tmp/mySessionTest.ear WAS-Stammverzeichnis/session/samples/splicer.properties`

Der Servletfilter, der verbunden wird, verwaltet Standardkonfigurationswerte. Sie können diese Standardwerte mit Konfigurationsoptionen überschreiben, die Sie in der Eigenschaftendatei im zweiten Argument angeben. Eine Liste der Parameter, die Sie verwenden können, finden Sie im Abschnitt „Initialisierungsparameter für den Servlet-Kontext" auf Seite 329.

Sie können die Beispieldatei `splicer.properties` ändern und verwenden, die mit der Installation von eXtreme Scale bereitgestellt wird. Sie können auch das Script "**`addObjectGridServlets`**" verwenden, das den Sitzungsmanager einfügt, indem Sie jedes Servlet erweitern. Das empfohlene Script ist jedoch das Script "**`addObjectGridFilter`**".

- **Anwendung manuell mit dem Ant-Build-Script verbinden**

  Im Lieferumfang von WebSphere eXtreme Scale ist eine Datei `build.xml` enthalten, das von Apache Ant verwendet werden kann, und im Ordner *WAS-Stammverzeichnis*`/bin` einer Installation von WebSphere Application Server enthalten ist. Sie können die Datei `build.xml` bearbeiten, um die Konfigurationseigenschaften des Sitzungsmanagers zu ändern. Die Konfigurationseigenschaften sind identisch mit den Eigenschaftsnamen in der Datei `splicer.properties`. Sie bearbeiten die Datei `build.xml` und rufen dann den Ant-Prozess mit dem folgenden Befehl auf:

  – UNIX `ant.sh, ws_ant.sh`

  – Windows `ant.bat, ws_ant.bat`

  (UNIX) bzw. (Windows).

- **Webdeskriptor manuell aktualisieren**

  Editieren Sie die mit der Webanwendung gepackte Datei `web.xml`, um die Filterdeklaration, die Servlet-Zuordnung und die Initialisierungsparameter für den Servlet-Kontext zu integrieren. Verwenden Sie diese Methode nicht, weil sie fehleranfällig ist.

Eine Liste der Parameter, die Sie verwenden können, finden Sie im Abschnitt „Initialisierungsparameter für den Servlet-Kontext" auf Seite 329.

2. Implementieren Sie die Anwendung. Führen Sie dazu die Schritte aus, die Sie gewöhnlich für einen Server oder Cluster verwenden. Nach der Implementierung der Anwendung können Sie die Anwendung starten.
3. Greifen Sie auf die Anwendung zu. Sie können jetzt auf die Anwendung zugreifen, die mit dem Sitzungsmanager und mit WebSphere eXtreme Scale interagiert.

Nächste Schritte

Sie können die meisten Konfigurationsparameter des Sitzungsmanagers ändern, wenn Sie Ihre Anwendung für die Verwendung des Sitzungsmanagers instrumentieren. Zu diesen Attributen gehören die synchrone oder asynchrone Replikation, die Größe der speicherinternen Sitzungstabelle usw. Abgesehen von den Attributen, die während der Instrumentierung der Anwendung geändert werden können, sind die einzigen Attribute, die Sie nach der Anwendungsimplementierung ändern können, die Attribute, die sich auf die WebSphere eXtreme Scale-Serverclustertopologie beziehen, und die Art und Weise, in der Clients (Sitzungsmanager) eine Verbindung zu diesen Servern herstellen.

Verhalten bei einem fernen Szenario: Wenn das gesamte Datengrid, in dem die Anwendungssitzungsdaten gehostet werden, über den Web-Container-Client nicht verfügbar ist, verwendet der Client stattdessen den Basis-Web-Container in WebSphere Application Server für die Sitzungsverwaltung. Das Datengrid kann in den folgenden Szenarien nicht erreichbar sein:

- Es besteht ein Netzproblem zwischen dem Web-Container und den fernen Container-Servern.
- Die fernen Container-Server-Prozesse wurden gestoppt.

Die Anzahl der im Speicher verwalteten Sitzungsreferenzen, die mit dem Parameter **sessionTableSize** angegeben wird, wird auch auch dann beibehalten, wenn die Sitzungen im Basis-Web-Container gespeichert werden. Die Sitzungen, die am längsten nicht mehr verwendet wurden, werden aus dem Sitzungscache des Web-Containers entfernt, wenn der Wert von **sessionTableSize** überschritten wird. Wenn das ferne Datengrid wieder verfügbar ist, können Sitzungen, die aus dem Web-Container-Cache entfernt wurden, Daten aus dem fernen Datengrid abrufen und die Daten in eine neue Sitzung laden. Wenn das gesamte ferne Datengrid nicht verfügbar ist und die Sitzung aus dem Sitzungscache entfernt wird, gehen die Sitzungsdaten des Benutzers verloren. Aufgrund dieses Problems sollten Sie nicht das gesamte Produktionsdatengrid beenden, wenn das System unter Last ausgeführt wird.

Anwendungen für die HTTP-Sitzungsverwaltung in WebSphere Application Server:

Sie können Ihre Anwendung von WebSphere Application Server für die persistente Speicherung von Sitzungen in einem Datengrid konfigurieren. Dieses Datengrid kann in einem integrierten Container-Server, der in WebSphere Application Server ausgeführt wird, oder in einem fernen Datengrid enthalten sein.

Vorbereitende Schritte

Bevor Sie die Konfiguration in WebSphere Application Server ändern, müssen folgende Voraussetzungen erfüllt sein:

- Sie müssen den Namen des des Sitzungsdatengrids kennen, das Sie verwenden möchten. Weitere Informationen zum Erstellen eines Sitzungsdatengrids finden Sie unter „HTTP-Sitzungsmanager mit WebSphere Application Server konfigurieren" auf Seite 308.
- Wenn der Katalogservice, den Sie für die Verwaltung Ihrer Sitzungen verwenden möchten, außerhalb der Zelle befindet, in der Sie Ihre Sitzungsanwendung installieren, müssen Sie eine Katalogservicedomäne erstellen. Weitere Informationen finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen" auf Seite 264.

- Wenn Sie eine Katalogservicedomäne konfigurieren, müssen Sie möglicherweise die Clientsicherheit in der Katalogservicedomäne aktivieren, falls die Container-Server eine Authentifizierung erfordern. Diese Einstellungen teilen der Laufzeitumgebung mit, welche Implementierung CredentialGenerator verwendet werden muss. Diese Implementierung generiert einen Berechtigungsnachweis, der an das ferne Datengrid übergeben wird. Weitere Informationen zum Konfigurieren dieser Einstellungen finden Sie unter „Clientsicherheit in einer Katalogservicedomäne konfigurieren” auf Seite 536.
- Die globale Sicherheit muss in der Administrationskonsole von WebSphere Application Server aktiviert werden, wenn eines der folgenden Szenarien unterstützt werden soll:
  - In den Katalogservern in Ihrer Katalogservicedomäne ist Secure Sockets Layer (SSL) aktiviert.
  - Sie möchten SSL für eine Katalogservicedomäne mit SSL-Unterstützung verwenden.

  Sie legen die SSL-Anforderung für einen Katalogserver fest, indem Sie das Attribut **transportType** in der Servereigenschaftendatei auf `SSL-Required` setzen. Weitere Informationen zum Konfigurieren der globalen Sicherheit finden Sie unter Globale Sicherheitseinstellungen konfigurieren.
- Wenn Sie Version 7.1.0.3 oder höher verwenden, können Sie Sitzungen, die URL-Umschreibung oder Cookies als Sitzungsüberwachungsmechanismus verwenden, persistent im Datengrid speichern. Sitzungen, die URL-Umschreibung als Sitzungsüberwachungsmechanismus verwenden, können für Releases vor Version 7.1.0.3 nicht persistent gespeichert werden. Zum Aktivieren der Persistenz von Sitzungen, die URL-Umschreibung verwenden, setzen Sie nach der automatischen Verbindung der Anwendung die Eigenschaft **useURLEncoding** in der Datei `splicer.properties` auf `true`.
- 7.1.1+ Wenn Sie Anwendungen für die HTTP-Sitzungsverwaltung in WebSphere Application Server automatisch verbinden, ist in allen Anwendungsservern, die die Webanwendung hosten, die angepasste Eigenschaft **HttpSessionIdReuse** des Web-Containers Webcontainers web auf `true` gesetzt. Diese Eigenschaft ermöglicht Sitzungen, die von einem Anwendungsserver in einen anderen übernommen wurden bzw. in einem fernen Szenario im speicherinternen Sitzungscache ungültig gemacht wuden, ihre Sitzungs-ID über mehrere Anforderungen hinweg beizubehalten. Wenn Sie dieses Verhalten nicht wünschen, setzen Sie die angepasste Eigenschaft des Web-Containers in allen fraglichen Anwendungsservern auf `false`, bevor Sie die Sitzungsverwaltung für die Anwendungen konfigurieren. Weitere Informationen zu dieser angepassten Eigenschaft finden Sie unter „Fehlerbehebung bei der Cacheintegration” auf Seite 554.

Vorgehensweise

- **Gehen Sie wie folgt vor, um die Sitzungsverwaltung bei der Installation der Anwendung zu konfigurieren:**
  1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Anwendungen** > **Neue Anwendung** > **Neue Unternehmensanwendung**. Wählen Sie den Pfad **Detailliert** für die Erstellung der Anwendung aus, und führen Sie die ersten Schritte im Assistenten aus.
  2. Konfigurieren Sie im Schritt **eXtreme Scale - Einstellungen für die Sitzungsverwaltung** das Datengrid, das Sie verwenden möchten. Wählen Sie **Fernes eXtreme-Scale-Datengrid** oder **Integriertes eXtreme-Scale-Datengrid** aus.
     - Für die Option **Fernes eXtreme-Scale-Datengrid** wählen Sie die Katalogservicedomäne aus, die das Sitzungsdatengrid verwaltet. Wählen Sie anschließend in der Liste aktiver Sitzungsdatengrids ein Datengrid aus.

– Für die Option **Integriertes eXtreme-Scale-Datengrid** wählen Sie die ObjectGrid-Standardkonfiguration aus, oder Sie geben Sie Position Ihrer ObjectGrid-Konfigurationsdateien an.

3. Führen Sie die Schritte im Assistenten aus, um Ihre Anwendung zu installieren.

Sie können die Anwendung auch mit einem wsadmin-Script installieren. Im folgenden Beispiel erstellt der Parameter **-SessionManagement** dieselbe Konfiguration, die Sie auch über die Administrationskonsole erstellen können:

Für die Konfiguration eines fernen eXtreme-Scale-Datengrids:

```
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]]]')
```

Für das integrierte eXtreme-Scale-Szenario mit Standardkonfiguration:

```
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]]]')
```

Für das integrierte eXtreme-Scale-Szenario mit einer angepassten Konfiguration:

```
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]]]')
```

- **Gehen Sie wie folgt vor, um die Sitzungsverwaltung in einer vorhandenen Anwendung über die Administrationskonsole von WebSphere Application Server zu konfigurieren:**

  1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Anwendungen** > **Anwendungstypen** > **WebSphere-Unternehmensanwendungen** > *Anwendungsname* > **Eigenschaften des Webmoduls** > **Sitzungsverwaltung** > **eXtreme Scale - Einstellungen für die Sitzungsverwaltung**.
  2. Aktualisieren Sie die Felder, um die Sitzungspersistenz in einem Datengrid zu aktivieren.

Sie können die Anwendung auch mit einem wsadmin-Script aktualisieren. Im folgenden Beispiel erstellt der Parameter **-SessionManagement** dieselbe Konfiguration, die Sie auch über die Administrationskonsole erstellen können:

Für die Fernkonfiguration des eXtreme-Scale-Datengrids:

```
AdminApp.edit('DefaultApplication','[-SessionManagement[[true
XSRemoteSessionManagement cs0:!:grid0]]]')
```

Die übergebenen Zeichen **:!:** werden als Begrenzungszeichen verwendet. Die folgenden Werte werden übergeben:

Katalogservicename:!:*Gridname*

Für das integrierte eXtreme-Scale-Szenario mit Standardkonfiguration:

```
AdminApp.edit('DefaultApplication','[-SessionManagement[[true
XSEmbeddedSessionManagement :!:!:!:default]]]')
```

Die übergebenen Zeichen **:!:** werden als Begrenzungszeichen verwendet. Die folgenden Werte werden übergeben:

Katalogservicename:!:*Gridname*:!:default:!:
absoluter_Pfad_zur_ObjectGrid-XML-Datei:!:*absoluter_Pfad_zur_XML-Implementierungsdatei*

Für das integrierte eXtreme-Scale-Szenario mit einer angepassten Konfiguration:

```
AdminApp.edit('DefaultApplication','[-SessionManagement[[true
XSEmbeddedSessionManagement
:!:!:!:custom:!:c:\XS\objectgrid.xml:!:c:\XS\objectgriddeployment.xml]]]')
```

Die übergebenen Zeichen **:!:** werden als Begrenzungszeichen verwendet. Die folgenden Werte werden übergeben:

Katalogservicename:!:*Gridname*:!:custom:!:
absoluter_Pfad_zur_ObjectGrid-XML-Datei:!:*absoluter_Pfad_zur_XML-Implementierungsdatei*

Wenn Sie die Änderungen speichern, verwendet die Anwendung das konfigurierte Datengrid für die Sitzungspersistenz auf dem Gerät.

- **Gehen Sie wie folgt vor, um die Sitzungsverwaltung in einem vorhandenen Server zu konfigurieren:**
  1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Server** > **Servertypen** > **WebSphere-Anwendungsserver** > *Servername* > **Sitzungsverwaltung** > **eXtreme Scale - Einstellungen für die Sitzungsverwaltung**.
  2. Aktualisieren Sie die Felder, um die Sitzungspersistenz zu aktivieren.

  Sie können die Sitzungsverwaltung auch mit den folgenden Befehlen des Tools "wsadmin" in einem vorhandenen Server konfigurieren:

  Für die Konfiguration eines fernen eXtreme-Scale-Datengrids:

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSRemoteSessionManagement -XSRemoteSessionManagement
[-catalogService cs0 -csGridName grid0]]')
```

  Für die integrierte eXtreme-Scale-Konfiguration:

  – Standardkonfiguration bei Verwendung der XML-Standarddateien:

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement [-embeddedGridType default -objectGridXML -objectGridDeploymentXML ]]')
```

  – Angepasste Konfiguration bei Verwendung angepasster XML-Dateien:

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement
[-embeddedGridType custom -objectGridXML c:\XS\objectgrid.xml -objectGridDeploymentXML
c:\XS\objectgriddeployment.xml]]')
```

Wenn Sie die Änderungen speichern, verwendet der Server das konfigurierte Datengrid für die Sitzungspersistenz mit allen Anwendungen, die im Server ausgeführt werden.

- Wenn Sie weitere Aspekte der HTTP-Sitzungskonfiguration ändern möchten, können Sie die Datei `splicer.properties` bearbeiten. Sie können die Pfadposition der Datei `splicer.properties` über die angepasste Eigenschaft **`sessionFilterProps`** abrufen. Wenn Sie Sitzungspersistenz auf Serverbene konfiguriert haben, ist der Name der angepassten Eigenschaft `com.ibm.websphere.xs.sessionFilterProps`. Wenn Sie Sitzungspersistenz auf Anwendungsebene konfiguriert haben, ist der Name der angepassten Eigenschaft *<Anwendungsname>*`,com.ibm.websphere.xs.sessionFilterProps`. Diese angepassten Eigenschaften können an den folgenden Positionen angegeben werden:
  - In einer Umgebung von WebSphere Application Server Network Deployment: Ändern Sie die Datei `splicer.properties` im Pfad des Deployment-Manager-Profils.
  - In einer eigenständigen Umgebung von WebSphere Application Server: angepasste Eigenschaft im Anwendungsserver

  Sie können die angegebene Datei öffnen, Änderungen vornehmen und die Knoten dann synchronisieren, so dass die aktualisierte Eigenschaftendatei an die anderen Knoten in der Konfiguration weitergegeben wird. Alle Anwendungsserverknoten setzen voraus, dass sich die Datei `splicer.properties` im angegebenen Pfad befindet, damit die Sitzungen ordnungsgemäß persistent gespeichert werden können.

  Achtung: Wenn Sie die Persistenz für Sitzungen aktivieren möchten, die URL-Umschreibung verwenden, setzen Sie die Eigenschaft **`useURLEncoding`** in der Datei `splicer.properties` auf `true`.

  Weitere Informationen zu den Eigenschaften in der Datei `splicer.properties` finden Sie unter „Datei `splicer.properties`" auf Seite 332.

Ergebnisse

Sie haben den HTTP-Sitzungsmanager so konfiguriert, dass die Sitzungen in einem Datengrid persistent gespeichert werden. Es werden Einträge aus dem Datengrid entfernt, wenn die Sitzungen das zulässige Zeitlimit überschreiten. Weitere Informationen zum Aktualisieren des Sitzungszeitlimits in der Administrationskonsole von WebSphere Application Server finden Sie unter Einstellungen für die Sitzungsverwaltung.

Einstellungen für die Sitzungsverwaltung in eXtreme Scale:

Sie können Ihre Anwendungen von WebSphere Application Server so konfigurieren, dass sie WebSphere eXtreme Scale oder WebSphere DataPower XC10 Appliance für die Sitzungspersistenz verwenden.

Sie können diese Einstellungen im Installationsassistent für Unternehmensanwendungen oder auf den Detailseiten der Anwendung oder des Servers bearbeiten:

- Version 6.1: Klicken Sie auf **Anwendungen** > **Neue Anwendung installieren**.
- Version 6.1: Klicken Sie auf **Anwendungen** > **Unternehmensanwendungen** > ***Anwendungsname***.
- Version 6.1: Klicken Sie auf **Server** > **Anwendungsserver** > *Servername* > **Einstellungen des Webcontainers** > **Sitzungsverwaltung**.

- Version 7.0: Klicken Sie auf **Anwendungen** > **Neue Anwendung** > **Neue Unternehmensanwendung**, und wählen Sie den detaillierten Pfad für die Erstellung der Anwendung aus.
- Version 7.0: Klicken Sie auf **Anwendungen** > **Anwendungstypen** > **WebSphere-Unternehmensanwendungen** > *Anwendungsname* > **Eigenschaften des Webmoduls** > **Sitzungsverwaltung** > **Einstellungen für die Sitzungsverwaltung**.
- Version 7.0: Klicken Sie auf **Server** > **Servertypen** > **WebSphere-Anwendungsserver** > *Servername* > **Containereinstellungen** > **Einstellungen für die Sitzungsverwaltung**.

Sitzungsverwaltung aktivieren:

Ermöglicht der Sitzungsverwaltung, ein integriertes oder fernes WebSphere-eXtreme-Scale-Datengrid oder WebSphere DataPower XC10 Appliance für die Sitzungspersistenz zu verwenden.

Sitzungspersistenz verwalten mit:

Gibt an, wie die Sitzungspersistenz verwaltet wird. Sie können eine der folgenden Optionen auswählen:

- WebSphere DataPower XC10 Appliance
- Fernes eXtreme-Scale-Datengrid
- Integriertes eXtreme-Scale-Datengrid

Die verbleibenden Einstellungen, die Sie konfigurieren, richten sich nach dem ausgewählten Mechanismus für die Sitzungspersistenz.

Spezielle Einstellungen für WebSphere DataPower XC10 Appliance:

Die folgenden Einstellungen sind speziell für die Konfiguration von WebSphere DataPower XC10 Appliance für die Sitzungspersistenz bestimmt.

IP-Adresse oder Hostname von WebSphere DataPower XC10 Appliance:

Gibt die IP-Adresse oder den Hostnamen des Geräts an, das für das persistente Speichern von Sitzungen verwendet werden soll.

Verwaltungsberechtigungsnachweise für IBM WebSphere DataPower XC10 Appliance:

Gibt die Kombination von **Benutzername** und **Kennwort** an, die Sie verwenden, um sich bei der Benutzerschnittstelle von DataPower XC10 Appliance anzumelden. Klicken Sie auf **Verbindung testen...**, um die Verbindung zu Ihrem Gerät zu testen.

Benutzervorgabe für Sitzungspersistenz:

Gibt das Datengrid an, in dem Sitzungen persistent gespeichert werden. Sie können eine der folgenden Optionen auswählen:

- **Sitzungen in einem neuen Datengrid in IBM WebSphere DataPower XC10 Appliance persistent speichern**. Nach der Auswahl dieser Option können Sie den **Namen eines Datengrids** angeben.
- **Sitzungen in einem vorhandenen Datengrid in IBM WebSphere DataPower XC10 Appliance persistent speichern**. Nach der Auswahl dieser Option können Sie den **Namen eines vorhandenen Datengrids** eingeben oder suchen.

Konfiguration eines fernen eXtreme-Scale-Datengrids:

Die folgenden Einstellungen sind speziell für die Konfiguration des fernen eXtreme-Scale-Grids für Sitzungspersistenz bestimmt.

Katalogservicedomäne, die das ferne Sitzungsdatengrid verwaltet:

Gibt den Namen der Katalogservicedomäne an, die Sie für die Verwaltung Ihrer Sitzungen verwenden möchten.

Wenn keine Katalogservicedomänen angezeigt werden oder wenn Sie eine neue Katalogservicedomäne erstellen möchten, klicken Sie auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen**.

Fernes Datengrid, in dem Sitzungsinformationen gespeichert werden sollen:

Gibt den Namen des Datengrids in der Katalogservicedomäne an, in der Sie Ihre Sitzungsinformationen speichern möchten. Die Liste aktiver ferner Grids wird gefüllt, wenn Sie einen Katalogservice auswählen. Das ferne Datengrid muss bereits in der eXtreme-Scale-Konfiguration vorhanden sein.

Konfiguration eines integrierten eXtreme-Scale-Datengrids:

Die folgenden Einstellungen sind speziell für die Konfiguration einer integrierten eXtreme-Scale-Konfiguration bestimmt. Im integrierten eXtreme-Scale-Szenario werden die eXtreme-Scale-Prozesse in Prozessen von WebSphere Application Server ausgeführt.

Konfiguration eines integrierten eXtreme-Scale-Datengrids:

- **ObjectGrid-Standardkonfiguration verwenden**
- **Angepasste ObjectGrid-Konfigurationsdateien angeben**

  Vollständiger Pfad der in die Konfiguration zu kopierenden Datei `objectgrid.xml`
  Gibt den vollständigen Pfad der Datei `objectgrid.xml` für die zu verwendende Konfiguration an.

  Vollständiger Pfad der in die Konfiguration zu kopierenden Datei `objectgriddeployment.xml`
  Gibt den vollständigen Pfad der Datei `objectgriddeployment.xml` für die zu verwendende Konfiguration an.

WebSphere eXtreme Scale für die Verwaltung von SIP-Sitzungen verwenden

Sie können WebSphere eXtreme Scale als Alternative zum Datenreplikationsservice (DRS) als SIP-Replikationsmechanismus (Session Initiation Protocol) für die Replikation von SIP-Sitzungen verwenden.

Konfiguration des SIP-Sitzungsmanagements

Wenn Sie WebSphere eXtreme Scale als SIP-Replikationsmechanismus verwenden möchten, definieren Sie die angepasste Eigenschaft "com.ibm.sip.ha.replicator.type". Wählen Sie in der Administrationskonsole für jeden Server, dem Sie die angepasste Eigenschaft hinzufügen möchten, **Anwendungsserver** > ***mein_Anwendungsserver*** > **SIP-Container** > **Angepasste Eigenschaften** aus. Geben Sie `com.ibm.sip.ha.replicator.type` im Feld "Name" und `OBJECTGRID` im Feld "Wert" ein.

Verwenden Sie die folgenden Eigenschaften, um das Verhalten des ObjectGrids anzupassen, das zum Speichern von SIP-Sitzungen verwendet wird. Klicken Sie in

der Administrationskonsole für jeden Server, dem Sie die angepasste Eigenschaft hinzufügen möchten, auf **Anwendungsserver** > *mein_Anwendungsserver* > **SIP-Container** > **Angepasste Eigenschaften**. Füllen Sie die Felder **Name** und **Wert** aus. Für jeden Server müssen dieselben Eigenschaften definiert werden, damit sie ordnungsgemäß funktionieren.

Tabelle 23. Angepasste Eigenschaften für das SIP-Sitzungsmanagement mit ObjectGrid

Eigenschaft	Wert	Standardwert
com.ibm.sip.ha.replicator.type	`OBJECTGRID`: ObjectGrid als SIP-Sitzungsspeicher verwenden	
min.synchronous.replicas	Mindestanzahl synchroner Replikate	0
max.synchronous.replicas	Maximale Anzahl synchroner Replikate	0
max.asynchronous.replicas	Maximale Anzahl asynchroner Replikate	1
auto.replace.lost.shards	Weitere Informationen finden Sie im Abschnitt „Verteilte Implementierungen konfigurieren" auf Seite 242.	true
development.mode	• `true`: Verwendung aktiver Replikate auf demselben Knoten wie primäre Entität zulassen. • `false`: Replikate dürfen sich nicht zusammen mit der primären Entität auf demselben Knoten befinden.	false

HTTP-Sitzungsmanager mit WebSphere Portal konfigurieren

Sie können HTTP-Sitzungen über WebSphere Portal persistent in einem Datengrid speichern.

Vorbereitende Schritte

Ihre Umgebungen von WebSphere eXtreme Scale und WebSphere Portal müssen die folgenden Voraussetzungen erfüllen:

- Wie Sie WebSphere eXtreme Scale installieren, richtet sich nach Ihrem Implementierungsszenario. Sie können die Container-Server, die die Datengrids hosten, innerhalb oder außerhalb der Zelle von WebSphere Application Server ausführen:
  - Ausführung der Container-Server in der Zelle von WebSphere Application Server **(integriertes Szenario)**: Installieren Sie Client und Server von WebSphere eXtreme Scale auf Ihren Knoten von WebSphere Application Server und WebSphere Portal.
  - Ausführung der Container-Server außerhalb der Zelle von WebSphere Application Server **(fernes Szenario)**: Installieren Sie WebSphere eXtreme Scale Client auf Ihren Knoten von WebSphere Application Server und WebSphere Portal.

  Weitere Informationen finden Sie im Abschnitt „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren" auf Seite 167.
- WebSphere Portal Version 7 oder höher.
- Angepasste Portlets müssen in WebSphere Portal konfiguriert werden. Die Verwaltungsportlets, die mit WebSphere Portal bereitgestellt werden, können momentan nicht mit Datengrids integriert werden.

Informationen zu diesem Vorgang

Die Einführung von WebSphere eXtreme Scale in eine Umgebung von WebSphere Portal kann in den folgenden Szenarien hilfreich sein:

Wichtig: Obwohl die folgenden Szenarien Vorteile bieten, kann die Einführung von WebSphere eXtreme Scale in die Umgebung zu einer erhöhten Prozessorauslastung auf der WebSphere-Portal-Schicht führen.

- **Wenn Sitzungspersistenz erforderlich ist.**

  Wenn die Sitzungsdaten aus Ihren angepassten Portlets bei einem Ausfall von WebSphere Portal Server verfügbar bleiben müssen, können Sie die HTTP-Sitzungen persistent im Datengrid von WebSphere eXtreme Scale festschreiben. Die Daten werden auf vielen Servern repliziert, was die Datenverfügbarkeit erhöht.

- **In einer Topologie mit mehreren Rechenzentren.**

  Wenn sich Ihre Topologie über mehrere Rechenzentren an verschiedenen physischen Standorten erstreckt, können Sie die HTTP-Sitzungen von he WebSphere Portal im Datengrid von WebSphere eXtreme Scale persistent speichern. Die Sitzungen werden in den Datengrids in den Rechenzentren repliziert. Wenn ein Rechenzentrum ausfällt, werden die Sitzungen von einem anderen Rechenzentrum übernommen, das eine Kopie der Daten im Datengrid besitzt.

- **Speicherbedarf auf der WebSphere-Portal-Server-Schicht verringern.**

  Durch die Auslagerung der Sitzungsdaten in eine ferne Schicht von Container-Servern befindet sich ein Teil der Sitzungen auf den Servern von sWebSphere Portal. Dieser Datenauslagerung verringert den Speicherbedarf der WebSphere-Portal-Server-Schicht.

Vorgehensweise

1. Verbinden Sie die Anwendung `wps` von WebSphere Portal und alle angepassten Portlets, damit die Sitzungen im Datengrid gespeichert werden können.

   Sie können die Anwendung verbinden, indem Sie das HTTP-Sitzungsmanagement konfigurieren, wenn Sie die Anwendung implementieren. Sie können aber auch angepasste Eigenschaften verwenden, damit Ihre Anwendungen automatisch verbunden werden. Weitere Informationen zum Verbinden der Anwendung finden Sie unter „HTTP-Sitzungsmanager mit WebSphere Application Server konfigurieren” auf Seite 308.

2. Wenn Sie das ferne Szenario verwenden, in dem sich Ihre Container-Server außerhalb von WebSphere Application Server befinden, starten Sie die fernen eXtreme-Scale-Container für die HTTP-Sitzungspersistenz explizit. Starten Sie die Container mit den Konfigurationsdateien `XS/ObjectGrid/session/samples/objectGridStandAlone.xml` und `objectGridDeploymentStandAlone.xml`. Sie können beispielsweise den folgenden Befehl verwenden:

   ```
   startOgServer.sh xsContainer1 -catalogServiceEndPoints <Host>:<Port>
   -objectgridFile XS/ObjectGrid/session/samples/objectGridStandAlone.xml -deploymentPolicyFile
   XS/ObjectGrid/session/samples/objectGridDeploymentStandAlone.xml
   ```

   Weitere Informationen zum Starten von Container-Servern finden Sie unter „Container-Server starten” auf Seite 406. Wenn Sie ein integriertes Szenario verwenden, finden Sie im Artikel „Container-Server in WebSphere Application Server konfigurieren” auf Seite 282 weitere Informationen zum Konfigurieren und Starten von Container-Servern.

3. Starten Sie die Server von WebSphere Portal erneut. Weitere Informationen finden Sie unter WebSphere Portal Version 7: Starting and stopping servers, deployment managers, and node agents.

Ergebnisse

Sie können auf WebSphere Portal Server zugreifen, und die HTTP-Sitzungsdaten für die konfigurierten angepassten Portlets werden persistent im Datengrid gespeichert.

Wenn das gesamte Datengrid, in dem die Anwendungssitzungsdaten gehostet werden, über den Web-Container-Client nicht verfügbar ist, verwendet der Client stattdessen den Basis-Web-Container in WebSphere Application Server für die Sitzungsverwaltung. Das Datengrid kann in den folgenden Szenarien nicht erreichbar sein:

- Es besteht ein Netzproblem zwischen dem Web-Container und den fernen Container-Servern.
- Die fernen Container-Server-Prozesse wurden gestoppt.

Die Anzahl der im Speicher verwalteten Sitzungsreferenzen, die mit dem Parameter **sessionTableSize** angegeben wird, wird auch auch dann beibehalten, wenn die Sitzungen im Basis-Web-Container gespeichert werden. Die Sitzungen, die am längsten nicht mehr verwendet wurden, werden aus dem Sitzungscache des Web-Containers entfernt, wenn der Wert von **sessionTableSize** überschritten wird. Wenn das ferne Datengrid wieder verfügbar ist, können Sitzungen, die aus dem Web-Container-Cache entfernt wurden, Daten aus dem fernen Datengrid abrufen und die Daten in eine neue Sitzung laden. Wenn das gesamte ferne Datengrid nicht verfügbar ist und die Sitzung aus dem Sitzungscache entfernt wird, gehen die Sitzungsdaten des Benutzers verloren. Aufgrund dieses Problems sollten Sie nicht das gesamte Produktionsdatengrid beenden, wenn das System unter Last ausgeführt wird.

HTTP-Sitzungsmanager für verschiedene Anwendungsserver konfigurieren

WebSphere eXtreme Scale ist mit einer Sitzungsverwaltungsimplementierung gebündelt, die den Standardsitzungsmanager für einen Web-Container überschreibt. Diese Implementierung bietet Sitzungsreplikation, hohe Verfügbarkeit, eine bessere Skalierbarkeit und Konfigurationsoptionen. Sie können den Sitzungsreplikationsmanager von WebSphere eXtreme Scale und den generischen Start integrierter ObjectGrid-Container aktivieren.

Informationen zu diesem Vorgang

Sie können den HTTP-Sitzungsmanager mit anderen Anwendungsservern verwenden, in denen WebSphere Application Server nicht ausgeführt wird, z. B. WebSphere Application Server Community Edition. Zum Konfigurieren des Datengrids in anderen Anwendungsservern müssen Sie Ihre Anwendung verbinden und die JAR-Dateien von WebSphere eXtreme Scale in Ihre Anwendung integrieren.

Vorgehensweise

1. Fügen Sie Ihre Anwendung so zusammen, dass sie den Sitzungsmanager verwenden kann. Wenn Sie den Sitzungsmanager verwenden möchten, müssen Sie die entsprechenden Filterdeklarationen den Webimplementierungsdeskriptoren für die Anwendung hinzufügen. Außerdem werden die Konfigurationsparameter des Sitzungsmanagers in Form von Initialisierungsparametern für den Servlet-Kontext in den Implementierungsdeskriptoren an den Sitzungsmanager übergeben. Es gibt drei Methoden, mit denen Sie diese Informationen in Ihre Anwendung einführen können:
   - Script **addObjectGridFilter**:

Verwenden Sie ein Befehlszeilenscript, das mit eXtreme Scale bereitgestellt wird, um eine Anwendung mit Filterdeklarationen und Konfigurationen in Form von Initialisierungsparametern für den Servlet-Kontext zusammenzufügen. Das Script *WXS-Ausgangsverzeichnis*/session/bin/addObjectGridFilter.sh|bat akzeptiert zwei Parameter: den absoluten Pfad zur EAR-Datei oder WAR-Datei, die Sie verbinden möchten, und den absoluten Pfad zur Datei "splicer.properties", die verschiedene Konfigurationseigenschaften enthält. Das Syntaxformat dieses Scripts ist wie folgt:

Windows

```
addObjectGridFilter.bat <EAR-_oder_WAR-Datei> <Datei_splicer.properties>
```

UNIX

```
addObjectGridFilter.sh <EAR-_oder_WAR-Datei> <Datei_splicer.properties>
```

UNIX Beispiel mit einer Installation von eXtreme Scale in einem eigenständigen Verzeichnis unter UNIX:

a. `cd WXS-Ausgangsverzeichnis/session/bin`
b. `addObjectGridFilter.sh /tmp/mySessionTest.ear WXS-Ausgangsverzeichnis/session/samples/splicer.properties`

Der Servlet-Filter, der eingefügt wird, verwaltet die Standardwerte für Konfigurationswerte. Sie können diese Standardwerte mit Konfigurationsoptionen überschreiben, die Sie in der Eigenschaftendatei im zweiten Argument angeben. Eine Liste der Parameter, die Sie verwenden können, finden Sie im Abschnitt „Initialisierungsparameter für den Servlet-Kontext" auf Seite 329.

Sie können die Musterdatei `splicer.properties` ändern und verwenden, die mit der Installation von eXtreme Scale bereitgestellt wird. Sie können auch das Script "**addObjectGridServlets**" verwenden, das den Sitzungsmanager einfügt, indem Sie jedes Servlet erweitern. Das empfohlene Script ist jedoch das Script "**addObjectGridFilter**".

- Ant-Build-Script:

  Im Lieferumfang von WebSphere eXtreme Scale ist eine Datei `build.xml` enthalten, das von Apache Ant verwendet werden kann, und im Ordner *WAS-Stammverzeichnis*/bin einer Installation von WebSphere Application Server enthalten ist. Sie können die Datei `build.xml` bearbeiten, um die Konfigurationseigenschaften des Sitzungsmanagers zu ändern. Die Konfigurationseigenschaften sind identisch mit den Eigenschaftsnamen in der Datei `splicer.properties`. Rufen Sie nach der Änderung der Datei `build.xml` den Ant-Prozess auf, indem Sie `ant.sh`, `ws_ant.sh` (UNIX) bzw. `ant.bat`, `ws_ant.bat` (Windows) aufrufen.

- Webdeskriptor manuell aktualisieren:

  Editieren Sie die mit der Webanwendung gepackte Datei `web.xml`, um die Filterdeklaration, die Servlet-Zuordnung und die Initialisierungsparameter für den Servlet-Kontext zu integrieren. Verwenden Sie diese Methode nicht, weil sie fehleranfällig ist.

Eine Liste der Parameter, die Sie verwenden können, finden Sie im Abschnitt „Initialisierungsparameter für den Servlet-Kontext" auf Seite 329.

2. Integrieren Sie die JAR-Dateien des Sitzungsreplikationsmanagers von WebSphere eXtreme Scale in Ihre Anwendung ein. Sie können die Dateien in das Verzeichnis `WEB-INF/lib` des Anwendungsmoduls oder in den Klassenpfad des Anwendungsservers integrieren. Die erforderlichen JAR-Dateien variieren je nach Typ der verwendeten Container:
   - Ferne Container-Server: `ogclient.jar` und `sessionobjectgrid.jar`

- Integrierte Container-Server: `objectgrid.jar` und `sessionobjectgrid.jar`

3. Optional: Wenn Sie ferne Container-Server verwenden, starten Sie die Container-Server. Einzelheiten finden Sie im Abschnitt „Container-Server starten" auf Seite 406.
4. Implementieren Sie die Anwendung. Führen Sie dazu die Schritte aus, die Sie gewöhnlich für einen Server oder Cluster verwenden. Nach der Implementierung der Anwendung können Sie die Anwendung starten.
5. Greifen Sie auf die Anwendung zu. Sie können jetzt auf die Anwendung zugreifen, die mit dem Sitzungsmanager und mit WebSphere eXtreme Scale interagiert.

Nächste Schritte

Sie können die meisten Konfigurationsparameter des Sitzungsmanagers ändern, wenn Sie Ihre Anwendung für die Verwendung des Sitzungsmanagers instrumentieren. Zu diesen Attributen gehören Varianten des Replikationstyps (synchron oder asynchron), die Größe der speicherinternen Sitzungstabelle usw. Abgesehen von den Attributen, die während der Instrumentierung der Anwendung geändert werden können, sind die einzigen Attribute, die Sie nach der Anwendungsimplementierung ändern können, die Attribute, die sich auf die WebSphere eXtreme Scale-Serverclustertopologie beziehen, und die Art und Weise, in der Clients (Sitzungsmanager) eine Verbindung zu diesen Servern herstellen.

Verhalten bei einem fernen Szenario: Wenn das gesamte Datengrid, in dem die Anwendungssitzungsdaten gehostet werden, über den Web-Container-Client nicht verfügbar ist, verwendet der Client stattdessen den Basis-Web-Container des Anwendungsservers für die Sitzungsverwaltung. Das Datengrid kann in den folgenden Szenarien nicht erreichbar sein:

- Es besteht ein Netzproblem zwischen dem Web-Container und den fernen Container-Servern.
- Die fernen Container-Server-Prozesse wurden gestoppt.

Die Anzahl der im Speicher verwalteten Sitzungsreferenzen, die mit dem Parameter **`sessionTableSize`** angegeben wird, wird auch auch dann beibehalten, wenn die Sitzungen im Basis-Web-Container gespeichert werden. Die Sitzungen, die am längsten nicht mehr verwendet wurden, werden aus dem Sitzungscache des Web-Containers entfernt, wenn der Wert von **`sessionTableSize`** überschritten wird. Wenn das ferne Datengrid wieder verfügbar ist, können Sitzungen, die aus dem Web-Container-Cache entfernt wurden, Daten aus dem fernen Datengrid abrufen und die Daten in eine neue Sitzung laden. Wenn das gesamte ferne Datengrid nicht verfügbar ist und die Sitzung aus dem Sitzungscache entfernt wird, gehen die Sitzungsdaten des Benutzers verloren. Aufgrund dieses Problems sollten Sie nicht das gesamte Produktionsdatengrid beenden, wenn das System unter Last ausgeführt wird.

XML-Dateien für die Konfiguration des HTTP-Sitzungsmanagers

Wenn Sie einen Container-Server starten, der HTTP-Sitzungsdaten speichert, können Sie die Standard-XML-Dateien verwenden, oder Sie können angepasste XML-Dateien verwenden. Diese Dateien erstellen bestimmte ObjectGrid-Namen, die Anzahl der Replikate usw.

Position der Beispieldateien

Diese XML-Dateien werden für eine eigenständige Installation in *WXS-Installationsstammverzeichnis*`/ObjectGrid/session/samples` und für eine Instal-

lation von WebSphere eXtreme Scale in einer Zelle von WebSphere Application Server in *WAS-Stammverzeichnis*/optionalLibraries/ObjectGrid/session/samples gepackt.

Integriertes XML-Paket

Wenn Sie ein integriertes Szenario konfigurieren, wird der Container-Server auf der Web-Container-Schicht gestartet. Verwenden Sie die Datei objectGrid.xml und die Datei objectGridDeployment.xml, die standardmäßig bereitgestellt werden. Sie können diese Dateien aktualisieren, um das Verhalten des HTTP-Sitzungsmanagers anzupassen.

```
<?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>
```

Abbildung 35. Datei objectGrid.xml

Werte, die Sie ändern können:

ObjectGrid-Attribut "name"
Der Wert muss mit den folgenden Werten in anderen Konfigurationsdateien übereinstimmen:

- Eigenschaft **objectGridName** in der Datei splicer.properties, die verwendet wird, um die Webanwendung zu binden
- Attribut **objectgridName** in der Datei objectGridDeployment.xml

Wenn Sie mehrere Anwendungen haben und möchten, dass die Sitzungsdaten in unterschiedlichen Datengrids gespeichert werden, müssen diese Anwendungen unterschiedliche Wert für das ObjectGrid-Attribut "name" haben.

7.1.1+ **ObjectGrid-Attribut "txTimeout"**
Dieser Wert bestimmt, wie viele Sekunden eine Transaktion offen bleiben darf, bevor der Container-Server die Überschreitung des Transaktionszeitlimits auslöst. Der Standardwert sind 30 Sekunden und kann je nach Umgebung geändert werden. Wenn die HTTP-Sitzungspersistenz mit einem höheren Wert als null für den Servletkontextinitialisierungsparameter **replicationInterval** definiert wird, werden Transaktionen in einem Thread zu Stapeln zusammengefasst. Die Dauer der Transaktion wird durch die Größe und die Anzahl der HTTP-Sitzungen bestimmt. Der Standardwert sind 10 Sekunden, und 30 Sekunden reicht für die Ausführung von Transaktionen aus. Wenn die Eigenschaft **replicationInterval** auf 0 gesetzt ist, wird eine Transaktion gewöhnlich gestartet, wenn eine Webanwendung ein gültiges HttpSession-Objekt abruft. Die Transaktion wird am Ende der Webanwendungsanforderung festgeschrieben. Wenn in Ihrer Um-

gebung Anforderungen ausgeführt werden, die länger als 30 Sekunden dauern, setzen Sie diesen Wert entsprechend.

Werte, die Sie nicht ändern können:

ObjectGridEventListener
Die ObjectGridEventListener-Zeile kann nicht geändert werden und wird intern verwendet.

objectgridSessionMetadata
Die objectgridSessionMetadata-Zeile verweist auf die Map, in der die Metadaten der HTTP-Sitzung gespeichert sind. Es gibt einen Eintrag für jede in diesem Datengrid gespeicherte HTTP-Sitzung in dieser Map.

objectgridSessionTTL.*
Dieser Wert kann nicht geändert werden und ist für spätere Verwendung bestimmt.

objectgridSessionAttribute.*
Der `objectgridSessionAttribute.*`-Text definiert eine dynamische Map. Dieser Wert wird verwendet, um die Map zu erstellen, in der HTTP-Sitzungsattribute gespeichert werden, wenn der Parameter **`fragmentedSession`** in der Datei `splicer.properties` auf `true` gesetzt ist. Diese dynamische Map hat den Namen `objectgridSessionAttribute`. Auf der Basis dieser Schablone wird eine weitere Map mit dem Namen `objectgridSessionAttributeEvicted` erstellt, in der Sitzungen gespeichert werden, die das zulässige Zeitlimit überschritten haben, aber vom Webcontainer nicht ungültig gemacht wurden.

Die **`MapEventListener`**-Zeile ist intern und kann nicht geändert werden.

```
<?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>
```

Abbildung 36. Datei `objectGridDeployment.xml`

Werte, die Sie ändern können:

ObjectGrid-Attribut "name"
Der Wert muss mit den folgenden Werten in anderen Konfigurationsdateien übereinstimmen:

- Eigenschaft **`objectGridName`** in der Datei `splicer.properties`, die verwendet wird, um die Webanwendung zu binden
- Das ObjectGrid-Attribut **`name`** in der Datei `objectGrid.xml`.

Wenn Sie mehrere Anwendungen haben und möchten, dass die Sitzungsdaten in unterschiedlichen Datengrids gespeichert werden, müssen diese Anwendungen unterschiedliche Wert für das ObjectGrid-Attribut "name" haben.

Attribute des Elements "mapSet"

Sie können alle mapSet-Eigenschaften mit Ausnahme des Attributs "placementStrategy" ändern.

Name Kann in jeden Wert aktualisiert werden.

numberOfPartitions
Gibt die Anzahl primärer Partitionen an, die in jedem Server gestartet werden, der die Webanwendung enthält. Wenn Sie Partitionen hinzufügen, werden die Daten für den Fall eines Failovers breiter verteilt. Der Standardwert sind 5 Partitionen und reicht für die meisten Anwendungen aus.

minSyncReplicas, maxSyncReplicas und maxAsyncReplicas
Gibt die Anzahl und den Typ der Replikate an, in denen die HTTP-Sitzungsdaten gespeichert werden. Standardmäßig wird 1 asynchrones Replikat verwendet, das für die meisten Anwendungen ausreicht. Synchrone Replikation findet im Anforderungspfad statt, was die Antwortzeiten für die Webanwendung erhöhen kann.

developmentMode
Informiert den eXtreme-Scale-Verteilungsservice darüber, ob die Replikat-Shards für eine Partition auf demselben Knoten wie das primäre Shards verteilt werden können. Sie können den Wert in einer Entwicklungsumgebung auf "true" setzen, sollten diese Funktion in einer Produktionsumgebung aber inaktivieren, weil der Ausfall eines Knotens zum Verlust von Sitzungsdaten führen könnte.

placementStrategy
Ändern Sie den Wert dieses Attributs nicht.

Der Rest der Datei verweist auf dieselben Map-Namen, die auch in der Datei `objectGrid.xml` enthalten sind. Diese Namen können nicht geändert werden.

Werte, die Sie nicht ändern können:

- Attribut "placementStrategy" im Element "mapSet"

Fernes XML-Paket

Wenn Sie den Fernmodus verwenden, in dem die Container als eigenständige Prozesse ausgeführt werden, müssen Sie die Datei `objectGridStandAlone.xml` und die Datei `objectGridDeploymentStandAlone.xml` verwenden, um die Prozesse zu starten. Sie können diese Dateien aktualisieren, um die Konfiguration zu ändern.

```
<?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>
```

Abbildung 37. Datei `objectGridStandAlone.xml`

Werte, die Sie ändern können:

ObjectGrid-Attribut "name"
Der Wert muss mit den folgenden Werten in anderen Konfigurationsdateien übereinstimmen:

- Eigenschaft **`objectGridName`** in der Datei `splicer.properties`, die verwendet wird, um die Webanwendung zu binden
- Attribut **`objectgridName`** in der Datei `objectGridStandAlone.xml`

Wenn Sie mehrere Anwendungen haben und möchten, dass die Sitzungsdaten in unterschiedlichen Datengrids gespeichert werden, müssen diese Anwendungen unterschiedliche Wert für das ObjectGrid-Attribut "name" haben.

7.1.1+ **ObjectGrid-Attribut "txTimeout"**
Dieser Wert bestimmt, wie viele Sekunden eine Transaktion offen bleiben darf, bevor der Container-Server die Überschreitung des Transaktionszeitlimits auslöst. Der Standardwert sind 30 Sekunden und kann je nach Umgebung geändert werden. Wenn die HTTP-Sitzungspersistenz mit einem höheren Wert als null für den Servletkontextinitialisierungsparameter **`replicationInterval`** definiert wird, werden Transaktionen in einem Thread zu Stapeln zusammengefasst. Die Dauer der Transaktion wird durch die Größe und die Anzahl der HTTP-Sitzungen bestimmt. Der Standardwert sind 10 Sekunden, und 30 Sekunden reicht für die Ausführung von Transaktionen aus. Wenn die Eigenschaft **`replicationInterval`** auf `0` gesetzt ist, wird eine Transaktion gewöhnlich gestartet, wenn eine Webanwendung ein gültiges HttpSession-Objekt abruft. Die Transaktion wird am Ende der Webanwendungsanforderung festgeschrieben. Wenn in Ihrer Umgebung Anforderungen ausgeführt werden, die länger als 30 Sekunden dauern, setzen Sie diesen Wert entsprechend.

Werte, die Sie nicht ändern können:

ObjectGridEventListener
Die ObjectGridEventListener-Zeile kann nicht geändert werden und wird intern verwendet.

objectgridSessionMetadata
Die objectgridSessionMetadata-Zeile verweist auf die Map, in der die Metadaten der HTTP-Sitzung gespeichert sind. Es gibt einen Eintrag für jede in diesem Datengrid gespeicherte HTTP-Sitzung in dieser Map.

objectgridSessionTTL.*
Dieser Wert kann nicht geändert werden und ist für spätere Verwendung bestimmt.

objectgridSessionAttribute.*
Der `objectgridSessionAttribute.*`-Text definiert eine dynamische Map. Dieser Wert wird verwendet, um die Map zu erstellen, in der HTTP-Sitzungsattribute gespeichert werden, wenn der Parameter **`fragmentedSession`** in der Datei `splicer.properties` auf `true` gesetzt ist. Diese dynamische Map hat den Namen `objectgridSessionAttribute`. Auf der Basis dieser Schablone wird eine weitere Map mit dem Namen `objectgridSessionAttributeEvicted` erstellt, in der Sitzungen gespeichert werden, die das zulässige Zeitlimit überschritten haben, aber vom Webcontainer nicht ungültig gemacht wurden.

Die **`MetadataMapListener`**-Zeile ist intern und kann nicht geändert werden.

```
<?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>
```

Abbildung 38. Datei `objectGridDeploymentStandAlone.xml`

Werte, die Sie ändern können:

ObjectGrid-Attribut "name"
Der Wert muss mit den folgenden Werten in anderen Konfigurationsdateien übereinstimmen:

- Eigenschaft **`objectGridName`** in der Datei `splicer.properties`, die verwendet wird, um die Webanwendung zu binden
- Das ObjectGrid-Attribut **`name`** in der Datei `objectGrid.xml`.

Wenn Sie mehrere Anwendungen haben und möchten, dass die Sitzungsdaten in unterschiedlichen Datengrids gespeichert werden, müssen diese Anwendungen unterschiedliche Wert für das ObjectGrid-Attribut "name" haben.

Attribute des Elements "mapSet"
Sie können alle mapSet-Eigenschaften mit Ausnahme des Attributs "placementStrategy" ändern.

Name Kann in jeden Wert aktualisiert werden.

numberOfPartitions
Gibt die Anzahl primärer Partitionen an, die in jedem Server gestartet werden, der die Webanwendung enthält. Wenn Sie Partitionen hinzufügen, werden die Daten für den Fall eines Failovers breiter verteilt. Der Standardwert sind 5 Partitionen und reicht für die meisten Anwendungen aus.

minSyncReplicas, maxSyncReplicas und maxAsyncReplicas
Gibt die Anzahl und den Typ der Replikate an, in denen die HTTP-Sitzungsdaten gespeichert werden. Standardmäßig wird 1 asynchrones Replikat verwendet, das für die meisten Anwendungen ausreicht. Synchrone Replikation findet im Anforderungspfad statt, was die Antwortzeiten für die Webanwendung erhöhen kann.

developmentMode
Informiert den eXtreme-Scale-Verteilungsservice darüber, ob die Replikat-Shards für eine Partition auf demselben Knoten wie das primäre Shards verteilt werden können. Sie können den Wert in einer Entwicklungsumgebung auf "true" setzen, sollten diese Funktion in einer Produktionsumgebung aber inaktivieren, weil der Ausfall eines Knotens zum Verlust von Sitzungsdaten führen könnte.

placementStrategy
Ändern Sie den Wert dieses Attributs nicht.

Der Rest der Datei verweist auf dieselben Map-Namen, die auch in der Datei `objectGrid.xml` enthalten sind. Diese Namen können nicht geändert werden.

Werte, die Sie nicht ändern können:

- Attribut "placementStrategy" im Element "mapSet"

Initialisierungsparameter für den Servlet-Kontext

Die folgende Liste mit Initialisierungsparametern für den Servlet-Kontext können in der Datei "splicer.properties" abhängig von der ausgewählten Verbindungsmethode (Splicing) angegeben werden.

Parameter

`objectGridType`

Die gültigen Werte für diesen Parameter sind die Zeichenfolgewerte `REMOTE` und `EMBEDDED`. Der Standardwert ist `REMOTE`.

Wenn Sie diesen Parameter auf `REMOTE` setzen, werden die Sitzungsdaten außerhalb des Servers gespeichert, in dem die Webanwendung ausgeführt wird.

Wenn dieser Parameter auf `EMBEDDED` gesetzt wird, wird ein integrierter eXtreme-Scale-Container in dem Anwendungsserverprozess gestartet, in dem die Webanwendung ausgeführt wird.

`objectGridName`

Ein Zeichenfolgewert, der den Namen der ObjectGrid-Instanz definiert, die für eine bestimmte Webanwendung verwendet wird. Der Standardname ist "session".

Diese Eigenschaft muss den objectGridName-Wert in der ObjectGrid-XML-Datei und in der XML-Implementierungsdatei widerspiegeln, die zum Starten der Container-Server von eXtreme Scale verwendet werden.

`catalogHostPort`

Der Katalogserver kann eine Verbindung zu einem Katalogserver aufgebaut werden, um eine clientseitige ObjectGrid-Instanz abzurufen. Der Wert muss im Format `Host:Port<,Host:Port>` angegeben werden. Der Host ist der Listener-Host, auf dem der Katalogserver ausgeführt wird. Der Port ist der Listener-Port für diesen Katalogserverprozess. Diese Liste kann beliebig lang sein und

wird nur für Bootstrapping verwendet. Die erste geeignete Adresse wird verwendet. Sie ist in WebSphere Application Server optional, wenn die Eigenschaft **catalog.services.cluster** konfiguriert ist.

replicationInterval

Ein ganzzahliger Wert (in Sekunden), der das Intervall definiert, in dem aktualisierte Sitzungen in das ObjectGrid geschrieben werden. Der Standardwert sind 10 Sekunden. Die gültigen Werte sind 0 bis 60. 0 bedeutet, dass aktualisierte Sitzungen am Ende des Methodenaufrufs des Servlet-Service für jede Anforderung in das ObjectGrid geschrieben werden. Ein höherer **replicationInterval**-Wert verbessert die Leistung, weil weniger Aktualisierungen in das Datengrid geschrieben werden. Die Konfiguration ist bei einem höheren Wert jedoch weniger fehlertolerant.

Diese Einstellung gilt nur, wenn "objectGridType" auf REMOTE gesetzt ist.

sessionTableSize

Ein ganzzahliger Wert, der die Anzahl der Sitzungsreferenzen definiert, die im Speicher verwaltet werden. Der Standardwert ist 1000.

Diese Einstellung gilt nur für REMOTE-Topologien, weil die EMBEDDED-Topologie die Sitzungsdaten bereits auf derselben Schicht wie den Webcontainer enthält.

Sitzungen werden auf der Basis der LRU-Logik aus der speicherinternen Tabelle entfernt. Wenn eine Sitzung aus der speicherinternen Tabelle entfernt wird, wird sie im im Web-Container ungültig gemacht. Die Daten werden jedoch nicht aus dem Grid entfernt, sodass nachfolgende Anforderungen für diese Sitzungen die Daten weiterhin abrufen können. Dieser Wert muss höher sein als der Wert für die maximale Größe des Thread-Pools des Web-Containers, damit Konkurrenzsituationen im Sitzungscache verringert werden.

fragmentedSession

Die gültigen Werte für diesen Parameter sind die Zeichenfolgewerte true und false. Der Standardwert ist true. Mit dieser Einstellung können Sie steuern, ob das Produkt Sitzungsdaten als vollständigen Eintrag oder jedes Attribut einzeln speichert.

Setzen Sie den Parameter "fragmentedSession" auf true, wenn die Webanwendungssitzung viele Attribute oder große Attribute hat. Setzen Sie den Parameter "fragmentedSession" nur dann auf false, wenn eine Sitzung wenig Attribute hat, weil alle Attribute unter demselben Schlüssel im Datengrid gespeichert werden.

In der früheren filterbasierten Implementierung hat diese Eigenschaft den Namen "persistenceMechanism" mit den möglichen Werten "ObjectGridStore" (fragmentiert) und "ObjectGridAtomicSessionStore" (nicht fragmentiert).

securityEnabled

Die gültigen Werte für diesen Parameter sind die Zeichenfolgewerte true und false. Der Standardwert ist false. Diese Einstellung aktiviert die Sicherheit für eXtreme-Scale-Clients. Sie muss mit der Einstellung **securityEnabled** in der eXtreme-Scale-Servereigenschaftendatei übereinstimmen. Stimmen die Einstellungen nicht überein, tritt eine Ausnahme ein.

credentialGeneratorClass

Der Name der Klasse, die die Schnittstelle com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator implementiert. Diese Klasse wird verwendet, um Berechtigungsnachweise für Clients abzurufen.

credentialGeneratorProps

Die Eigenschaften für die CredentialGenerator-Implementierungsklasse. Die Eigenschaften werden mit der Methode "setProperties(String)" auf das Objekt gesetzt. Der credentialGeneratorProps-Wert wird nur verwendet, wenn der Wert der Eigenschaft **credentialGeneratorClass** ungleich null ist.

objectGridXML

Die Position der Datei `objectgrid.xml`. Die integrierte XML-Datei in der eXtreme-Scale-Bibliothek wird automatisch geladen, wenn objectGridType=EMBEDDED und die Eigenschaft **objectGridXML** nicht angegeben sind.

objectGridDeploymentXML

Gibt die Position der XML-Datei für die ObjectGrid-Implementierungsrichtlinie an. Die integrierte XML-Datei in der eXtreme-Scale-Bibliothek wird automatisch geladen, wenn objectGridType=EMBEDDED und die Eigenschaft **objectGridDeploymentXML** nicht angegeben sind.

traceSpec

Gibt die IBM WebSphere-Tracespezifikation als Zeichenfolgewert an. Verwenden Sie diese Einstellung für andere Anwendungsserver als WebSphere Application Server.

traceFile

Gibt die Position der Tracedatei als Zeichenfolgewert an. Verwenden Sie diese Einstellung für andere Anwendungsserver als WebSphere Application Server.

cookieDomain

Gibt an, ob Sitzungen auf mehreren Hosts zugänglich sein müssen. Setzen Sie den Wert auf den Namen der gemeinsamen Domäne der Hosts.

reuseSessionID

Setzen Sie diese Eigenschaft auf `true`, wenn der zugrunde liegende Web-Container Sitzungs-IDs für Anforderungen an verschiedene Hosts wiederverwendet. Der Standardwert ist `false`. Der Wert dieser Eigenschaft muss mit dem Wert im Web-Container übereinstimmen. Wenn Sie WebSphere Application Server verwenden und die HTTP-Sitzungspersistenz für eXtreme Scale über die Administrationskonsole oder Scripting mit dem Tool **wsadmin** konfigurieren, wird die angepasste Eigenschaft `HttpSessionIdReuse=true` des Web-Containers standardmäßig hinzugefügt. Die Eigenschaft **reuseSessionID** wird auch auf `true` gesetzt. Wenn Sie nicht möchten, dass die Sitzungs-IDs wiederverwendet werden, setzen Sie die angepasste Eigenschaft `HttpSessionIdReuse=false` im Webcontainer, bevor Sie die Sitzungspersistenz für eXtreme Scale konfigurieren.

shareSessionsAcrossWebApps

Gibt an, ob Sitzungen von mehreren Webanwendungen gemeinsam genutzt werden. Die gültigen Werte sind die Zeichenfolgewerte `true` und `false`. Die Standardeinstellung ist `false`. Gemäß der Servletspezifikation können HTTP-Sitzungen nicht von mehreren Webanwendungen gemeinsam genutzt werden. Damit diese gemeinsame Nutzung zugelassen wird, wird eine Erweiterung der Servletspezifikation bereitgestellt.

useURLEncoding

Setzen Sie diese Eigenschaft auf `true`, wenn Sie URL-Umschreibung aktivieren möchten. Der Standardwert ist `false` und gibt an, dass Cookies zum Speichern

von Sitzungsdaten verwendet werden. Der Wert dieses Parameters muss mit den Einstellungen des Web-Containers für die Sitzungsverwaltung übereinstimmen.

Datei splicer.properties

Die Datei splicer.properties enthält alle Konfigurationsoptionen für die Konfiguration eines auf Servletfiltern basierenden Sitzungsmanagers.

Beispieldatei "splicer.properties"

Wenn Sie sich für die Verwendung einer der zusätzlichen Eigenschaften entscheiden, die in dieser Datei beschrieben werden, müssen Sie die Kommentarzeichen in den Zeilen mit den Eigenschaften entfernen, die Sie aktivieren möchten.

```
# Eigenschaftendatei, die alle Konfigurationsoptionen
# enthält, die für einen ObjectGrid-Sitzungsmanager, der auf
# Servlet-Filtern basiert, konfiguriert werden können.
#
# In dieser Eigenschaftendatei können alle Standardwerte
# eingefügt werden, die diesen Konfigurationseinstellungen zugeordnet
# werden sollen, und einzelne Einstellungen können mit
# Ant-Task-Eigenschaften überschrieben werden, wenn diese
# Eigenschaftendatei mit der ANT-Task filtersplicer verwendet wird.

# Zeichenfolgewert: "REMOTE" oder "EMBEDDED". Der Standardwert ist REMOTE.
# Wenn diese Einstellung auf "REMOTE" gesetzt wird, werden die Sitzungsdaten
# außerhalb des Servers gespeichert, auf dem die Webanwendung ausgeführt wird.
# Wenn die Eigenschaft auf "EMBEDDED" gesetzt ist, wird ein integrierter
# WebSphere-eXtreme-Scale-Container in dem Anwendungsserverprozess gestartet,
# in dem die Webanwendung ausgeführt wird.

objectGridType = REMOTE

# Ein Zeichenfolgewert, der den Namen der ObjectGrid-Instanz definiert,
# die für eine bestimmte Webanwendung verwendet wird.
# Der Standardname ist "session". Diese Eigenschaft muss den objectGridName
# in den Dateien objectgrid.xml und deployment.xml widerspiegeln, die
# zum Starten der eXtreme-Scale-Container verwendet werden.


objectGridName = session

# Es kann eine Verbindung zu einem Katalogserver aufgebaut werden, um
# eine clientseitige ObjectGrid-Instanz abzurufen. Der Wert muss im
# Format "Host:Port<,Host:Port>" angegeben werden, wobei "Host"
# für den Listener-Host steht, auf dem der Katalogserver ausgeführt wird,
# und "Port" für den Listener-Port für diesen Katalogserverprozesse.
# Diese Liste kann beliebig lang sein und wird nur für Bootstrapping
# verwendet.
# Die erste geeignete Adresse wird verwendet. Sie ist in WebSphere
# optional, wenn die Eigenschaft "catalog.services.cluster" konfiguriert ist.

# catalogHostPort = Host:Port<,Host:Port>

# Ein ganzzahliger Wert (in Sekunden), der das Intervall definiert, in
# dem aktualisierte Sitzungen in das ObjectGrid geschrieben werden. Der
# Standardwert ist 10. Diese Eigenschaft wird nur verwendet, wenn
# objectGridType auf REMOTE gesetzt ist. Die gültigen Werte sind
# 0 bis 60. 0 bedeutet, dass aktualisierte Sitzungen am Ende des
# Methodenaufrufs des Servlet-Service für jede Anforderung in das
# ObjectGrid geschrieben werden.

replicationInterval = 10

# Ein ganzzahliger Wert, der die Anzahl der Sitzungsreferenzen definiert,
```

```
# die im Speicher verwaltet werden. Der Standardwert ist 1000. Diese
# Eigenschaft wird nur verwendet, wenn objectGridType auf REMOTE gesetzt
# ist. Wenn die Anzahl der gespeicherten Sitzungen im Hauptspeicher
# im Webcontainer diesen Wert überschreitet, wird die am längsten nicht
# verwendete Sitzung im Webcontainer ungültig gemacht.  Wenn eine
# Anforderung für diese Sitzung eingeht, nachdem sie ungültig gemacht
# wurde, wird eine neue Sitzung erstellt (mit einer neuer Sitzungs-ID,
# wenn reuseSessionId=false definiert ist), die mit den Attributen der
# ungültig gemachten Sitzung gefüllt wird. Dieser Wert muss immer auf
# einen Wert gesetzt werden, der höher ist als die maximale Größe des
# Thread-Pools des Webcontainers, damit Konkurrenzsituationen in diesem
# Sitzungscache vermieden werden.

sessionTableSize = 1000

# Ein Zeichenfolgewert. Die gültigen Werte sind "true" und "false".
# Der Standardwert ist "true".
# Mit diesem Wert wird gesteuert, ob Sitzungsdaten als vollständiger
# Eintrag oder jedes Attribut gesondert gespeichert wird.
# Diese Eigenschaft wurde in der früheren filterbasierten Implementierung
# als persistenceMechanism bezeichnet und hatte die gültigen Werte
# ObjectGridStore (fragmentiert) und ObjectGridAtomicSessionStore
# (nicht fragmentiert).

fragmentedSession = true

# Ein Zeichenfolgewert. Die gültigen Werte sind "true" und "false".
# Der Standardwert ist "false".
# Aktiviert die eXtreme-Scale-Clientsicherheit. Diese Einstellung muss
# mit der securityEnabled-Einstellung in der Servereigenschaftendatei
# von eXtreme Scale übereinstimmen. Stimmen die Einstellungen nicht
# überein, tritt eine Ausnahme ein.

securityEnabled = false

# Gibt die Unterstützung für die Authentifizierung mit
# Clientberechtigungsnachweisen an.
#   Die gültigen Werte sind im Folgenden beschrieben:
#   Never      - Der Client unterstützt keine Authentifizierung mit Berechtigungsnachweisen.
#   Supported* - Der Client unterstützt die Authentifizierung mit Berechtigungsnachweisen
#                nur dann, wenn diese auch vom Server unterstützt wird.
#   Required   - Der Client setzt die Authentifizierung mit Berechtigungsnachweisen voraus.
#   Der Standardwert ist "Supported".

# credentialAuthentication =

# Gibt den Wiederholungszähler für die Authentifizierung an, wenn der
# Berechtigungsnachweis abgelaufen ist. Wenn die Einstellung auf 0
# gesetzt ist, findet keine Authentifizierungswiederholung statt.

# authenticationRetryCount =

# Gibt den Namen der Klasse an, die die Schnittstelle
# "com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator"
# implementiert. Diese Klasse wird verwendet, um Berechtigungsnachweise
# für Clients abzurufen.

# credentialGeneratorClass =

# Gibt die Eigenschaften für die Implementierungsklasse
# "CredentialGenerator" an. Die Eigenschaften werden mit der Methode
# "setProperties(String)" auf das Objekt gesetzt. Der Wert von
# "credentialGeneratorProps" wird nur verwendet, wenn der Wert der
# Eigenschaft "credentialGeneratorClass" nicht null ist.

# credentialGeneratorProps =
```

```
# Die Dateiposition der Datei "objectgrid.xml".
# Die integrierte XML-Datei, die in die eXtreme-Scale-Bibliothek
# gepackt wurde, wird automatisch geladen, wenn diese Eigenschaft
# nicht angegeben und wenn objectGridType=EMBEDDED definiert ist.

# objectGridXML =

# Die Dateiposition der ObjectGrid-XML-Implementierungsrichtliniendatei.
# Die integrierte XML-Datei, die in die eXtreme-Scale-Bibliothek
# gepackt wurde, wird automatisch geladen, wenn diese Eigenschaft
# nicht angegeben und wenn objectGridType=EMBEDDED definiert ist.

# objectGridDeploymentXML =

# Eine Zeichenfolge für die IBM WebSphere-Tracespezifikation, die
# für alle anderen Anwendungsserver neben WebSphere hilfreich ist.

# traceSpec =

# Eine Zeichenfolge für die Position der Tracedatei, die für alle
# anderen Anwendungen neben WebSphere hilfreich ist.

# traceFile=

# Diese Eigenschaft muss gesetzt werden, wenn die Sitzungen
# auf mehreren Hosts zugänglich sein müssen. Der Wert ist
# der Name der gemeinsamen Domäne der Hosts.

# cookieDomain=

# Setzen Sie diese Eigenschaft auf "true", wenn der zugrunde liegende
# Webcontainer Sitzungs-IDs für Anforderungen an verschiedene Hosts
# wiederverwendet. Der Standardwert ist "false". Der Wert dieser
# Einstellung muss mit dem Wert übereinstimmen, der im Webcontainer
# definiert ist.

# reuseSessionId=

# Zeichenfolgewert: "true" oder "false". Der Standardwert ist
# "false". Gemäß der Servletspezifikation können HTTP-Sitzungen nicht
# von mehreren Webanwendungen gemeinsam genutzt werden. Damit diese
# gemeinsame Nutzung zugelassen wird, wird eine Erweiterung der
# Servletspezifikation bereitgestellt.

# shareSessionsAcrossWebApps = false

# Setzen Sie diese Einstellung auf "true", wenn Sie URL-Umschreibung aktivieren
# möchten. Der Standardwert ist "false", d. h., es werden Cookies zum Speichern
# von Daten verwendet. Der Wert dieser Einstellung muss der Einstellung
# entsprechen, die in den Einstellungen des Webcontainers für die
# Sitzungsverwaltung definiert ist.

# useURLEncoding = false
```

Dynamischen Cache-Provider für WebSphere eXtreme Scale konfigurieren

Die Installation und Konfiguration des dynamischen Cache-Providers für eXtreme Scale richtet sich nach Ihren Anforderungen und Ihrer konfigurierten Umgebung.

Vorbereitende Schritte

- Zur Verwendung des dynamischen Cache-Providers muss WebSphere eXtreme Scale über die Knotenimplementierungen von WebSphere Application Server (einschließlich des Deployment-Manager-Knotens) installiert werden. Weitere In-

formationen finden Sie im Abschnitt „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client mit WebSphere Application Server installieren" auf Seite 167.

- Die globale Sicherheit muss in der Administrationskonsole von WebSphere Application Server aktiviert werden, wenn in den Katalogservern in Ihrer Katalogservicedomäne Secure Sockets Layer (SSL) aktiviert ist oder wenn Sie SSL für eine Katalogservicedomäne mit SSL-Unterstützung verwenden möchten. Sie legen die SSL-Anforderung für einen Katalogserver fest, indem Sie das Attribut "transportType" in der Servereigenschaftendatei auf `SSL-Required` setzen. Weitere Informationen zum Konfigurieren der globalen Sicherheit finden Sie unter Globale Sicherheitseinstellungen konfigurieren.

Informationen zu diesem Vorgang

Informationen zur Verwendung des dynamischen Cache-Providers von eXtreme Scale mit IBM WebSphere Commerce finden Sie in den folgenden Artikeln in der Dokumentation zu IBM WebSphere Commerce:

- Dynamischen Cacheservice und Servlet-Caching aktivieren
- Datencache von WebSphere Commerce aktivieren

Wenn Sie Ihr Caching nicht an eine spezielle definierte Objektcache- oder Servlet-Cache-Instanz weiterleiten, werden die Aufrufe der Dynamic-Cache-API wahrscheinlich von der Basiscacheinstanz bearbeitet. Wenn Sie den dynamischen Cache-Provider von eXtreme Scale für JSP-, Web-Service- oder Befehlscaching verwenden möchten, müssen Sie die Basiscacheinstanz für die Verwendung des dynamischen Cache-Providers von eXtreme Scale konfigurieren. Für die Konfiguration der Basiscacheinstanz werden dieselben Konfigurationseigenschaften verwendet. Denken Sie daran, dass diese Konfigurationseigenschaften als angepasste JVM-Eigenschaften (Java Virtual Machine) konfiguriert werden müssen. Dieser Vorbehalt gilt mit Ausnahme des Servlet-Cachings für jede Cachekonfigurationseigenschaft, die in diesem Abschnitt beschrieben wird. Wenn Sie eXtreme Scale mit dem dynamischen Cache-Provider für Servlet-Caching verwenden möchten, müssen Sie diese Aktivierung in den Systemeigenschaften und nicht in den angepassten Eigenschaften konfigurieren.

Vorgehensweise

1. Dynamischen Cache-Provider von eXtreme Scale aktivieren.
   - **WebSphere Application Server Version 7.0 und höher:**

     Sie können den dynamischen Cache-Service über die Administrationskonsole für die Verwendung des dynamischen Cache-Providers von eXtreme Scale konfigurieren. Nach der Installation von eXtreme Scale ist der dynamische Cache-Provider von eXtreme Scale sofort als Option unter **Cache-Provider** in der Administrationskonsole verfügbar. Weitere Informationen finden Sie im Information Center von WebSphere Application Server Version 7.0: Cache-Service-Provider auswählen.
   - **WebSphere Application Server Version 6.1:**

     Verwenden Sie eine angepasste Eigenschaft, um den dynamischen Cache-Service für die Verwendung des dynamischen Cache-Providers von eXtreme Scale zu konfigurieren. Sie können diese angepassten Eigenschaften auch in WebSphere Application Server Version 7.0 und höher verwenden. Zum Erstellen einer angepassten Eigenschaft in einer Cacheinstanz klicken Sie auf **Ressourcen** > **Cacheinstanzen** > **Typ_der_Cacheinstanz** > **Name_der_Cacheinstanz** > **Angepasste Eigenschaften** > **Neu**. Wenn Sie die Basiscacheinstanz verwenden, erstellen Sie die angepassten Eigenschaften in der JVM.

com.ibm.ws.cache.CacheConfig.cacheProviderName
Zur Verwendung des dynamischen Cache-Providers von eXtreme Scale setzen Sie die Eigenschaft auf den Wert `com.ibm.ws.objectgrid.dynacache.CacheProviderImpl`. Sie können diese angepasste Eigenschaft in einer dynamischen Cacheinstanz oder in der Basiscacheinstanz erstellen. Wenn Sie die angepasste Eigenschaft in der Basiscacheinstanz konfigurieren, verwenden alle anderen Cacheinstanzen im Server standardmäßig den Provider von eXtreme Scale. Alle Konfigurationseigenschaften des dynamischen Cache-Providers von eXtreme Scale, die für die Basiscacheinstanz konfiguriert werden, sind die Standardkonfigurationseigenschaften für alle Cacheinstanzen, die durch eXtreme Scale gestützt werden. Zum Überschreiben der Basiscacheinstanz und Konfigurieren einer bestimmten dynamischen Cacheinstanz für die Verwendung des standardmäßig zu verwendenden dynamischen Cache-Providers erstellen Sie die angepasste Eigenschaft "com.ibm.ws.cache.CacheConfig.cacheProviderName" in der dynamischen Cacheinstanz und setzen diese auf den Wert `default`.

2. Optional: Wenn Sie replizierte Cacheinstanzen verwenden, konfigurieren Sie die Replikationseinstellung für den Cache.

   Mit dem dynamischen Cache-Provider von eXtreme Scale ist es möglich, lokale Cacheinstanzen und replizierte Cacheinstanzen zu verwenden. Wenn Sie nur lokale Cacheinstanzen verwenden, können Sie diesen Schritt überspringen.

   Verwenden Sie eine der folgenden Methoden, um den replizierten Cache zu konfigurieren:

   - Cachereplikation über die Administrationskonsole aktivieren. Sie können die Cachereplikation jederzeit in WebSphere Application Server Version 7.0 aktivieren. In WebSphere Application Server Version 6.1 müssen Sie eine DRS-Replikationsdomäne erstellen.
   - Cachereplikation mit der angepassten Eigenschaft "com.ibm.ws.cache.CacheConfig.enableCacheReplication" aktivieren, damit der Cache meldet, dass er ein replizierter Cache ist, obwohl ihm keine DRS-Replikationsdomäne zugeordnet ist. Setzen Sie diese angepasste Eigenschaft auf den Wert `true`. Konfigurieren Sie diese angepasste Eigenschaft in der Cacheinstanz, wenn Sie einen Objektcache oder Servlet-Cache verwenden, bzw. n der JVM, wenn Sie die Basiscacheinstanz verwenden.

3. Optional: Wenn Sie eXtreme Scale als JSP-Fragmentcache verwenden, setzen Sie die angepasste Eigenschaft "com.ibm.ws.cache.CacheConfig.disableTemplateInvalidation" auf `true`, um schablonenbasierte Invalidierungen während des erneuten Ladens von JSPs (JSP-Reload) zu inaktivieren.

4. Topologie für den dynamischen Cache-Service konfigurieren.

   Der einzige erforderliche Konfigurationsparameter für den dynamischen Cache-Provider von eXtreme Scale ist die Cachetopologie. Setzen Sie die angepasste Eigenschaft in der Cacheinstanz bzw. für den dynamischen Cache-Service, wenn Sie die Basiscacheinstanz verwenden. Geben Sie den Namen der angepassten Eigenschaft ein: `com.ibm.websphere.xs.dynacache.topology`.

   Im Folgenden sind drei gültige Werte für diese Eigenschaft aufgelistet. Sie müssen einen der zulässigen Werte verwenden:

   - `embedded`
   - `embedded_partitioned`
   - `remote`

   Wenn Sie integrierte (embedded) oder integrierte partitionierte (embedded_partitioned) Topologien verwenden, sollten Sie die angepasste Eigenschaft

`com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent` auf `true` setzen, um Serialisierungskosten einzusparen. Setzen Sie diese angepasste Eigenschaft in der Cacheinstanz bzw. in der JVM, wenn Sie die Basiscacheinstanz verwenden.

5. Optional: Wenn Sie eine integrierte partitionierte Topologie verwenden, konfigurieren Sie die anfängliche Containeranzahl für den dynamischen Cache-Service.

   Sie können die Leistung von Caches, die die integrierte partitionierte Topologie verwenden, maximieren, indem Sie die anfängliche Containeranzahl konfigurieren. Konfigurieren Sie die Variable als Systemeigenschaft in der Java Virtual Machine von WebSphere Application Server.

   Geben Sie den Namen der Eigenschaft an: `com.ibm.websphere.xs.dynacache.num_initial_containers`.

   Der empfohlene Wert für diese Konfigurationseigenschaft ist eine ganze Zahl, die der Gesamtanzahl der Instanzen von WebSphere Application Server entspricht, die auf diese verteilte Cacheinstanz zugreifen, bzw. geringfügig darunter liegt. Wenn ein dynamischer Cache-Service von Datengrid-Membern gemeinsam genutzt wird, muss der Wert auf die Anzahl der Grid-Member gesetzt werden.

   Für integrierte (embedded) und integrierte partitionierte (embedded_partitioned) Topologien müssen Sie Version 7.0 von WebSphere Application Server verwenden. Definieren Sie die folgende angepasste Eigenschaft im JVM-Prozess, um sicherzustellen, dass die Anfangscontainer sofort verfügbar sind.

   ```
   com.ibm.ws.cache.CacheConfig.createCacheAtServerStartup=true
   ```

6. Konfigurieren Sie das Katalogservice-Grid von eXtreme Scale.

   Wenn Sie eXtreme Scale als dynamischen Cache-Provider für eine verteilte Cacheinstanz verwenden, müssen Sie eine Katalogservicedomäne von eXtreme Scale konfigurieren.

   Eine einzelne Katalogservicedomäne kann mehrere dynamische Cacheserviceprovider bedienen, die von eXtreme Scale gestützt werden.

   Ein Katalogservice kann innerhalb und außerhalb von Prozessen von WebSphere Application Server ausgeführt werden. Wenn Sie ab eXtreme Scale Version 7.1 die Administrationskonsole verwenden, um Katalogservicedomänen zu konfigurieren, verwendet der dynamische Cache diese Einstellungen. Es ist nicht erforderlich, weitere Schritte zum Konfigurieren eines Katalogservice auszuführen. Weitere Informationen finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen” auf Seite 264.

7. Konfigurieren Sie angepasste Schlüsselobjekte.

   Wenn Sie angepasste Objekte als Schlüssel verwenden, müssen die Objekte die Schnittstelle "Serializable" oder "Externalizable" implementieren. Wenn Sie die integrierte oder integrierte partitionierte Topologie verwenden, müssen Sie die Objekte in den gemeinsam genutzten Bibliothekspfad von WebSphere stellen, wie es auch der Fall ist, wenn sie mit dem dynamischen Standardcacheprovider verwendet werden. Weitere Einzelheiten finden Sie im Artikel "Schnittstellen 'DistributedMap' und 'DistributedObjectCache' für den dynamischen Cache verwenden" im Information Center von WebSphere Application Server Network Deployment.

   Wenn Sie die ferne Topologie verwenden, müssen Sie die angepassten Schlüsselobjekte für die eigenständigen Container von eXtreme Scale in den Klassenpfad stellen. Weitere Informationen finden Sie unter „Container-Server starten” auf Seite 406.

8. Optional: Wenn Sie eine ferne Topologie verwenden, konfigurieren Sie die Container-Server von eXtreme Scale.
   - **Integrierte oder integrierte partitionierte Topologie:**

     Die zwischengespeicherten Daten werden in den Container-Servern von WebSphere eXtreme Scale gespeichert. Container-Server können innerhalb und außerhalb von Prozessen von WebSphere Application Server ausgeführt werden. Der Provider von eXtreme Scale erstellt automatisch Container innerhalb des WebSphere-Prozesses, wenn Sie integrierte oder integriert partitionierte Topologien für eine Cacheinstanz verwenden. Für diese Topologien ist keine weitere Konfiguration erforderlich.
   - **Ferne Topologie:**

     Wenn Sie die ferne Topologie verwenden, müssen Sie eigenständige eXtreme-Scale-Container-Server vor den Instanzen von WebSphere Application Server starten, die auf die Cacheinstanz zugreifen. Weitere Informationen finden Sie in den Schritten zum Starten eigenständiger Container-Server in der Veröffentlichung *Verwaltung*. Stellen Sie sicher, dass alle Container-Server für einen bestimmten dynamischen Cacheservice auf dieselben Katalogserviceendpunkte zeigen.

     Die XML-Konfigurationsdateien für die eigenständigen Container des dynamischen Cache-Providers von eXtreme Scale befinden sich entweder im Verzeichnis *WXS-Installationsstammverzeichnis*`/customLibraries/ObjectGrid/dynacache/etc` (für Installationen in WebSphere Application Server) bzw. im Verzeichnis *WXS-Installationsstammverzeichnis*`/ObjectGrid/dynacache/etc` (für eigenständige Installationen). Die Dateien haben die Namen `dynacache-remote-objectgrid.xml` und `dynacache-remote-definition.xml`. Erstellen Sie Kopien dieser Dateien, die Sie bearbeiten und verwenden, wenn Sie eigenständige Container für den dynamischen Cache-Provider von eXtreme Scale starten. Der Parameter **numInitialContainers** in der Datei **`dynacache-remote-deployment.xml`** muss der Anzahl der aktiven Containerprozesse entsprechen. Beachten Sie, dass das Attribut **`numberOfPartitions`** in der Datei `dynacache-remote-objectgrid.xml` den Standardwert `47` hat.

     Anmerkung: Die Gruppe der Container-Server-Prozesse muss genügend freien Speicher haben, um alle dynamischen Cacheinstanzen zu bedienen, die für die ferne Topologie konfiguriert wurden. Alle Prozesse von WebSphere Application Server, die dieselben oder äquivalente Werte für die angepasste Eigenschaft "catalog.services.cluster" verwenden, müssen dieselbe Gruppe eigenständiger Container verwenden. Die Anzahl der Container und die Anzahl der Server, in denen sich die Container befinden, müssen entsprechend dimensioniert werden. Weitere Einzelheiten finden Sie im Abschnitt „Kapazitätsplanung für den dynamischen Cache” auf Seite 59.

     Im Folgenden sehen Sie einen Befehlszeileneintrag, der einen eigenständigen Container für den dynamischen Cache-Provider von eXtreme Scale startet:

     UNIX

     ```
     startOgServer.sh container1 -objectGridFile
     ../dynacache/etc/dynacache-remote-objectgrid.xml -deploymentPolicyFile
     ../dynacache/etc/dynacache-remote-deployment.xml -catalogServiceEndPoints
     MyServer1.company.com:2809
     ```

9. Für verteilte und integrierte Topologien aktivieren Sie den Agenten für Messungen, um die Schätzungen für die Speicherbelegung zu verbessern.

   Der Agent für Messungen schätzt die Speicherbelegung (Statistik "usedBytes"). Der Agent erfordert eine JVM der Java Version 5 oder höher.

   Laden Sie den Agenten, indem Sie der JVM-Befehlszeile das folgende Argument hinzufügen:

`-javaagent:WXS-Bibliotheksverzeichnis/wxssizeagent.jar`

Für eine integrierte Topologie fügen Sie das Argument der Befehlszeile für den Prozess von WebSphere Application Server hinzu.

Für eine verteilte Topologie fügen Sie das Argument der Befehlszeile der Prozesse von eXtreme Scale (Container) und des Prozesses von WebSphere Application Server hinzu.

JPA-L2-Cache-Plug-in

WebSphere eXtreme Scale enthält Cache-Plug-ins der Stufe 2 (L2) für die JPA-Provider OpenJPA und Hibernate. Wenn Sie eines dieser Plug-ins verwenden, verwendet Ihre Anwendung die JPA-API. Es wird ein Datengrid zwischen Anwendung und Datenbank eingeführt, das die Antwortzeiten verbessert.

Die Verwendung von eXtreme Scale als L2-Cache-Provider erhöht die Leistung beim Lesen und Abfragen von Daten und reduziert die Last der Datenbank. WebSphere eXtreme Scale bietet im Vergleich mit integrierten Cacheimplementierungen verschiedene Vorteile, weil der Cache automatisch in allen Prozessen repliziert wird. Wenn ein Client einen Wert zwischenspeichert, können alle anderen Clients den zwischengespeicherten Wert, der sich lokal im Speicher befindet, verwenden.

Sie können die Topologie und die Eigenschaften für den L2-Cache-Provider in der Datei `persistence.xml` konfigurieren. Weitere Informationen zum Konfigurieren dieser Eigenschaften finden Sie unter „Konfigurationseigenschaften des JPA-Caches" auf Seite 346.

Tipp: Das JPA-L2-Cache-Plug-in erfordert eine Anwendung, die die JPA-APIs verwendet. Wenn Sie APIs von WebSphere eXtreme Scale für den Zugriff auf eine JPA-Datenquelle verwenden möchten, verwenden Sie den JPA-Loader. Weitere Informationen finden Sie unter JPA-Loader.

Hinweise zur JPA-L2-Cachetopologie

Die folgenden Faktoren haben Auswirkungen auf den zu konfigurierenden Topologietyp:

1. **Wie viele Daten werden schätzungsweise zwischengespeichert?**
   - Wenn die Daten in einen einzigen JVM-Heapspeicher passen, verwenden Sie „Integrierte Topologie" auf Seite 341 oder „Domäneninterne Topologie" auf Seite 340.
   - Wenn die Daten nicht in einen einzigen JVM-Heapspeicher passen, verwenden Sie „Integrierte, partitionierte Topologie" auf Seite 342 oder „Ferne Topologie" auf Seite 344.
2. **Welches Verhältnis zwischen Lese- und Schreiboperationen erwarten Sie?**

   Das Verhältnis zwischen Lese- und Schreiboperationen wirkt sich auf die Leistung des L2-Caches aus. Jede Topologie verarbeitet Lese- und Schreiboperationen anders.
   - „Integrierte Topologie" auf Seite 341: lokale lesen, fern schreiben
   - „Domäneninterne Topologie" auf Seite 340: lokal lesen, lokal schreiben
   - „Integrierte, partitionierte Topologie" auf Seite 342: Partitioniert: fern lesen, fern schreiben
   - „Ferne Topologie" auf Seite 344: fern lesen, fern schreiben

Anwendungen, die größtenteils schreibgeschützt sind, sollten, sofern möglich, integrierte und domäneninterne Topologien verwenden. Anwendungen, die mehr Schreiboperationen durchführen, sollten domäneninterne Topologien verwenden.

3. **Wie ist der Prozentsatz abgefragter Daten im Vergleich zum Prozentsatz anhand eines Schlüssels gefundener Daten?**

   Wenn der JPA-Abfragecache aktiviert ist, wird er von Abfrageoperationen genutzt. Aktivieren Sie den JPA-Abfragecache nur für Anwendungen mit einem hohen Lese/Schreib-Verhältnis, z. B., wenn Stand der Leseoperationen 99 % erreicht. Wenn Sie den JPA-Abfragecache verwenden, müssen Sie den „Integrierte Topologie" auf Seite 341 oder den „Domäneninterne Topologie" verwenden.

   Die Find-by-key-Operation (Suchen nach Schlüsseln) ruft eine Zielentität ab, wenn die Zielentität keine Beziehung hat. Wenn die Zielentität Beziehungen mit dem Abruftyp EAGER hat, werden diese Beziehungen zusammen mit der Zielentität abgerufen. Im JPA-Datencache verursacht der Abruf dieser Beziehungen einige wenige Cachetreffer, um alle Beziehungsdaten abzurufen.

4. **Welcher Veraltungsstand der Daten wird toleriert?**

   In einem System mit wenigen JVMs treten Latenzzeiten bei Schreiboperationen während der Datenreplikation auf. Das Ziel des Caches ist die Verwaltung einer synchronisierten Datenansicht in allen JVMs. Wenn Sie die domäneninterne Topologie verwenden, treten bei Schreiboperationen Verzögerungen während der Datenreplikation auf. Anwendungen, die diese Topologie verwenden, müssen veraltete Leseoperationen und gleichzeitige Schreiboperationen tolerieren, die Daten überschreiben.

7.1.1+
Domäneninterne Topologie

Bei einer domäneninternen Topologie werden primäre Shards an jeden Container-Server in der Topologie verteilt. Diese primären Shards enthalten die vollständigen Daten für die Partition. Alle primären Shards können auch Schreiboperationen im Cache ausführen. Diese Konfiguration schaltet Engpässe in der integrierten Topologie aus, in der alle Schreiboperationen im Cache über ein einziges primäres Shard erfolgen.

In einer domäneninternen Topologie werden keine Replikat-Shards erstellt, selbst wenn Sie Replikate in Ihren Konfigurationsdateien definiert haben. Jedes redundante primäre Shard enthält eine vollständige Kopie der Daten, sodass jedes primäre Shard auch als Replikat-Shard betrachtet werden kann. Diese Konfiguration verwendet ähnlich wie in der integrierten Topologie eine einzige Partition.

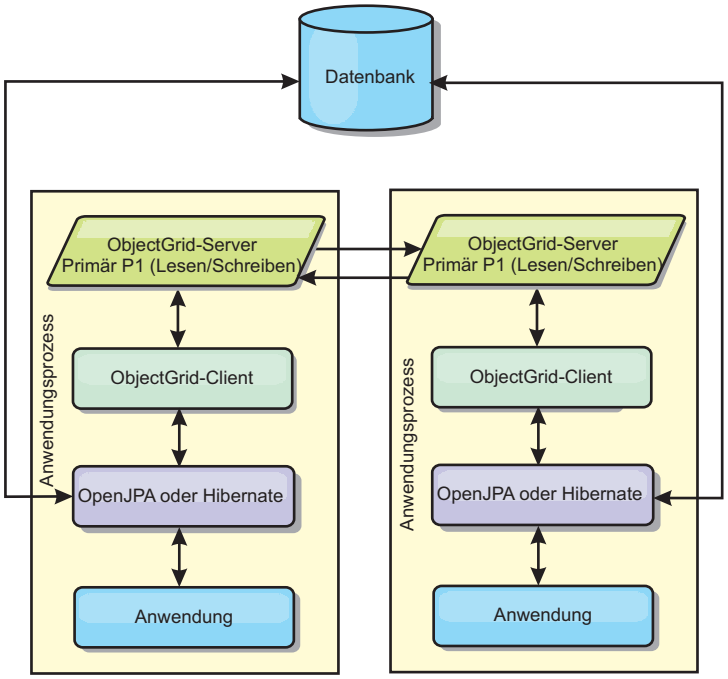


Abbildung 39. Domäneninterne JPA-Topologie

Zugehörige JPA-Cachekonfigurationseigenschaften für die domäneninterne Topologie:

```
ObjectGridName=ObjectGrid-Name,ObjectGridType=EMBEDDED,PlacementScope=CONTAINER_SCOPE,PlacementScopeTopology=HUB | RING
```

Vorteile:

- Lese- und Aktualisierungsoperationen im Cache sind lokal.
- Die Konfiguration ist einfach.

Einschränkungen:

- Diese Topologie eignet sich optimal, wenn die Container-Server alle Partitionsdaten enthalten.
- Replikat-Shards werden, selbst wenn sie konfiguriert sind, nie verteilt, weil jeder Container-Server ein primäres Shard hostet. Alle primären Shards werden auf den anderen primären Shards repliziert, so dass diese primären Shards zu gegenseitigen Replikaten werden.

Integrierte Topologie

Tipp: Für eine optimale Leistung sollten Sie eine domäneninterne Topologie in Erwägung ziehen.

Eine integrierte Topologie erstellt einen Container-Server im Prozessbereich jeder Anwendung. OpenJPA und Hibernate lesen die Speicherkopie des Caches direkt und schreiben in alle anderen Kopien. Sie können die Schreibleistung durch den Einsatz asynchroner Replikation verbessern. Diese Standardtopologie liefert die beste Leistung, wenn die zwischengespeicherte Datenmenge in einen einzigen Prozess passt. Bei einer integrierten Topologie erstellen Sie eine einzige Partition für die Daten.

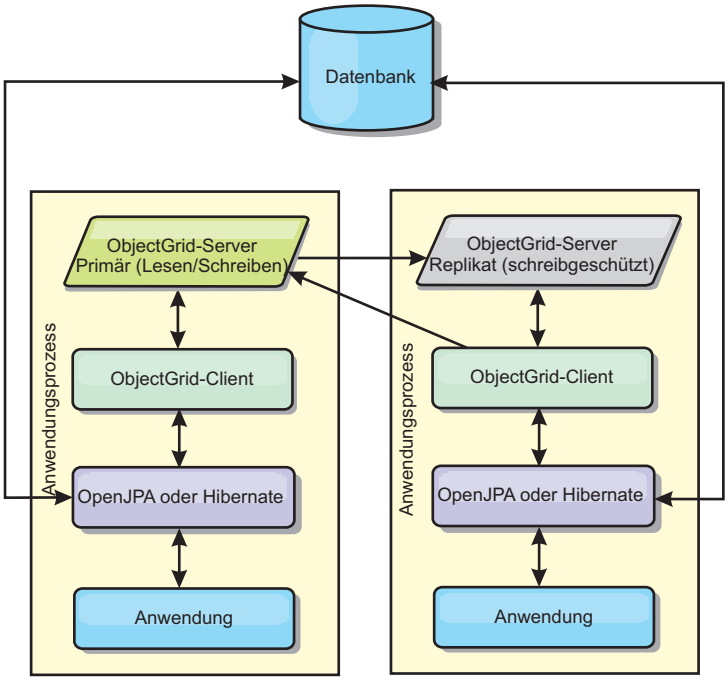


Abbildung 40. Integrierte JPA-Topologie

Zugehörige JPA-Cachekonfigurationseigenschaften für die integrierte Topologie:

```
ObjectGridName=ObjectGrid-Name,ObjectGridType=EMBEDDED,MaxNumberOfReplicas=Anzahl_der_Replikate,ReplicaMode=SYNC | ASYNC | NONE
```

Vorteile:

- Alle Leseoperationen im Cache sind schnelle lokale Zugriffe.
- Die Konfiguration ist einfach.

Einschränkungen:

- Das Datenvolumen ist auf die Größe des Prozesses beschränkt.
- Alle Cacheaktualisierungen werden über ein einziges primäres Shard gesendet, woraufhin ein Engpass entsteht.

Integrierte, partitionierte Topologie

Tipp: Für eine optimale Leistung sollten Sie eine domäneninterne Topologie in Erwägung ziehen.

Vorsicht:
Verwenden Sie den JPA-Abfragecache nicht für eine integrierte partitionierte Topologie. Im Abfragecache werden Abfrageergebnisse gespeichert, die eine Sammlung von Entitätsschlüsseln sind. Der Abfragecache verwendet den Datencache, um alle Entitätsdaten abzufragen. Da der Datencache auf mehrere Prozesse verteilt ist, können diese zusätzlichen Aufrufe die Vorteile des L2-Caches aufheben.

Wenn die zwischengespeicherten Daten nicht in einen einzigen Prozess passen, können Sie die integrierte partitionierte Topologie verwenden. In dieser Topologie werden die Daten auf mehrere Prozesse verteilt. Die Daten werden so auf die primären Shards verteilt, dass jedes primäre Shard einen Teil der Daten enthält. Sie können diese Option auch verwenden, wenn die Latenzzeit der Datenbank hoch

ist.

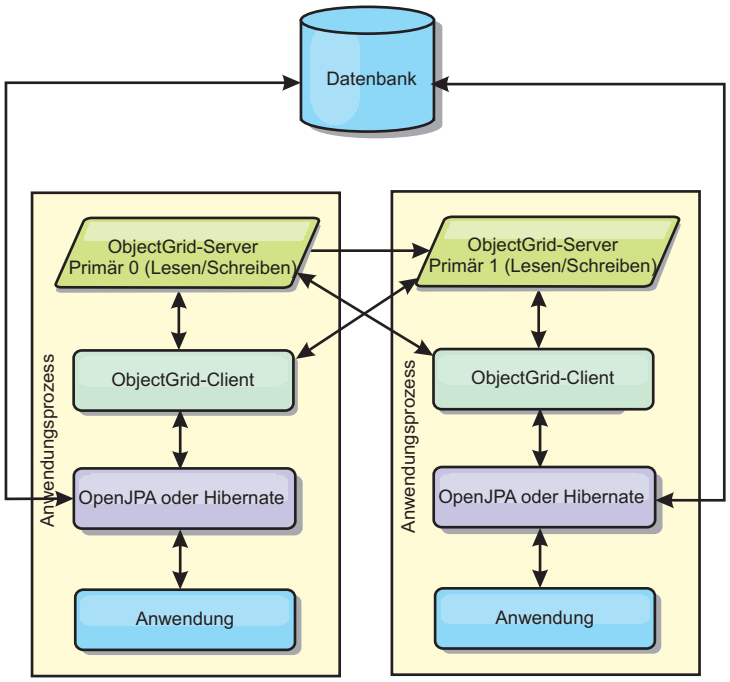


Abbildung 41. Integrierte, partitionierte JPA-Topologie

Zugehörige JPA-Cachekonfigurationseigenschaften für die integrierte partitionierte Topologie:

```
ObjectGridName=ObjectGrid-Name,ObjectGridType=EMBEDDED_PARTITION,ReplicaMode=SYNC | ASYNC | NONE,
NumberOfPartitions=Anzahl_Partitionen,ReplicaReadEnabled=TRUE | FALSE
```

Vorteile:

- Es können große Datenvolumen gespeichert werden.
- Die Konfiguration ist einfach.
- Cacheaktualisierungen werden auf mehrere Prozesse verteilt.

Einschränkungen:

- Die meisten Lese- und Aktualisierungsoperationen im Cache werden über Fernzugriff durchgeführt.

Um beispielsweise 10 GB Daten mit maximal 1 GB pro JVM zu speichern, sind zehn Java Virtual Machines erforderlich. Die Anzahl der Partitionen muss daher auf mindestens 10 gesetzt werden. Im Idealfall wird die Anzahl der Partitionen auf eine Primzahl gesetzt, so dass in jedem Shard eine angemessene Speichermenge zugeteilt wird. Gewöhnlich entspricht der Wert der Einstellung "numberOfPartitions" der Anzahl der Java Virtual Machines. Bei dieser Einstellung enthält jede JVM eine Partition. Wenn Sie die Replikation aktivieren, müssen Sie die Anzahl der Java Virtual Machines im System erhöhen. Andernfalls wird in jeder JVM zusätzlich eine Replikatpartition gespeichert, die genauso viel Speicher belegt wie eine primäre Partition.

Lesen Sie die Informationen zur Berechnung der Speicherkapazität und der Partitionsanzahl in der Veröffentlichung *Verwaltung*, um die Leistung der von Ihnen ausgewählten Konfiguration zu maximieren.

In einem System mit vier Java Virtual Machines und einem numberOfPartitions-Wert von 4 beispielsweise enthält jede JVM eine primäre Partition. Bei einer Leseoperation besteht eine Chance von 25 %, dass die Daten aus einer lokal verfügbaren Partition abgerufen werden, was im Vergleich mit dem Abruf der Daten aus einer fernen JVM wesentlich schneller ist. Wenn eine Leseoperation, z. B. eine Abfrage, eine Sammlung von Daten abrufen muss, die gleichmäßig auf vier Partitionen verteilt sind, sind 75 % der Aufrufe ferne und 25 % der Aufrufe lokale Aufrufe. Wenn die Einstellung "ReplicaMode" auf `SYNC` oder `ASYNC` und die Einstellung "ReplicaReadEnabled" auf `true` gesetzt wird, werden vier Relikatpartitionen erstellt und auf vier Java Virtual Machines verteilt. Jede JVM enthält eine primäre Partition und eine Replikatpartition. Die Chance, dass die Leseoperation lokal ausgeführt wird, erhöht sich auf 50 %. Die Leseoperation, die eine Sammlung von Daten abgerufen muss, die gleichmäßig auf vier Partitionen verteilt sind, hat 50 % ferne Aufrufe und 50% lokale Aufrufe. Lokale Aufrufe sind wesentlich schneller als ferne Aufrufe. Mit jedem fernen Aufruf nimmt die Leistung ab.

Ferne Topologie

Vorsicht:
Verwenden Sie den JPA-Abfragecache nicht für eine ferne Topologie. Im Abfragecache werden Abfrageergebnisse gespeichert, die eine Sammlung von Entitätsschlüsseln sind. Der Abfragecache verwendet den Datencache, um alle Entitätsdaten abzufragen. Da der Datencache fern ist, können diese zusätzlichen Aufrufe die Vorteile des L2-Caches aufheben.

Tipp: Für eine optimale Leistung sollten Sie eine domäneninterne Topologie in Erwägung ziehen.

In einer fernen Topologie werden alle zwischengespeicherten Daten in einem oder mehreren gesonderten Prozessen gespeichert, was die Speicherbelegung der Anwendungsprozesse verringert. Sie können Ihre Daten auf unterschiedliche Prozesse verteilen, indem Sie ein partitioniertes, repliziertes eXtreme-Scale-Datengrid implementieren. Im Gegensatz zu den integrierten und integrierten partitionierten Konfigurationen, die in den vorherigen Abschnitten beschrieben wurden, müssen Sie ein fernes Datengrid unabhängig von der Anwendung und vom JPA-Provider verwalten.

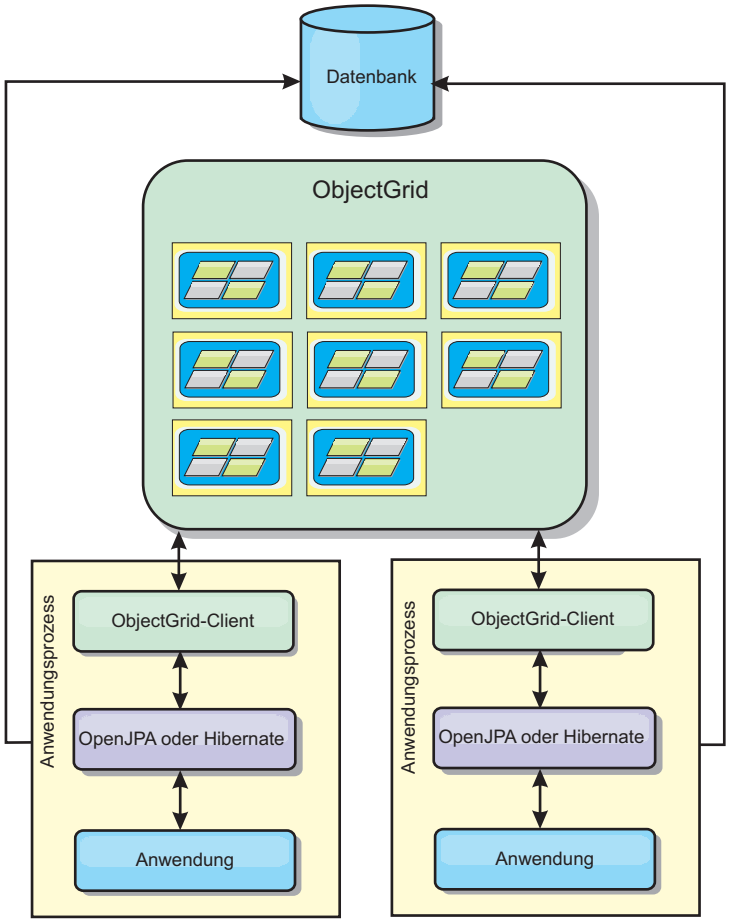


Abbildung 42. Ferne JPA-Topologie

Zugehörige JPA-Cachekonfigurationseigenschaften für die ferne Topologie:
`ObjectGridName=`*`ObjectGrid-Name`*`,ObjectGridType=REMOTE`

Der ObjectGrid-Typ REMOTE erfordert keine Eigenschaftseinstellungen, weil das ObjectGrid und die Implementierungsrichtlinie gesondert von der JPA-Anwendung definiert werden. Das JPA-Cache-Plug-in stellt über Fernzugriff eine Verbindung zu einem vorhandenen fernen ObjectGrid her.

Da alle Interaktionen mit dem ObjectGrid über Fernzugriff erfolgen, hat diese Topologie die geringste Leistung von allen ObjectGrid-Typen.

Vorteile:

- Es können große Datenvolumen gespeichert werden.
- Der Anwendungsprozess ist frei von zwischengespeicherten Daten.
- Cacheaktualisierungen werden auf mehrere Prozesse verteilt.
- Flexible Konfigurationsoptionen

Einschränkungen:

- Alle Lese- und Aktualisierungsoperationen im Cache werden über Fernzugriff durchgeführt.

Konfigurationseigenschaften des JPA-Caches

WebSphere eXtreme Scale enthält Cache-Plug-ins der Stufe 2 für die JPA-Provider OpenJPA und Hibernate. Zum Konfigurieren des L2-Cache-Plug-ins müssen Sie Eigenschaften in der Datei `persistence.xml` aktualisieren.

Tipp: Das JPA-L2-Cache-Plug-in erfordert eine Anwendung, die die JPA-APIs verwendet. Wenn Sie APIs von WebSphere eXtreme Scale für den Zugriff auf eine JPA-Datenquelle verwenden möchten, verwenden Sie den JPA-Loader. Weitere Informationen finden Sie im Abschnitt „JPA-Loader konfigurieren" auf Seite 361.

Eigenschaftenposition

Sie können diese Eigenschaften in der Datei `persistence.xml` konfigurieren. Die Syntax für die Angabe der Eigenschaften in dieser Datei ist davon abhängig, ob Sie OpenJPA oder Hibernate verwenden:

- **OpenJPA:** Sie können die Eigenschaften in DataCache oder QueryCache definieren:

  ```
  <property name="openjpa.DataCache"
   value="<object_grid_datacache_class(<Eigenschaft>=<Wert>,...)"/>
  ```

  oder

  ```
  <property name="openjpa.QueryCache"
   value="<object_grid_querycache_class(<Eigenschaft>=<Wert>,...)"/>
  ```

- **Hibernate:**

  ```
  <property name="objectgrid.configuration" value="<Eigenschaft>=<Wert>,..." />
  ```

Standardtopologie und -eigenschaften

Die folgenden Standardeigenschaftswerte werden verwendet, wenn Sie keine Werte in der Konfiguration angeben:

- **ObjectGridName**: Name der Persistenzeinheit
- **ObjectGridType**: `EMBEDDED`
- **NumberOfPartitions**: 1 (kann nicht geändert werden, wenn der ObjectGrid-Typ `EMBEDDED` ist)
- **ReplicaMode**: `SYNC`
- **ReplicaReadEnabled**: `TRUE` (kann nicht geändert werden, wenn der ObjectGrid-Typ `EMBEDDED` ist)
- **MaxUsedMemory**: `TRUE`
- **MaxNumberOfReplicas**: `47` (muss kleiner-gleich der Anzahl Java Virtual Machines in einem verteilten System sein)

Eigenschaften

Sie können JPA-Cache-Plug-ins mit den folgenden Eigenschaften konfigurieren.

ObjectGridName
: Gibt den eindeutigen ObjectGrid-Namen an. Der Standardwert ist der Name der definierten Persistenzeinheit. Wenn der Name der Persistenzeinheit nicht über den JPA-Provider verfügbar ist, wird ein generierter Name verwendet.

ObjectGridType
: Gibt den Typ des ObjectGrids an.

  Gültige Werte:

EMBEDDED
Der Standard- und empfohlene Konfigurationstyp. Zu den Standardeinstellungen gehören `NumberOfPartitions=1`, `ReplicaMode=SYNC`, `ReplicaReadEnabled=true` und `MaxNumberOfReplicas=47`. Verwenden Sie den Parameter **ReplicaMode**, um den Replikationsmodus festzulegen, und den Parameter **MaxNumberOfReplicas**, um die maximale Anzahl an Replikaten festzulegen. Wenn ein System mehr als 47 Java Virtual Machines hat, setzen Sie **MaxNumberOfReplicas** auf die Anzahl der Java Virtual Machines.

EMBEDDED_PARTITION
Der zu verwendende Typ, wenn das System hohe Datenvolumen in einem verteilten System zwischenspeichern muss. Die Standardanzahl an Partitionen ist 47 beim Replikationsmodus `NONE`. In einem kleinen System, das nur wenige Java Virtual Machines hat, setzen Sie **NumberOfPartitions** auf einen Wert kleiner-gleich der Anzahl Java Virtual Machines. Sie können Werte für **ReplicaMode**, **NumberOfPartitions** und **ReplicaReadEnabled** angeben, um das System zu optimieren.

REMOTE Der Cache versucht, über den Katalogservice eine Verbindung zu einem fernen, verteilten ObjectGrid herzustellen.

MaxNumberOfReplicas
Gibt die maximale Anzahl der für den Cache zu verwendenden Replikate an. Dieser Wert gilt nur für den Typ `EMBEDDED`. Der muss größer-gleich der Anzahl an Java Virtual Machines in einem System sein. Der Standardwert ist 47.

Gültige Werte: Größer-gleich 1

MaxUsedMemory

Gültige Werte: `TRUE` oder `FALSE`Aktiviert das Entfernen von Cacheeinträgen, wenn ein Speicherengpass auftritt. Der Standardwert ist `TRUE` und sorgt dafür, dass Daten entfernt werden, wenn die Auslastung des JVM-Heapspeichers den Schwellenwert von 70 % überschreitet. Sie können den Prozentsatz für den Schwellwert für die Auslastung des JVM-Heapspeichers ändern, indem Sie die Eigenschaft "memoryThresholdPercentage" in der Datei `objectGridServer.properties` definieren und diese Datei in den Klassenpfad stellen. Weitere Einzelheiten zu Bereinigungsprogrammen (Evictor) finden Sie in Plug-ins für die Bereinigung von Cacheobjektenden Informationen zu Bereinigungsprogrammen in der Veröffentlichung *Produktübersicht*. Weitere Informationen zur Servereigenschaftendatei finden Sie im Abschnitt Servereigenschaftendatei.

NumberOfPartitions

Gültige Werte: Größer-gleich 1Gibt die Anzahl der für den Cache zu verwendenden Partitionen an. Diese Eigenschaft gilt, wenn `EMBEDDED_PARTITION` als Wert für ObjectGridType angegeben ist. Der Standardwert ist 47. Für den Typ `EMBEDDED` ist der Wert von **NumberOfPartitions** immer 1.

7.1.1+ **PlacementScope**
Gibt die Granularität einer einzelnen Instanz eines MapSets an.

Gültige Werte:

DOMAIN_SCOPE
(Standardeinstellung) Verteilt ein primäres Shard für jede Partition

an einen Container-Server in der Katalogservicedomäne. Replikat-Shards für jede Partition werden an die anderen Container-Server in der Katalogservicedomäne verteilt.

CONTAINER_SCOPE
Verteilt ein primäres Shard an jeden Container-Server in der Katalogservicedomäne.

7.1.1+ **PlacementScopeTopology**
Definiert die Verbindungstopologie der Container-Server in der Katalogservicedomäne. Dieser Wert wird nur verwendet, wenn der PlacementScope-Wert auf einen anderen Wert als `DOMAIN_SCOPE` gesetzt ist.

Gültige Werte:

HUB (Standardeinstellung) Wenn die Hubtopologie ausgewählt ist, wird ein einziges Datengrid als Hub ausgewählt. Jedes andere Datengrid stellt eine Verbindung zum Hub her. Diese Topologie ist relativ gut skalierbar, weil die Peripheriedomänen eine einzige Verbindung haben. Der Hub kann zu einem Engpass und Single Point of Failure werden. Der Hub wird in einen anderen Container-Server verlagert, wenn er ausfällt. Diese Konfiguration hat den Vorteil, dass komplexerer Arbitrierungscode geschrieben werden kann, der einem einzigen Punkt, dem Hub, ermöglicht, alle Kollisionen zu behandeln.

RING Wenn Sie die Ringtopologie auswählen, wird jedes Datengrid mit zwei anderen Datengrids verbunden. Die Reihenfolge der Verbindungen ist nicht garantiert. Jeder gestartete Container wird jedoch wahrscheinlich mit dem ersten Container und dem letzten Container verbunden, der dem Ring hinzugefügt wird. Diese Topologie ist die Topologie mit der höchsten Skalierbarkeit, aber es können nur zwei Verbindungen ausfallen, bevor die Topologie vorübergehend stillgelegt wird. Wenn die Container-Server ausfallen, werden Verbindungen zwischen den verbleibenden Servern hergestellt, sobald der Fehler erkannt wurde.

ReplicaMode

Gültige Werte: `SYNC/ASYNC/NONE`Gibt die Methode an, die verwendet wird, um den Cache in die Replikate zu kopieren. Diese Eigenschaft gilt, wenn Sie `EMBEDDED` oder `EMBEDDED_PARTITION` als Wert für ObjectGridType festgelegt haben. Der Standardwert ist `NONE` für den Typ `EMBEDDED_PARTITION` und `SYNC` für den Typ `EMBEDDED`. Wenn Sie **ReplicaMode** auf `NONE` und `EMBEDDED` für ObjectGridType angeben, verwendet der Typ `EMBEDDED` weiterhin den **ReplicaMode**-Wert `SYNC`.

ReplicaReadEnabled

Gültige Werte: `TRUE` oder `FALSE`Wenn Sie diese Eigenschaft aktivieren, lesen Clients aus Replikaten. Diese Eigenschaft gilt für den Typ `EMBEDDED_PARTITION`. Der Standardwert ist FALSE für den Typ `EMBEDDED_PARTITION`. Beim Typ `EMBEDDED` wird **ReplicaReadEnabled** immer auf `TRUE` gesetzt.

writeBehind

Nur für Hibernate-Provider: Wenn writeBehind aktiviert ist, werden Aktualisierungen vorübergehend in einem JVM-spezifischen Datenspeicher gespeichert, bis die writeBehindInterval- bzw. writeBehindMaxBatchSize-Bedingung erfüllt ist.

Achtung: Wenn writeBehind nicht aktiviert ist, werden weitere Write-behind-Konfigurationseinstellungen ignoriert.

Wichtig: Verwenden Sie die Write-behind-Funktion mit Vorsicht. Write-behind-Konfigurationen führen zu längeren Latenzzeiten bei der Datensynchronisation in allen JVMs und zu einem höheren Risiko von Aktualisierungsverlusten. In einem System, in dem die Write-behind-Konfiguration mit vier oder mehr JVMs aktiviert ist, wird die in einer JVM durchgeführte Aktualisierung mit einer Verzögerung von ca. 15 Sekunden anderen JVMs bereitgestellt. Wenn zwei JVMs denselben Eintrag aktualisieren, verliert die JVM, die die Aktualisierung zuerst mit einer Flush-Operation in den Cache schreibt, ihre Aktualisierung.

Gültige Werte: TRUE und FALSE

Standardwert: FALSE

writeBehindInterval

Nur für Hibernate-Provider: Gibt das Zeitintervall (in Millisekunden) an, in dem Aktualisierungen mit einer Flush-Operation in den Cache geschrieben werden.

Gültige Werte: Größer-gleich 1

Standardwert 5000 (5 Sekunden)

writeBehindPoolSize

Nur für Hibernate-Provider: Gibt die maximale Größe des Thread-Pools an, der für das Schreiben von Aktualisierungen in den Cache verwendet wird.

Gültige Werte: Größer-gleich 1

Standardwert: 5

writeBehindMaxBatchSize

Nur für Hibernate-Provider: Gibt die maximale Stapelgröße pro Regionscache für das Schreiben von Aktualisierungen in den Cache an. Wenn die Größe beispielsweise auf 1000 gesetzt ist und die Anzahl der Aktualisierungen, die im Write-behind-Speicher eines Regionscaches den Wert 1000 überschreitet, werden die Aktualisierungen mit einer Flush-Operation in den Cache geschrieben und zwar auch dann, wenn die angegebene writeBehindInterval-Bedingung nicht erfüllt ist. Aktualisierungen werden entweder alle fünf Sekunden oder bei mehr als 1000 Einträgen im Write-behind-Speicher jedes Regionscaches mit einer Flush-Operation in den Cache geschrieben. Wenn die writeBehindMaxBatchSize-Bedingung erfüllt ist, schreibt nur der Regionscache, der diese Bedingung erfüllt, seine Aktualisierungen im Write-behind-Speicher mit einer Flush-Operation in den Cache. Ein Regionscache entspricht gewöhnlich einer Entität oder einer Abfrage.

Gültige Werte: Größer-gleich 1

Standardwert: 1000

OpenJPA-Cache-Plug-in konfigurieren

Sie können DataCache- und QueryCache-Implementierungen für OpenJPA konfigurieren.

Vorbereitende Schritte

- Sie müssen die JPA-Cache-Plug-in-Topologie bestimmen, die Sie verwenden möchten. Weitere Informationen zu den verschiedenen Konfigurationen und den Eigenschaften, die Sie für jede Topologie definieren können, finden Sie unter „JPA-L2-Cache-Plug-in" auf Seite 339.
- Sie müssen eine Anwendung haben, die die JPA-APIs verwendet. Wenn Sie die APIs von WebSphere eXtreme Scale für den Datenzugriff mit JPA verwenden möchten, verwenden Sie den JPA-Loader. Weitere Informationen finden Sie im Abschnitt „JPA-Loader konfigurieren" auf Seite 361.

Vorgehensweise

1. Definieren Sie Eigenschaften in der Datei `persistence.xml` definieren, um das OpenJPA-Cache-Plug-in zu konfigurieren: Sie könenn diese Eigenschaften in der DataCache- oder QueryCache-Implementierung definieren.

   DataCache- und QueryCache-Konfigurationen sind voneinander unabhängig. Sie können beide Konfigurationen aktivieren. Wenn Sie jedoch beide Konfigurationen aktivieren, verwendet QueryCache dieselbe Konfiguration wie DataCache, und die Konfiguration von QueryCache wird verworfen.

   ```
   <property name="openjpa.DataCache"
             value="<object_grid_datacache_class(<Eigenschaft>=<Wert>,...)"/>
   ```

   oder

   ```
   <property name="openjpa.QueryCache"
             value="<object_grid_querycache_class(<Eigenschaft>=<Wert>,...)"/>
   ```

   Anmerkung: Sie können die QueryCache-Konfiguration nur für integrierte und integrierte domäneninerne Topologien aktivieren.

   Sie können die Eigenschaft "ObjectGridName", die Eigenschaft "ObjectGridType" und andere einfache auf die Implementierungsrichtlinie bezogene Eigenschaften in der Eigenschaftenliste der ObjectGrid-Cacheklasse angeben, um die Cachekonfiguration anzupassen. Es folgt ein Beispiel:

   ```
   <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"/>
   ```

   Eine Liste der Eigenschaften, die Sie definieren können, finden Sie unter „Konfigurationseigenschaften des JPA-Caches" auf Seite 346.
2. In der Datei `persistence.xml` müssen Sie die Eigenschaft "openjpa.RemoteCommitProvider" auf `sjvm` setzen.

   ```
   <property name="openjpa.RemoteCommitProvider" value="sjvm"/>
   ```
3. Optional: Zu weiteren Anpassung des vom Cache verwendeten Datengrids können Sie weitere Einstellungen mit XML-Dateien angeben.

   Für die meisten Szenarien reicht die Definition von Cacheeigenschaften aus. Wenn Sie das vom Cache verwendete ObjectGrid weiter anpassen möchten, können Sie OpenJPA-ObjectGrid-XML-Konfigurationsdateien im Verzeichnis `META-INF` bereitstellen, wie z. B. die Datei `persistence.xml`. Während der Initialisierung versucht der Cache, diese XML-Dateien zu finden und sie dann zu verarbeiten.

   Es gibt drei Typen von ObjectGrid-XML-Konfigurationsdateien für OpenJPA:

   - `openjpa-objectGrid.xml` (ObjectGrid-Konfiguration)

Dateipfad: `META-INF/openjpa-objectGrid.xml`

Diese Datei wird verwendet, um die ObjectGrid-Konfiguration für die Typen EMBEDDED und EMBEDDED_PARTITION anzupassen. Für den Typ REMOTE wird diese Datei ignoriert. Standardmäßig wird jede Entitätsklasse einer eigenen BackingMap-Konfiguration zugeordnet, die denselben Namen wie die Entitätsklasse in der ObjectGrid-Konfiguration hat. Die Entitätsklasse "com.mycompany.Employee" wird beispielsweise der BackingMap "com.mycompany.Employee" zugeordnet. Die BackingMap-Standardkonfiguration hat die Einstellungen `readOnly="false"`, `copyKey="false"`, `lockStrategy="NONE"` und `copyMode="NO_COPY"`. Sie können einige BackingMaps mit der ausgewählten Konfiguration anpassen. Sie können das reservierte Schlüsselwort `ALL_ENTITY_MAPS` verwenden, um alle Maps darzustellen mit Ausnahme anderer angepasster Maps, die in der Datei `openjpa-objectGrid.xml` aufgelistet sind. BackingMaps, die nicht in der Datei `openjpa-objectGrid.xml` aufgelistet sind, verwenden die Standardkonfiguration. Wenn für angepasste BackingMaps kein Attribut "BackingMaps" oder keine Eigenschaften definiert sind, diese Attribute aber in der Standardkonfiguration angegeben sind, werden die Attributwerte aus der Standardkonfiguration angewendet. Ist eine Entitätsklasse beispielsweise mit `timeToLive=30` annotiert, enthält die BackingMap-Standardkonfiguration für diese Entität ebenfalls `timeToLive=30`. Wenn die angepasste Datei `openjpa-objectGrid.xml` diese BackingMap ebenfalls enthält, aber den timeToLive-Wert nicht definiert, wird für die angepasste BackingMap standardmäßig der Wert `timeToLive=30` verwendet. Die Datei `openjpa-objectGrid.xml` soll die Standardkonfiguration überschreiben oder erweitern.

- `openjpa-objectGridDeployment.xml` (Implementierungsrichtlinie)

  Dateipfad: `META-INF/openjpa-objectGridDeployment.xml`

  Diese Datei wird verwendet, um die Implementierungsrichtlinie anzupassen. Wenn Sie bei der Anpassung der Implementierungsrichtlinie die Datei `openjpa-objectGridDeployment.xml` bereitstellen, wird die Standardimplementierungsrichtlinie verworfen. Alle Attributwerte für die Implementierungsrichtlinie stammen aus der bereitgestellten Datei `openjpa-objectGridDeployment.xml`.

- `openjpa-objectGrid-client-override.xml` (ObjectGrid-Clientkorrekturkonfiguration)

  Dateipfad: `META-INF/openjpa-objectGrid-client-override.xml`

  Diese Datei wird verwendet, um ein clientseitiges ObjectGrid anzupassen. Standardmäßig wendet der ObjectGrid-Cache eine ObjectGrid-Standardkonfiguration für Clientkorrekturwerte an, die einen nahen Cache inaktiviert. Wenn eine Anwendung einen nahen Cache erfordert, können Sie diese Datei bereitstellen und `numberOfBuckets="xxx"` angeben. Der Standardclientkorrekturwert inaktiviert den nahen Cache mit `numberOfBuckets="0"`. Der nahe Cache kann aktiv sein, wenn das Attribut "numberOfBuckets" über die Datei `openjpa-objectGrid-client-override.xml` auf einen Wert größer als 0 zurückgesetzt wird. Die Datei `openjpa-objectGrid-client-override.xml` funktioniert ähnlich wie die Datei `openjpa-objectGrid.xml`. Sie überscheibt oder erweitert die Standardkonfiguration für ObjectGrid-Clientkorrekturwerte.

Je nach konfigurierter eXtreme-Scale-Topologie können Sie jede dieser drei XML-Dateien verwenden, um diese Topologie anzupassen.

Für die Typen EMBEDDED und EMBEDDED_PARTITION können Sie jede der drei XML-Dateien für die Anpassung des ObjectGrids, der Implementierungsrichtlinie oder der Konfiguration der ObjectGrid-Clientkorrekturwerte verwenden.

Bei ObjectGrids des Typs REMOTE erstellt der ObjectGrid-Cache kein dynamisches ObjectGrid. Vielmehr ruft der Cache nur ein clientseitiges ObjectGrid vom Katalogservice ab. Zum Anpassen der Konfiguration für die ObjectGrid-Clientkorrekturwerte können Sie nur eine Datei `openjpa-objectGrid-client-override.xml` verwenden.

4. Optional: (Nur für ferne Konfigurationen) Sie müssen ein externes eXtreme-Scale-System einrichten, wenn Sie einen Cache mit dem ObjectGrid-Typ REMOTE konfigurieren möchten.

   Sie müssen ein externes eXtreme-Scale-System einrichten, wenn Sie einen Cache mit dem ObjectGrid-Typ REMOTE konfigurieren möchten. Sie benötigen ObjectGrid- und ObjectGridDeployment-XML-Konfigurationsdateien, die auf der Datei `persistence.xml` basieren, um ein externes System einrichten zu können. Beispiele für diese Konfigurationsdateien finden Sie unter „Beispiel: OpenJPA-ObjectGrid-XML-Dateien" auf Seite 353.

Ergebnisse

EMBEDDED-, EMBEDDED_PARTITION- oder domäneninterne Konfiguration:

Wenn eine Anwendung gestartet wird, erkennt oder startet das Plug-in automatisch einen Katalogservice, startet einen Container-Server und stellt die Verbindung zum Katalogservice her. Das Plug-in kommuniziert anschließend mit dem ObjectGrid-Container und seinen Peers, die in anderen Anwendungsserverprozessen ausgeführt werden, über die Clientverbindung.

REMOTE-Konfiguration:

Die Implementierungsrichtlinie wird gesondert von der JPA-Anwendung spezifiziert. Ein externes ObjectGrid-System hat Katalogservice- und Container-Server-Prozesse. Sie müssen einen Katalogservice starten, bevor Sie Container-Server starten. Weitere Informationen finden Sie unter „Eigenständige Server starten" auf Seite 403 und unter „Container-Server starten" auf Seite 406.

Nächste Schritte

- Entwickeln Sie eine OpenJPA-Anwendung, die die Konfiguration verwendet. Weitere Informationen finden Sie im Abschnitt Beispiel: Hibernate-Plug-in zum vorherigen Laden von Daten in den ObjectGrid-Cache verwenden.
- Erstellen Sie in einer Produktionsumgebung Katalogservicedomänen für Ihre automatisch erstellten Prozesse für die EMBEDDED- bzw. EMBEDDED_PARTITION-Konfiguration.
  - Eigenständige Umgebung:

    Wenn Sie Ihre Server nicht in einem Prozess von WebSphere Application Server ausführen, werden die Hosts und Ports der Katalogservicedomäne über eine Eigenschaftendatei mit dem Namen `objectGridServer.properties` angegeben. Diese Datei muss im Klassenpfad der Anwendung gespeichert werden und die definierte Eigenschaft **catalogServiceEndPoints** haben. Die Katalogservicedomäne wird unabhängig von den Anwendungsprozessen gestartet und muss vor den Anwendungsprozessen gestartet werden.

    Das Format der Datei `objectGridServer.properties` ist wie folgt:

    ```
    catalogServiceEndPoints=<Hostname1>:<Port1>,<Hostname2>:<Port2>
    ```

  - Umgebung mit WebSphere Application Server:

Wenn Sie in einem Prozess von WebSphere Application Server arbeiten, stellt das JPA-Cache-Plug-in automatisch eine Verbindung zum Katalogservice bzw. zur Katalogservicedomäne her, der bzw. das für die Zelle von WebSphere Application Server definiert ist.

- Wenn Sie den ObjectGridTyp-Wert `EMBEDDED` oder `EMBEDDED_PARTITION` in einer Java-SE-Umgebung verwenden, verwenden Sie am Ende des Programms die Methode System.exit(0), um den integrierten eXtreme-Scale-Server zu stoppen. Andernfalls reagiert das Programm möglicherweise nicht mehr.

Beispiel: OpenJPA-ObjectGrid-XML-Dateien:

OpenJPA-ObjectGrid-XML-Dateien müssen auf der Basis der Persistenzeinheit erstellt werden.

Datei `persistence.xml`

Im Folgenden sehen Sie eine Beispieldatei `persistence.xml`, die die Konfiguration einer Persistenzeinheit darstellt:

```
<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>
```

Datei `openjpa-objectGrid.xml`

Die Datei `openjpa-objectGrid.xml` wird verwendet, um die ObjectGrid-Konfiguration für die Typen EMBEDDED und EMBEDDED_PARTITION anzupassen. Es folgt die Datei `openjpa-objectGrid.xml`, die der Datei `persistence.xml` entspricht:

```
<?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>
```

Wichtig:

1. Jede Entität wird einer BackingMap zugeordnet, die denselben Namen wie die vollständig qualifizierte Entitätsklasse hat.

Standardmäßig sind Entitäten Teil des L2-Caches. In den Entity-Klassen, die vom Caching ausgeschlossen werden müssen, können Sie die Annotation `@DataCache(enabled=false)` in der Entitätsklasse einfügen, die Sie vom L2-Cache ausschließen möchten.

```
import org.apache.openjpa.persistence.DataCache;
@Entity
@DataCache(enabled=false)
public class OpenJPACacheTest { ... }
```

2. Wenn Entitätsklassen in einer Vererbungshierarchie enthalten sind, werden untergeordnete Klassen der BackingMap der übergeordneten Klasse zugeordnet. Das bedeutet, dass eine Vererbungshierarchie eine einzige BackingMap nutzt.
3. Die ObjectGridQueryCache-Map ist für die Unterstützung von QueryCache erforderlich.
4. Die backingMapPluginCollection für jede Entitäts-Map erfordert, dass der ObjectTransformer die Klasse "com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" verwendet.
5. Die backingMapPluginCollection für ObjectGridQueryCache muss einen Schlüsselindex mit dem Namen "QueryCacheKeyIndex" haben, wie im folgenden Beispiel gezeigt wird.
6. Der Evictor ist für jede Map optional.

Datei `openjpa-objectGridDeployment.xml`

Die Datei `openjpa-objectGridDeployment.xml` wird verwendet, um die Implementierungsrichtlinie anzupassen. Es folgt die Datei `openjpa-objectGridDeployment.xml`, die der Datei `persistence.xml` entspricht:

`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>
```

Anmerkung: Die ObjectGridQueryCache-Map ist für die Unterstützung von QueryCache erforderlich.

Hibernate-Cache-Plug-in konfigurieren

Sie können den Cache für die Verwendung des Hibernate-Cache-Plug-in aktivieren, indem Sie Eigenschaftendateien angeben.

Vorbereitende Schritte

- Sie müssen die JPA-Cache-Plug-in-Topologie bestimmen, die Sie verwenden möchten. Weitere Informationen zu den verschiedenen Konfigurationen finden Sie unter „JPA-L2-Cache-Plug-in" auf Seite 339.
- Sie müssen eine Anwendung haben, die die JPA-APIs verwendet. Wenn Sie die APIs von WebSphere eXtreme Scale für den Datenzugriff mit JPA verwenden

möchten, verwenden Sie den JPA-Loader. Weitere Informationen finden Sie im Abschnitt „JPA-Loader konfigurieren" auf Seite 361.

Vorgehensweise

1. Wenn Sie WebSphere Application Server verwenden, speichern Sie die JAR-Dateien (Java-Archiv) an den entsprechenden Positionen für Ihre Konfiguration.

   Das Hibernate-Cache-Plug-in ist in der Datei `oghibernate-cache.jar` gepackt und im Verzeichnis *WAS-Stammverzeichnis*`/optionalLibraries/ObjectGrid` installiert. Wenn Sie das Hibernate-Cache-Plug-in verwenden möchten, müssen Sie die Datei `oghibernate-cache.jar` in die Hibernate-Bibliothek einschließen. Schließen Sie die Hibernate-Bibliothek beispielsweise in Ihre Anwendung ein, müssen Sie auch die Datei `oghibernate-cache.jar` einschließen. Wenn Sie eine gemeinsam genutzte Bibliothek definieren, in die Sie die Hibernate-Bibliothek einschließen, müssen Sie die Datei `oghibernate-cache.jar` dem Verzeichnis der gemeinsam genutzten Bibliothek hinzufügen.

   eXtreme Scale installiert die Datei `cglib.jar` nicht in der Umgebung von WebSphere Application Server. Wenn Sie vorhandene Anwendungen oder gemeinsam genutzte Bibliotheken haben wie Hibernate, die von der Datei `cglib.jar` abhängig sind, suchen Sie die Datei `cglib.jar`, und fügen Sie sie dem Klassenpfad hinzu. Enthält Ihre Anwendung beispielsweise alle JAR-Dateien für die Hibernate-Bibliothek, aber nicht die für Hibernate verfügbare Datei `cglib.jar`, müssen Sie die Datei `cglib.jar` von Hibernate in Ihre Anwendung einfügen.

2. Definieren Sie Eigenschaften in Ihrer Datei `persistence.xml`, um das Hibernate-Cache-Plug-in zu konfigurieren.

   Die Syntax für die Definition von Eigenschaften in der Datei `persistence.xml` ist wie folgt:

   ```
   <property name="hibernate.cache.provider_class"
           value="com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider" />
   <property name="hibernate.cache.use_query_cache" value="true"/>
   <property name="objectgrid.configuration" value="<Eigenschaft>=<Wert>,..." />
   <property name="objectgrid.hibernate.regionNames" value="<Regionsname>,.." />
   ```

   - **`hibernate.cache.provider_class`**: Der Wert der Eigenschaft **`provider_class`** ist die Klasse com.ibm.websphere.objectgrid.hibernate.cache. ObjectGridHibernateCacheProvider.
   - **`hibernate.cache.use_query_cache`**: Zum Aktivieren des Abfragecaches setzen Sie den Wert der Eigenschaft **`use_query_cache`** auf `true`.

     Anmerkung: Sie können den Abfragecache nur für integrierte und integrierte domäneninterne Topologien aktivieren.
   - **`objectgrid.configuration`**: Verwenden Sie die Eigenschaft "objectgrid.configuration", um die Cachekonfigurationseingenschaften von eXtreme Scale zu definieren, einschließlich des Attributs "ObjectGridType", das angibt, wie die Shards im Datengrid verteilt werden.

     Sie müssen einen eindeutigen Wert für die Eigenschaft "ObjectGridName" angeben, um potenzielle Namenskonflikte zu vermeiden. Die anderen Konfigurationseigenschaften für den eXtreme-Scale-Cache sind optional.

     Zum Aktivieren des Write-behind-Cachings verwenden Sie die folgenden Write-behind-Attribute in der Eigenschaft "objectgrid.configuration". Wenn Write-behind-Caching aktiviert ist, werden Aktualisierungen vorübergehend in einem JVM-bezogenen Datenspeicher gespeichert, bis die writeBehindInterval- oder writeBehindMaxBatchSize-Bedingungen erfüllt sind, wenn die Daten mit einer Flushoperation in den Cache geschrieben werden.

     ```
     writeBehind=true, writeBehindInterval=5000, writeBehindPoolSize=10, writeBehindMaxBatchSize=1000
     ```

Achtung: Wenn writeBehind nicht aktiviert ist, werden weitere Write-behind-Konfigurationseinstellungen ignoriert.

Weitere Informationen zu den Werten, die Sie in der Eigenschaft **objectgrid.configuration** festlegen können, finden Sie unter „Konfigurationseigenschaften des JPA-Caches” auf Seite 346.

- **objectgrid.hibernate.regionNames**: Die Eigenschaft "objectgrid.hibernate.regionNames" ist optional und muss angegeben werden, wenn nach der Initialisierung des eXtreme-Scale-Caches regionsNames-Werte definiert werden. Angenommen, eine Entitätsklasse ist einem regionName-Wert zugeordnet, und die Entitätsklasse ist weder in der Datei `persistence.xml` noch in der Hibernate-Zuordnungsdatei enthalten. Weiter angenommen, die Entitätsklasse hat eine Annotation "Entity". In diesem Fall wird der regionName-Wert für diese Entitätsklasse beim Laden der Klassen aufgelöst, wenn der eXtreme-Scale-Cache initialisiert wird. Ein weiteres Beispiel ist die Methode "Query.setCacheRegion(String regionName)", die nach der Initialisierung des Caches von eXtreme Scale ausgeführt wird. In diesen Situationen müssen Sie alle möglichen dynamisch bestimmten Regionsnamen (regionNames) in die Eigenschaft "objectgrid.hibernate.regionNames" einfügen, damit der eXtreme-Scale-Cache BackingMaps für alle Regionsnamen vorbereiten kann.

3. Optional: Zu weiteren Anpassung des vom Cache verwendeten Datengrids können Sie weitere Einstellungen mit XML-Dateien angeben.

   Für die meisten Szenarien reicht die Definition von Cacheeigenschaften aus. Wenn Sie das vom Cache verwendete ObjectGrid weiter anpassen möchten, können Sie Hibernate-ObjectGrid-XML-Konfigurationsdateien im Verzeichnis `META-INF` bereitstellen, wie z. B. die Datei `persistence.xml`. Während der Initialisierung versucht der Cache, diese XML-Dateien zu finden und sie dann zu verarbeiten.

   Es gibt drei Typen von ObjectGrid-XML-Konfigurationsdateien für Hibernate:

   - `hibernate-objectGrid.xml` (ObjectGrid-Konfiguration)

     Dateipfad: `META-INF/hibernate-objectGrid.xml`

     Standardmäßig hat jede Entitätsklasse einen zugeordneten Regionsnamen (standardmäßig den Namen der Entitätsklasse), der einer BackingMap-Konfiguration zugeordnet ist, die nach dem Regionsnamen in der ObjectGrid-Konfiguration benannt ist. Die Entitätsklasse "com.mycompany.Employee" hat beispielsweise standardmäßig einen zugeordneten Regionsnamen, der der BackingMap "com.mycompany.Employee" zugeordnet ist. Die BackingMap-Standardkonfiguration hat die Einstellungen readOnly="false", copyKey="false", lockStrategy="NONE" und `copyMode="NO_COPY"`. Sie können einige BackingMaps mit einer ausgewählten Konfiguration anpassen. Das reservierte Schlüsselwort "ALL_ENTITY_MAPS" kann verwendet werden, um alle Maps mit Ausnahme der angepassten Maps aus der Datei `hibernate-objectGrid.xml` darzustellen. BackingMaps, die nicht in der Datei `hibernate-objectGrid.xml` aufgelistet sind, verwenden die Standardkonfiguration.

   - `hibernate-objectGridDeployment.xml` (Implementierungsrichtlinie)

     Dateipfad: `META-INF/hibernate-objectGridDeployment.xml`

     Diese Datei wird verwendet, um die Implementierungsrichtlinie anzupassen. Wenn Sie bei der Anpassung der Implementierungsrichtlinie die Datei `hibernate-objectGridDeployment.xml` bereitstellen, wird die Standardimplementierungsrichtlinie verworfen. Alle Attributwerte für die Implementierungsrichtlinie werden der bereitgestellten Datei `hibernate-objectGridDeployment.xml` entnommen.

- `hibernate-objectGrid-client-override.xml` (ObjectGrid-Clientkorrekturkonfiguration)

  Dateipfad: `META-INF/hibernate-objectGrid-client-override.xml`

  Diese Datei wird verwendet, um ein clientseitiges ObjectGrid anzupassen. Standardmäßig wendet der ObjectGrid-Cache eine Standardkonfiguration für Clientkorrekturwerte an, die den nahen Cache inaktiviert. Wenn eine Anwendung einen nahen Cache erfordert, können Sie diese Datei bereitstellen und `numberOfBuckets="xxx"` angeben. Der Standardclientkorrekturwert inaktiviert den nahen Cache mit `numberOfBuckets="0"`. Der nahe Cache kann aktiv sein, wenn das Attribut "numberOfBuckets" über die Datei `hibernate-objectGrid-client-override.xml` auf einen Wert größer als 0 zurückgesetzt wird. Die Funktionsweise der Datei `hibernate-objectGrid-client-override.xml` gleicht der der Datei `hibernate-objectGrid.xml`: Sie überscheibt oder erweitert die Standardkonfiguration für ObjectGrid-Clientkorrekturwerte.

Je nach konfigurierter eXtreme-Scale-Topologie können Sie jede dieser drei XML-Dateien verwenden, um diese Topologie anzupassen.

Für die Typen EMBEDDED und EMBEDDED_PARTITION können Sie jede der drei XML-Dateien für die Anpassung des ObjectGrids, der Implementierungsrichtlinie oder der Konfiguration der ObjectGrid-Clientkorrekturwerte verwenden.

Bei ObjectGrids des Typs REMOTE erstellt der Cache kein dynamisches ObjectGrid. Vielmehr ruft der Cache ein clientseitiges ObjectGrid vom Katalogservice ab. Zum Anpassen der Konfiguration für die ObjectGrid-Clientkorrekturwerte können Sie nur eine Datei `hibernate-objectGrid-client-override.xml` verwenden.

4. Optional: (Nur für ferne Konfigurationen) Sie müssen ein externes eXtreme-Scale-System einrichten, wenn Sie einen Cache mit dem ObjectGrid-Typ REMOTE konfigurieren möchten.

   Sie müssen ein externes eXtreme-Scale-System einrichten, wenn Sie einen Cache mit dem ObjectGrid-Typ REMOTE konfigurieren möchten. Sie benötigen ObjectGrid- und ObjectGridDeployment-XML-Konfigurationsdateien, die auf der Datei `persistence.xml` basieren, um ein externes System einrichten zu können. Beispiele für diese Konfigurationsdateien finden Sie unter „Beispiel: Hibernate-ObjectGrid-XML-Dateien” auf Seite 359.

Ergebnisse

EMBEDDED- oder EMBEDDED_PARTITION-Konfiguration:

Wenn eine Anwendung gestartet wird, erkennt oder startet das Plug-in automatisch einen Katalogservice, startet einen Container-Server und stellt die Verbindung zum Katalogservice her. Das Plug-in kommuniziert anschließend mit dem ObjectGrid-Container und seinen Peers, die in anderen Anwendungsserverprozessen ausgeführt werden, über die Clientverbindung.

Jeder JPA-Entität wird über den Klassennamen der Entität eine unabhängige BackingMap zugeordnet. Jede BackingMap hat die folgenden Attribute:

- `readOnly="false"`
- `copyKey="false"`
- `lockStrategy="NONE"`
- `copyMode="NO_COPY"`

REMOTE-Konfiguration:

Die Implementierungsrichtlinie wird gesondert von der JPA-Anwendung spezifiziert. Ein externes ObjectGrid-System hat Katalogservice- und Container-Server-Prozesse. Sie müssen einen Katalogservice starten, bevor Sie Container-Server starten. Weitere Informationen finden Sie unter „Eigenständige Server starten” auf Seite 403 und unter „Container-Server starten” auf Seite 406.

Nächste Schritte

- Entwickeln Sie eine Hibernate-Anwendung, die die Konfiguration verwendet. Weitere Informationen finden Sie im Abschnitt Beispiel: Hibernate-Plug-in zum vorherigen Laden von Daten in den ObjectGrid-Cache verwenden.
- Erstellen Sie in einer Produktionsumgebung Katalogservicedomänen für Ihre automatisch erstellten Prozesse für die EMBEDDED- bzw. EMBEDDED_PARTITION-Konfiguration.
  - Eigenständige Umgebung:

    Wenn Sie Ihre Server nicht in einem Prozess von WebSphere Application Server ausführen, werden die Hosts und Ports der Katalogservicedomäne über eine Eigenschaftendatei mit dem Namen `objectGridServer.properties` angegeben. Diese Datei muss im Klassenpfad der Anwendung gespeichert werden und die definierte Eigenschaft **catalogServiceEndPoints** haben. Die Katalogservicedomäne wird unabhängig von den Anwendungsprozessen gestartet und muss vor den Anwendungsprozessen gestartet werden.

    Das Format der Datei `objectGridServer.properties` ist wie folgt:

    ```
    catalogServiceEndPoints=<Hostname1>:<Port1>,<Hostname2>:<Port2>
    ```
  - Umgebung mit WebSphere Application Server:

    Wenn Sie in einem Prozess von WebSphere Application Server arbeiten, stellt das JPA-Cache-Plug-in automatisch eine Verbindung zum Katalogservice bzw. zur Katalogservicedomäne her, der bzw. das für die Zelle von WebSphere Application Server definiert ist.
- Wenn Sie den ObjectGridTyp-Wert `EMBEDDED` oder `EMBEDDED_PARTITION` in einer Java-SE-Umgebung verwenden, verwenden Sie am Ende des Programms die Methode System.exit(0), um den integrierten eXtreme-Scale-Server zu stoppen. Andernfalls reagiert das Programm möglicherweise nicht mehr.

Beispiel: Hibernate-ObjectGrid-XML-Dateien:

Hibernate-ObjectGrid-XML-Dateien müssen auf der Basis einer Persistenzeinheit erstellt werden.

Datei `persistence.xml`

Im Folgenden sehen Sie eine Beispieldatei `persistence.xml`, die die Konfiguration einer Persistenzeinheit darstellt:

```
<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.provider_class"
        value="com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider" />
      <property name="hibernate.cache.use_query_cache" value="true" />
      <property name="objectgrid.configuration" value="ObjectGridType=EMBEDDED,
               ObjectGridName=Annuity, MaxNumberOfReplicas=4" />
    </properties>
  </persistence-unit>

</persistence>
```

Datei hibernate-objectGridDeployment.xml

Verwenden Sie die Datei hibernate-objectGridDeployment.xml, um die Richtliniendatei optional anzupassen. Wenn Sie diese Datei im Verzeichnis META-INF/hibernate-objectGridDeployment.xml bereitstellen, wird die Standardimplementierungsrichtlinie durch die Konfiguration in dieser Datei überschrieben.

```
<?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="defaultCacheMap" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <map ref="org.hibernate.cache.UpdateTimestampsCache" />
      <map ref="org.hibernate.cache.StandardQueryCache" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Datei hibernate-objectGrid.xml

Wenn Sie Hibernate nicht mit Java Persistence API (JPA) verwenden, verwenden Sie die folgende Beispieldatei hibernate-objectGrid.xml, um Ihre Hibernate-Konfiguration zu erstellen:

```
<?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="defaultCacheMap" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="defaultCacheMap" />
      <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.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.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.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="org.hibernate.cache.UpdateTimestampsCache" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="org.hibernate.cache.UpdateTimestampsCache" />
      <backingMap name="org.hibernate.cache.StandardQueryCache" readOnly="false" copyKey="false"
                  lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                  pluginCollectionRef="org.hibernate.cache.StandardQueryCache" />
    </objectGrid>
```

```
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="defaultCacheMap">
      <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="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
      <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="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
      <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="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="org.hibernate.cache.UpdateTimestampsCache">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>

    </backingMapPluginCollection>
    <backingMapPluginCollection id="org.hibernate.cache.StandardQueryCache">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Anmerkung: Die Maps "org.hibernate.cache.UpdateTimestampsCache", "org.hibernate.cache.StandardQueryCache" und "defaultCacheMap" sind erforderlich.

Datenbankintegration konfigurieren

Sie können WebSphere eXtreme Scale verwenden, um die Last der Datenbanken zu verringern. Sie können eine Java Persistence API (JPA) zwischen WebSphere eXtreme Scale und der Datenbank verwenden, um Änderungen als Ladeprogramm zu integrieren.

Vorbereitende Schritte

Eine Zusammenfassung der verschiedenen Topologien, die Sie mit einer Datenbank erstellen können, finden Sie im Artikel „Datenbankintegration: Write-behind, Inline- und Neben-Caching” auf Seite 17.

JPA-Loader konfigurieren

Ein JPA-Loader (Java Persistence API (JPA)) ist eine Plug-in-Implementierung, die JPA für die Interaktion mit der Datenbank verwendet.

Vorbereitende Schritte

- Sie müssen eine JPA-Implementierung wie Hibernate oder OpenJPA haben.
- Als Datenbank kann jedes Back-End verwendet werden, das vom ausgewählten JPA-Provider unterstützt wird.
- Legen Sie fest, ob Sie das Plug-in JPALoader oder das Plug-in JPAEntityLoader verwenden. Verwenden Sie das JPALoader-Plug-in, wenn Sie Daten mit der API "ObjectMap" speichern. Verwenden Sie das JPAEntityLoader-Plug-in, wenn Sie Daten mit der API "EntityManager" speichern.

Anmerkung: Wenn Sie die JPA-APIs für den Zugriff auf die JPA-Datenquelle verwenden, verwenden Sie das Plug-in für den JPA-L2-Cache. Das Cache-Plug-in führt das Datengrid zwischen Ihrer Anwendung und der JPA-Datenquelle ein, während weiterhin eine JPA-Anwendung verwendet wird. Weitere Informationen finden Sie unter „JPA-L2-Cache-Plug-in" auf Seite 339.

Informationen zu diesem Vorgang

Weitere Informationen zur Funktionsweise von Java Persistence API (JPA) Loader finden Sie unter JPA-Loader.

Vorgehensweise

1. Konfigurieren Sie die Parameter, die JPA erfordert, um mit einer Datenbank zu interagieren.

   Die folgenden Parameter sind erforderlich. Diese Parameter werden in der Bean JPALoader oder JPAEntityLoader und in der Bean JPATxCallback konfiguriert.

   - **persistenceUnitName**: Gibt den Namen der Persistenzeinheit an. Dieser Parameter wird für zwei Zwecke benötigt: zum Erstellen einer JPA-EntityManagerFactory und für das Suchen der JPA-Entitätsmetadaten in der Datei `persistence.xml`. Dieses Attribut wird in der Bean JPATxCallback gesetzt.
   - **JPAPropertyFactory**: Gibt die Factory zum Erstellen einer Map für Persistenzeigenschaften an, mit denen die Standardpersistenzeigenschaften überschrieben werden sollen. Dieses Attribut wird in der Bean JPATxCallback gesetzt. Um dieses Attribut zu setzen, ist eine Spring-Konfiguration erforderlich.
   - **entityClassName**: Gibt den Namen der Entitätsklasse an, die erforderlich ist, um JPA-Methoden wie EntityManager.persist, EntityManager.find usw. zu verwenden. Das Plug-in JPALoader erfordert diesen Parameter, aber der Parameter ist für JPAEntityLoader optional. Wenn der Parameter **entityClassName** für das Plug-in JPAEntityLoader nicht konfiguriert ist, wird die in der ObjectGrid-Entitäts-Map konfigurierte Entitätsklasse verwendet. Sie müssen für den EntityManager von eXtreme Scale und den JPA-Provider dieselbe Klasse verwenden. Dieses Attribut wird in der Bean JPALoader oder JPAEntityLoader gesetzt.
   - **preloadPartition**: Gibt die Partition an, bei der der Preload-Prozess für die Map gestartet wird. Wenn die Preload-Partitionsnummer kleiner als null oder größer als die Gesamtanzahl der Partitionen minus 1 ist, wird der Preload-Prozess für die Map nicht gestartet. Der Standardwert ist -1 und bedeutet, dass der Preload-Prozess standardmäßig nicht gestartet wird. Dieses Attribut wird in der Bean JPALoader oder JPAEntityLoader gesetzt.

   Neben den JPA-Parametern, die in eXtreme Scale konfiguriert werden müssen, werden JPA-Metadaten verwendet, um den Schlüssel von den JPA-Entitäten abzurufen. Die JPA-Metadaten können als Annotation oder in einer Datei `orm.xml` konfiguriert werden, die in der Datei `persistence.xml` angegeben wird. Sie sind nicht Teil der Konfiguration von eXtreme Scale.

2. Konfigurieren Sie XML-Dateien für die JPA-Konfiguration.

   Informationen zum Konfigurieren des Plug-ins JPALoader oder JPAEntityLoader finden Sie unter Plug-ins für die Kommunikation mit Datenbanken.

   Konfigurieren Sie ein JPATxCallback-Transaktions-Callback zusammen mit der Loader-Konfiguration. Das folgende Beispiel ist eine ObjectGrid-XML-Deskriptordatei (`objectgrid.xml`), in der ein JPAEntityLoader und ein JPATxCallback konfiguriert sind:

   Loader mit Callback konfigurieren - XML-Beispiel

```
<?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>
```

Wenn Sie eine JPAPropertyFactory konfigurieren möchten, müssen Sie eine Spring-Konfiguration verwenden. Im Folgenden sehen Sie eine XML-Beispielkonfigurationsdatei mit dem Namen `JPAEM_spring.xml`, in der eine Spring-Bean für die Konfigurationen von eXtreme Scale konfiguriert wird.

Loader mit JPA-Eigenschaften-Factory konfigurieren - XML-Beispiel

```
<?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>
```

Die XML-Konfigurationsdatei `Objectgrid.xml` folgt. Beachten Sie, dass der ObjectGrid-Name `JPAEM` ist und dem ObjectGrid-Namen in der Spring-Konfigurationsdatei `JPAEM_spring.xml` entspricht.

JPAEM-Loader-Konfiguration - XML-Beispiel

```
<?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>
```

Eine Entität kann mit den JPA-Annotationen und mit den EntityManager-Annotationen von eXtreme Scale annotiert werden. Jede Annotation hat ein funktio-

nal entsprechendes XML-Element, das verwendet werden kann. Deshalb wurde in eXtreme Scale der Spring-Namespace hinzugefügt. Sie können diese Annotationen auch über die Spring-Namespace-Unterstützung konfigurieren. Weitere Informationen finden Sie im Abschnitt Übersicht über das Spring-Framework.

Zeitbasierte JPA-Aktualisierungskomponente konfigurieren

Sie können eine zeitbasierte Datenbankaktualisierung mit XML für eine lokale oder verteilte Konfiguration von eXtreme Scale konfigurieren. Eine lokale Konfiguration kann auch programmgesteuert konfiguriert werden.

Informationen zu diesem Vorgang

Weitere Informationen zur Funktionsweise der zeitbasierten JAP-Datenaktualisierungskomponente finden Sie im Abschnitt Zeitbasierte JPA-Datenaktualisierungskomponente.

Vorgehensweise

Erstellen Sie eine timeBasedDBUpdate-Konfiguration.

- **Mit einer XML-Datei:**

  Das folgende Beispiel zeigt eine Datei `objectgrid.xml`, die eine timeBasedDBUpdate-Konfiguration enthält:

  Zeitbasierte JPA-Aktualisierungskomponente - XML-Beispiel

```
<?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>
```

  In diesem Beispiel wird die Map "user" mit einer zeitbasierten Datenbankaktualisierung konfiguriert. Der Datenbankaktualisierungsmodus ist INVALIDATE_ONLY, und das Zeitmarkenfeld ist "rowChgTs".

  Wenn das verteilte ObjectGrid "changeOG" im Container-Server gestartet wird, wird automatisch ein Thread für die zeitbasierte Datenbankaktualisierung in Partition 0 gestartet.

- **Programmgesteuert:**

  Wenn Sie ein lokales ObjectGrid erstellen, können Sie auch ein TimeBasedDBUpdateConfig-Objekt erstellen und in der BackingMap-Instanz definieren:

```
public void setTimeBasedDBUpdateConfig(TimeBasedDBUpdateConfig dbUpdateConfig);
```

  Weitere Informationen zum Definieren eines Objekts in der BackingMap-Instanz finden Sie in den Informationen zur Schnittstelle "BackingMap" in der API-Dokumentation.

  Alternativ können Sie das Zeitmarkenfeld in der Entitätsklasse mit der Annotation "com.ibm.websphere.objectgrid.jpa.dbupdate.annotation.Timestamp" annotieren. Wenn Sie den Wert in der Klasse konfigurieren, müssen Sie das Zeitmarkenfeld nicht in der XML-Konfiguration konfigurieren.

Nächste Schritte

Starten Sie die zeitbasierte JPA-Datenaktualisierungskomponente. Weitere Informationen finden Sie unter Zeitbasierte JPA-Aktualisierungskomponente starten.

REST-Datenservices konfigurieren

Sie können den REST-Datenservice von WebSphere eXtreme Scale mit WebSphere Application Server Version 7.0, WebSphere Application Server Community Edition und Apache Tomcat verwenden.

Informationen zu diesem Vorgang

Das enthaltene Beispiel setzt sich aus Quellcode und kompilierten Binärdateien für die Ausführung eines partitionierten Datengrids zusammen. Dieses Beispiel veranschaulicht, wie ein einfaches Datengrid, das Modell und die Daten mit Entitäten erstellt werden und stellt zwei Befehlszeilenclientanwendungen bereit, mit denen Entitäten unter Verwendung von Java oder C# hinzugefügt und abgefragt werden können.

Der Java-Beispielclient verwendet die Java-API "EntityManager", um Daten im Datengrid persistent zu speichern und Daten abzufragen. Dieser Client kann in Eclipse oder über ein Befehlszeilenscript ausgeführt werden. Der Java-Beispielclient demonstriert den REST-Datenservice zwar nicht, ermöglicht aber die Aktualisierung der Daten im Grid, so dass ein Webbrowser oder andere Clients die Daten lesen können.

Der in C# geschriebene Microsoft-WCF-Data-Services-Beispielclient kommuniziert mit dem eXtreme-Scale-Datengrid über den REST-Datenservice unter Verwendung des .NET-Frameworks. Der WCF-Data-Services-Client kann zum Aktualisieren und Abfragen des Datengrids verwendet werden.

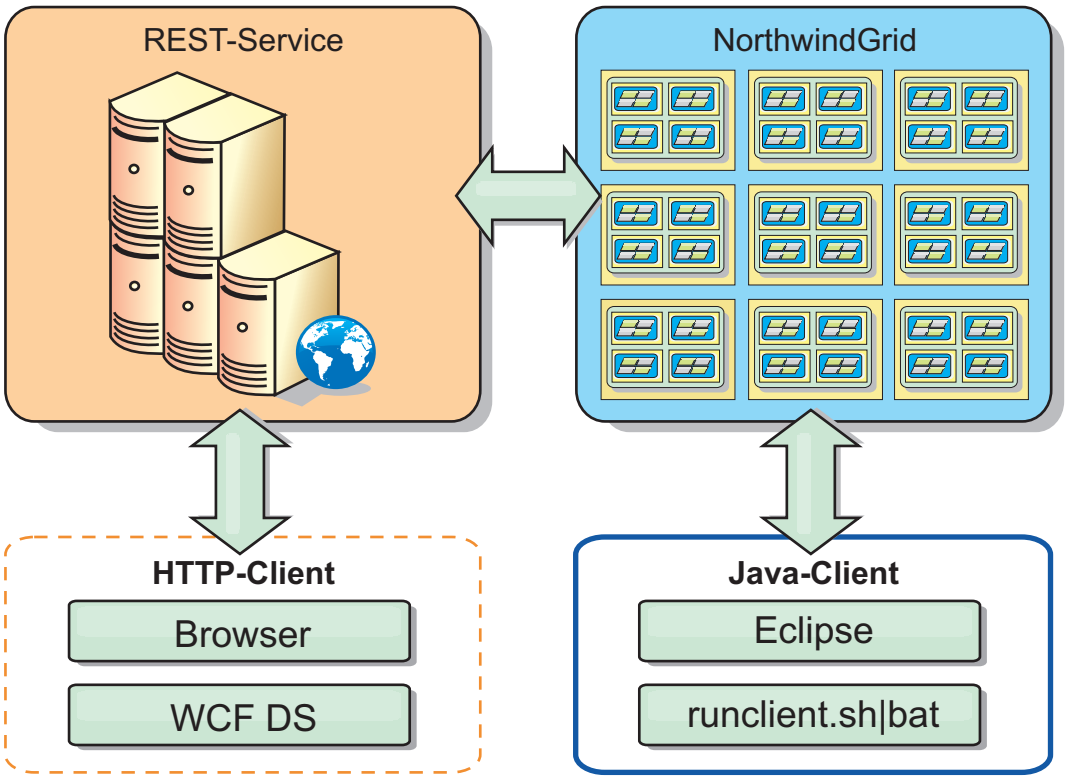


Abbildung 43. Beispieltopologie zur Einführung. HTTP-Clients, die den REST-Datenservice verwenden, und Java-Clients können auf dasselbe Datengrid zugreifen.

Vorgehensweise

1. eXtreme-Scale-Datengrid konfigurieren und starten. Weitere Informationen finden Sie unter „REST-Datenservice aktivieren".
2. Konfigurieren und starten Sie den REST-Datenservice in einem Webserver. Weitere Informationen finden Sie unter „Anwendungsserver für den REST-Datenservice konfigurieren" auf Seite 375.
3. Führen Sie einen Client aus, um mit dem REST-Datenservice zu interagieren. Es sind zwei Optionen verfügbar:
   a. Führen Sie den Java-Beispielclient aus, um das Grid unter Verwendung der API "EntityManager" mit Daten zu füllen und die Daten im Grid unter Verwendung eines Webbrowsers und des REST-Datenservice von eXtreme Scale abzufragen. Lesen Sie hierzu den Artikel „Java-Client mit REST-Datenservices verwenden" auf Seite 392.
   b. Führen Sie den in C# geschriebenen WCF-Data-Services-Beispielclient aus. Lesen Sie hierzu den Artikel „WCF-Client von Visual Studio 2008 mit dem REST-Datenservice" auf Seite 394.

REST-Datenservice aktivieren

Der REST-Datenservice kann Entitätsmetadaten von WebSphere eXtreme Scale für die Darstellung jeder Entität als EntitySet darstellen.

eXtreme-Scale-Beispieldatengrid starten

Im Allgemeinen starten Sie das eXtreme-Scale-Datengrid, bevor Sie den REST-Datenservice starten. Mit den folgenden Schritten werden ein einziger Katalogserviceprozess von eXtreme Scale und zwei Container-Serverprozesse gestartet.

Für die Installation von WebSphere eXtreme Scale stehen drei verschiedene Methoden zur Verfügung:

- Testinstallation
- Eigenständige Implementierung
- Integrierte Implementierung von WebSphere Application Server

Skalierbares Datenmodell in eXtreme Scale

Im Beispiel "Microsoft Northwind" wird die Tabelle "Order Detail" verwendet, um eine N:N-Assoziation zwischen "Orders" und "Products" herzustellen.

Mit Hilfe von ORM-Spezifikationen (Object to Relational Mapping) wie ADO.NET Entity Framework und Java Persistence API (JPA) können die Tabellen und Beziehungen über Entitäten zugeordnet werden. Diese Architektur ist jedoch nicht skalierbar. Alle Komponenten müssen sich auf derselben Maschine oder in einem kostenintensiven Maschinencluster finden, um eine angemessene Leistung zu erzielen.

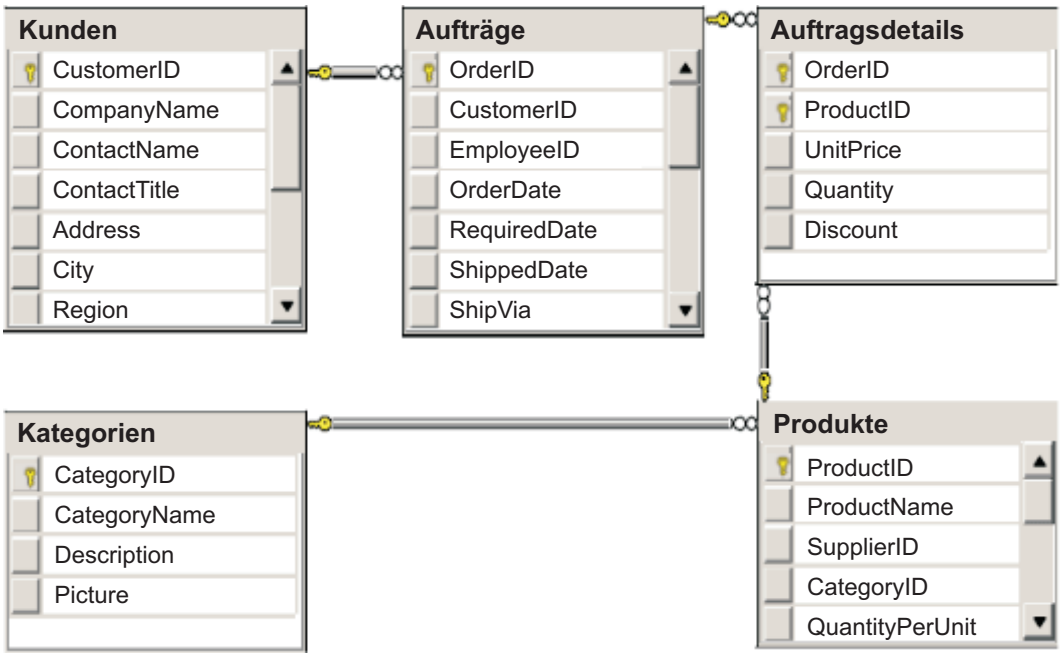


Abbildung 44. Schemadiagramm zum Beispiel "Microsoft SQL Server Northwind"

Zum Erstellen einer skalierbaren Version des Beispiels müssen die Entitäten so modelliert werden, dass jede Entität oder Gruppe zusammengehöriger Entitäten auf der Basis eines einzigen Schlüssels partitioniert werden können. Durch die Erstellung von Partitionen auf der Basis eines einzigen Schlüssels können Anforderungen auf mehrere unabhängige Server verteilt werden. Für diese Konfiguration wurden die Entitäten in zwei Baumstrukturen aufgeteilt, in die Baumstruktur "Customer und Order" und in die Baumstruktur "Product und Category". In diesem Modell können beide Baumstrukturen unabhängig voneinander partitioniert werden und deshalb unterschiedlich anwachsen, was die Skalierbarkeit erhöht.

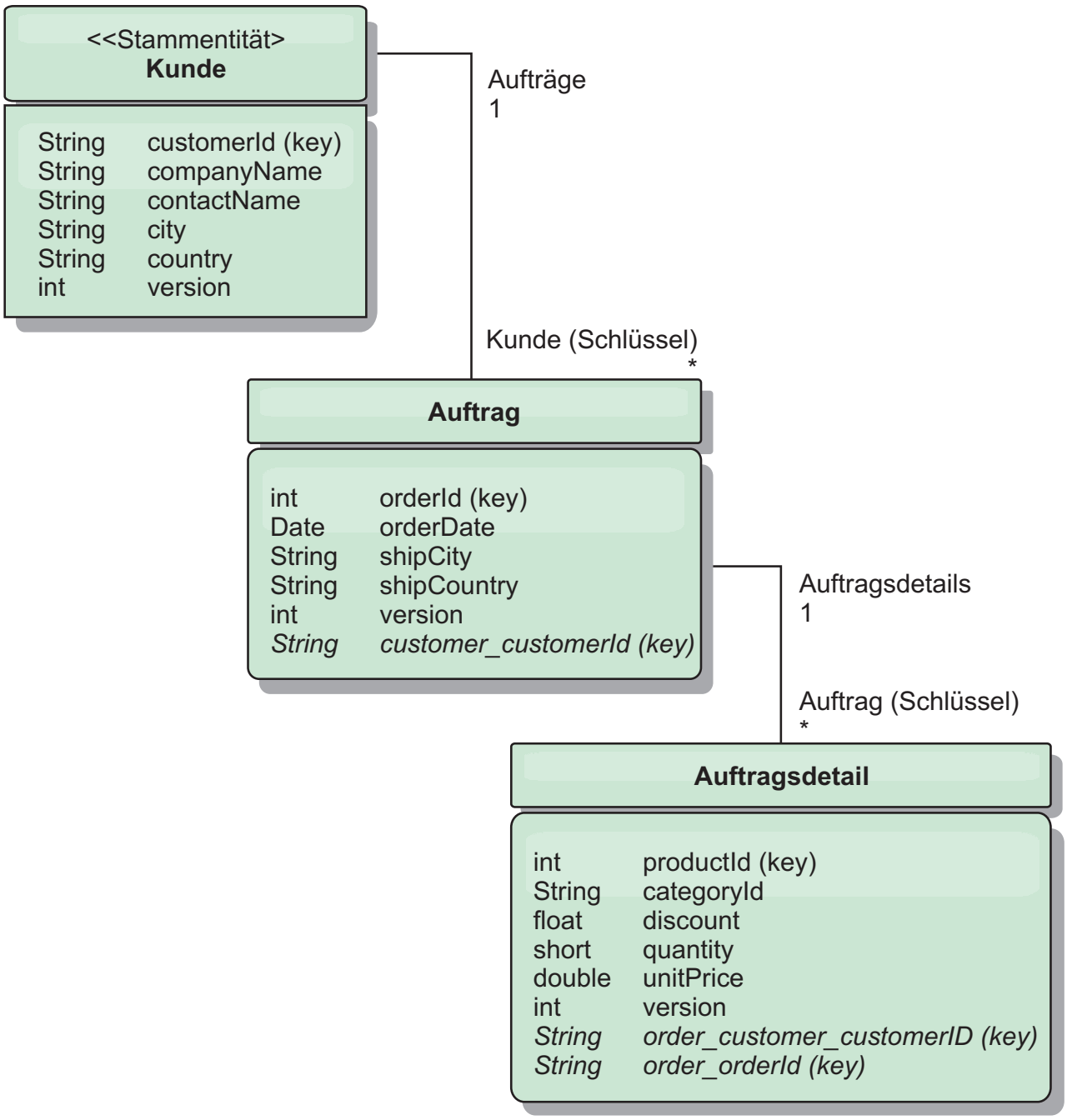


Abbildung 45. Entitätsschemadiagramm "Customer und Order"

Order und Product haben eindeutige, separate ganze Zahlen als Schlüssel. Die Tabellen "Order" und "Product" sind also voneinander unabhängig. Stellen Sie sich beispielsweise die Auswirkungen der Größe eines Katalogs, d. h. die Anzahl der von Ihnen vertriebenen Produkte, mit der Gesamtanzahl an Bestellungen vor. Intuitiv geht man davon aus, dass das Vorhandensein vieler Produkte bedeutet, dass auch viele Bestellungen vorhanden sind, aber das muss nicht unbedingt der Fall sein. Würde diese Annahme stimmen, könnten Sie Ihre Verkaufszahlen problemlos steigern, indem Sie Ihrem Katalog einfach weitere Produkte hinzufügen. Bestellungen und Produkte haben eigene voneinander unabhängige Tabellen. Sie können dieses Konzept noch erweitern und für Bestellungen und Produkte jeweils eigene separate Datengrids verwenden. Mit unabhängigen Datengrids können Sie die An-

zahl der Partitionen und Servern sowie die Größe jedes Datengrids gesondert steuern, so dass Ihre Anwendung skalieren kann. Wenn Sie die Größe Ihres Katalogs verdoppeln, müssen Sie das Produktdatengrid verdoppeln, aber das Grid für die Bestellungen bleibt unverändert. Der umgekehrte Fall gilt für eine Bestellspitze oder eine erwartete Bestellspitze.

In dem Schema hat ein Kunde (Customer) null oder mehr Bestellungen (Order), und eine Bestellung Bestellpositionen (OrderDetail), jeweils mit einem bestimmten Produkt. Ein Produkt (Product) wird in jedem OrderDetail anhand seiner ID (dem Produktschlüssel) identifiziert. Customer, Order und OrderDetails werden in einzigen Datengrid gespeichert, mit Customer als Stammentität des Datengrids. Sie können Kunden nach ID abrufen, müssen Bestellungen aber anhand der Kunden-ID abrufen. Deshalb wird die Kunden-ID der Bestellung im Schlüssel hinzugefügt. Analog dazu sind Kunden-ID und Bestell-ID Teil der OrderDetail-ID.

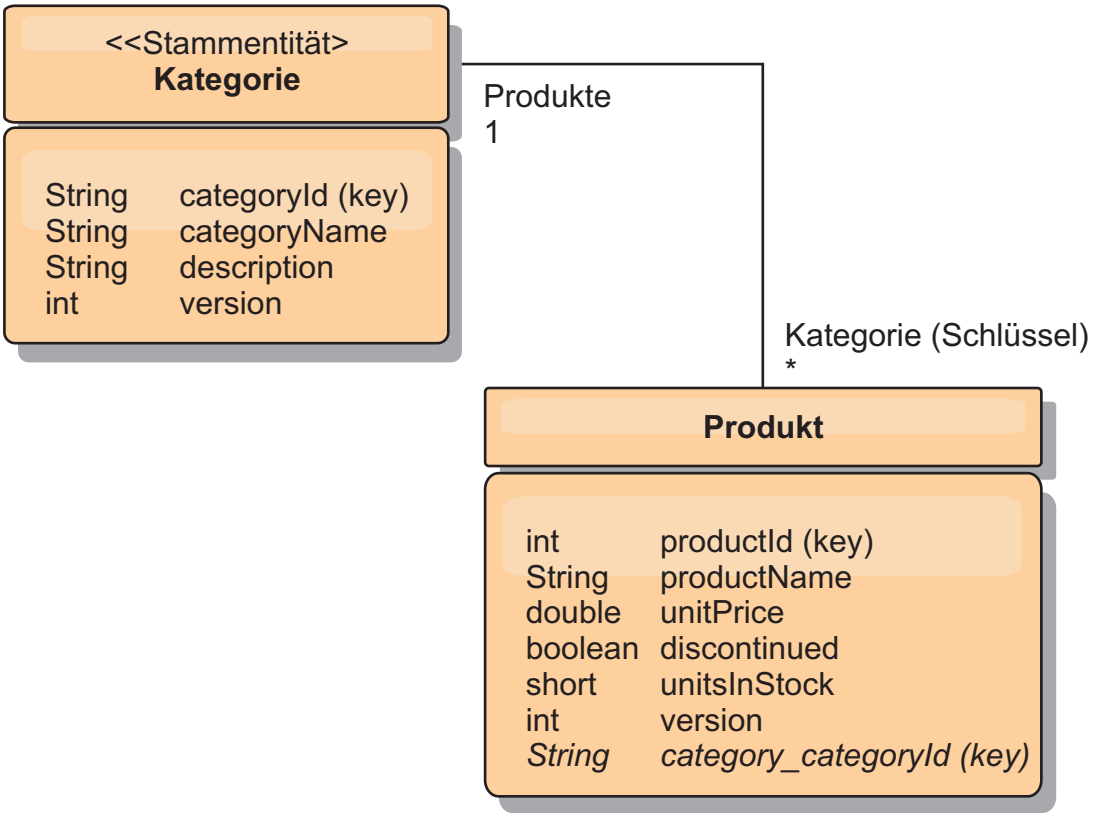


Abbildung 46. Entitätsschemadiagramm "Category und Product"

Im Schema "Category and Product" ist "Category" der Schemastamm. Mit diesem Schema können Kunden Produkte nach Kategorie abfragen. Weitere Einzelheiten zu Schlüsselassoziationen und deren Bedeutung finden Sie unter „Daten mit REST abrufen und aktualisieren”.

Daten mit REST abrufen und aktualisieren

Das Protokoll "OData" erfordert, dass alle Entitäten über ihre kanonische Form adressiert werden können. Das bedeutet, dass jede Entität den Schlüssel der partitionierten Stammentität, den Schemastamm, enthalten muss.

Im Folgenden sehen Sie ein Beispiel für die Verwendung der Assoziation von einer Stammentität für die Adressierung einer untergeordneten Entität in:

```
/Customer('ACME')/order(100)
```

In WCF Data Services muss die untergeordnete Entität direkt adressierbar sein, d. h., der Schlüssel im Schemastamm muss Teil des Schlüssels der untergeordneten Entität sein: `/Order(customer_customerId='ACME', orderId=100)`. Dies wird erreicht, indem eine Assoziation zur Stammentität erstellt wird, wobei die 1:1- bzw. N:1-Assoziation zur Stammentität auch als Schlüssel bezeichnet wird. Wenn Entitäten in den Schlüssel eingeschlossen werden, werden die Attribute der übergeordneten Entität als Schlüsseleigenschaften bereitgestellt.

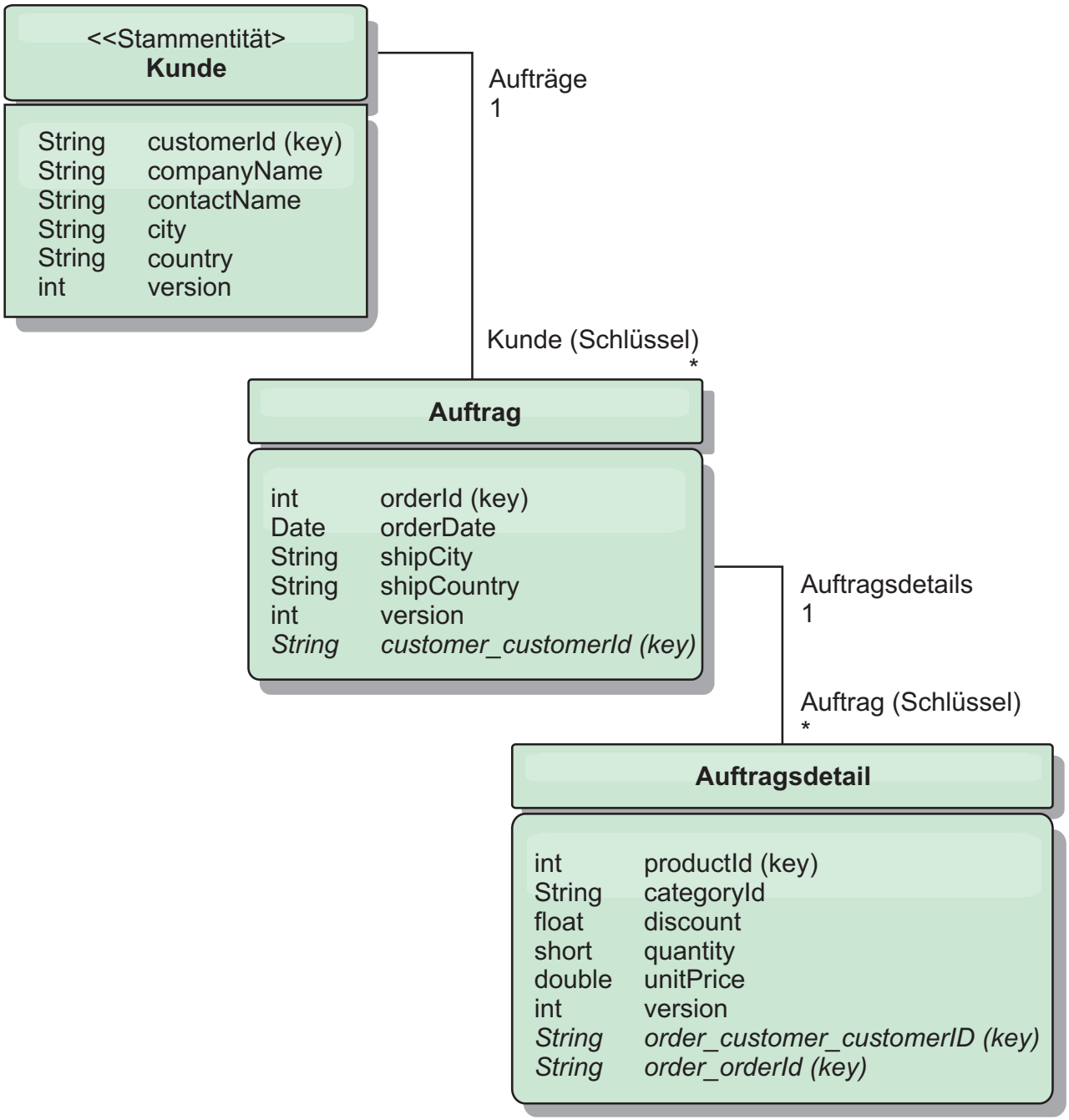


Abbildung 47. Entitätsschemadiagramm "Customer und Order"

Das Customer/Order-Entitätsschemadiagramm veranschaulicht, wie jede Entität unter Verwendung des Customer partitioniert wird. Die Entität "Order" enthält den Customer als Teil ihres Schlüssels und ist deshalb direkt adressierbar. Der REST-Datenservice stellt alle Schlüsselassoziationen als einzelne Eigenschaften bereit: Order hat customer_customerId, und OrderDetail hat order_customer_customerId und order_orderId.

Mit der API "EntityManager" können Sie die Order-Entität anhand der Customer- und Order-ID suchen:

```
transaction.begin();
// Order mit Customer suchen. Die ID wird erst dann in die Klasse
// Customer eingeschlossen, wenn die OrderId-Schlüsselinstanz erstellt wird.
Order order = (Order) em.find(Order.class,
    new OrderId(100, new Customer('ACME')));
...
transaction.commit();
```

Mit dem REST-Datenservice kann die Order-Entität über beide der folgenden URLs abgerufen werden:

- `/Order(orderId=100, customer_customerId='ACME')`
- `/Customer('ACME')/orders?$filter=orderId eq 100`

Auf den Kundenschlüssel wird über den Attributnamen der Customer-Entität, ein Unterstreichungszeichen und den Attributnamen der Customer-ID, customer_customerId, zugegriffen.

Eine Entität kann auch eine Entität, die keine Stammentität ist, in ihrem Schlüssel enthalten, wenn alle Vorgänger dieser Entität Assoziationen zur Stammentität haben. In diesem Beispiel hat OrderDetail eine Schlüsselassoziation zu Order und Order eine Schlüsselassoziation zur Customer-Stammentität. Verwendung der API "EntityManager":

```
transaction.begin();
// OrderDetailId-Schlüsselinstanz erstellen. Sie
// enthält Order und Customer, bei denen nur
// der Schlüssel definiert ist.
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);
...
```

Der REST-Datenservice lässt die direkte Adressierung von OrderDetail zu:

```
/OrderDetail(productId=500, order_customer_customerId='ACME', order_orderId =100)
```

Die Assoziation von der Entität "OrderDetail" zur Entität "Product" wurde unterbrochen, um die unabhängige Partitionierung von Order- und Product-Inventar zuzulassen. Die Entität "OrderDetail" speichert die Kategorie und die Produkt-ID anstelle einer festen Beziehung. Durch die Entkopplung der beiden Entitätsschemas wird jeweils nur auf eine einzige Partition zugegriffen.

Das im Diagramm dargestellte Category/Product-Schema zeigt, dass "Category" die Stammentität ist und dass jedes "Product" eine Assoziation zu einer Entität "Category" hat. Die Entität "Category" ist in der Product-ID enthalten. Der REST-Datenservice stellt eine Schlüsseleigenschaft bereit, category_categoryId, die über die das Product direkt adressiert werden kann.

Da Category die Stammentität ist, muss die Category in einer partitionierten Umgebung bekannt sein, damit das Product gefunden wird. Wenn die API "EntityManager" verwendet wird, muss die Transaktion vor der Suche des Product an die Entität "Category" gebunden werden.

Verwendung der API "EntityManager":

```
transaction.begin();
// Stammentität Category nur mit dem Schlüssel erstellen. Auf diese
// Weise kann eine ProductId erstellt werden, ohne zuerst die
// Category suchen zu müssen. Die Transaktion ist jetzt an die
// Partition gebunden, in der die Category "COMP" gespeichert ist.
Category cat = new Category("COMP");
Product product = (Product) em.find(Product.class,
    new ProductId(500, cat));
...
```

Der REST-Datenservice ermöglicht die direkte Adressierung des Product:

```
/Product(productId=500, category_categoryId='COMP')
```

Eigenständiges Datengrid für REST-Datenservices starten

Führen Sie die folgenden Schritte aus, u das Beispieldatengrind für den REST-Service von WebSphere eXtreme Scale für eine eigenständige Implementierung von eXtreme Scale zu starten.

Vorbereitende Schritte

Installieren Sie die Testversion von WebSphere eXtreme Scale oder das vollständige Produkt:

- Installieren Sie die eigenständige Version des Produkts, und wenden Sie alle nachfolgenden Fixes an.
- Laden Sie die Testversion von WebSphere eXtreme Scale Version 7.1 oder höher herunter, die den REST-Datenservice von WebSphere eXtreme Scale enthält, und entpacken Sie sie.

Informationen zu diesem Vorgang

Starten Sie das Beispieldatengrid von WebSphere eXtreme Scale.

Vorgehensweise

1. Starten Sie den Katalogserviceprozess. Öffnen Sie eine Befehlszeile oder ein Terminalfenster, und setzen Sie die Umgebungsvariable JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=`*`Java-Ausgangsverzeichnis`*
   - Windows `set JAVA_HOME=`*`Java-Ausgangsverzeichnis`*
2. `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
3. Starten Sie den Katalogserviceprozess. Wenn Sie den Service *ohne* eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Wenn Sie den Service mit eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`

4. Starten Sie zwei Container-Serverprozesse. Öffnen Sie eine weitere Befehlszeile oder ein Terminalfenster, und setzen Sie die Umgebungsvariable JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=Java-Ausgangsverzeichnis`
   - Windows `set JAVA_HOME=Java-Ausgangsverzeichnis`
5. `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
6. Starten Sie einen Container-Serverprozess:

   Wenn Sie den Server ohne eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
   - Linux UNIX `./runcontainer.sh container0`
   - Windows `runcontainer.bat container0`

   Wenn Sie den Server mit eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
   - Linux UNIX `./runcontainer_secure.sh container0`
   - Windows `runcontainer_secure.bat container0`
7. Öffnen Sie eine weitere Befehlszeile oder ein Terminalfenster, und setzen Sie die Umgebungsvariable JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=Java-Ausgangsverzeichnis`
   - Windows `set JAVA_HOME=Java-Ausgangsverzeichnis`
8. `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
9. Starten Sie einen zweiten Container-Serverprozess.

   Wenn Sie den Server ohne eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle.
   - Linux UNIX `./runcontainer.sh container1`
   - Windows `runcontainer.bat container1`

   Wenn Sie den Server mit eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
   - Linux UNIX `./runcontainer_secure.sh container1`
   - Windows `runcontainer_secure.bat container1`

Ergebnisse

Warten Sie, bis die eXtreme-Scale-Container bereit sind, bevor Sie mit den nächsten Schritten fortfahren. Die Container-Server sind bereit, wenn die folgende Nachricht im Terminalfenster angezeigt wird:

```
CWOBJ1001I: Der ObjectGrid-Server Containername ist für die Verarbeitung
von Anforderungen bereit.
```

Containername steht für den Namen des gestarteten Containers.

Datengrid für REST-Datenservices in WebSphere Application Server starten

Führen Sie die folgenden Schritte aus, um ein eigenständiges Beispieldatengrid für den REST-Service von WebSphere eXtreme Scale für eine in WebSphere Application Server integrierte Implementierung von WebSphere eXtreme Scale zu starten. Ob-

wohl WebSphere eXtreme Scale mit WebSphere Application Server integriert ist, starten Sie mit diesen Schritten einen eigenständigen Katalogserviceprozess und -container von WebSphere eXtreme Scale.

Vorbereitende Schritte

Installieren Sie das Produkt in einem Installationsverzeichnis von WebSphere Application Server Version 7.0.0.5 oder höher mit aktivierter Sicherheit. Erweitern Sie mindestens ein Anwendungsserverprofil.

Informationen zu diesem Vorgang

Starten Sie das Beispieldatengrid von WebSphere eXtreme Scale.

Vorgehensweise

1. Starten Sie den Katalogserviceprozess. Öffnen Sie eine Befehlszeile oder ein Terminalfenster, und setzen Sie die Umgebungsvariable JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=`*`Java-Ausgangsverzeichnis`*
   - Windows `set JAVA_HOME=`*`Java-Ausgangsverzeichnis`*

   `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
2. Starten Sie den Katalogserviceprozess.

   Wenn Sie den Server ohne eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle.
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Wenn Sie den Server mit eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
3. Starten Sie zwei Container-Serverprozesse. Öffnen Sie eine weitere Befehlszeile oder ein Terminalfenster, und setzen Sie die Umgebungsvariable JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=`*`Java-Ausgangsverzeichnis`*
   - Windows `set JAVA_HOME=`*`Java-Ausgangsverzeichnis`*
4. Starten Sie einen Container-Serverprozess.

   Wenn Sie den Server ohne eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle.
   a. Öffnen Sie ein Befehlszeilenfenster.
   b. `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
   c. Zum Starten des Servers *ohne* eXtreme-Scale-Sicherheit verwenden Sie die folgenden Befehle:
      - Linux UNIX `./runcontainer.sh container0`
      - Windows `runcontainer.bat container0`
   d. Wenn Sie den Server mit eXtreme-Scale-Sicherheit starten möchten, verwenden Sie die folgenden Befehle:
      - Linux UNIX `./runcontainer_secure.sh container0`
      - Windows `runcontainer_secure.bat container0`

5. Starten Sie einen zweiten Container-Serverprozess.
   a. Öffnen Sie ein Befehlszeilenfenster.
   b. `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
   c. Zum Starten des Servers *ohne* eXtreme-Scale-Sicherheit verwenden Sie die folgenden Befehle:
      - Linux UNIX `./runcontainer.sh container1`
      - Windows `runcontainer.bat container1`
   d. Zum Starten des Servers *mit* eXtreme-Scale-Sicherheit verwenden Sie die folgenden Befehle:
      - Linux UNIX `./runcontainer_secure.sh container1`
      - Windows `runcontainer_secure.bat container1`

Ergebnisse

Warten Sie, bis die Container-Server bereit sind, bevor Sie mit den nächsten Schritten fortfahren. Die Container-Server sind bereit, wenn die folgende Nachricht angezeigt wird:

`CWOBJ1001I: Der ObjectGrid-Server` *`Containername`* `ist für die Verarbeitung von Anforderungen bereit.`

Containername steht für den Namen des im vorherigen Schritt gestarteten Containers.

Anwendungsserver für den REST-Datenservice konfigurieren

Sie können verschiedene Anwendungsserver konfigurieren, um den REST-Datenservice zu verwenden.

REST-Datenservice in WebSphere Application Server implementieren

In diesem Abschnitt wird beschrieben, wie Sie den REST-Datenservice von WebSphere eXtreme Scale in WebSphere Application Server oder WebSphere Application Server Network Deployment Version 6.1.0.25 oder höher konfigurieren. Diese Anweisungen gelten auch für Implementierungen, in denen WebSphere eXtreme Scale mit der Implementierung von WebSphere Application Server integriert ist.

Vorbereitende Schritte

Sie müssen eine der folgenden Umgebungen auf Ihrem System haben, um den REST-Datenservice für WebSphere eXtreme Scale zu konfigurieren und zu implementieren.

- WebSphere Application Server mit eigenständigem Client von WebSphere eXtreme Scale:
  - WebSphere eXtreme Scale Trial Version 7.1 mit dem REST-Datenservice wurde heruntergeladen und entpackt, oder das Produkt WebSphere eXtreme Scale Version 7.1.0.0 mit dem kumulativen Fix 2 wurde in einem eigenständigen Verzeichnis installiert.
  - WebSphere Application Server Version 6.1.0.25 oder 7.0.0.5 oder höher ist installiert und aktiv.
- WebSphere Application Server mit WebSphere eXtreme Scale integriert:

WebSphere eXtreme Scale Version 7.1.0.0 mit dem kumulativen Fix 2 oder höher wurde in WebSphere Application Server Version 6.1.0.25 oder 7.0 oder höher installiert.

Tipp: Der REST-Datenservice von WebSphere eXtreme Scale setzt nur voraus, dass die Clientoption von WebSphere eXtreme Scale installiert ist. Das Profil muss nicht erweitert werden.

Informationen zum Aktivieren der Java-2-Sicherheit finden Sie im Information Center von WebSphere Application Server.

Vorgehensweise

1. Konfigurieren und starten Sie ein Datengrid.
   a. Einzelheiten zum Konfigurieren eines Datengrids für den REST-Datenservice finden Sie unter „Datengrid für REST-Datenservices in WebSphere Application Server starten" auf Seite 373.
   b. Vergewissern Sie sich, dass ein Client eine Verbindung zum Grid herstellen und auf die Entitäten im Datengrid zugreifen kann. Ein Beispiel finden Sie unter „Lernprogramm: Einführung in WebSphere eXtreme Scale" auf Seite 1.
2. Erstellen Sie die Konfigurations-JAR-Datei oder das Konfigurationsverzeichnis für den REST-Datenservice von eXtreme Scale. Informationen zum Packen und Implementieren des REST-Service finden Sie unter „REST-Datenservice installieren" auf Seite 205.
3. Fügen Sie die Konfigurations-JAR-Datei bzw. das Konfigurationsverzeichnis für den REST-Datenservice dem Klassenpfad des Anwendungsservers hinzu:
   a. Öffnen Sie die Administrationskonsole von WebSphere Application Server.
   b. Navigieren Sie zu **Umgebung** > **Gemeinsam genutzte Bibliotheken**.
   c. Klicken Sie auf **Neu**.
   d. Fügen Sie den entsprechenden Feldern die folgenden Einträge hinzu:
      - Name: `extremescale_rest_configuration`
      - Klassenpfad: <Konfigurations-JAR oder -Verzeichnis für REST-Service>
   e. Klicken Sie auf **OK**.
   f. Speichern Sie die Änderungen in der Masterkonfiguration.
4. Fügen Sie die JAR-Datei für die eXtreme-Scale-Clientlaufzeitumgebung, die Datei `wsogclient.jar` und die Konfigurations-JAR-Datei bzw. das Konfigurationsverzeichnis für den REST-Datenservice dem Klassenpfad des Anwendungsservers hinzu. Dieser Schritt ist nicht erforderlich, wenn WebSphere eXtreme Scale mit der Installation von WebSphere Application Server integriert ist.
   a. Öffnen Sie die Administrationskonsole von WebSphere Application Server.
   b. Navigieren Sie zu **Umgebung** > **Gemeinsam genutzte Bibliotheken**.
   c. Klicken Sie auf **Neu**.
   d. Fügen Sie den Feldern die folgenden Einträge hinzu:
      - Name: `extremescale_client_v71`
      - Klassenpfad: `WXS-Ausgangsverzeichnis/lib/wsogclient.jar`

      Hinweis: Fügen Sie jeden Pfad in einer gesonderten Zeile hinzu.
   e. Klicken Sie auf **OK**.
   f. Speichern Sie die Änderungen in der Masterkonfiguration.
5. Installieren Sie die EAR-Datei des REST-Datenservice `wxsrestservice.ear` über die Administrationskonsole in WebSphere Application Server:

a. Öffnen Sie die Administrationskonsole von WebSphere Application Server.
b. Klicken Sie auf **Anwendungen** > **Neue Anwendung**.
c. Navigieren Sie zur Datei `/lib/wxsrestservice.ear` im Dateisystem, wählen Sie sie aus, und klicken Sie auf **Weiter**.
   - Wenn Sie WebSphere Application Server Version 7.0 verwenden, klicken Sie auf "Weiter".
   - Wenn Sie WebSphere Application Server Version 6.1 verwenden, geben Sie einen Kontextstammwert mit dem Namen `/wxsrestservice` ein, und fahren Sie mit dem nächsten Schritt fort.
d. Wählen Sie die Option für detaillierte Installation aus, und klicken Sie auf **Weiter**.
e. Klicken Sie in der Anzeige mit Anwendungssicherheitswarnungen auf **Weiter**.
f. Wählen Sie die Standardinstallationsoptionen aus, und klicken Sie auf **Weiter**.
g. Wählen Sie einen Server aus, dem Sie die Anwendung zuordnen möchten, und klicken Sie auf **Weiter**.
h. Verwenden Sie auf der Seite für das erneute Laden von JSP-Dateien die Standardeinstellungen, und klicken Sie auf **Weiter**.
i. Ordnen Sie auf der Seite "Gemeinsam genutzte Bibliotheken" das Modul `wxsrestservice.war` den definierten gemeinsam benutzten Bibliotheken zu:
   - `extremescale_rest_configuration`
   - `extremescale_client_v71`

   Tipp: Diese gemeinsam genutzte Bibliothek ist nur erforderlich, wenn WebSphere eXtreme Scale nicht mit WebSphere Application Server integriert ist.
j. Verwenden Sie auf der Seite für die Zuordnung von Beziehungen zu gemeinsam genutzten Bibliotheken die Standardeinstellungen, und klicken Sie auf **Weiter**.
k. Verwenden Sie auf der Seite für die Zuordnung virtueller Hosts die Standardeinstellungen, und klicken Sie auf **Weiter**.
l. Setzen Sie auf der Seite für die Zuordnung der Kontextstammelemente das Kontextstammelement auf "wxsrestservice".
m. Klicken Sie in der Anzeige "Zusammenfassung" auf **Fertig stellen**, um die Installation durchzuführen.
n. Speichern Sie die Änderungen in der Masterkonfiguration.

6. Starten Sie die REST-Datenserviceanwendung `wxsrestservice`.
   a. Wechseln Sie in der Administrationskonsole zu der Anwendung.
      - WebSphere Application Server Version 7.0: Klicken Sie in der Administrationskonsole auf **Anwendungen** > **Anwendungstypen** > **WebSphere-Anwendungen**.
      - WebSphere Application Server Version 6.1: Klicken Sie in der Administrationskonsole auf **Anwendungen** > **Enterprise-Anwendungen**.
   b. Wählen Sie das Kontrollkästchen neben der Anwendung `wxsrestservice` aus, und klicken Sie auf **Starten**.
   c. Sehen Sie sich die Datei `SystemOut.log` für das Anwendungsserverprofil an. Wenn der REST-Datenservice erfolgreich gestartet wurde, wird die folgende Nachricht in der Datei `SystemOut.log` für das Serverprofil angezeigt:

      ```
      CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde
      gestartet.
      ```

7. Vergewissern Sie sich, dass der REST-Datenservice funktioniert: Die Portnummer finden Sie, in dem Sie in der Datei `SystemOut.log` im Verzeichnis "logs" des Anwendungsserverprofils nach dem ersten Port suchen, der für die Nachrichten-ID SRVE0250I angezeigt wird. Der Standardport ist 9080.

   Beispiel: `http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/` Das AtomPub-Servicedokument wird angezeigt.

   Beispiel: `http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata`. Das EDMX-Dokument (Entity Model Data Extensions) wird angezeigt.
8. Zum Stoppen der Datengridprozesse verwenden Sie die Tastenkombination `STRG+C` im entsprechenden Befehlsfenster.

REST-Datenservices mit WebSphere eXtreme Scale integrated in WebSphere Application Server 7.0 starten:

In diesem Abschnitt wird beschrieben, wie der REST-Datenservice von eXtreme Scale mit dem Produkt WebSphere Application Server Version 7.0, das mit WebSphere eXtreme Scale integriert und erweitert wurde, konfiguriert und gestartet wird.

Vorbereitende Schritte

Vergewissern Sie sich, dass das eigenständige eXtreme-Scale-Beispieldatengrid gestartet ist. Einzelheiten zum Starten des Datengrids finden Sie unter „REST-Datenservice aktivieren” auf Seite 366.

Informationen zu diesem Vorgang

Führen Sie zum Einstieg in die Verwendung des REST-Datenservice von WebSphere eXtreme Scale mit WebSphere Application Server die folgenden Schritte aus:

Vorgehensweise

1. Fügen Sie die Konfigurations-JAR-Datei des Beispiels für den REST-Datenservice von WebSphere eXtreme Scale dem Klassenpfad hinzu:
   a. Öffnen Sie die WebSphere-Administrationskonsole.
   b. Navigieren Sie zu "Umgebung -> Gemeinsam genutzte Bibliotheken".
   c. Klicken Sie auf "Neu".
   d. Fügen Sie den entsprechenden Feldern die folgenden Einträge hinzu:
      1) Name: extremescale_gettingstarted_config
      2) Klassenpfad
         - `Ausgangsverzeichnis_des_REST-Service/gettingstarted/restclient/bin`
         - `Ausgangsverzeichnis_des_REST-Service//gettingstarted/common/bin`

         Hinweis: Jeder Pfad muss in einer gesonderten Zeile angegeben werden.
   e. Klicken Sie auf **OK**.
   f. Speichern Sie die Änderungen in der Masterkonfiguration.
2. Installieren Sie die EAR-Datei des REST-Datenservice "wxsrestservice.ear" über die WebSphere-Administrationskonsole in WebSphere Application Server:
   a. Öffnen Sie die WebSphere-Administrationskonsole.

b. Navigieren Sie zu "Anwendungen -> Neue Anwendung".
c. Navigieren Sie zur Datei `Ausgangsverzeichnis_des_REST-Service/lib/wxsrestservice.ear` im Dateisystem. Wählen Sie die Datei aus, und klicken Sie auf **Weiter**.
d. Wählen Sie die detaillierten Installationsoptionen aus, und klicken Sie auf **Weiter**.
e. Klicken Sie in der Anzeige mit Anwendungssicherheitswarnungen auf **Weiter**.
f. Wählen Sie die Standardinstallationsoptionen aus, und klicken Sie auf **Weiter**.
g. Wählen Sie einen Server aus, dem Sie das Modul "wxsrestservice.war" zuordnen möchten, und klicken Sie auf **Weiter**.
h. Verwenden Sie auf der Seite für das erneute Laden von JSP-Dateien die Standardeinstellungen, und klicken Sie auf **Weiter**.
i. Ordnen Sie auf der Seite "Gemeinsam genutzte Bibliotheken" das Modul "wxsrestservice.war" den folgenden gemeinsam genutzten Bibliotheken zu, die in Schritt 1 definiert wurden: `extremescale_ gettingstarted _config`.
j. Verwenden Sie auf der Seite für die Zuordnung von Beziehungen zu gemeinsam genutzten Bibliotheken die Standardeinstellungen, und klicken Sie auf **Weiter**.
k. Verwenden Sie auf der Seite für die Zuordnung virtueller Hosts die Standardeinstellungen, und klicken Sie auf **Weiter**.
l. Setzen Sie auf der Seite für die Zuordnung der Kontextstammelemente das Kontextstammelement auf "wxsrestservice".
m. Klicken Sie in der Anzeige "Zusammenfassung" auf **Fertig stellen**, um die Installation durchzuführen.
n. Speichern Sie die Änderungen in der Masterkonfiguration.

3. Wenn das eXtreme-Scale-Datengrid mit aktivierter eXtreme-Scale-Sicherheit gestartet wurde, setzen Sie die folgende Eigenschaft in der Datei `Ausgangsverzeichnis_des_REST-Service/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=Ausgangsverzeichnis_des_REST-Service/gettingstarted/security/security.ogclient.properties
```

4. Starten Sie den Anwendungsserver und die REST-Datenserviceanwendung "wxsrestservice" von eXtreme Scale.

   Suchen Sie nach dem Start der Anwendung in der Protokolldatei "SystemOut.log" des Anwendungsservers die folgende Nachricht: `CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde gestartet.`

5. Vergewissern Sie sich, dass der REST-Datenservice funktioniert:
   a. Öffnen Sie einen Browser, und navigieren Sie zur folgenden Adresse:

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid

      Das Servicedokument für das Grid NorthwindGrid wird angezeigt.
   b. Navigieren Sie zur folgenden Adresse:

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Das EDMX-Dokument (Entity Model Data Extensions) wird angezeigt.

6. Zum Stoppen der Datengridprozesse verwenden Sie die Tastenkombination STRG+C im entsprechenden Befehlsfenster.

REST-Datenservice in WebSphere Application Server Community Edition implementieren

Sie können den REST-Datenservice von eXtreme Scale in WebSphere Application Server Community Edition Version 2.1.1.3 oder höher konfigurieren.

Vorbereitende Schritte

- Eine JRE oder ein JDK von IBM (empfohlen) oder Sun der Version 5 oder höher ist installiert, und eine Umgebungsvariable JAVA_HOME ist definiert.
- Laden Sie WebSphere Application Server Community Edition Version 2.1.1.3 oder höher herunter, und installieren Sie das Produkt im `WASCE-Stammverzeichnis`, z. B. `/opt/IBM/wasce`. Informationen zu Version 2.1.1 bzw. anderen Versionen finden Sie in den Installationsanweisungen.

Vorgehensweise

1. Datengrid konfigurieren und starten.
   a. Einzelheiten zum Konfigurieren eines Datengrids von eXtreme Scale für den REST-Datenservice finden Sie unter „Eigenständiges Datengrid für REST-Datenservices starten" auf Seite 372.
   b. Vergewissern Sie sich, dass ein eXtreme-Scale-Client eine Verbindung zum Grid herstellen und auf die Entitäten im Datengrid zugreifen kann. Ein Beispiel finden Sie unter „Lernprogramm: Einführung in WebSphere eXtreme Scale" auf Seite 1.
2. Erstellen Sie die Konfigurations-JAR-Datei oder das Konfigurationsverzeichnis für den REST-Datenservice von eXtreme Scale. Einzelheiten finden Sie in den Informationen zum Packen und Implementieren im Abschnitt „REST-Datenservice installieren" auf Seite 205.
3. Starten Sie den Server von WebSphere Application Server Community Edition:
   a. Führen Sie den folgenden Befehl aus, um den Server ohne aktivierte Java-SE-Sicherheit zu starten:

      UNIX Linux `WASCE-Stammverzeichnis/bin/startup.sh`

      Windows `WASCE-Stammverzeichnis/bin/startup.bat`

   b. Führen Sie die folgenden Schritte aus, um den Server mit aktivierter Java-SE-Sicherheit zu starten: UNIX Linux
      1) Öffnen Sie eine Befehlszeile oder ein Terminalfenster, und führen Sie den folgenden Kopierbefehl aus (oder kopieren Sie den Inhalt der angegebenen Richtliniendatei in Ihre vorhandene Richtlinie): `cp Ausgangsverzeichnis_des_REST-Service/gettingstarted/wasce/geronimo.policy WASCE-Stammverzeichnis/bin`).
      2) Bearbeiten Sie die Datei `wasce_root/bin/setenv.sh`.
      3) Fügen Sie hinter der Zeile mit `"WASCE_JAVA_HOME="` Folgendes hinzu:
      ```
      export JAVA_OPTS="-Djava.security.manager
      -Djava.security.policy=geronimo.policy".
      ```

      Windows
      1) Öffnen Sie ein Befehlszeilenfenster, und führen Sie den folgenden Kopierbefehl aus, oder kopieren Sie den Inhalt der angegebenen Richtliniendatei in Ihre vorhandene Richtlinie:
      ```
      copy Ausgangsverzeichnis_des_REST-Service\gettingstarted\wasce\
      geronimo.policy\bin
      ```
      2) Bearbeiten Sie die Datei `wasce_root\bin\setenv.bat`.

3) Fügen Sie hinter der Zeile mit "set WASCE_JAVA_HOME=" Folgendes hinzu:

```
set JAVA_OPTS="-Djava.security.manager
-Djava.security.policy=geronimo.policy"
```

4. Fügen Sie die JAR-Datei für die ObjectGrid-Clientlaufzeitumgebung dem Repository von WebSphere Application Server Community Edition hinzu:
   a. Öffnen Sie die Administrationskonsole von WebSphere Application Server Community Edition, und melden Sie sich an. Der Standard-URL ist `http://localhost:8080/console`, die Standardbenutzer-ID ist `system`, und das Kennwort ist `manager`.
   b. Klicken Sie im Ordner **Services** auf der linken Seite des Konsolfensters auf den Link **Repository**.
   c. Tragen Sie im Abschnitt **Archiv dem Repository hinzufügen** Folgendes in die Eingabetextfelder ein:

Tabelle 24. Archiv dem Repository hinzufügen

Textfeld	Wert
Datei	`WXS-Ausgangsverzeichnis/lib/ogclient.jar`
Gruppe	com.ibm.websphere.xs
Artefakt	ogclient
Version	7.1
Typ	JAR

   d. Klicken Sie auf die Schaltfläche "Installieren".

   Suchen Sie im folgenden technischen Hinweis nach Einzelheiten zu den verschiedenen Methoden für die Konfiguration von Klassen- und Bibliotheksabhängigkeiten: Specifying external dependencies to applications running on WebSphere Application Server Community Edition.
5. Implementieren Sie das Modul des REST-Datenservice, die Datei `wxsrestservice.war`, im Server von WebSphere Application Server Community Edition, und starten Sie es.
   a. Kopieren und bearbeiten Sie die XML-Datei mit dem Beispielimplementierungsplan `Ausgangsverzeichnis_des_REST-Service/gettingstarted/wasce/geronimo-web.xml`. Fügen Sie die Pfadabhängigkeiten zur Konfigurations-JAR bzw. zum Konfigurationsverzeichnis Ihres REST-Datenservice hinzu. Sehen Sie sich das Beispiel zum Definieren des Klassenpfads an, um Ihre Datei `wxsRestService.properties` sowie andere Konfigurationsdateien und Metadatenklassen hinzuzufügen.
   b. Öffnen Sie die Administrationskonsole von WebSphere Application Server Community Edition, und melden Sie sich an.

      Tipp: Der Standard-URL ist `http://localhost:8080/console`. Die Standardbenutzer-ID ist `system`, und das Kennwort ist `manager`.
   c. Klicken Sie auf den Link **Neu Implementieren** auf der linken Seite des Konsolfensters.
   d. Geben Sie auf der Seite **Neue Anwendungen installieren** die folgenden Werte in die Textfelder ein:

Tabelle 25. Neue Anwendungen installieren

Textfeld	Wert
Archiv	`Ausgangsverzeichnis_des_REST-Service/lib/wxsrestservice.war`

Tabelle 25. Neue Anwendungen installieren (Forts.)

Textfeld	Wert
Plan	`Ausgangsverzeichnis_des_REST-Service/gettingstarted/wasce/geronimo-web.xml`

Tipp: Verwenden Sie den Pfad zur Datei `geronimo-web.xml`, die Sie in Schritt 3 kopiert und bearbeitet haben.

e. Klicken Sie auf die Schaltfläche "Installieren". Auf der Konsolseite sollte daraufhin angezeigt werden, dass die Anwendung erfolgreich installiert und gestartet werden.

f. Untersuchen Sie das Systemausgabeprotokoll oder die Konsole von WebSphere Application Server Community Edition, um sicherzustellen, dass der REST-Datenservice erfolgreich gestartet wurde. Die folgende Nachricht muss vorhanden sein:

```
CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde gestartet.
```

6. Starten Sie den Server von WebSphere Application Server Community Edition mit dem folgenden Befehl:
   - UNIX Linux `WASCE-Stammverzeichnis/bin/startup.sh`
   - Windows `WASCE-Stammverzeichnis/bin/startup.bat`
7. Installieren Sie den REST-Datenservice von eXtreme Scale und das bereitgestellte Beispiel im Server von WebSphere Application Server Community Edition:
   a. Fügen Sie die JAR-Datei für die ObjectGrid-Clientlaufzeitumgebung dem Repository von WebSphere Application Server Community Edition hinzu:
      1) Öffnen Sie die Administrationskonsole von WebSphere Application Server Community Edition, und melden Sie sich an. Der Standard-URL ist `http://localhost:8080/console`. Die Standardbenutzer-ID ist `system`, und das Kennwort ist `manager`.
      2) Klicken Sie im Ordner "Services" auf der linken Seite des Konsolfensters auf den Link "**Repository**".
      3) Tragen Sie im Abschnitt **Archiv dem Repository hinzufügen** Folgendes in die Eingabetextfelder ein:

Tabelle 26. Archiv dem Repository hinzufügen

Textfeld	Wert
Datei	WXS-Ausgangsverzeichnis/lib/ogclient.jar
Gruppe	com.ibm.websphere.xs
Artefakt	ogclient
Version	7.1
Typ	JAR

      4) Klicken Sie auf die Schaltfläche "Installieren".

      Tipp: Suchen Sie im folgenden technischen Hinweis nach Einzelheiten zu den verschiedenen Methoden für die Konfiguration von Klassen- und Bibliotheksabhängigkeiten: Specifying external dependencies to applications running on WebSphere Application Server Community Edition.

   b. Implementieren Sie das REST-Datenservicemodul `wxsrestservice.war` im Server von WebSphere Application Server Community Edition.

1) Bearbeiten Sie die XML-Beispielimplementierungsdatei `Ausgangsverzeichnis_des_REST-Service/gettingstarted/wasce/geronimo-web.xml`, und fügen Sie den Klassenpfadverzeichnissen für das Einführungsmuster Pfadabhängigkeiten hinzu.
   - Ändern Sie die classesDirs-Pfade für die beiden GBeans des Getting-Started-Clients:

   Der classesDirs-Pfad für die GBean "GettingStarted_Client_SharedLib" muss auf `Ausgangsverzeichnis_des_REST-Service/Ggettingstarted/restclient/bin` gesetzt werden.

   Der classesDirs-Pfad für die GBean "GettingStarted_Common_Shared-Lib" muss auf `Ausgangsverzeichnis_des_REST-Service/gettingstarted/common/bin` gesetzt werden.
2) Öffnen Sie die Administrationskonsole von WebSphere Application Server Community Edition, und melden Sie sich an.
3) Klicken Sie auf den Link **Neu Implementieren** auf der linken Seite des Konsolfensters.
4) Geben Sie auf der Seite **Neue Anwendungen installieren** die folgenden Werte in die Textfelder ein:

Tabelle 27. Neue Anwendungen installieren

Textfeld	Wert
Archiv	*Ausgangsverzeichnis_des_REST-Service*/lib/wxsrestservice.war
Plan	*Ausgangsverzeichnis_des_REST-Service*/gettingstarted/wasce/geronimo-web.xml

5) Klicken Sie auf die Schaltfläche **Installieren**.

   Auf der Konsolseite sollte daraufhin angezeigt werden, dass die Anwendung erfolgreich installiert und gestartet werden.
6) Überprüfen Sie anhand des Systemausgabeprotokolls von WebSphere Application Server Community Edition, ob der REST-Datenservice erfolgreich gestartet wurde, indem Sie nach der folgenden Nachricht suchen:

   `CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde gestartet.`

8. Vergewissern Sie sich, dass der REST-Datenservice funktioniert:

   Öffnen Sie einen Webbrowser, und navigieren Sie zum folgenden URL:
   `http://<Host>:<Port>/<Kontextstammelement >/restservice/<Gridname>`

   Der Standardport für WebSphere Application Server Community Edition ist 8080 und wird mit der Eigenschaft "HTTPPort" in der Datei `/var/config/config-substitutions.properties` definiert.

   Beispiel:http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/

Ergebnisse

Das Dokument des AtomPub-Service wird angezeigt.

REST-Datenservice in WebSphere Application Server Community Edition starten:

In diesem Abschnitt wird beschrieben, wie der REST-Datenservice von eXtreme Scale mit WebSphere Application Server Community Edition konfiguriert und gestartet wird.

Vorbereitende Schritte

Vergewissern Sie sich, dass das Beispieldatengrid gestartet ist. Einzelheiten zum Starten des Grids finden Sie unter „REST-Datenservice aktivieren” auf Seite 366.

Vorgehensweise

1. Laden Sie WebSphere Application Server Community Edition Version 2.1.1.3 oder höher herunter, und installieren Sie das Produkt im `WASCE-Stammverzeichnis`, z. B. `/opt/IBM/wasce`.
2. Starten Sie den Server von WebSphere Application Server Community Edition mit dem folgenden Befehl:
   - Linux UNIX `WASCE-Stammverzeichnis/bin/startup.sh`
   - Windows `WASCE-Stammverzeichnis/bin/startup.bat`
3. Wenn das eXtreme-Scale-Grid mit aktivierter eXtreme-Scale-Sicherheit gestartet wurde, setzen Sie die folgenden Eigenschaften in der Datei `Ausgangsverzeichnis_des_REST-Service/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=Ausgangsverzeichnis_des_REST-Service/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Installieren Sie den REST-Datenservice von eXtreme Scale und das bereitgestellte Beispiel im Server von WebSphere Application Server Community Edition:
   a. Fügen Sie die JAR-Datei für die ObjectGrid-Clientlaufzeitumgebung dem Repository von WebSphere Application Server Community Edition hinzu:
      1) Öffnen Sie die Administrationskonsole von WebSphere Application Server Community Edition, und melden Sie sich an.

         Tipp: Der Standard-URL ist `http://localhost:8080/console`. Die Standardbenutzer-ID ist `system`, und das Kennwort ist `manager`.
      2) Klicken Sie im Ordner "Services" auf **Repository**.
      3) Tragen Sie im Abschnitt **Archiv dem Repository hinzufügen** Folgendes in die Eingabetextfelder ein:

Tabelle 28. Archivierung im Repository

Textfeld	Wert
Datei	WXS-Ausgangsverzeichnis/lib/ogclient.jar
Gruppe	com.ibm.websphere.xs
Artefakt	ogclient
Version	7.0
Typ	jar

      4) Klicken Sie auf die Schaltfläche "Installieren".

         Tipp: Suchen Sie im folgenden technischen Hinweis nach Einzelheiten zu den verschiedenen Methoden von Konfigurationsklassen- und Bibliotheksabhängigkeiten: Specifying external dependencies to applications running on WebSphere Application Server Community Edition.
   b. Implementieren Sie das Modul des REST-Datenservice, das in der Datei `wxsrestservice.war` enthalten ist, im Server von WebSphere Application Server Community Edition.
      1) Bearbeiten Sie die XML-Beispielimplementierungsdatei `Ausgangsverzeichnis_des_REST-Service/gettingstarted/wasce/`

geronimo-web.xml, und fügen Sie den Klassenpfadverzeichnissen für das Einführungsmuster Pfadabhängigkeiten hinzu.

Ändern Sie die classesDirs-Pfade für die beiden GBeans des GettingStarted-Clients:

- Der classesDirs-Pfad für die GBean "GettingStarted_Client_SharedLib" muss auf Ausgangsverzeichnis_des_REST-Service/Ggettingstarted/restclient/bin gesetzt werden.
- Der classesDirs-Pfad für die GBean "GettingStarted_Common_SharedLib" muss auf Ausgangsverzeichnis_des_REST-Service/gettingstarted/common/bin gesetzt werden.

2) Öffnen Sie die Administrationskonsole von WebSphere Application Server Community Edition, und melden Sie sich an.

   Tipp: Der Standard-URL ist http://localhost:8080/console. Die Standardbenutzer-ID ist system, und das Kennwort ist manager.

3) Klicken Sie auf **Neu implementieren**.
4) Geben Sie auf der Seite **Neue Anwendungen installieren** die folgenden Werte in die Textfelder ein:

Tabelle 29. Installationswerte

Textfeld	Wert
Archiv	Ausgangsverzeichnis_des_REST-Service/lib/wxsrestservice.war
Plan	Ausgangsverzeichnis_des_REST-Service/gettingstarted/wasce/geronimo-web.xml

5) Klicken Sie auf die Schaltfläche "Installieren".

   Auf der Konsolseite sollte angezeigt werden, dass die Anwendung erfolgreich installiert und gestartet werden.

6) Überprüfen Sie anhand des Systemausgabeprotokolls von WebSphere Application Server Community Edition oder der Konsole, ob der REST-Datenservice erfolgreich gestartet wurde, indem Sie nach der folgenden Nachricht suchen:

   ```
   CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale
   wurde gestartet.
   ```

5. Vergewissern Sie sich, dass der REST-Datenservice funktioniert:
   a. Öffnen Sie den folgenden Link in einem Browserfenster: http://localhost:8080/wxsrestservice/restservice/NorthwindGrid. Das Servicedokument für das Grid "NorthwindGrid" wird angezeigt.
   b. Öffnen Sie den folgenden Link in einem Browserfenster: http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata. Das EDMX-Dokument (Entity Model Data Extensions) wird angezeigt.
6. Zum Stoppen der Gridprozesse verwenden Sie die Tastenkombination STRG+C im entsprechenden Befehlsfenster.
7. Verwenden Sie zum Stoppen von WebSphere Application Server Community Edition den folgenden Befehl:
   - UNIX Linux `WASCE-Stammverzeichnis/bin/shutdown.sh`
   - Windows `WASCE-Stammverzeichnis\bin\shutdown.bat`

   Tipp: Die Standardbenutzer-ID ist system, und das Kennwort ist manager. Wenn Sie einen angepassten Port verwenden, verwenden Sie die Option -port.

REST-Datenservice in Apache Tomcat implementieren

In diesem Artikel wird beschrieben, wie Sie den REST-Datenservice von WebSphere eXtreme Scale in Apache Tomcat Version 5.5 oder höher konfigurieren.

Informationen zu diesem Vorgang

- Eine JRE oder ein JDK von IBM oder Sun der Version 5 oder höher ist installiert, und eine Umgebungsvariable JAVA_HOME ist definiert.
- Apache Tomcat Version 5.5 oder höher ist installiert. Einzelheiten zur installation von Tomcat finden Sie auf der Webseite Apache Tomcat.
- Sie haben eine eigenständige Installation von WebSphere eXtreme Scale.

Vorgehensweise

1. Wenn Sie eine Sun JRE oder ein Sun JDK verwenden, installieren Sie IBM ORB in Tomcat:
   a. Tomcat Version 5.5:

      Kopieren Sie alle JAR-Dateien aus dem

      Verzeichnis *WXS-Ausgangsverzeichnis*/lib/endorsed

      in das

      Verzeichnis *Tomcat-Stammverzeichnis*/common/endorsed.
   b. Tomcat Version 6.0:

      Erstellen Sie ein Verzeichnis "endorsed":

      UNIX Linux `mkdir Tomcat-Stammverzeichnis/endorsed`

      Windows `md Tomcat-Stammverzeichnis/endorsed`

      Kopieren Sie alle JAR-Dateien aus dem

      Verzeichnis wxs_home/lib/endorsed

      in das

      Verzeichnis tomcat_root/common/endorsed.
2. Konfigurieren und starten Sie das Datengrid.
   a. Einzelheiten zum Konfigurieren eines Datengrids für den REST-Datenservice finden Sie in Kapitel 6, „Konfigurieren”, auf Seite 229.
   b. Vergewissern Sie sich, dass ein eXtreme-Scale-Client eine Verbindung zum Grid herstellen und auf die Entitäten im Grid zugreifen kann. Ein Beispiel finden Sie unter „REST-Datenservices konfigurieren” auf Seite 365.
3. Erstellen Sie die Konfigurations-JAR-Datei oder das Konfigurationsverzeichnis für den REST-Datenservice von eXtreme Scale. Einzelheiten finden Sie in den Informationen zum Packen und Implementieren unter „REST-Datenservice installieren” auf Seite 205.
4. Implementieren Sie das REST-Datenservicemodul "wxsrestservice.war" im Tomcat-Server.

   Kopieren Sie die Datei wxsrestservice.war von

   Ausgangsverzeichnis_des_REST-Service/lib

   nach

   Tomcat-Stammverzeichnis/webapps
5. Fügen Sie die JAR-Datei für die ObjectGrid-Clientlaufzeitumgebung und die Anwendungs-JAR-Datei dem gemeinsam genutzten Klassenpfad in Tomcat hinzu:
   a. Bearbeiten Sie die Datei *Tomcat-Stammverzeichnis*/conf/catalina.properties.

b. Fügen Sie die folgenden Pfadnamen am Ende der Eigenschaft "shared.loader" hinzu, indem Sie die einzelnen Pfadnamen durch Kommas trennen:
   - *WXS-Ausgangsverzeichnis*/lib/ogclient.jar
   - *Ausgangsverzeichnis_des_REST-Service*/gettingstarted/restclient/bin
   - *Ausgangsverzeichnis_des_REST-Service*/gettingstarted/common/bin

6. Wenn Sie die Java-2-Sicherheit verwenden, fügen Sie der Tomcat-Richtliniendatei Sicherheitsberechtigungen hinzu:
   - Bei der Verwendung von Tomcat Version 5.5:

     Führen Sie den Inhalt der Beispieldatei catalina policy der Version 5.5 in *Ausgangsverzeichnis_des_REST-Service*/gettingstarted/tomcat/catalina-5_5.policy mit der Datei *Tomcat-Stammverzeichnis*/conf/catalina.policy.
   - Bei der Verwendung von Tomcat Version 6.0:

     Führen Sie den Inhalt der Beispieldatei catalina policy der Version 6.0 in *Ausgangsverzeichnis_des_REST-Service*/gettingstarted/tomcat/catalina-6_0.policy mit der Datei *Tomcat-Stammverzeichnis*/conf/catalina.policy
7. Starten Sie den Tomcat-Server:
   - **Bei der Verwendung von Tomcat 5.5 unter UNIX oder Windows oder der Verwendung von Tomcat 6.0 durch Verteilung über eine ZIP-Datei**
     a. cd *Tomcat-Stammverzeichnis*/bin
     b. Starten Sie den Server:
        – Ohne aktivierte Java-2-Sicherheit:

          UNIX Linux ./catalina.sh run

          Windows catalina.bat run
        – Mit aktivierter Java-2-Sicherheit:

          UNIX Linux ./catalina.sh run -security

          Windows catalina.bat run -security
     c. Die Apache-Tomcat-Protokolle werden in der Konsole angezeigt. Wenn der REST-Datenservice erfolgreich gestartet wurde, wird die folgende Nachricht in der Administrationskonsole angezeigt:

        CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde gestartet.
   - **Bei der Verwendung von Tomcat 6.0 unter Windows über Verteilung durch das Windows-Installationsprogramm:**
     a. cd /bin
     b. Starten Sie das Konfigurationstool von Apache Tomcat 6:

        tomcat6w.exe
     c. Java-2-Sicherheit aktivieren (optional):

        Fügen Sie den Java-Optionen auf der Registerkarte "Java" im Eigenschaftsfenster von Apache Tomcat 6 die folgenden Einträge hinzu:

        -Djava.security.manager

        -Djava.security.policy=\conf\catalina.policy
     d. Klicken Sie im Eigenschaftsfenster von Apache Tomcat 6 auf die Startschaltfläche, um den Tomcat-Server zu starten.
     e. Sehen Sie sich die folgenden Protokolle an, um sich zu vergewissern, dass der Tomcat-Server erfolgreich gestartet wurde:
        – *Tomcat-Stammverzeichnis*/bin/catalina.log

Zeigt den Status der Tomcat-Server-Engine an.

– *Tomcat-Stammverzeichnis*/bin/stdout.log

Zeigt das Systemausgabeprotokoll an.

f. Wenn der REST-Datenservice erfolgreich gestartet wurde, wird die folgende Nachricht im Systemausgabeprotokoll angezeigt:

```
CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde gestartet.
```

8. Vergewissern Sie sich, dass der REST-Datenservice funktioniert.

Öffnen Sie einen Webbrowser, und navigieren Sie zum folgenden URL:

http://*Host:Port*/*Kontextstammelement*/restservice/*Gridname*

Der Standardport für Tomcat ist 8080 und wird in der Datei *Tomcat-Stammverzeichnis*/conf/server.xml im Element <Connector> definiert.

Beispiel:

```
http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
```

Ergebnisse

Das Dokument des AtomPub-Service wird angezeigt.

REST-Datenservices in Apache Tomcat starten:

In diesem Abschnitt wird beschrieben, wie der REST-Datenservice von eXtreme Scale mit Apache Tomcat Version 5.5 oder höher konfiguriert und gestartet wird.

Vorbereitende Schritte

Vergewissern Sie sich, dass das eXtreme-Scale-Beispieldatengrid gestartet ist. Einzelheiten zum Starten des Datengrids finden Sie unter „REST-Datenservice aktivieren” auf Seite 366.

Vorgehensweise

1. Laden Sie Apache Tomcat Version 5.5 oder höher in das Tomcat-Installationsverzeichnis herunter, und installieren Sie diese Version. Beispiel: /opt/tomcat
2. Installieren Sie den REST-Datenservice von eXtreme Scale und das bereitgestellte Beispiel wie folgt im Tomcat-Server:
   a. Wenn Sie eine Sun JRE oder ein Sun JDK verwenden, müssen Sie den IBM ORB in Tomcat installieren:
      - Für Tomcat Version 5.5

        Kopieren Sie alle JAR-Dateien von

        WXS-Ausgangsverzeichnis/lib/endorsed

        nach

        Tomcat-Stammverzeichnis/common/endorsed
      - Für Tomcat Version 6.0
        1) Erstellen Sie ein Verzeichnis "endorsed".
           – UNIX Linux `mkdir Tomcat-Stammverzeichnis/endorsed`
           – Windows `md Tomcat-Stammverzeichnis/endorsed`
        2) Kopieren Sie alle JAR-Dateien von

           WXS-Ausgangsverzeichnis/lib/endorsed

           nach

Tomcat-Stammverzeichnis/endorsed

b. Implementieren Sie das REST-Datenservicemodul "wxsrestservice.war" im Tomcat-Server.

   Kopieren Sie die Datei "wxsrestservice.war" von

   Ausgangsverzeichnis_des_REST-Service/lib

   nach

   Tomcat-Stammverzeichnis/webapps

c. Fügen Sie die JAR-Datei für die ObjectGrid-Clientlaufzeitumgebung und die Anwendungs-JAR-Datei dem gemeinsam genutzten Klassenpfad in Tomcat hinzu:

   1) Bearbeiten Sie die Datei Tomcat-Stammverzeichnis/conf/catalina.properties.

   2) Fügen Sie die folgenden Pfadnamen am Ende der Eigenschaft "shared.loader" in Form einer durch Kommas begrenzten Liste hinzu:

      - WXS-Ausgangsverzeichnis/lib/ogclient.jar
      - Ausgangsverzeichnis_des_REST-Service/gettingstarted/restclient/bin
      - Ausgangsverzeichnis_des_REST-Service/gettingstarted/common/bin

      Wichtig: Das Pfadtrennzeichen muss ein **Schrägstrich** sein.

3. Wenn das eXtreme-Scale-Datengrid mit aktivierter eXtreme-Scale-Sicherheit gestartet wurde, setzen Sie die folgenden Eigenschaften in der Datei Ausgangsverzeichnis_des_REST-Service/gettingstarted/restclient/bin/wxsRestService.properties.

```
ogClientPropertyFile=Ausgangsverzeichnis_des_REST-Service/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Starten Sie den Tomcat-Server mit dem REST-Datenservice:

   - Wenn Sie Tomcat 5.5 unter UNIX or Windows oder Tomcat 6.0 unter UNIX verwenden, gehen Sie wie folgt vor:

     a. cd Tomcat-Stammverzeichnis/bin

     b. Starten Sie den Server:

        – UNIX Linux ./catalina.sh run

        – Windows catalina.bat run

     c. In der Konsole werden daraufhin die Apache-Tomcat-Protokolle angezeigt. Wenn der REST-Datenservice erfolgreich gestartet wurde, wird die folgende Nachricht in der Administrationskonsole angezeigt:

        CWOBJ4000I: Der REST-Datenservice von WebSphere eXtreme Scale wurde gestartet.

   - Wenn Sie Tomcat 6.0 unter Windows verwenden, gehen Sie wie folgt vor:

     a. cd Tomcat-Stammverzeichnis/bin

     b. Starten Sie das Konfigurationstool von Apache Tomcat 6 mit dem folgenden Befehl: tomcat6w.exe

     c. Klicken Sie im Eigenschaftsfenster von Apache Tomcat 6 auf die Startschaltfläche, um den Tomcat-Server zu starten.

     d. Sehen Sie sich die folgenden Protokolle an, um sich zu vergewissern, dass der Tomcat-Server erfolgreich gestartet wurde:

        – Tomcat-Stammverzeichnis/bin/catalina.log

          Zeigt den Status der Tomcat-Server-Engine an.

        – Tomcat-Stammverzeichnis/bin/stdout.log

Zeigt das Systemausgabeprotokoll an.

e. Wenn der REST-Datenservice erfolgreich gestartet wurde, wird die folgende Nachricht im Systemausgabeprotokoll angezeigt: `CWOBJ4000I: Der REST-Service von WebSphere eXtreme Scale wurde gestartet.`

5. Vergewissern Sie sich, dass der REST-Datenservice funktioniert:
   a. Öffnen Sie einen Browser, und navigieren Sie zur folgenden Adresse:

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid

      Das Servicedokument für das Grid NorthwindGrid wird angezeigt.
   b. Navigieren Sie zur folgenden Adresse:

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Das EDMX-Dokument (Entity Model Data Extensions) wird angezeigt.
6. Zum Stoppen der Datengridprozesse verwenden Sie die Tastenkombination STRG+C im entsprechenden Befehlsfenster.
7. Zum Stoppen von Tomcat verwenden Sie die Tastenkombination STRG+C in dem Fenster, in dem Sie Tomcat gestartet haben.

Web-Browser für den Zugriff auf ATOM-Feeds des REST-Datenservice konfigurieren

Der REST-Datenservice von eXtreme Scale erstellt bei der Verwendung eines Webbrowsers standardmäßig ATOM-Feeds. Das ATOM-Feed-Format ist möglicherweise mit älteren Browsern nicht kompatibel, oder es kann so interpretiert werden, dass die Daten nicht als XML angezeigt werden können. Sie können Internet Explorer Version 8 und Firefox Version 3 für die Anzeige der ATOM-Feeds und der XML im Browser konfigurieren.

Informationen zu diesem Vorgang

Der REST-Datenservice von eXtreme Scale erstellt bei der Verwendung eines Webbrowsers standardmäßig ATOM-Feeds. Das ATOM-Feed-Format ist möglicherweise mit älteren Browsern nicht kompatibel, oder es kann so interpretiert werden, dass die Daten nicht als XML angezeigt werden können. Bei älteren Browsern werden Sie aufgefordert, die Dateien auf der Platte zu speichern. Verwenden Sie nach dem Download der Dateien den von Ihnen bevorzugten XML-Reader, um die Dateien anzuzeigen. Die generierte XML wird für die Anzeige nicht formatiert, und deshalb wird alles in einer einzigen Zeile ausgegeben. Die meisten Programme zum Lesen von XML, wie z. B. Eclipse, unterstützten eine Neuformatierung der XML in ein lesbares Format.

Bei modernen Browsern wie Microsoft Internet Explorer Version 8 und Firefox Version 3 können die ATOM-XML-Dateien nativ im Browser angezeigt werden. Die folgenden Abschnitte enthalten Details zur Konfiguration von Internet Explorer Version 8 und Firefox Version 3 für die Anzeige von ATOM-Feeds und XML im Browser.

Vorgehensweise

Internet Explorer Version 8 konfigurieren

- Verwenden Sie die folgenden Schritte, um Internet Explorer für das Lesen der vom REST-Datenservice generierten ATOM-Feeds zu aktivieren:
  1. Klicken Sie auf **Extras** > **Internetoptionen**-
  2. Wählen Sie das Register **Inhalte** aus.

3. Klicken Sie im Abschnitt **Feeds und Web Slices** auf die Schaltfläche **Einstellungen**.
4. Wählen Sie das Feld "Feedleseanzeige einschalten" ab.
5. Klicken Sie auf **OK**, um zum Browser zurückzukehren.
6. Starten Sie Internet Explorer erneut.

Firefox Version 3 konfigurieren

- Firefox zeigt Seiten mit dem Inhaltstyp "application/atom+xml" nicht automatisch an. Wenn eine solche Seite zum ersten Mal angezeigt wird, fordert Firefox Sie zum Speichern der Datei auf. Zum Anzeigen der Seite öffnen Sie die Datei in Firefox wie folgt:
  1. Wählen Sie im Dialogfenster für die Anwendungsauswahl das Optionsfeld "Öffnen mit" aus, und klicken Sie auf die Schaltfläche **Durchsuchen**.
  2. Navigieren Sie zum Firefox-Installationsverzeichnis. Beispiel: `C:\Program Files\Mozilla Firefox`
  3. Wählen Sie `firefox.exe` aus, und klicken Sie auf die Schaltfläche **OK**.
  4. Wählen Sie das Markierungsfeld "Für Dateien dieses Typs immer diese Aktion ausführen" aus.
  5. Klicken Sie auf die Schaltfläche **OK**.
  6. Anschließend zeigt Firefox die ATOM-XML-Seite in einem neuen Browserfenster oder auf einer neuen Registerkarte an.
- Firefox gibt ATOM-Feeds automatisch in einem lesbaren Format wieder. Die Feeds, die vom REST-Datenservice erstellt werden, enthalten jedoch XML. Firefox kann die XML nur anzeigen, wenn Sie den Feed-Renderer inaktivieren. Anders als in Internet Explorer muss in Firefox das Plug-in für die Wiedergabe von ATOM-Feeds explizit geändert werden. Zum Konfigurieren von Firefox für das Lesen von ATOM-Feeds als XML-Dateien gehen Sie wie folgt vor:
  1. Öffnen Sie die folgende Datei in einem Texteditor: `<Firefox-Installationsstammverzeichnis>\components\FeedConverter.js`. In dem Pfad steht `<Firefox-Installationsstammverzeichnis>` für das Stammverzeichnis, in dem Firefox installiert ist.

     Bei Windows-Betriebssystemen ist das Standardverzeichnis `C:\Program Files\Mozilla Firefox`.
  2. Suchen Sie das Snippet, das wie folgt aussieht:

     ```
     // show the feed page if it wasn't sniffed and we have a document,
     // or we have a document, title, and link or id
     if (result.doc && (!this._sniffed ||
         (result.doc.title && (result.doc.link || result.doc.id)))) {
     ```
  3. Setzen Sie die beiden Zeilen, die mit `if` und `result` beginnen, auf Kommentar, indem Sie `//` (zwei Schrägstriche) an den Anfang der Zeilen setzen.
  4. Fügen Sie dem Snippet die folgende Anweisung an: `if(0) {`.
  5. Der Text sollte anschließend wie folgt aussehen:

     ```
     // show the feed page if it wasn't sniffed and we have a document,
     // or we have a document, title, and link or id
     //if (result.doc && (!this._sniffed ||
     //    (result.doc.title && (result.doc.link || result.doc.id)))) {
     if(0) {
     ```
  6. Speichern Sie die Datei.
  7. Starten Sie Firefox erneut.
  8. Jetzt kann Firefox automatisch alle Feeds im Browser anzeigen.
- Testen Sie Ihr Setup, indem Sie verschiedene URLs ausprobieren.

Beispiel

In diesem Abschnitt werden einige Beispiel-URLs beschrieben, die Sie verwenden können, um die Daten anzuzeigen, die von dem Einführungsmuster hinzugefügt wurden, das mit dem REST-Datenservice bereitgestellt wird. Fügen Sie vor der Verwendung der folgenden URLs mit dem Java-Beispielclient oder dem Beispielclient für Visual Studio WCF Data Services den Standarddatensatz zum eXtreme-Scale-Beispieldatengrid hinzu.

In den folgenden Beispielen wird der Port 8080 angenommen, der aber variieren kann. Lesen Sie den Abschnitt zum Konfigurieren des REST-Datenservice in verschiedenen Anwendungsservern.

- Einen einzigen Kunden (customer) mit der ID "ACME" anzeigen:
  `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')`
- Alle Bestellungen (order) für den Kunden "ACME" anzeigen:
  `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')/orders`
- Kunden "ACME" und die zugehörigen Bestellungen anzeigen:
  `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')?$expand=orders`
- Bestellung 1000 für den Kunden "ACME" anzeigen:
  `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')`
- Bestellung 1000 für den Kunden "ACME" und den zugeordnten Kunden anzeigen:
  `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')?$expand=customer`
- Bestellung 1000 für den Kunden "ACME" sowie den zughörigen Kunden und Bestelldetails (OrderDetails) anzeigen:
  `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')?$expand=customer,orderDetai`
- Alle Bestellungen für den Kunden "ACME" für den Monat Oktober 2009 (GMT) anzeigen:
  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer(customerId='ACME')/orders?
  ge datetime'2009-10-01T00:00:00'
  and orderDate lt datetime'2009-11-01T00:00:00'
  ```
- Die ersten drei Bestellungen und die zugehörigen Bestelldetails für den Kunden "ACME" für den Monat Oktober 2009 (GMT) anzeigen:
  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer(customerId='ACME')/orders?
  ge datetime'2009-10-01T00:00:00'
  and orderDate lt datetime'2009-11-01T00:00:00'
  &$orderby=orderDate&$top=3&$expand=orderDetails
  ```

Java-Client mit REST-Datenservices verwenden

Die Java-Clientanwendung verwendet die eXtreme-Scale-API "EntityManager", um Daten in das Grid einzufügen.

Informationen zu diesem Vorgang

In den vorherigen Abschnitten wurde beschrieben, wie ein eXtreme-Scale-Datengrid erstellt und mit dem REST-Datenservice von eXtreme Scale konfiguriert und gestartet wird. Die Java-Clientanwendung verwendet die eXtreme-Scale-API "EntityManager", um Daten in das Grid einzufügen. Die demonstriert nicht, wie die REST-Schnittstellen verwendet werden. Dieser Client soll veranschaulichen, wie die API "EntityManager" verwendet wird, um mit dem eXtreme-Scale-Datengrid zu interagieren und somit Daten im Grid zu ändern. Zum Anzeigen der Daten im Grid über den REST-Datenservice verwenden Sie einen Webbrowser oder Clientanwendung von Visual Studio 2008.

Vorgehensweise

Führen Sie den folgenden Befehl aus, um dem eXtreme-Scale-Datengrid schnell Inhalt hinzuzufügen:

1. Öffnen Sie eine Befehlszeile oder ein Terminalfenster, und setzen Sie die Umgebungsvariable JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=Java-Ausgangsverzeichnis`
   - Windows `set JAVA_HOME=Java-Ausgangsverzeichnis`
2. `cd Ausgangsverzeichnis_des_REST-Service/gettingstarted`
3. Fügen Sie einige Daten in das Grid ein. Die eingefügten Daten werden später mit einem Webbrowser und dem REST-Datenservice abgerufen.

   Wenn das Datengrid *ohne* eXtreme-Scale-Sicherheit gestartet wurde, verwenden Sie die folgenden Befehle:
   - UNIX Linux `./runclient.sh load default`
   - Windows `runclient.bat load default`

   Wenn das Datengrid *mit* eXtreme-Scale-Sicherheit gestartet wurde, verwenden Sie die folgenden Befehle:
   - UNIX Linux `./runclient_secure.sh load default`
   - Windows `runclient_secure.bat load default`

   Für einen Java-Client verwenden Sie die folgende Befehlssyntax:
   - UNIX Linux `runclient.sh Befehl`
   - Windows `runclient.bat Befehl`

   Die folgenden Befehle sind verfügbar:
   - `load default`

     Lädt einen vordefinierten Satz von Customer-, Category- und Product-Entitäten in das Datengrid und erstellt einen zufälligen Satz von Bestellungen (Order) für jeden Kunden (Customer).
   - `load category Kategorie-ID Kategoriename erste_Produkt-ID Anzahl_Produkte`

     Erstellt eine Produktkategorie und eine festgelegte Anzahl an Produktentitäten im Datengrid. Der Parameter "erste_Produkt-ID" gibt die ID-Nummer des ersten Produkts an, und jedem nachfolgenden Produkt wird die jeweils nächste ID zugeordnet, bis die angegebene Anzahl an Produkten erstellt wurde.
   - `load customer Firmencode KontaktnameFirmenname Anzahl_Bestellungen erste_Bestellungs-IDLieferstadt max_Artikel Skonto`

     Lädt einen neuen Kunden (Customer) in das Datengrid und erstellt einen festen Satz an Bestellungen (Order-Entitäten) für ein zufälliges Produkt, das derzeit im Grid geladen ist. Die Anzahl der Bestellungen wird mit dem Parameter <Anzahl_Bestellungen> bestimmt. Jede Bestellung hat eine zufällige Anzahl an OrderDetail-Entitäten (bis <max-Artikel>).
   - `display customer Firmencode`

     Zeigt eine Customer-Entität und die zugehörigen Order- und OrderDetail-Entitäten an.
   - `display category Kategorie-ID`

     Zeigt eine Category-Entität eines Produkts und die zugehörigen Product-Entitäten an.

Ergebnisse

- `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`

Beispieldatengrid und Java-Client mit Eclipse ausführen und erstellen

Das Einführungsbeispiel zum REST-Datenservice kann mit Eclipse aktualisiert und erweitert werden. Einzelheiten zum Konfigurieren Ihrer Eclipse-Umgebung finden Sie im Textdokument `Ausgangsverzeichnis_des_REST-Service/gettingstarted/ECLIPSE_README.txt`.

Nach dem Importieren des Projekts "WXSRestGettingStarted" in Eclipse und nach der erfolgreichen Erstellung des Beispiels wird das Beispiel automatisch neu kompiliert, und die zum Starten des Container-Servers und -clients verwendeten Scriptdateien verwenden automatisch die Klassendateien und XML-Dateien. Der REST-Datenservice erkennt automatisch alle Änderungen, die nach der Konfiguration des Webservers vorgenommen wurden, damit die Eclipse-Build-Verzeichnisse automatisch gelesen werden.

Wichtig: Wenn Quellen- oder Konfigurationsdateien geändert werden, müssen der eXtreme-Scale-Container und die REST-Datenserviceanwendung erneut gestartet werden. Der eXtreme-Scale-Container-Server muss vor der Webanwendung des REST-Datenservice gestartet werden.

WCF-Client von Visual Studio 2008 mit dem REST-Datenservice

Das Einführungsmuster zum REST-Datenservice von eXtreme Scale enthält einen WCF-Data-Services-Client, der mit dem REST-Datenservice von eXtreme Scale interagieren kann. Das Beispiel ist als Befehlszeilenanwendung in C# geschrieben.

Softwarevoraussetzungen

Der in C# geschriebene Beispielclient von WCF Data Services setzt Folgendes voraus:

- Betriebssystem
  - Microsoft Windows XP
  - Microsoft Windows Server 2003
  - Microsoft Windows Server 2008
  - Microsoft Windows Vista
- Microsoft Visual Studio 2008 mit Service-Pack 1

  Tipp: Zusätzliche Hardware- und Softwarevoraussetzungen finden Sie unter dem vorherigen Link.
- Microsoft .NET Framework 3.5 Service Pack 1
- Microsoft Support: An update for the .NET Framework 3.5 Service Pack 1 is available

Einführungslink erstellen und ausführen

Der Beispielclient von WCF Data Services enthält ein Visual-Studio-2008-Projekt sowie eine Lösung und den Quellcode für die Ausführung des Beispiels. Das Beispiel muss in Visual Studio 2008 geladen und in ein ausführbares Windows-Programm kompiliert werden, bevor es ausgeführt werden kann. Informationen zum Erstellen und Ausführen des Beispiels finden Sie im folgenden Textdokument: `Ausgangsverzeichnis_des_REST-Service/gettingstarted/VS2008_README.txt`.

Befehlssyntax für den in C# geschriebenen WCF-Data-Service-Client

Windows `WXSRESTGettingStarted.exe <Service-URL> <Befehl>`

<Service-URL> steht für den URL des zuvor konfigurierten REST-Datenservice von eXtreme Scale.

Die folgenden Befehle sind verfügbar:

- `load default`

  Lädt einen vordefinierten Satz von Customer-, Category- und Product-Entitäten in das Datengrid und erstellt einen zufälligen Satz von Bestellungen (Order) für jeden Kunden (Customer).
- `load category <Kategorie-ID> <Kategoriename> <erste_Produkt-ID> <Anzahl_Produkte>`

  Erstellt eine Produktkategorie und eine festgelegte Anzahl an Produktentitäten im Datengrid. Der Parameter "erste_Produkt-ID" gibt die ID-Nummer des ersten Produkts an, und jedem nachfolgenden Produkt wird die jeweils nächste ID zugeordnet, bis die angegebene Anzahl an Produkten erstellt wurde.
- `load customer <Firmencode> <Kontaktname> <Firmenname> <Anzahl_Bestellungen> <erste_Bestellungs-ID> <Lieferstadt> <max_Artikel> <Skonto>`

  Lädt einen neuen Kunden (Customer) in das Datengrid und erstellt einen festen Satz an Bestellungen (Order-Entitäten) für ein zufälliges Produkt, das derzeit im Datengrid geladen ist. Die Anzahl der Bestellungen wird mit dem Parameter <Anzahl_Bestellungen> bestimmt. Jede Bestellung hat eine zufällige Anzahl an OrderDetail-Entitäten (bis <max-Artikel>).
- `display customer <Firmencode>`

  Zeigt eine Customer-Entität und die zugehörigen Order- und OrderDetail-Entitäten an.
- `display category <Kategorie-ID>`

  Zeigt eine Category-Entität eines Produkts und die zugehörigen Product-Entitäten an.
- `unload`

  Entfernt alle Entitäten, die mit dem Befehl "default load" geladen wurden.

Die folgenden Beispiele veranschaulichen verschiedene Befehle.

- `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`

Server für OSGi konfigurieren

WebSphere eXtreme Scale enthält ein OSGi-Server-Bundle, das Ihnen ermöglicht, Server und Container in einem OSGi-Framework zu starten und zu konfigurieren. In den Konfigurationsartikeln wird beschrieben, wie Sie das Server-Bundle von eXtreme Scale, den OSGi-Blueprint-Service und die eXtreme-Scale-Konfiguration für die Ausführung der eXtreme-Server in einem Eclipse-Equinox-OSGi-Framework verwenden.

Informationen zu diesem Vorgang

Die folgenden Aufgaben müssen zum Starten eines eXtreme-Scale-Servers in Eclipse Equinox ausgeführt werden:

Vorgehensweise

1. OSGi-Bundle erstellen, in dem die Plug-ins von eXtreme Scale gespeichert werden, wo sie als Services bereitgestellt werden, und die ObjectGrid-XML-Deskriptordatei so aktualisieren, dass sie auf die Services verweist
2. OSGi zum Starten eines Container-Servers von eXtreme Scale starten
3. Server-Bundle von eXtreme Scale installieren und im OSGi-Framework starten
4. OSGi-Bundle installieren und starten, das die Plug-ins von eXtreme Scale enthält

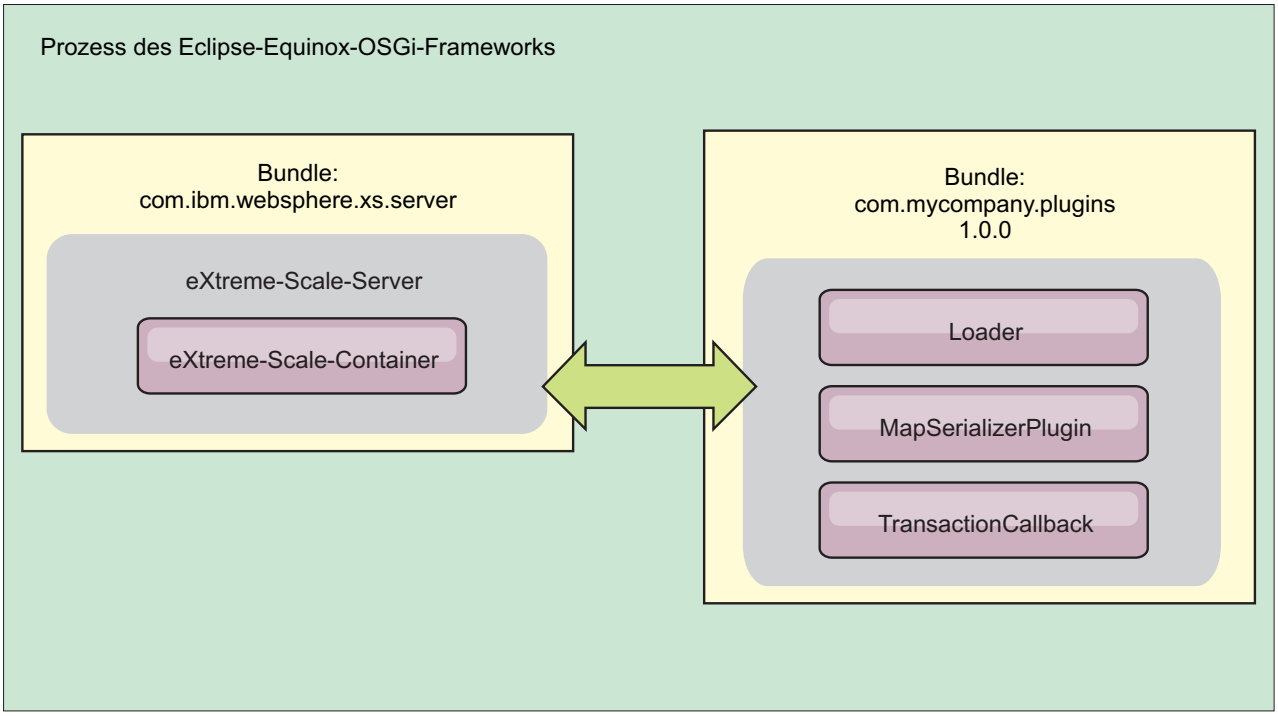


Abbildung 48. Eclipse-Equinox-Prozess für die Installation und das Starten von OSGi-Bundles mit Plug-ins von eXtreme Scale

eXtreme-Scale-Plug-ins mit OSGi Blueprint konfigurieren

Alle ObjectGrid- und BackingMap-Plug-ins von eXtreme Scale können mit dem OSGi-Blueprint-Service, der mit Eclipse Gemini und Apache Aries bereitgestellt wird, als OSGi-Beans und -Services definiert werden.

Vorbereitende Schritte

Bevor Sie Ihre Plug-ins als OSGi-Services konfigurieren können, müssen Sie Ihre Plug-ins in ein OSGi-Bundle packen und sich mit den grundlegenden Prinzipien der erforderlichen Plug-ins vertraut machen. Das Bundle muss die Server- bzw. Clientpakete von WebSphere eXtreme Scale sowie weitere abhängige Pakete, die von den Plug-ins benötigt werden, importieren oder eine Bundleabhängigkeit in den Server- bzw. Client-Bundles von eXtreme Scale erstellen. In diesem Artikel wird beschrieben, wie Sie die Blueprint-XML konfigurieren, um Plug-in-Beans zu erstellen und diese als OSGi-Services für eXtreme Scale bereitzustellen.

Informationen zu diesem Vorgang

Beans und Services werden in einer Blueprint-XML-Datei definiert, und der Blueprint-Container erkennt, erstellt und verbindet die Beans miteinander und stellt diese dann als Services bereit. Durch diesen Prozess werden die Beans anderen OSGi-Bundles, einschließlich den Server- und Client-Bundles von eXtreme Scale, zur Verfügung gestellt.

Wenn Sie angepasste Plug-in-Services für eXtreme Scale erstellen, muss das Bundle, in dem die Plug-ins gehostet werden sollen, für die Verwendung von Blueprint konfiguriert werden. Außerdem muss eine Blueprint-XML-Datei erstellt und im Bundle gespeichert werden. Informationen zum allgemeinen Verständnis der Spezi-

fikation finden Sie unter Building OSGi applications with the Blueprint Container specification.

Vorgehensweise

1. Erstellen Sie eine Blueprint-XML-Datei. Sie können die Datei beliebig nennen. Sie müssen jedoch den Blueprint-Namespace einschließen.

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
   ...
   </blueprint>
   ```

2. Erstellen Sie Bean-Definitionen in der Blueprint-XML-Datei für jedes eXtreme-Scale-Plug-in.

   Beans werden mit dem Element `<bean>` definiert, können mit anderen Bean-Referenzen verbunden werden und können Initialisierungsparameter enthalten.

   Wichtig: Beim Definieren einer Bean müssen Sie den richtigen Geltungsbereich verwenden. Blueprint unterstützt die Geltungsbereiche "Singleton" und "Prototyp". eXtreme Scale unterstützt auch einen angepassten Shard-Geltungsbereich.

   Definieren Sie die meisten eXtreme-Scale-Plug-ins als Beans mit dem Geltungsbereich "Prototyp" oder "Shard", weil alle Beans für jedes ObjectGrid-Shard bzw. jede BackingMap-Instanz, dem bzw. der sie zugeordnet ist, eindeutig sein muss. Beans mit dem Geltungsbereich "Shard" können hilfreich sein, wenn die Beans in anderen Kontexten verwendet werden, damit die richtige Instanz abgerufen wird.

   Zum Definieren einer Bean mit dem Geltungsbereich "Prototyp" verwenden Sie das Attribut `scope="prototype"` in der Bean:

   ```
   <bean id="myPluginBean" class="com.mycompany.MyBean" scope="prototype">
   ...
   </bean>
   ```

   Zum Definieren einer Bean mit dem Geltungsbereich "Shard" müssen Sie dem XML-Schema den Namespace `objectgrid` hinzufügen und das Attribut `scope="objectgrid:shard"` in der Bean verwenden:

   ```
   <?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. Erstellen Sie PluginServiceFactory-Bean-Definitionen für jede Plug-in-Bean. Alle eXtreme-Scale-Beans müssen eine definierte PluginServiceFactory-Bean haben, damit der richtige Bean-Geltungsbereich angewendet werden kann. eXtreme Scale enthält eine BlueprintServiceFactory, die Sie verwenden können. Sie enthält zwei Eigenschaften, die definiert werden müssen. Sie müssen die Eigenschaft blueprintContainer auf die `blueprintContainer`-Referenz und die Eigenschaft beanId auf den Bean-ID-Namen gesetzt werden. Wenn eXtreme Scale den Service für die Instanziierung der entsprechenden Beans sucht, sucht der Server die Bean-Komponenteninstanz mithilfe des Blueprint-Containers.

```
bean id="myPluginBeanFactory"
    class="com.ibm.websphere.objectgrid.plugins.osgi.BluePrintServiceFactory">
      <property name="blueprintContainer" ref="blueprintContainer"/>
<property name="beanId" value="myPluginBean" />
</bean>
```

4. Erstellen Sie einen Servicemanager für jede PluginServiceFactory-Bean. Jeder Servicemanager stellt die PluginServiceFactory-Bean mithilfe des Elements `<service>` bereit. Das Element "service" gibt den Namen an, unter dem die Bean OSGi bereitgestellt wird, die Referenz auf die PluginServiceFactory-Bean, die bereitzustellende Schnittstelle und das Ranking des Service. eXtreme Scale verwendet das Service-Manager-Ranking, um Service-Upgrades durchzuführen, wenn das eXtreme-Scale-Grid aktiv ist. Wenn das Ranking nicht angegeben wird, nimmt das OSGi-Framework das Ranking 0 an. Weitere Informationen finden Sie im Abschnitt zum Aktualisieren von Service-Rankings.

   Blueprint enthält mehrere Optionen für die Konfiguration von Service-Managern. Zum Definieren eines einfachen Service-Managers für eine PluginServiceFactory-Bean erstellen Sie ein Element `<service>` für jede PluginServiceFactory-Bean:

   ```
   <service ref="myPluginBeanFactory"
       interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
       ranking="1">
   </service>
   ```

5. Speichern Sie die Blueprint-XML-Datei im Plug-in-Bundle. Die Blueprint-XML-Datei muss im Verzeichnis `OSGI-INF/blueprint` gespeichert werden, damit sie vom Blueprint-Container erkannt wird.

   Wenn Sie die Blueprint-XML-Datei in einem anderen Verzeichnis speichern möchten, müssen Sie den folgenden Bundle-Blueprint-Manifestheader angeben:

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

Ergebnisse

Die eXtreme-Scale-Plug-ins sind jetzt für die Bereitstellung in einem OSGi-Blueprint-Container konfiguriert. Außerdem wurde die ObjectGrid-XML-Deskriptordatei so konfiguriert, dass sie auf die Plug-ins über den OSGi-Blueprint-Service verweist.

Server mit OSGi Blueprint konfigurieren

Sie können Container-Server von WebSphere eXtreme Scale mit einer OSGi-Blueprint-XML-Datei konfigurieren, was das Packen und die Entwicklung eigenständiger Server-Bundles vereinfacht.

Vorbereitende Schritte

In diesem Artikel wird davon ausgegangen, dass die folgenden Aufgaben ausgeführt wurden:

- Das Eclipse-Equinox-OSGi-Framework wurde installiert und mit dem Eclipse-Gemini- oder Apache-Aries-Blueprint-Container gestartet.
- Das eXtreme-Scale-Server-Bundle wurde installiert und gestartet.
- Das Bundle mit den dynamischen eXtreme-Scale-Plug-ins wurde erstellt.
- Die ObjectGrid-XML-Deskriptordatei und die XML-Implementierungsrichtliniendatei von eXtreme Scale wurden erstellt.

Informationen zu diesem Vorgang

In dieser Aufgabe wird beschrieben, wie Sie einen eXtreme-Scale-Server mit einem Container über eine Blueprint-XML-Datei konfigurieren. Das Ergebnis dieser Prozedur ist ein Container-Bundle. Wenn das Container-Bundle gestartet wird, überwacht das eXtreme-Scale-Server-Bundle das Bundle, parst die Server-XML und startet Server und Container.

Ein Container-Bundle kann optional mit der Anwendung und den eXtreme-Scale-Plug-ins kombiniert werden, wenn dynamische Plug-in-Aktualisierungen nicht erforderlich sind oder die Plug-ins keine dynamische Aktualisierung unterstützen.

Vorgehensweise

1. Blueprint-XML-Datei mit eingeschlossenem `objectgrid`-Namespace erstellen. Sie können die Datei beliebig nennen. Sie muss jedoch den Blueprint-Namespace enthalten.

   ```
   <?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">
   ...
   </blueprint>
   ```

2. XML-Definition für den eXtreme-Scale-Server mit den entsprechenden Servereigenschaften hinzufügen. Einzelheiten zu allen verfügbaren Konfigurationseigenschaften finden Sie in der Spring-XML-Deskriptordatei. Sehen Sie sich das folgende XML-Definitionsbeispiel an:

   ```
   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. XML-Definition für den eXtreme-Scale-Container mit der Referenz auf die Serverdefinition sowie die im Bundle integrierten ObjectGrid-XML-Deskriptor- und ObjectGrid-XML-Implementierungsdateien hinzufügen, z. B.:

   ```
   <objectgrid:container id="container"
       objectgridxml="/META-INF/objectGrid.xml"
       deploymentxml="/META-INF/objectGridDeployment.xml"
       server="xsServer" />
   ```

4. Blueprint-XML-Datei im Container-Bundle speichern. Die Blueprint-XML-Datei muss im Verzeichnis `OSGI-INF/blueprint` gespeichert werden, damit der Blueprint-Container gefunden wird.

   Wenn Sie die Blueprint-XML-Datei in einem anderen Verzeichnis speichern möchten, müssen Sie den Manifestheader "Bundle-Blueprint" angeben, z. B.:

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

5. Dateien in eine einzige Bundle-JAR-Datei packen. Sehen Sie sich das folgende Beispiel für eine Bundleverzeichnishierarchie an:

   ```
   MyBundle.jar
       /META-INF/manifest.mf
       /META-INF/objectGrid.xml
       /META-INF/objectGridDeployment.xml
       /OSGI-INF/blueprint/blueprint.xml
   ```

Ergebnisse

Es wurde ein eXtreme-Scale-Container-Bundle erstellt, das in Eclipse Equinox installiert werden kann. Wenn das Container-Bundle gestartet wird, startet die Laufzeitumgebung des eXtreme-Scale-Servers automatisch den eXtreme-Scale-Singleton-Server mit den im Bundle definierten Parametern und startet einen Container-Server. Das Bundle kann gestoppt und gestartet werden, was dazu führt, dass der Container gestoppt bzw. gestartet wird. Der Server ist ein Singleton und wird nicht gestoppt, wenn das Bundle zum ersten Mal gestartet wird.

Server mit der OSGI-Konfigurationsverwaltung konfigurieren

Sie können den OSGi-Konfigurationsverwaltungsservice verwenden, um Container-Server von WebSphere eXtreme Scale zu konfigurieren.

Informationen zu diesem Vorgang

Zum Konfigurieren eines Servers wird die ManagedService-PID (persistente ID) "com.ibm.websphere.xs.server" gesetzt, dass auf die ObjectGrid-Servereigenschaftendatei im Dateisystem verwiesen wird. Zum Konfigurieren eines Containers wird die ManagedServiceFactory-PID "com.ibm.websphere.xs.container" gesetzt, um auf die ObjectGrid-XML-Implementierungsdatei und die ObjectGrid-XML-Implementierungsrichtliniendatei im Dateisystem zu verweisen.

Wenn die beiden PIDs im Konfigurationsverwaltungsservice gesetzt werden, initialisiert der eXtreme-Scale-Server-Service automatisch den Server und startet den Container mit den angegebenen Konfigurationsdateien. Die Konfigurationsverwaltungs-PIDs werden im OSGi-Konfigurationsverzeichnis persistent gespeichert. Wenn die Konfiguration nicht bereinigt wird, bleiben die Einstellungen über Frameworkneustarts hinweg aktiv.

Es gibt mehrere Dienstprogramme anderer Anbieter für die Festlegung der Konfigurationsverwaltungseigenschaften. Die folgenden Dienstprogramme sind Beispiele für Tools, die vom Produkt unterstützt werden:

- Der Befehlszeilenclient Luminis OSGi Configuration Admin ermöglicht Ihnen eine Konfiguration über die Befehlszeile.
- Mit Apache Felix File Install können Sie die PID-Einstellungen für die Konfigurationsverwaltung in Standardeigenschaftendateien angeben.

Zum Konfigurieren von eXtreme-Scale-Container-Servern mit dem Befehlszeilenclient Lumis OSGi Configuration Administration gehen Sie wie folgt vor:

Vorgehensweise

1. Erstellen Sie eine Managed-Service-PID für die Eigenschaftendatei des ObjectGrid-Servers in der OSGi-Konsole, indem Sie die folgenden Befehle ausführen:

   ```
   osgi> cm create com.ibm.websphere.xs.server
   osgi> cm put com.ibm.websphere.xs.server objectgrid.server.props /mypath/server.properties
   ```

2. Erstellen Sie eine Managed-Service-Factory-PID für den ObjectGrid-Container in der OSGi-Konsole, indem Sie die folgenden Befehle ausführen:

   Achtung: Verwenden Sie die PID, die mit dem Konfigurationsverwaltungsbefehl **createf** erstellt wurde. Die PID, die im folgenden Code-Snippet verwendet wird, ist nur ein Beispiel.

   ```
   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
   ```

Ergebnisse

Die eXtreme-Scale-Container-Server sind jetzt konfiguriert und können in einem Eclipse-Equinox-OSGi-Framework gestartet werden.

Nächste Schritte

Die Container-Server können auch über das Programm mit der API ServerFactory und den OSGi-Bundle-Aktivatoren erstellt werden. Einzelheiten zur Verwendung der API ServerFactory finden Sie in der API-Dokumentation.

Kapitel 7. Verwalten

Die Verwaltung und der Betrieb der Produktumgebung umfassen das Starten und Stoppen von Servern, die Verwaltung der Verfügbarkeit des Datengrids und die Wiederherstellung nach Ausfällen im Rechenzentrum. Nach der Konfiguration Ihrer Katalogserver und Container-Server können Sie die Server mit verschiedenen Methoden starten und stoppen. Die Methode, die Sie zum Starten und Stoppen von Servern verwenden, richtet sich danach, ob Sie eine integrierte Topologie oder eine Topologie verwenden, die in WebSphere Application Server ausgeführt wird.

Eigenständige Server starten und stoppen

Sie können eigenständige Katalog- und Container-Server mit den Scripts **startOgServer** und **stopOgServer** oder mit der integrierten Server-API starten und stoppen.

Vorbereitende Schritte

Wenn Sie Server in einer eigenständigen Umgebung starten oder stoppen, die einen externen Clientsicherheitsprovider verwendet, müssen Sie die Umgebungsvariable *CLIENT_AUTH_LIB* setzen, bevor Sie das Script **startOgServer** oder **stopOgServer** ausführen. Weitere Informationen zum Setzen dieser Umgebungsvariablen finden Sie unter „Sichere Server in einer eigenständigen Umgebung starten" auf Seite 538.

Eigenständige Server starten

Wenn Sie eine eigenständige Konfiguration verwenden, setzt sich die Umgebung aus Katalogservern, Container-Servern und Clientprozessen zusammen. Server von WebSphere eXtreme Scale können mit Hilfe der integrierten Server-API auch in vorhandene Java-Anwendungen integriert werden. Sie müssen diese Prozesse manuell konfigurieren und starten.

Vorbereitende Schritte

Sie können Server von WebSphere eXtreme Scale in einer Umgebung ohne WebSphere Application Server starten. Wenn Sie WebSphere Application Server verwenden, lesen Sie den Abschnitt „WebSphere eXtreme Scale mit WebSphere Application Server konfigurieren" auf Seite 263.

Eigenständigen Katalogservice starten

Sie müssen den Katalogservice manuell starten, wenn Sie eine verteilte Umgebung von WebSphere eXtreme Scale ohne WebSphere Application Server verwenden.

Vorbereitende Schritte

- Wenn Sie WebSphere Application Server verwenden, wird der Katalogservice automatisch in den vorhandenen Prozessen gestartet. Weitere Informationen finden Sie unter Katalogservice in WebSphere Application Server starten.

Informationen zu diesem Vorgang

Starten Sie den Katalogservice mit dem Script **startOgServer**. Wenn Sie den Startbefehl aufrufen, verwenden Sie das Script **startOgServer.sh** auf UNIX-Plattformen bzw. das Script **startOgServer.bat** unter Windows.

Der Katalogservice kann in einem einzelnen Prozess ausgeführt werden, oder es können mehrere Katalogserver zu einer Katalogservicedomäne zusammengefasst werden. Eine Katalogservicedomäne ist in einer Produktionsumgebung für hohe Verfügbarkeit erforderlich. Weitere Einzelheiten zu Katalogservicedomänen finden Sie in den Informationen zu Katalogservicedomänen in der Veröffentlichung *Produktübersicht*. Außerdem können Sie zusätzliche Scriptparameter angeben, um den ORB (Object Request Broker) an eine bestimmte Host/Port-Kombination zu binden, die Domäne anzugeben oder die Sicherheit zu aktivieren.

Vorgehensweise

- **Einen einzigen Katalogserverprozess starten.**

  Geben Sie zum Starten eines einzelnen Katalogservers die folgenden Befehle in der Befehlszeile ein:

  1. Navigieren Sie zum Verzeichnis `bin`.

     ```
     cd ObjectGrid-Stammverzeichnis/bin
     ```

  2. Führen Sie den Befehl **startOgServer** aus.

     ```
     startOgServer.bat|sh catalogServer
     ```

  Eine Liste aller verfügbaren Befehlszeilenparameter finden Sie im Abschnitt „Script **startOgServer**" auf Seite 409. Wenn Sie den Katalogservice in einer Produktionsumgebung ausführen, verwenden Sie keine einzelne Java Virtual Machine (JVM). Wenn der Katalogservice fehlschlägt, können in diesem Fall keine neuen Clients Anforderungen an die implementierte eXtreme-Scale-Umgebung weitergeleitet werden, und es können keine neuen ObjectGrid-Instanzen zur Domäne hinzugefügt werden. Aus diesen Gründen sollten Sie eine Gruppe von Java Virtual Machines starten, um eine Katalogservicedomäne auszuführen.

- **Katalogservicedomäne starten, die sich aus mehreren Endpunkten zusammensetzt.**

  Wenn Sie eine Gruppe von Servern zum Ausführen eines Katalogservice starten möchten, müssen Sie die Option **-catalogServiceEndPoints** im Script `startOgServer` verwenden. Dieses Argument akzeptiert eine Liste mit Katalogserviceendpunkten im Format *Servername*:*Hostname*:*Client-Port*:*Peer-Port*.Im folgenden Beispiel wird gezeigt, wie die erste von drei Java Virtual Machines für einen Katalogservice gestartet wird:

  1. Navigieren Sie zum Verzeichnis "bin".

     ```
     cd WXS-Installationsstammverzeichnis/bin
     ```

  2. Führen Sie den Befehl **startOgServer** aus.

     ```
     startOgServer.bat|sh cs1 -catalogServiceEndPoints
     cs1:MyServer1.company.com:6601:6602,
     cs2:MyServer2.company.com:6601:6602,
     cs3:MyServer3.company.com:6601:6602
     ```

     In diesem Beispiel wird der Server `cs1` auf dem Host `MyServer1.company.com` gestartet. Dieser Servername ist das erste Argument, das an das Script übergeben wird. Während der Initialisierung von Server `cs1` werden die catalogServiceEndpoints-Parameter untersucht, um die für diesen Prozess zugeord-

neten Ports zu bestimmen. Die Liste wird auch verwendet, um Server cs1 das Annehmen von Verbindungen von anderen Servern (cs2 und cs3) zu ermöglichen.

3. Zum Starten der verbleibenden Katalogserver in der Liste, übergeben Sie die folgenden Argumente an das Script "startOgServer". Der Server cs2 wird auf dem Host MyServer2.company.com gestartet.

```
startOgServer.bat|sh cs2 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

Es soll der Server cs3 auf dem Host MyServer3.company.com gestartet werden:

```
startOgServer.bat|sh cs3 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

Wichtig: Mindestens zwei Katalogserver parallel starten.

Sie müssen Katalogserver, die in einem Datengrid enthalten sind, parallel starten, weil jeder Server wartet, bis die anderen Katalogserver der Stammgruppe beitreten. Ein für ein Datengrid konfigurierter Katalogserver wird erst gestartet, wenn er die anderen Member in der Gruppe identifiziert. Der Katalogserver überschreitet das zulässige Zeitlimit, wenn keine anderen Server verfügbar werden.

- **ORB an einen bestimmten Host und einen bestimmten Port binden.**

  Neben den Ports, die mit dem Argument **catalogServiceEndpoints** definiert werden, verwendet jeder Katalogservice einen Object Request Broker (ORB), um Verbindungen von Clients und Containern zu akzeptieren. Standardmäßig ist der ORB an Port 2809 des lokalen Hosts empfangsbereit. Wenn Sie den ORB an eine bestimmte Host/Port-Kombination in der JVM des Katalogservice binden möchten, verwenden Sie dazu die Argumente **-listenerHost** und **-listenerPort**. Im folgenden Beispiel wird gezeigt, wie Sie einen Katalogserver mit einer einzelnen JVM starten und den zugehörigen ORB an Port 7000 auf MyServer1.company.com binden:

```
startOgServer.sh catalogServer -listenerHost MyServer1.company.com
-listenerPort 7000
```

  Jedem eXtreme-Scale-Container und -Client müssen die ORB-Endpunktdaten des Katalogservice bereitgestellt werden. Clients benötigen nur einen Teil dieser Daten, aber Sie sollten für eine hohe Verfügbarkeit mindestens zwei Endpunkte verwenden.

- Optional: **Katalogservicedomäne benennen.**

  Ein Katalogservicedomänenname ist nicht erforderlich, wenn Sie einen Katalogservice starten. Wenn Sie jedoch die Multimasterreplikation oder mehrere Katalogservicedomänen in derselben Gruppe von Prozessen verwenden, müssen Sie einen eindeutigen Katalogservicedomänennamen definieren. Der Standarddomänenname ist DefaultDomain. Wenn Sie Ihre Domäne benennen möchten, verwenden Sie die Option **-domain**. Im folgenden Beispiel wird demonstriert, wie Sie einen Katalogservice mit einer einzelnen JVM und dem Domänennamen myDomain starten.

```
startOgServer.sh catalogServer -domain myDomain
```

Weitere Informationen zum Konfigurieren der Multimasterreplikation finden Sie unter „Topologien mit mehreren Rechenzentren konfigurieren” auf Seite 288.

- **Sicheren Katalogservice starten.** Weitere Informationen finden Sie unter „Sichere Server in einer eigenständigen Umgebung starten” auf Seite 538.
- **Katalogservice über das Programm starten.**

  Jede JVM, die von der Methode CatalogServerProperties.setCatalogServer markiert wird, kann den Katalogservice für eXtreme Scale ausführen. Diese Methode weist die Laufzeitumgebung des eXtreme-Scale-Servers an, den Katalogservice zu instanziieren, wenn der Server gestartet wird. Der folgende Code veranschaulicht, wie der Katalogserver von eXtreme Scale instanziiert wird:

  ```
  CatalogServerProperties catalogServerProperties =
   ServerFactory.getCatalogProperties();
  catalogServerProperties.setCatalogServer(true);

  // Die Methode getInstance() startet den Katalogservice.
  Server server = ServerFactory.getInstance();
  ```

  Weitere Informationen zum Starten von Servern über das Programm finden Sie unter „Integrierte Server-API zum Starten und Stoppen von Servern verwenden” auf Seite 418.

Container-Server starten

Sie können Container-Server über die Befehlszeile unter Verwendung einer Implementierungstopologie oder einer Datei `server.properties` starten.

Informationen zu diesem Vorgang

Zum Starten eines Containerprozesses benötigen Sie eine ObjectGrid-XML-Datei. Die ObjectGrid-XML-Datei gibt an, welche eXtreme-Scale-Server im Container enthalten sind. Stellen Sie sicher, dass Ihr Container jedes ObjectGrid in der XML aufnehmen kann, die Sie übergeben. Alle Klassen, die diese ObjectGrids voraussetzen, müssen im Klassenpfad des Containers enthalten sein. Weitere Informationen zur ObjectGrid-XML-Datei finden Sie im Abschnitt Datei `objectGrid.xsd`.

Vorgehensweise

- **Starten Sie den Containerprozess über die Befehlszeile.**
  1. Navigieren Sie über die Befehlszeile zum Verzeichnis "bin":

     ```
     cd WXS-Installationsstammverzeichnis/bin
     ```

  2. Führen Sie den folgenden Befehl aus:

     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Wichtig: Im Container wird die Option **-catalogServiceEndPoints** verwendet, um auf den ORB-Host (Object Request Broker) und -Port des Katalogservice zu verweisen. Der Katalogservice verwendet die Optionen **-listenerHost** und **-listenerPort**, um den Host und den Port für die ORB-Bindung anzugeben, oder er akzeptiert die Standardbindung. Wenn Sie einen Container starten, verwenden Sie die Option **-catalogServiceEndPoints**, um die Werte zu referenzieren, die an die Optionen **-listenerHost** und **-listenerPort** im Katalogservice übergeben werden. Wenn die Optionen **-listenerHost** und **-listenerPort** beim Starten des Katalogservice nicht verwendet werden, stellt der ORB eine Bindung zu Port 2809 auf dem lokalen Host für den Katalogservice her. Verwenden Sie die Option **-catalogServiceEndPoints** nicht, um die Hosts und Ports zu referenzieren, die an die Option **-catalogServiceEndPoints** im Katalogservice übergeben wurden. Im Katalogservice wird die Option **-catalogServiceEndPoints** verwendet, um die erforderlichen Ports für die statische Serverkonfiguration

anzugeben.
Dieser Prozess wird mit c0 identifiziert, dem ersten Argument, das an das Script übergeben wird. Verwenden Sie die Datei `companyGrid.xml`, um den Container zu starten. Wenn Ihr Katalogserver-ORB auf einem anderen Host ausgeführt wird als Ihr Container oder wenn er einen vom Standard abweichenden Port verwendet, müssen Sie das Argument **-catalogServiceEndPoints** verwenden, um die Verbindung zum ORB herzustellen. Für dieses Beispiel wird angenommen, dass ein einzelner Katalogservice an Port `2809` auf `MyServer1.company.com` aktiv ist.

- **Starten Sie den Container über eine Implementierungsrichtlinie.**

  Obwohl dies nicht erforderlich ist, wird eine Implementierungsrichtlinie für den Containerstart empfohlen. Die Implementierungsrichtlinie wird verwendet, um die Partitionierung und Replikation für eXtreme Scale zu konfigurieren. Die Implementierungsrichtlinie kann auch verwendet werden, um das Verteilungsverhalten zu beeinflussen. Da im vorherigen Beispiel keine Implementierungsrichtliniendatei angegeben wurde, empfängt das Beispiel alle Standardwerte für Replikation, Partitionierung und Verteilung. Deshalb sind die Maps im CompanyGrid in einem einzigen MapSet enthalten. Das MapSet wird nicht partitioniert oder repliziert. Weitere Informationen zu Implementierungsrichtliniendateien finden Sie im Abschnitt XML-Deskriptordatei für Implementierungsrichtlinie. Im folgenden Beispiel wird die Datei `companyGridDpReplication.xml` verwendet, um eine Container-JVM, die JVM "c0", zu starten:

  1. Navigieren Sie über die Befehlszeile zum Verzeichnis "bin":

     ```
     cd WXS-Installationsstammverzeichnis/bin
     ```

  2. Führen Sie den folgenden Befehl aus:

     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Anmerkung: Wenn Sie Java-Klassen haben, die in einem bestimmten Verzeichnis gespeichert sind, können Sie den Server wie folgt mit Argumenten starten, anstatt das Script "StartOgServer" zu ändern: `-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`
  . In der Datei `companyGridDpReplication.xml` enthält ein einziges MapSet alle Maps. Dieses MapSet wird in 10 Partitionen aufgeteilt. Jede Partition hat ein synchrones Replikat und keine asynchronen Replikate. Jeder Container, der die Implementierungsrichtlinie `companyGridDpReplication.xml` in Kombination mit der ObjectGrid-XML-Datei `companyGrid.xml` verwendet, kann CompanyGrid-Shards aufnehmen. Starten Sie eine weitere Container-JVM, die JVM "c1":

  1. Navigieren Sie über die Befehlszeile zum Verzeichnis "bin":

     ```
     cd WXS-Installationsstammverzeichnis/bin
     ```

  2. Führen Sie den folgenden Befehl aus:

     ```
     startOgServer.sh c1 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Jede Implementierungsrichtlinie enthält ein oder mehrere Elemente "objectgridDeployment". Wenn ein Container gestartet wird, veröffentlicht er seine Implementierungsrichtlinie im Katalogservice. Der Katalogservice untersucht jedes Element "objectgridDeployment". Wenn der Wert des Attributs "objectgridName" mit dem Wert des Attributs "objectgridName" eines zuvor empfangenen Elements "objectgridDeployment" übereinstimmt, wird das letzte Element "objectgridDeployment" ignoriert. Das erste Element "objectgridDeployment", das für ein bestimmtes Attribut "objectgridName" empfangen wird, wird als Master ver-

wendet. Angenommen, die JVM "c2" verwendet eine Implementierungsrichtlinie, die das MapSet in eine andere Anzahl von Partitionen aufteilt:

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>
```

Jetzt können Sie eine dritte JVM, die JVM "c2", starten.

1. Navigieren Sie über die Befehlszeile zum Verzeichnis "bin":

   ```
   cd WXS-Installationsstammverzeichnis/bin
   ```

2. Führen Sie den folgenden Befehl aus:

   ```
   startOgServer.sh c2 -objectGridFile ../xml/companyGrid.xml
   -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
   -catalogServiceEndPoints MyServer1.company.com:2809
   ```

Der Container in der JVM "c2" wird mit einer Implementierungsrichtlinie gestartet, die 5 Partitionen für mapSet1 angibt. Der Katalogservice enthält jedoch bereits die Masterkopie des objectgridDeployment-Objekt für das CompanyGrid. Als die JVM "c0" gestartet wurde, waren 10 Partitionen für dieses MapSet vorhanden. Weil dies der erste Container war, der gestartet wurde und seine Implementierungsrichtlinie veröffentlicht hat, wurde seine Implementierungsrichtlinie als Master definiert. Deshalb wird jeder objectgridDeployment-Attributwert, der `CompanyGrid` entspricht, in einer nachfolgenden Implementierungsrichtlinie ignoriert.

- **Starten Sie einen Container über eine Servereigenschaftendatei.**

  Sie können eine Servereigenschaftendatei verwenden, um die Traceerstellung zu und die Sicherheit für einen Container zu konfigurieren. Führen Sie die folgenden Befehle aus, um Container "c3" über eine Servereigenschaftendatei zu starten:

  1. Navigieren Sie über die Befehlszeile zum Verzeichnis "bin":

     ```
     cd WXS-Installationsstammverzeichnis/bin
     ```

  2. Führen Sie den folgenden Befehl aus:

     ```
     startOgServer.sh c3 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     -serverProps ../serverProps/server.properties
     ```

  Im Folgenden sehen Sie eine Beispieldatei `server.properties`:

  server.properties

  ```
  workingDirectory=
  traceSpec=*=all=disabled
  systemStreamToFileEnabled=trueenableMBeans=true
  memoryThresholdPercentage=50
  ```

Dies ist die Basisservereigenschaftendatei, in der die Sicherheit nicht aktiviert ist. Weitere Informationen zur Datei `server.properties` finden Sie im Abschnitt Servereigenschaftendatei.

- **Starten Sie einen Container-Server über das Programm.**

  Weitere Informationen zum Starten von Container-Servern über das Programm finden Sie unter „Integrierte Server-API zum Starten und Stoppen von Servern verwenden” auf Seite 418.

Script `startOgServer`

Das Script **`startOgServer`** startet Container- und Katalogserver. Sie können beim Starten Ihrer Server eine Vielzahl von Parametern verwenden, um die Traceerstellung zu aktivieren, Portnummern anzugeben usw.

Zweck

Sie können das Script **`startOgServer`** verwenden, um Server zu starten.

Position

Sie finden das Script **`startOgServer`** im Verzeichnis `bin` des Stammverzeichnisses, z. B.:

```
cd WXS-Installationsstammverzeichnis/bin
```

Anmerkung: Wenn Sie Java-Klassen haben, die in einem bestimmten Verzeichnis gespeichert sind, können Sie den Server wie folgt mit Argumenten starten, anstatt das Script "startOgServer" zu ändern: `-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`

.

Syntax für Katalogserver

Verwenden Sie zum Starten eines Katalogservers die folgenden Befehle:

Windows

```
startOgServer.bat <Server> [Optionen]
```

UNIX

```
startOgServer.sh <Server>[Optionen]
```

Verwenden Sie zum Starten des konfigurierten Standardkatalogservers die folgenden Befehle:

Windows

```
startOgServer.bat catalogServer
```

UNIX

```
startOgServer.sh catalogServer
```

Optionen für das Starten von Katalogservern

Die folgenden Parameter sind alle optional.

Parameter für das Starten eines Katalogservers:

-catalogServiceEndPoints <Servername:Hostname:Client-Port:Peer-Port>
Verweist im Container auf den ORB-Host (Object Request Broker) und ORB-Port im Katalogservice. Die Attribute sind wie folgt definiert:

Servername
Gibt einen Namen an, der den Prozess identifiziert, den Sie starten.

Hostname
Gibt den Hostnamen des Computers an, auf dem der Server gestartet wird.

Client-Port
Gibt den Port an, der für die Kommunikation mit dem Peerkatalogservice verwendet wird.

Peer-Port
Dieser Wert entspricht dem HA-Manager-Port. Er gibt den Port an, der für die Kommunikation mit dem Peerkatalogservice verwendet wird.

Im folgenden Beispiel wird der Katalogserver "cs1" gestartet, der sich in derselben Katalogservicedomäne wie die Server "cs2" und "cs3" befindet:

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

-clusterSecurityFile <XML-Sicherheitsdatei des Clusters>
Gibt die Datei `objectGridSecurity.xml` auf der Festplatte an, die die Sicherheitseigenschaften beschreibt, die allen Servern (einschließlich Katalogservern und Container-Servern) gemein sind. Ein Beispiel für eine solche Eigenschaft ist die Authentifikatorkonfiguration, die die Benutzerregistry und das Authentifizierungsverfahren darstellt.

Beispiel:`/opt/xs/ogsecurity.xml`

-clusterSecurityUrl <URL der Sicherheits-XML des Clusters>
Gibt die Datei `objectGridSecurity.xml` als URL zur Datei auf der Festplatte bzw. im Netz an, die die Sicherheitseigenschaften beschreibt, die allen Servern (einschließlich Katalogservern und Container-Servern) gemein sind. Ein Beispiel für eine solche Eigenschaft ist die Authentifikatorkonfiguration, die die Benutzerregistry und das Authentifizierungsverfahren darstellt.

Beispiel: `file:///opt/xs/ogsecurity.xml`

-domain <Domänenname>
Gibt den Namen der Katalogservicedomäne für diesen Katalogserver an. Die Katalogservicedomäne erstellt eine Gruppe hoch verfügbarer Katalogserver. Jeder Katalogserver für eine Domäne muss denselben Wert für den Parameter **-domain** angeben.

-JMXConnectorPort <Port>
Definiert den SSL-Port (Secure Sockets Layer), an den der JMX-Service (Java Management Extensions) gebunden wird.

-haManagerPort <Port>
Synonym mit Peer-Port. Gibt die Nummer des vom High Availability Manager verwendeten Ports an. Wenn Sie diese Eigenschaft nicht definieren, generiert der Katalogservice automatisch einen verfügbaren Port. Diese Eigenschaft gilt für den Container-Server und den Katalogservice. (Nur erforderlich für Umgebungen von WebSphere Application Server.)

-JMXServicePort <Port>
Gibt die Nummer des Ports an, über den der MBean-Server mit Java Management Extensions (JMX) kommuniziert. Sie müssen für jede JVM in Ihrer Konfiguration eine andere Portnummer verwenden. Wenn Sie JMX/RMI verwenden, geben Sie die Option **JMXServicePort** und die Portnummer explizit an, selbst wenn Sie den Standardportwert verwenden möchten. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.

Standardeinstellung: `1099`

-jvmArgs <JVM-Argumente>
Gibt eine Gruppe von JVM-Argumenten an. Alle Optionen hinter der Option **-jvmArgs** werden verwendet, um die Server-JVM zu starten. Wenn der Parameter **-jvmArgs** verwendet wird, müssen Sie sicherstellen, dass er das letzte optionale Scriptargument ist, das angegeben wird.

Beispiel:-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <Hostname>

Gibt den Hostnamen an, an den der Object Request Broker (ORB) für die Kommunikation mit Internet Inter-ORB Protocol (IIOP) gebunden wird. Der Wert muss ein vollständig qualifizierter Domänenname oder eine IP-Adresse sein. Wenn Ihre Konfiguration mehrere Netzkarten enthält, definieren Sie den Listener-Host und Listener-Port, so dass der Object Request Broker in der JVM die IP-Adresse für die Bindung kennt. Wenn Sie die zu verwendende IP-Adresse nicht kennen, können Symptome wie Überschreitungen des Verbindungszeitlimits, ungewöhnliche API-Fehler und Clients, die zu blockieren scheinen, auftreten.**Standardeinstellung:** `localhost`

-listenerPort <Port>
Gibt die Portnummer an, an die der Object Request Broker (ORB) gebunden wird. Diese Einstellung konfiguriert Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. In WebSphere Application Server wird der Listener-Port von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.**Standardeinstellung:** `2809`

-quorum true|false
Aktiviert das Quorum im Katalogserver. Weitere Informationen finden Sie unter Katalogserverquorum.

-script <Scriptdatei>
Gibt die Position eines angepasstesn Script für Befehle an, die Sie angeben, um Katalogserver oder -container zu starten und dann wie gewünscht zu parametrisieren oder zu bearbeiten.

-serverProps <Servereigenschaftendatei>
Gibt die Servereigenschaftendatei an, die die serverspezifischen Sicherheitseigenschaften enthält. Der Dateiname für diese Eigenschaft kann im herkömmlichen Dateipfadformat angegeben werden, z. B. `c:/tmp/og/catalogserver.props`.

-traceSpec <Tracespezifikation>
Gibt eine Zeichenfolge an, die den Umfang des Trace angibt, der beim Serverstart aktiviert wird.

Beispiel:

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <Tracedatei>
Gibt den Pfad der Datei an, in der Traceinformationen gespeichert werden.

Beispiel: `../logs/c4Trace.log`

-timeout <Sekunden>
Gibt das Zeitlimit (in Sekunden) für den Serverstart an.

Syntax für Container-Server Windows

```
startOgServer.bat <Server> -objectgridFile <XML-Datei>
-deploymentPolicyFile <XML-Datei> [Optionen]
```

Windows

```
startOgServer.bat <Server> -objectgridUrl <XML-URL>
-deploymentPolicyUrl <XML-URL> [Optionen]
```

UNIX

```
startOgServer.sh <Server> -objectgridFile <XML-Datei>
-deploymentPolicyFile <XML-Datei> [Optionen]
```

UNIX

```
 startOgServer.sh <Server> -objectgridUrl <XML-URL>
-deploymentPolicyUrl <XML-URL> [Optionen]
```

Optionen für Container-Server

-catalogServiceEndPoints<Hostname:Port,Hostname:Port>
Gibt den ORB-Host und ORB-Port im Katalogservice an.

Standardeinstellung: localhost:2809

-deploymentPolicyFile <XML-Datei_für_Implementierungsrichtlinien>
Gibt den Pfad der Implementierungsrichtliniendatei auf der Festplatte an. Die Implementierungsrichtlinie wird für die Konfiguration der Partitionierung und der Replikation verwendet. Die Implementierungsrichtlinie kann auch verwendet werden, um das Verteilungsverhalten zu beeinflussen.

Beispiel: `../xml/SimpleDP.xml`

-deploymentPolicyUrl <URL_der_Implementierungsrichtlinie>
Gibt den URL für die Implementierungsrichtliniendatei auf der Festplatte bzw. im Netz an. Die Implementierungsrichtlinie wird für die Konfiguration der Partitionierung und der Replikation verwendet. Die Implementierungsrichtlinie kann auch verwendet werden, um das Verteilungsverhalten zu beeinflussen.

Beispiel: `file://xml/SimpleDP.xml`

-JMXConnectorPort <Port>
Definiert den SSL-Port (Secure Sockets Layer), an den der JMX-Service (Java Management Extensions) gebunden wird.

-JMXServicePort <Port>

Gibt die Nummer des Ports an, über den der MBean-Server mit Java Management Extensions (JMX) kommuniziert. Sie müssen für jede JVM in Ihrer Konfiguration eine andere Portnummer verwenden. Wenn Sie JMX/RMI verwenden, geben Sie die Option **JMXServicePort** und die Portnummer explizit an, selbst wenn Sie den Standardportwert verwenden möchten. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.**Standardeinstellung:** `1099`

-jvmArgs <JVM-Argumente>
Gibt eine Gruppe von JVM-Argumenten an. Alle Optionen hinter der Option **-jvmArgs** werden verwendet, um die Server-JVM zu starten. Wenn der Parameter **-jvmArgs** verwendet wird, müssen Sie sicherstellen, dass er das letzte optionale Scriptargument ist, das angegeben wird.

Beispiel:-jvmArgs `-Xms256M -Xmx1G`

-listenerHost <Hostname>

Gibt den Hostnamen an, an den der Object Request Broker (ORB) für die Kommunikation mit Internet Inter-ORB Protocol (IIOP) gebunden wird. Der Wert muss ein vollständig qualifizierter Domänenname oder eine IP-Adresse sein. Wenn Ihre Konfiguration mehrere Netzkarten enthält, definieren Sie den Listener-Host und Listener-Port, so dass der Object Request Broker in der JVM die IP-Adresse für die Bindung kennt. Wenn Sie die zu verwendende IP-Adresse nicht kennen, können Symptome wie Überschreitungen des Verbindungszeitlimits, ungewöhnliche API-Fehler und Clients, die zu blockieren scheinen, auftreten.**Standardeinstellung:** `localhost`

-listenerPort <Port>
Gibt die Portnummer an, an die der Object Request Broker (ORB) gebunden wird. Diese Einstellung konfiguriert Container und Clients für die Kommunikation mit dem Katalogservice über den ORB. In WebSphere Application Server wird der Listener-Port von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Diese Eigenschaft gilt für den Container-Server und den Katalogservice.**Standardeinstellung:** `2809`

-objectgridFile <ObjectGrid-XML-Deskriptordatei>
Gibt den Pfad der ObjectGrid-Deskriptordatei an. Die ObjectGrid-XML-Datei gibt an, welche eXtreme-Scale-Server der Container hostet.

-objectgridUrl <URL_des_ObjectGrid-Deskriptors>
Gibt einen URL für die ObjectGrid-Deskriptordatei an. Die ObjectGrid-XML-Datei gibt an, welche eXtreme-Scale-Server der Container hostet.

-script <Scriptdatei>
Gibt die Position eines angepasstesn Script für Befehle an, die Sie angeben, um Katalogserver oder -container zu starten und dann wie gewünscht zu parametrisieren oder zu bearbeiten.

-serverProps <Servereigenschaftendatei>
Gibt den Pfad der Servereigenschaftendatei an.

Beispiel:`../security/server.props`

-timeout <Sekunden>
Gibt das Zeitlimit (in Sekunden) für den Serverstart an.

-traceFile <Tracedatei>
Gibt den Pfad der Datei an, in der Traceinformationen gespeichert werden.

Beispiel: `../logs/c4Trace.log`

-traceSpec <Tracespezifikation>
Gibt eine Zeichenfolge an, die den Umfang des Trace angibt, der beim Serverstart aktiviert wird.

Beispiel:

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-zone <Zonenname>
Gibt die für alle Container im Server zu verwendende Zone an. Weitere Informationen zum Konfigurieren von Zonen finden Sie in „Routing an bevorzugte Zonen" auf Seite 250 den Informationen zu Zonen in der Veröffentlichung *Produktübersicht*.

Eigenständige Server stoppen

Sie können das Script `stopOgServer` verwenden, um eXtreme-Scale-Serverprozesse zu stoppen.

Informationen zu diesem Vorgang

Führen Sie das Script **stopOgServer** aus, indem Sie zum Verzeichnis `bin` navigieren:

```
cd WXS-Installationsstammverzeichnis/bin
```

Vorgehensweise

- **Einzelnen Container-Server stoppen.**

  Führen Sie das Script **stopOgServer** aus, um den Container-Server zu stoppen. Verwenden Sie diesen Befehl nur, wenn Sie einen einzelnen Container-Server stoppen. Wenn Sie den Stoppbefehl für einen einzelnen Katalogserver für mehrere Container-Server nacheinander ausführen, können Sie Leistungs- und Fluktuationsprobleme bei der Shard-Verteilung beobachten.

  ```
  stopOgServer Container-Server -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Achtung: Die Option **-catalogServiceEndPoints** muss mit dem Wert der Option **-catalogServiceEndPoints** übereinstimmen, die zum Starten des Containers verwendet wird. Wenn die Option **-catalogServiceEndPoints** nicht zum Starten des Containers verwendet wird, werden wahrscheinlich die Standardwerte "localhost" oder der Hostname und ORB-Port 2809 verwendet, um eine Verbindung zum Katalogservice herzustellen. Andernfalls verwenden Sie die an **-listenerHost** und **-listenerPort** im Katalogservice übergebenen Werte. Wenn die Optionen **-listenerHost** und **-listenerPort** beim Starten des Katalogservice nicht verwendet werden, stellt der ORB eine Bindung zu Port 2809 auf dem lokalen Host für den Katalogservice her.

- **Mehrere Container-Server stoppen.**

  Um Fluktuations- und Leistungsprobleme bei der Shard-Verteilung zu vermeiden, wenn Sie mehrere Container-Server gleichzeitig stoppen möchten, verwenden Sie das folgende Befehlsformat. Trennen Sie einzelnen Container-Server durch Kommas voneinander:

  ```
  stopOgServer containerServer0,containerServer1,containerServer2
  -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Wenn Sie alle Container in einer bestimmten Zone oder auf einem bestimmten Host stoppen möchten, können Sie den Parameter **-teardown** verwenden. Weitere Informationen finden Sie unter „Server mit dem Dienstprogramm **xscmd** normal stoppen" auf Seite 417.

- **Katalogserver stoppen.**

  Führen Sie das Script **stopOgServer** aus, um den Katalogserver zu stoppen.

  ```
  stopOgServer.sh Katalogserver -catalogServiceEndPoints MyServer1.company.com:2809
  ```

Achtung: Wenn Sie einen Katalogservice stoppen, verwenden Sie die Option **-catalogServiceEndPoints**, um auf den Host und den Port des Object Request Broker (ORB) im Katalogservice zu verweisen. Der Katalogservice verwendet die Optionen **-listenerHost** und **-listenerPort**, um den Host und den Port für die ORB-Bindung anzugeben, oder er akzeptiert die Standardbindung. Wenn die Optionen **-listenerHost** und **-listenerPort** beim Starten des Katalogservice nicht verwendet werden, stellt der ORB eine Bindung zu Port 2809 auf dem lokalen Host für den Katalogservice her. Beim Stoppen eines Katalogservice wird eine andere Option **-catalogServiceEndPoints** verwendet als beim Starten des Katalogservice.

Für das Starten eines Katalogservice sind Peerzugriffsports und Clientzugriffsports erforderlich, wenn die Standardports nicht verwendet werden. Für das Stoppen eines Katalogservice wird nur der ORB-Port benötigt.

- **Server der Webkonsole stoppen.** Zum Stoppen des Servers der Webkonsole führen Sie das Script **stopConsoleServer.bat|sh** aus. Das Script befindet sich im Verzeichnis *WXS-Installationsstammverzeichnis*/ObjectGrid/bin Ihrer Installation. Weitere Informationen finden Sie unter „Webkonsole starten und anmelden" auf Seite 451.
- **Traceerstellung für den Serverstoppprozess aktivieren.**

  Wenn ein Container nicht gestoppt werden kann, können Sie die Traceerstellung als Unterstützung für die Fehlerbehebung aktivieren. Zum Aktivieren der Traceerstellung beim Stoppen eines Servers fügen Sie den Stoppbefehlen die Parameter **-traceSpec** und **-traceFile** hinzu. Der Parameter **-traceSpec** gibt den Typ des zu aktivierenden Trace an, und der Parameter **-traceFile** gibt den Pfad und den Namen der für die Tracedaten zu erstellenden und zu verwendenden Datei an.

  1. Navigieren Sie über die Befehlszeile zum Verzeichnis "bin".

     ```
     cd WXS-Installationsstammverzeichnis/bin
     ```

  2. Führen Sie das Script **stopOgServer** mit aktivierter Traceerstellung aus.

     ```
     stopOgServer.sh c4 -catalogServiceEndPoints MyServer1.company.com:2809
      -traceFile ../logs/c4Trace.log -traceSpec ObjectGrid=all=enabled
     ```

     Suchen Sie nach der Traceerstellung nach Fehlern, die sich auf Portkonflikte, fehlende Klassen, fehlende oder ungültige XML-Dateien oder Stack-Traces beziehen. Empfohlene Tracespezifikation für den Start sind:

     – `ObjectGrid=all=enabled`

     – `ObjectGrid*=all=enabled`

     Informationen zu allen Optionen für die Tracespezifikation finden Sie im Abschnitt „Traceoptionen" auf Seite 546.
- **Integrierte Server über das Programm stoppen.**

  Weitere Informationen zum Stoppen integrierter Server über das Programm finde Sie unter „Integrierte Server-API zum Starten und Stoppen von Servern verwenden" auf Seite 418.

Script "stopOgServer"

Das Script **stopOgServer** stoppt Katalog- und Container-Server.

Zweck

Verwenden Sie das Script **stopOgServer**, um einen Server zu stoppen. Sie müssen den Namen des Servers und dessen Katalogserviceendpunkte angeben.

Position

Das Script **stopOgServer** befindet sich im Verzeichnis `bin` des Verzeichnisses `Stammverzeichnis`, z. B.:

```
cd WXS-Installationsstammverzeichnis/bin
```

Syntax

Katalog- oder Container-Server stoppen: Windows

```
stopOgServer.bat <Servername> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [Optionen]
```

UNIX

```
stopOgServer.sh <Servername> -catalogServiceEndPoints
<csHost:csListenerPort,csHost:csListenerPort> [Optionen]
```

Optionen

-catalogServiceEndPoints <csHost:csListenerPort, csHost:csListenerPort...>
Gibt den Host und den Portnummer des Object Request Broker (ORB) an.

Für Container-Server: Die Liste der Katalogserviceendpunkte muss mit der Liste übereinstimmen, die zum Starten des Container-Servers verwendet wurde. Wenn Sie diese Option beim Starten des Container-Servers nicht angegeben haben, verwenden Sie den Standardwert `localhost:2809`.

Für Katalogserver: Wenn Sie den Katalogserver stoppen, verwenden Sie die Werte, die Sie für die Optionen **-listenerHost** und **-listenerPort** beim Starten des Katalogservice angegeben haben. Wenn Sie diese Optionen beim Starten des Katalogservers nicht angegeben haben, verwenden Sie den Standardwert `localhost:2809`. Der Wert von **-catalogServiceEndPoints**, den Sie beim Stoppen des Katalogservice verwenden, ist anders als der Wert, den Sie beim Starten des Katalogservice angegeben haben.

-clientSecurityFile <Sicherheitseigenschaftendatei>
Gibt den Pfad zu der Clienteigenschaftendatei an, die die Sicherheitseigenschaften für den Client definiert. Weitere Informationen zu den Sicherheitseinstellungen in dieser Datei finden Sie unter Clienteigenschaftendatei.

-traceSpec <Tracespezifikation>
Gibt eine Zeichenfolge an, die den Umfang des Trace angibt, der beim Serverstart aktiviert wird.

Beispiel:

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <Tracedatei>
Gibt den Pfad der Datei an, in der Traceinformationen gespeichert werden.

Beispiel: `../logs/c4Trace.log`

-jvmArgs <JVM-Argumente>
Gibt eine Gruppe von JVM-Argumenten an. Alle Optionen hinter der Option **-jvmArgs** werden verwendet, um die Server-JVM zu starten. Wenn der Parameter **-jvmArgs** verwendet wird, müssen Sie sicherstellen, dass er das letzte optionale Scriptargument ist, das angegeben wird.

Beispiel:-jvmArgs `-Xms256M -Xmx1G`

Server mit dem Dienstprogramm xscmd normal stoppen

Sie können das Dienstprogramm **xscmd** mit dem Befehl **-c teardown** verwenden, um eine Liste oder Gruppe von Katalog- und Container-Servern zu stoppen. Dieser Befehl vereinfacht die Beendigung des gesamten Datengrids oder Teilen davon, indem unnötige Verteilungs- und Wiederherstellungsaktionen des Katalogservice vermieden werden, die normalerweise ausgeführt werden, wenn Prozesse gestoppt oder abgebrochen werden.

Vorgehensweise

- Definierte Liste von Servern stoppen.

  Geben Sie mit dem Parameter **-teardown** eine Liste von Servern an:

  ```
  xscmd –c teardown
  ```

- Alle Server in einer bestimmten Zone stoppen.

  Verwenden Sie den Parameter **-z**, und geben Sie den Namen der Zone an. Der Katalogserver bestimmt die Server, die in der Zone aktiv sind, und das Dienstprogramm **xscmd** fordert Sie zur Eingabe einer Liste von Servern in der ausgewählten Zone auf, bevor die Server beendet werden:

  ```
  xscmd –c teardown –z Zonenname
  ```

- Alle Server auf einem bestimmten Host stoppen.

  Verwenden Sie den Parameter **-hf**, und geben Sie den Namen des Hosts an. Wenn Sie beispielsweise alle Server auf dem Host `myhost.mycompany.com` beenden möchten, geben Sie `-hf myhost.mycompany.com` ein. Der Katalogserver bestimmt die Server, die auf dem Host aktiv sind, und das Dienstprogramm **xscmd** fordert Sie zur Eingabe einer Liste von Servern auf dem ausgewählten Host auf, bevor die Server beendet werden:

  ```
  xscmd –teardown –hf <Hostname>
  ```

Server in einer Umgebung von WebSphere Application Server starten und stoppen

Katalog- und Container-Server können in einer Umgebung von WebSphere Application Server oder WebSphere Application Server Network Deployment automatisch gestartet werden.

Vorbereitende Schritte

Konfigurieren Sie Katalogserver und Container-Server für die Ausführung in WebSphere Application Server:

- „Katalogservice in WebSphere Application Server konfigurieren" auf Seite 263
- „Container-Server in WebSphere Application Server konfigurieren" auf Seite 282

Informationen zu diesem Vorgang

Der Lebenszyklus von Katalog- und Container-Servern in WebSphere Application Server ist mit dem Prozess verknüpft, in dem diese Server ausgeführt werden.

Vorgehensweise

- **Katalogservices in WebSphere Application Server** starten:

  Der Lebenszyklus eines Katalogservers ist an den Prozess von WebSphere Application Server gebunden. Nach der Konfiguration der Katalogservicedomäne in WebSphere Application Server starten Sie jeden Server erneut, den Sie für die Katalogservicedomäne definiert haben. Der Katalogservice wird automatisch in

den Servern gestartet, die Sie der Katalogservicedomäne zugeordnet haben. Der Katalogservice kann je nach Edition von WebSphere Application Server auch in den folgenden Szenarien automatisch gestartet werden:

  - **WebSphere Application Server** (Basisedition): Sie können Ihre Anwendung so konfigurieren, dass automatisch ein Container-Server und ein Katalogservice gestartet wird. Dieses Feature vereinfacht die Komponententests in Entwicklungsumgebungen wie Rational Application Developer, weil Sie einen Katalogservice nicht explizit starten müssen. Weitere Informationen finden Sie im Artikel „Anwendungen von WebSphere Application Server für den automatischen Start von Container-Servern konfigurieren" auf Seite 282.
  - **WebSphere Application Server Network Deployment**: Der Katalogservice wird automatisch im Deployment-Manager-Prozess gestartet, wenn WebSphere eXtreme Scale auf dem Deployment-Manager-Knoten installiert ist und das Deployment-Manager-Profil erweitert wurde. Weitere Informationen finden Sie unter „Katalogservice in WebSphere Application Server konfigurieren" auf Seite 263.

- **Container-Server in WebSphere Application Server** starten:

  Der Lebenszyklus eines Container-Servers ist an die Anwendung von WebSphere Application Server gebunden. Wenn Sie die konfigurierte Anwendung starten, werden auch die Container-Server gestartet.

- **Gesamtes Datengrid von Servern stoppen**:

  Sie können Katalog- und Container-Server stoppen, indem Sie die Anwendungen und zugehörigen Anwendungsserver stoppen. Alternativ können Sie das gesamte Datengrid mit dem Dienstprogramm **xscmd** oder mit MBeans stoppen:

  - **Im Dienstprogramm xscmd:**

    Weitere Informationen zum Stoppen eines gesamten Datengrids finden Sie unter „Server mit dem Dienstprogramm **xscmd** normal stoppen" auf Seite 417.

  - **Mit MBeans:**

    Verwenden Sie die Operation tearDownServers in der MBean PlacementServiceMBean.

Integrierte Server-API zum Starten und Stoppen von Servern verwenden

Mit WebSphere eXtreme Scale können Sie eine programmgesteuerte API verwenden, um den Lebenszyklus integrierter Server und Container zu verwalten. Sie können den Server über das Programm mit jeder der Optionen konfigurieren, die Sie auch über die Befehlszeilenoptionen oder dateibasierten Servereigenschaften konfigurieren können. Sie können den integrierten Server als Container-Server und/oder Katalogservice konfigurieren.

Vorbereitende Schritte

Sie müssen eine Methode für die Ausführung von Code über eine bereits vorhandene Java Virtual Machine haben. Die eXtreme-Scale-Klassen müssen über die Baumstruktur der Klassenladeprogramme verfügbar sein.

Informationen zu diesem Vorgang

Viele Verwaltungs-Tasks können über die Verwaltungs-API ausgeführt werden. Die API wird häufig als interner Server für die Speicherung des Webanwendungsstatus eingesetzt. Der Webserver kann als integrierter WebSphere eXtreme Scale-Server gestartet werden, den Container-Server dem Katalogservice melden, und anschlie-

ßend wird der Server als Member eines größeren verteilten Grids hinzugefügt. Diese Verwendung kann aus einem ansonsten flüchtigen Datenspeicher einen skalierbaren und hoch verfügbaren Datenspeicher machen.

Sie können den vollständigen Lebenszyklus eines integrierten eXtreme-Scale-Servers über das Programm steuern. Die Beispiele sind so generisch wie möglich und enthalten nur direkte Codemuster für die beschriebenen Schritte.

Vorgehensweise

1. Rufen Sie das Objekt "ServerProperties" aus der Klasse "ServerFactory" ab, und konfigurieren Sie alle erforderlichen Optionen.

   Jeder eXtreme-Scale-Server besitzt eine Reihe konfigurierbarer Eigenschaften. Wenn ein Server über die Befehlszeile gestartet wird, werden diese Eigenschaften auf Standardwerte gesetzt, aber Sie können mehrere Eigenschaften überschreiben, indem Sie eine externe Quelle oder Datei angeben. Im integrierten Bereich können Sie die Eigenschaften direkt mit einem ServerProperties-Objekt setzen. Sie müssen diese Eigenschaften setzen, bevor Sie eine Serverinstanz aus der Klasse "ServerFactory" abrufen. Das folgende Beispiel-Snippet ruft ein ServerProperties-Objekt ab, setzt das Feld "CatalogServiceBootStrap" und initialisiert mehrere optionale Servereinstellungen. Eine Liste der konfigurierbaren Einstellungen finden Sie in der API-Dokumentation.

   ```
   ServerProperties props = ServerFactory.getServerProperties();
   props.setCatalogServiceBootstrap("host:port");
   // Für Verbindungsherstellung zu einem bestimmtem Katalogserver erforderlich
   props.setServerName("ServerOne"); // Server benennen
   props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // Tracespezifikation festlegen
   ```

2. Wenn der Server ein Katalogserver sein soll, rufen Sie das Objekt "CatalogServerProperties" ab.

   Jeder integrierte Server kann ein Katalogserver und/oder ein Container-Server sein. Der folgende Beispielcode ruft das Objekt "CatalogServerProperties" ab, aktiviert die Katalogserviceoption und konfiguriert verschiedene Einstellungen des Katalogservice.

   ```
   CatalogServerProperties catalogProps = ServerFactory.getCatalogProperties();
   catalogProps.setCatalogServer(true);
   // standardmäßig false; erforderlich für die Einstellung als Katalogservice
   catalogProps.setQuorum(true); // Quorum aktivieren/inaktivieren
   ```

3. Rufen Sie eine Server-Instanz aus der Klasse "ServerFactory" ab. Die Server-Instanz ist ein prozessbezogenes Singleton, das für die Verwaltung der Zugehörigkeiten im Grid zuständig ist. Nach der Instanziierung dieser Instanz ist dieser Prozess verbunden und zusammen mit den anderen Servern im Grid hoch verfügbar. Das folgende Beispiel veranschaulicht, wie die Server-Instanz erstellt wird:

   ```
   Server server = ServerFactory.getInstance();
   ```

   Wenn Sie sich das vorherige Beispiel ansehen, stellen Sie fest, dass die Klasse "ServerFactory" eine statische Methode bereitstellt, die eine Server-Instanz zurückgibt. Die Klasse "ServerFactory" ist die einzige geplante Schnittstelle für das Abrufen einer Server-Instanz. Deshalb stellt die Klasse sicher, dass die Instanz ein Singleton bzw. die einzige Instanz für jede JVM bzw. jedes isolierte Klassenladeprogramm ist. Die Methode "getInstance" initialisiert die Server-Instanz. Sie müssen alle Servereigenschaften konfigurieren, bevor Sie die Instanz initialisieren. Die Klasse "Server" ist für die Erstellung neuer Container-Instanzen zuständig. Sie können die Klassen "ServerFactory" und "Server" verwenden, um den Lebenszyklus der integrierten Serverinstanz zu verwalten.
4. Starten Sie eine Container-Instanz über die Serverinstanz.

Bevor Shards an einen integrierten Server verteilt werden können, müssen Sie einen Container im Server erstellen. Die Schnittstelle "Server" besitzt eine Methode "createContainer", die das Argument "DeploymentPolicy" akzeptiert. Im folgenden Beispiel wird die Serverinstanz, die Sie zum Erstellen eines Containers abgerufen haben, mit einer erstellten DeploymentPolicy-Datei verwendet. Beachten Sie, dass Container für die Serialisierung ein Klassenladeprogramm erfordern, dem die Binärdateien der Anwendung zur Verfügung stehen. Sie können diese Binärdateien bereitstellen, indem Sie die Methode "createContainer" aufrufen und in diesem Aufruf das Klassenladeprogramm für den Thread-Kontext auf das Klassenladeprogramm setzen, das Sie verwenden möchten.

```
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(new
   URL("file://urltodeployment.xml"),
 new URL("file://urltoobjectgrid.xml"));
Container container = server.createContainer(policy);
```

5. Entfernen und bereinigen Sie einen Container.

   Sie können einen Container-Server entfernen und bereinigen, indem Sie die Methode "teardown" für die abgerufene Containerinstanz ausführen. Bei der Ausführung der Methode "teardown" für einen Container wird der Container bereinigt und aus dem integrierten Server entfernt.

   Die Bereinigung des Containers beinhaltet die Verlagerung und Umrüstung aller Shards dieses Containers. Jeder Server kann mehrere Container und Shards enthalten. Die Bereinigung eines Containers hat keine Auswirkung auf den Lebenszyklus der übergeordneten Server-Instanz. Das folgende Beispiel veranschaulicht, wie die Methode "teardown" für einen Server ausgeführt wird. Die Methode "teardown" wird über die Schnittstelle "ContainerMBean" bereitgestellt. Wenn Sie keinen Zugriff mehr über das Programm auf diesen Container haben und die Schnittstelle "ContainerMBean" verwenden, können Sie den Container mit der zugehörigen MBean trotzdem bereinigen. Die Schnittstelle "Container" enthält auch eine Methode "terminate". Verwenden Sie diese Methode nur, wenn es unbedingt erforderlich ist. Diese Methode ist konsequenter und koordiniert die entsprechende Shard-Verlagerung und -Bereinigung nicht.

   ```
   container.teardown();
   ```

6. Stoppen Sie den integrierten Server.

   Wenn Sie einen integrierten Server stoppen, stoppen Sie auch alle Container und Shards, die im Server ausgeführt werden. Wenn Sie einen integrierten Server stoppen, müssen Sie alle offenen Verbindungen bereinigen und alle Shards verlagern oder umrüsten. Das folgende Beispiel veranschaulicht, wie ein Server gestoppt wird und die Methode "waitFor" in der Schnittstelle "Server" verwendet wird, um sicherzustellen, dass die Serverinstanz vollständig beendet wird. Ähnlich wie im Containerbeispiel wird die Methode "stopServer" über die Schnittstelle "ServerMBean" bereitgestellt. Mit dieser Schnittstelle können Sie einen Server über die entsprechende Managed Bean (MBean) stoppen.

   ```
   ServerFactory.stopServer();  // Factory zum Beenden des Server-Singletons
    // oder
   server.stopServer(); // Serverinstanz direkt verwenden
   server.waitFor();
   // Kehrt zurück, wenn die Beendigungsprozedur für den Server ordnungsgemäß abgeschlossen wurde
   ```

   Vollständiges Codebeispiel:

   ```
    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"); // Server benennen
            props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // TraceSpec

            /*
             * In den meisten Fällen dient der Server nur als Container-Server und
             * stellt eine Verbindung zu einem externen Katalogserver her. Auf diese
             * Weise wird eine höhere Verfügbarkeit erreicht. Der folgende, auf Kommentar
             * gesetze Codeauszug aktiviert diesen Server als Katalogserver.
             *
             *
             * CatalogServerProperties catalogProps =
             * ServerFactory.getCatalogProperties();
             * catalogProps.setCatalogServer(true); // Katalogservice aktivieren
             * catalogProps.setQuorum(true); // Quorum aktivieren
             */

            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);

            /*
             * Das Shard wird jetzt an diesen Container verteilt, wenn die
             * Implementierungsanforderungen erfüllt sind.
             * Dies umfasst die Erstellung des integrierten Servers und des Containers.
             *
             * Die folgenden Zeilen demonstrieren lediglich den Aufruf der Bereinigungsmethoden.
             */

            container.teardown();
            server.stopServer();
            int success = server.waitFor();

        } catch (ObjectGridException e) {
            // Container konnte nicht initialisiert werden.
        } catch (MalformedURLException e2) {
            // Ungültiger URL für XML-Datei(en)
        }

    }

}
```

Integrierte Server-API

WebSphere eXtreme Scale enthält Anwendungsprogrammierschnittstellen (APIs) und Systemprogrammierschnittstellen für die Integration von eXtreme-Scale-Servern und -Clients in vorhandenen Java-Anwendungen. Im folgenden Abschnitt werden die verfügbaren integrierten Server-APIs beschrieben.

eXtreme-Scale-Server instanziieren

Sie können verschiedene Eigenschaften verwenden, um die eXtreme-Scale-Serverinstanz zu konfigurieren, die Sie mit der Methode "ServerFactory.getServerProperties" abrufen können. Das ServerProperties-Objekt ist das Singleton, und deshalb ruft jeder Aufruf der Methode "getServerProperties" dieselbe Instanz ab.

Sie können einen neuen Server mit dem folgenden Code erstellen.

```
Server server = ServerFactory.getInstance();
```

Alle Eigenschaften, die vor ersten Aufruf von "getInstance" gesetzt werden, werden zum Initialisieren des Servers verwendet.

Servereigenschaften abrufen

Sie können die Servereigenschaften festlegen, bis die Methode "ServerFactory.getInstance" zum ersten Mal aufgerufen wird. Der erste Aufruf der Methode "getInstance" instanziiert den eXtreme-Scale-Server und liest alle konfigurierten Eigenschaften. Das Festlegen von Eigenschaften nach der Erstellung der Instanz hat keine Auswirkung. Das folgende Beispiel zeigt, wie Eigenschaften vor der Instanzierung einer Server-Instanz definiert werden.

```
// Servereigenschaften abrufen, die diesem Prozess zugeordnet sind.
ServerProperties serverProperties = ServerFactory.getServerProperties();

// Servernamen für diesen Prozess festlegen.
serverProperties.setServerName("EmbeddedServerA");

// Namen der Zone festlegen, in der sich dieser Prozess befindet.
serverProperties.setZoneName("EmbeddedZone1");

// Erforderliche Endpunktinformationen zum Booten des Katalogservice festlegen.
serverProperties.setCatalogServiceBootstrap("localhost:2809");

// Hostnamen des ORB-Listeners festlegen, zu dem die Bindung hergestellt werden soll.
serverProperties.setListenerHost("host.local.domain");

// ORB-Listener-Port festlegen, zu dem die Bindung hergestellt werden soll.
serverProperties.setListenerPort(9010);

// Alle MBeans für diesen Prozess inaktivieren.
serverProperties.setMBeansEnabled(false);

Server server = ServerFactory.getInstance();
```

Integrierter Katalogservice

Jede JVM, die von der Methode "CatalogServerProperties.setCatalogServer" markiert wird, kann den Katalogservice für eXtreme Scale ausführen. Diese Methode weist die Laufzeitumgebung des eXtreme-Scale-Servers an, den Katalogservice zu instanziieren, wenn der Server gestartet wird. Der folgende Code veranschaulicht, wie der Katalogserver von eXtreme Scale instanziiert wird:

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
```

eXtreme-Scale-Container integrieren

Setzen Sie die Methode "Server.createContainer" für jede JVM ab, die mehrere eXtreme-Scale-Container ausführen soll. Der folgende Code veranschaulicht, wie ein eXtreme-Scale-Container instanziiert wird:

```
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);
```

Eigenständiger Serverprozess

Sie können alle Services gemeinsam starten, was für die Entwicklung hilfreich und auch in einer Produktionsumgebung praktisch ist. Wenn die Services gemeinsam gestartet werden, führt ein einziger Prozess alle folgenden Aufgaben aus: Er startet

den Katalogservice, er startet eine Gruppe von Containern und er führt die Clientverbindungslogik aus. Mit dieser Art des Servicestarts können Programmierungsprobleme festgestellt werden, bevor die Services in einer verteilten Umgebung implementiert werden. Der folgende Code veranschaulicht, wie ein eigenständiger eXtreme-Scale-Server instanziiert wird:

```
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);
```

eXtreme Scale in WebSphere Application Server integrieren

Die Konfiguration für eXtreme Scale wird automatisch vorgenommen, wenn Sie eXtreme Scale in einer Umgebung mit WebSphere Application Server installieren. Sie müssen keine Eigenschaften festlegen, bevor Sie auf den Server zugreifen, um einen Container zu erstellen. Der folgende Code veranschaulicht, wie Sie einen eXtreme-Scale-Server in WebSphere Application Server instanziieren:

```
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);
```

Ein schrittweises Beispiel für das Starten eines integrierten Katalogservice und -containers über das Programm finden Sie unter „Integrierte Server-API zum Starten und Stoppen von Servern verwenden” auf Seite 418.

Verwaltung mit dem Dienstprogramm `xscmd`

Mit **xscmd** können Sie Verwaltungsaufgaben wie die folgenden in der Umgebung ausführen: Multimasterreplikationslinks konfigurieren, Quorum überschreiben und Gruppen von Servern mit dem Befehl "teardown" stoppen.

Vorbereitende Schritte

- Ihre Katalogserver und Container-Server müssen gestartet sein. Wenn Ihre Katalogserver in einer Katalogservicedomäne enthalten sind, müssen mindestens zwei Katalogserver gestartet sein.
- Vergewissern Sie sich, dass die Umgebungsvariable *JAVA_HOME* so gesetzt ist, dass die mit dem Produkte installierte Laufzeitumgebung verwendet wird. Wenn Sie die Testversion des Produkts verwenden, müssen Sie die Umgebungsvariable *JAVA_HOME* setzen.

Informationen zu diesem Vorgang

Das Dienstprogramm **xscmd** ersetzt das Beispieldienstprogramm **xsadmin** als vollständig unterstütztes Überwachungs- und Verwaltungstool. Sie können ähnliche Operationen mit dem Tool **xsadmin** ausführen, aber dieses Tool wird nicht unterstützt. Das Beispieldienstprogramm **xsadmin** stellt eine Methode für die Syntaxanalyse und die Erkennung aktuelle Implementierungsdaten bereit und kann als Grundlage für das Schreiben angepasster Dienstprogramme verwendet werden. Wenn Sie das Tool **xsadmin** zuvor für die Überwachung und Verwaltung verwendet haben, können Sie Ihre Scripts für die Verwendung des Dienstprogramms **xscmd**

aktualisieren. Informationen zum Zuordnen der **xsadmin**-Befehle zu den neuen **xscmd**-Befehlen finden Sie unter „Tool **xsadmin** auf das Tool **xscmd** migrieren" auf Seite 222.

Vorgehensweise

1. Öffnen Sie ein Befehlszeilenfenster. Setzen Sie in der Befehlszeile die entsprechenden Umgebungsvariablen.
   a. Setzen Sie die Umgebungsvariable *CLIENT_AUTH_LIB*:
      - Windows `set CLIENT_AUTH_LIB=<Pfad_zu_Sicherheits-JARs_oder_Klassen>`
      - UNIX `set CLIENT_AUTH_LIB=<Pfad_zu_Sicherheits-JARs_oder_Klassen> export CLIENT_AUTH_LIB`
2. Wechseln Sie in das Verzeichnis *WXS-Ausgangsverzeichnis*/bin.
   `cd WXS-Ausgangsverzeichnis/bin`
3. Zeigen Sie die Hilfe für die verschiedenen **xscmd**-Optionen an.
   - Führen Sie den folgenden Befehl aus, um die allgemeine Hilfe anzuzeigen:
     - UNIX `./xscmd.sh -h`
     - Windows `xscmd.bat -h`
   - Führen Sie den folgenden Befehl aus, um eine Liste aller Befehle anzuzeigen:
     - UNIX `./xscmd.sh -lc`
     - Windows `xscmd.bat -lc`
   - Führen Sie den folgenden Befehl aus, um die Hilfe für einen bestimmten Befehl anzuzeigen:
     - UNIX `./xscmd.sh -h Befehlsname`
     - Windows `xscmd.bat -h Befehlsname`
   - Führen Sie den folgenden Befehl aus, um eine Liste der Befehlsgruppen anzuzeigen:
     - UNIX `./xscmd.sh -lcg`
     - Windows `xscmd.bat -lcg`
   - Führen Sie den folgenden Befehl aus, um eine Liste der Befehle in einer Befehlsgruppe anzuzeigen:
     - UNIX `./xscmd.sh -lc Name_der_Befehlsgruppe`
     - Windows `xscmd.bat -lc Name_der_Befehlsgruppe`
4. Führen Sie Befehle aus, die Verbindungen zu bestimmten Katalogservern herstellen. Standardmäßig stellt der Befehl **xscmd** eine Verbindung zum Katalogserver auf dem lokalen Host unter Verwendung der Hostnamen/Port-Kombination `localhost:2809` her. Sie können auch eine Liste mit Hostnamen und Ports mit dem Befehl angeben, um Verbindungen zu Katalogservern auf anderen Hosts herzustellen. Das Dienstprogramm **xscmd** stellt eine Verbindung zu einem Host aus der Liste her. Die Hosts, die Sie angeben, müssen in derselben Katalogservicedomäne enthalten sein.
   - Geben Sie eine Liste eigenständiger Katalogserver an, zu denen eine Verbindung hergestellt werden soll:
     - UNIX `./xscmd.sh -c <Befehlsname> -cep Hostname:Port(,Hostname:Port)`

– Windows `xscmd.bat -c <Befehlsname> -cep Hostname:Port(,Hostname:Port)`

In den vorherigen Befehlen steht *Befehlsname* für den Namen des Befehls, den Sie ausführen. *Hostname:Port* steht für den Hostnamen und den Listener-Port des Katalogservers. Der Listener-Port-Wert in einem eigenständigen Katalogserver wird bei der Ausführung des Befehls **startOgServer** angegeben.

- Geben Sie eine Liste von Katalogservern von WebSphere Application Server an, zu denen eine verbindung hergestellt werden soll. Sie können keine Verbindung zu Katalogservern herstellen, die in WebSphere Application Server mit dem Standardwert "localhost" ausgeführt werden:

  – UNIX `./xscmd.sh -c <Befehlsname> -cep WAS-Hostname:Port(,Hostname:Port)`

  – Windows `xscmd.bat -c <efehlsname> -cep WAS-Hostname:Port(,Hostname:Port)`

  In den vorherigen Befehlen steht *Befehlsname* für den Namen des Befehls, den Sie ausführen. *WAS-Hostname* steht für den Hostnamen des Katalogservers in der Zelle von WebSphere Application Server. *Port* steht für den Listener-Port. Der Listener-Port-Wert in WebSphere Application Server wird von der Konfiguration des BOOTSTRAP_ADDRESS-Ports übernommen. Der Standardwert ist `9809`, wenn der Katalogserver im Deployment Manager ausgeführt wird. Wenn Sie den Katalogserver in einem Anwendungsserver ausführen, überprüfen Sie die Konfiguration des BOOTSTRAP_ADDRESS-Ports des Anwendungsservers, um die Portnummer zu bestimmen.

Server von eXtreme Scale mit dem Eclipse-Equinox-OSGi-Framework starten

Container-Server von WebSphere eXtreme Scale können mit verschiedenen Methoden in einem Eclipse-Equinox-OSGi-Framework gestartet werden.

Vorbereitende Schritte

Bevor Sie einen eXtreme-Scale-Container starten können, müssen Sie die folgenden Aufgaben ausgeführt haben:

1. Sie haben das Server-Bundle von WebSphere eXtreme Scale in Eclipse Equinox installiert.
2. Sie haben Ihre Anwendung als OSGi-Bundle gepackt.
3. Sie haben Ihre Plug-ins von WebSphere eXtreme Scale (sofern vorhanden) als OSGi-Bundle gepackt. Die Plug-ins können in dasselbe Bundle wie die Anwendung oder als separate Bundles gepackt werden.

Informationen zu diesem Vorgang

In dieser Aufgabe wird beschrieben, wie Sie einen eXtreme-Scale-Container-Server in einem Eclipse-Equinox-OSGi-Framework starten. Sie können jede der folgenden Methoden verwenden, um Container-Server mit der Eclipse-Equinox-Implementierung zu starten:

- OSGi-Blueprint-Service

  Sie können alle Konfigurations- und Metadaten in ein OSGi-Bundle einschließen. Sehen Sie sich die folgende Abbildung an, um sich mit dem Eclipse-Equinox-Prozess für diese Methode vertraut zu machen:

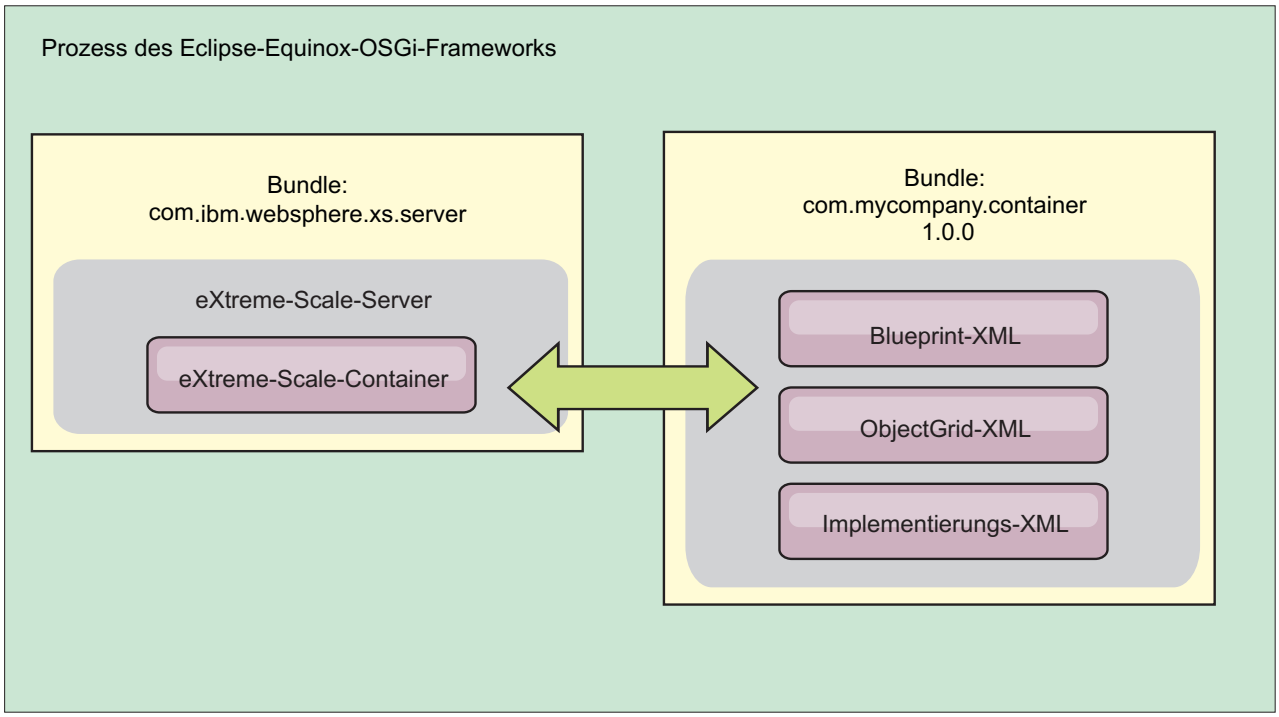


Abbildung 49. Eclipse-Equinox-Prozess für den Einschluss aller Konfigurations- und Metadaten in ein OSGi-Bundle

- Verwaltungsservice für OSGi-Konfiguration

  Sie können Konfigurations- und Metadaten außerhalb eines OSGi-Bundles angeben. Sehen Sie sich die folgende Abbildung an, um sich mit dem Eclipse-Equinox-Prozess für diese Methode vertraut zu machen:

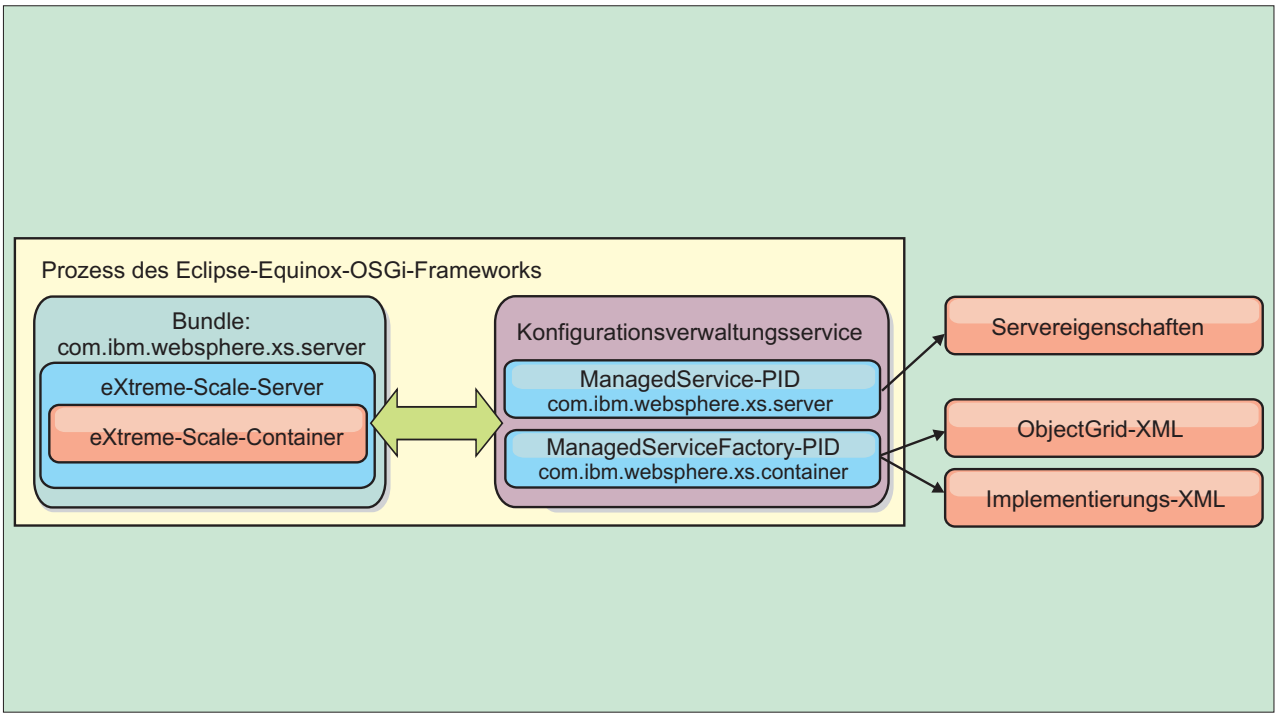


Abbildung 50. Eclipse-Equinox-Prozess für die Angabe von Konfigurations- und Metadaten außerhalb eines OSGi-Bundles

- Über das Programm
  Unterstützt angepasste Konfigurationslösungen.

In jedem Fall werden ein Server-Singleton von eXtreme Scale und mindestens ein Container konfiguriert.

Das Server-Bundle von eXtreme Scale, `objectgrid.jar`, enthält alle erforderlichen Bibliotheken zum Starten und Ausführen eines Grid-Containers von eXtreme Scale in einem OSGi-Framework. Die Laufzeitumgebung des Servers kommuniziert mit benutzerdefinierten Plug-ins und Datenobjekten über den OSGi-Service-Manager.

Wichtig: Nach dem Start eines Server-Bundles von eXtreme Scale und der Initialisierung des Servers von eXtreme Scale kann der Server nicht erneut gestartet werden. Zum erneuten Starten des Servers von eXtreme Scale muss der Eclipse-Equinox-Prozess erneut gestartet werden.

Sie können die Unterstützung für Spring-Namespaces von eXtreme Scale verwenden, um Container-Server von eXtreme Scale in einer Blueprint-XML-Datei zu konfigurieren. Wenn die XML-Elemente für Server und Container der Blueprint-XML-Datei hinzugefügt werden, startet der Namespace-Handler von eXtreme Scale automatisch einen Container-Server, wobei er die Parameter verwendet, die in der Blueprint-XML-Datei beim Start des Bundles definiert sind. Der Handler stoppt den Container, wenn das Bundle gestoppt wird.

Gehen Sie zum Konfigurieren von eXtreme-Scale-Container-Server mit der Blueprint-XML wie folgt vor:

Vorgehensweise

- Container-Server von eXtreme Scale mit OSGi-Blueprint starten.
  1. Erstellen Sie ein Container-Bundle.
  2. Installieren Sie das Container-Bundle im Eclipse-Equinox-OSGi-Framework. Weitere Informationen hierzu finden Sie im Artikel „OSGi-fähige Plug-ins installieren und starten”.
  3. Starten Sie das Container-Bundle.
- Container-Server von eXtreme Scale mit der OSGi-Konfigurationsverwaltung starten.
  1. Konfigurieren Sie den Server und den Container mit der Konfigurationsverwaltung.
  2. Wenn das Server-Bundle von eXtreme Scale gestartet wird oder die persistenten IDs mit der Konfigurationsverwaltung erstellt werden, werden Server und Container automatisch gestartet.
- Container-Server von eXtreme Scale mit der API "ServerFactory" starten. Weitere Informationen hierzu finden Sie in der Dokumentation der Server-APIs.
  1. Erstellen Sie eine Aktivatorklasse für OSGi-Bundles, und verwenden Sie die API "ServerFactory" von eXtreme Scale, um einen Server zu starten.

OSGi-fähige Plug-ins installieren und starten

In dieser Aufgabe installieren Sie das Bundle mit den dynamischen Plug-ins im OSGi-Framework. Anschließend starten Sie das Plug-in.

Vorbereitende Schritte

In diesem Artikel wird davon ausgegangen, dass die folgenden Aufgaben ausgeführt wurden:

- Sie haben das Server- oder Client-Bundle von eXtreme Scale im Eclipse-Equinox-OSGi-Framework installiert. Weitere Informationen finden Sie im Artikel „eXtreme-Scale-Bundles installieren” auf Seite 210.
- Sie haben mindestens ein dynamisches BackingMap- oder ObjectGrid-Plug-in implementiert. Weitere Informationen finden Sie im Artikel Dynamische eXtreme-Scale-Plug-ins erstellen.
- Sie haben die dynamischen Plug-ins als OSGi-Services in OSGi-Bundles gepackt.

Informationen zu diesem Vorgang

In dieser Aufgabe wird beschrieben, wie Sie das Bundle über die Eclipse-Equinox-Konsole installieren. Das Bundle kann mit verschiedenen Methoden installiert werden, z. B. durch Ändern der Konfigurationsdatei `config.ini`. Produkte, in denen Eclipse Equinox integriert ist, haben alternative Methoden für die Verwaltung von Bundles. Informationen zum Hinzufügen von Bundles in der Datei `config.ini` in Eclipse Equinox finden Sie unter Eclipse runtime options.

OSGi ermöglcht das Starten von Bundles, die doppelten Services haben. WebSphere eXtreme Scale verwendet das aktuellste Service-Ranking. Wenn Sie mehrere OSGi-Frameworks in einem eXtreme-Scale-Datengrid starten, müssen Sie sicherstellen, dass die richtigen Service-Rankings in jedem Server gestartet werden. Wenn Sie dies nicht tun, wird das Grid mit verschiedenen Versionen gestartet.

Um festzustellen, welche Versionen vom Datengrid verwendet werden, überprüfen Sie mit dem Dienstprogramm "xscmd" die aktuellen und verfügbaren Rankings.

Weitere Informationen zu den verfügbaren Service-Rankings finden Sie unter „OSGi-Services für eXtreme-Scale-Plug-ins mit **xscmd** aktualisieren” auf Seite 433.

Vorgehensweise

Plug-in-Bundle über die OSGi-Konsole im Eclipse-Equinox-OSGi-Framework installieren.

1. Starten Sie das Eclipse-Equinox-Framework mit aktivierter Konsole, z. B.:
   ```
   <java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
2. Installieren Sie das Plug-in-Bundle in der Equinox-Konsole.
   ```
   osgi> install file:///<Pfad_zum_Bundle>
   ```

   Equinox zeigt die Bundle-ID für das neu installierte Bundle an:
   ```
   Bundle id is 17
   ```
3. Geben Sie die folgende Zeile ein, um das Bundle in der Equinox-Konsole zu starten, wobei <ID> für die Bundle-ID steht, die dem Bundle bei der Installation zugeordnet wurde:
   ```
   osgi>  install <ID>
   ```
4. Rufen Sie den Servicestatus in der Equinox-Konsole ab, um sicherzustellen, dass das Bundle gestartet wurde:
   ```
   osgi> ss
   ```

   Wenn das Bundle erfolgreich gestartet wurde, wird der Status ACTIVE für das Bundle angezeigt, z. B.:
   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Plug-in-Bundle über die Datei "config.ini" im Eclipse-Equonix-OSGi-Framework installieren.

5. Kopieren Sie das Plug-in-Bundle in das Eclipse-Equinox-Plug-in-Verzeichnis, z. B.:
   ```
   <equinox_root>/plugins
   ```
6. Bearbeiten Sie die Eclipse-Equinox-Konfigurationsdatei `config.ini`, und fügen Sie das Bundle der Eigenschaft "osgi.bundles" hinzu, z. B.:
   ```
   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
   ```

   Wichtig: Vergewissern Sie sich, dass dem letzten Bundlenamen eine leere Zeile folgt. Jedes Bundle wird durch ein Komma abgetrennt.
7. Starten Sie das Eclipse-Equinox-Framework mit aktivierter Konsole, z. B.:
   ```
   <java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```
8. Rufen Sie den Servicestatus in der Equinox-Konsole ab, um sicherzustellen, dass das Bundle gestartet wurde, z. B.:
   ```
   osgi> ss
   ```

   Wenn das Bundle erfolgreich gestartet wurde, wird der Status ACTIVE für das Bundle angezeigt, z. B.:
   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Ergebnisse

Das Plug-in-Bundle ist jetzt installiert und gestartet. Der Container oder Client von eXtreme Scale kann jetzt gestartet werden. Weitere Informationen zum Entwickeln von eXtreme-Scale-Plug-ins finden Sie im Artikel System-APIs und Plug-ins.

OSGi-fähige Services mit dem Dienstprogramm `xscmd` verwalten

Sie können das Dienstprogramm **xscmd** verwenden, um Verwaltungsaufgaben wie das Anzeigen von Services und Rankings, die von jedem Container verwendet werden, und die Aktualisierung der Laufzeitumgebung zur Verwendung neuer Versionen der Bundles auszuführen.

Informationen zu diesem Vorgang

Mit dem Eclipse-Equinox-OSGi-Framework können Sie mehrere Versionen desselben Bundles installieren und diese Bundles zur Laufzeit aktualisieren. WebSphere eXtreme Scale ist eine verteilte Umgebung, in der die Container-Server in vielen OSGi-Framework-Instanzen ausgeführt werden.

Administratoren sind für das manuelle Kopieren, Installieren und Starten von Bundles im OSGi-Framework verantwortlich. eXtreme Scale enthält eine OSGi-Schnittstelle ServiceTrackerCustomizer, um Services zu überwachen, die als Plug-ins von eXtreme Scale in der ObjectGrid-XML-Deskriptordatei angegeben wurden. Verwenden Sie das Dienstprogramm **xscmd**, um festzustellen, welche Version des Plug-ins verwendet wird und welche Versionen verwendet werden können, und um Bundle-Upgrades durchzuführen.

eXtreme Scale verwendet die Service-Ranking-Nummer, um die Version jedes Service anzugeben. Wenn zwei oder mehr Services mit derselben Referenz geladen werden, verwendet eXtreme Scale automatisch den Service mit dem höchsten Ranking.

Vorgehensweise

- Führen Sie den Befehl **osgiCurrent** aus, und vergewissern Sie sich, dass jeder Server von eXtreme Scale das richtige Plug-in-Service-Ranking verwendet.

  Da eXtreme Scale automatisch die Servicereferenz mit dem höchsten Ranking auswählt, ist es möglich, dass das Datengrid mit mehreren Rankings eines Plug-in-Service gestartet wird.

  Wenn der Befehl eine Diskrepanz bei den Rankings feststellt oder einen Service nicht findet, wird eine Fehlerkategorie ungleich null gesetzt. Wird der Befehl erfolgreich ausgeführt, wird die Fehlerkategorie auf 0 gesetzt.

  Im folgenden Beispiel sehen Sie die Ausgabe des Befehls **osgiCurrent**, wenn zwei Plug-ins in demselben Grid mit vier Servern installiert sind. Das Plug-in loaderPlugin verwendet Ranking 1, und txCallbackPlugin verwendet Ranking 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
```

Im folgenden Beispiel sehen Sie die Ausgabe des Befehls **osgiCurrent**, wenn server2 mit einem neueren Ranking von loaderPlugin gestartet wurde:

```
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
```

- Führen Sie den Befehl **osgiAll** aus, um sicherzustellen, dass die Plug-in-Services in jedem Container-Server von eXtreme Scale ordnungsgemäß gestartet wurden.

  Wenn Bundles gestartet werden, die Services enthalten, auf die eine ObjectGrid-Konfiguration verweist, überwacht die eXtreme-Scale-Laufzeitumgebung das Plug-in automatisch, verwendet es aber nicht sofort. Der Befehl **osgiAll** zeigt an, welche Plug-ins für jeden Server verfügbar sind.

  Wenn Sie den Befehl ohne Parameter ausführen, werden alle Services für alle Grids und Server angezeigt. Es können zusätzliche Filter, einschließlich des Filters "**-serviceName** <Servicename>, angegeben werden, um die Ausgabe auf einen einzigen Service oder einen Teil des Datengrids zu beschränken.

  Im folgenden Beispiel sehen Sie die Ausgabe des Befehls **osgiAll**, wenn zwei Plug-ins in zwei Servern gestartet werden. Im Plug-in loaderPlugin sind beide Rankings (1 und 2) gestartet, und im Plug-in txCallbackPlugin ist nur Ranking 1 gestartet. Die Übersichtsnachricht am Ende der Ausgabe bestätigt, dass beide Server dieselben Service-Rankings sehen:

```
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.
```

  Im folgenden Beispiel sehen Sie die Ausgabe des Befehls **osgiAll**, wenn das Bundle, das loaderPlugin mit Ranking 1 1 enthält, in server1 gestoppt wird. Die Übersichtsnachricht am Ende der Ausgabe bestätigt, dass loaderPlugin mit Ranking 1 jetzt in server1 fehlt.

```
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
```

Im folgenden Beispiel sehen Sie die Ausgabe des Befehls, wenn der Servicename mit dem Argument **-sn** angegeben wird, aber der Service nicht vorhanden ist:

```
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.
```

- Führen Sie den Befehl **osgiCheck** aus, um zu prüfen, ob Gruppen von Plug-in-Services und -Rankings verfügbar sind.

  Der Befehl **osgiCheck** akzeptiert eine oder mehrere Gruppen von Service-Rankings im folgenden Format: `-serviceRankings <Servicename>;<Ranking>[,<Servicename>;<Ranking>]`

  Wenn alle Rankings verfügbar sind, kehrt die Methode mit der Fehlerkategorie 0 zurück. Ist mindestens ein Ranking nicht verfügbar, werden eine Fehlerkategorie ungleich 0 und eine Tabelle mit allen Servern ausgegeben, die die angegebenen Service-Rankings nicht enthalten. Es können zusätzliche Filter verwendet werden, um die Serviceprüfung auf einen Teil der verfügbaren Server in der eXtreme-Scale-Domäne zu beschränken.

  Wenn beispielsweise das angegebene Ranking oder der angegebene Service fehlt, wird die folgende Nachricht angezeigt:

  ```
  Server  OSGi Service Unavailable Rankings
  ------  ------------ --------------------
  server1 loaderPlugin 3
  server2 loaderPlugin 3
  ```

- Führen Sie den Befehl **osgiUpdate** aus, um das Ranking eines oder mehrerer Plug-ins für alle Server in einem einzelnen ObjectGrid und MapSet in einer einzigen Operation zu aktualisieren.

  Der Befehl akzeptiert eine oder mehrere Gruppen von Service-Rankings im folgenden Format: `-serviceRankings <Servicename>;<Ranking>[,<Servicename>;<Ranking>] -g <Gridname> -ms <MapSet-Name>`

  Mit diesem Befehl können Sie die folgenden Operationen ausführen:

  – Vergewissern Sie sich, dass die angegebenen Services zur Aktualisierung in allen Servern verfügbar sind.
  – Ändern Sie den Status des Grids mit der Schnittstelle StateManager in "offline". Weitere Informationen finden Sie unter „ObjectGrid-Verfügbarkeit verwalten” auf Seite 437. Dieser Prozess legt das Grid still und wartet, bis alle aktiven Transaktionen abgeschlossen sind, und verhindert, dass neue Transaktionen gestartet werden. Dieser Prozess weist außerdem alle ObjectGridLifecycleListener- und BackingMapLifecycleListener-Plug-ins an, alle transaktionsorientierten Aktivitäten einzustellen. Informationen zu Ereignislistener-Plug-ins finden Sie unter Plug-ins für die Bereitstellung von Ereignis-Listenern.
  – Aktualisieren Sie alle Container von eXtreme Scale, die in einem OSGi-Framework ausgeführt werden, so, dass sie die neuen Serviceversionen verwenden.
  – Ändern Sie den Status des Grids in "online", damit Transaktionen fortgesetzt werden können.

  Der Aktualisierungsprozess ist insofern idempotent, dass er in dem Fall, dass ein Client eine Task nicht ausführen kann, bewirkt, dass die Operation rückgängig gemacht wird. Wenn ein Client das Rollback nicht durchführen kann oder wäh-

rend des Aktualisierungsprozesses unterbrochen wird, kann derselbe Befehl erneut abgesetzt und beim entsprechenden Schritt fortgesetzt werden.

Wenn der Client seine Aktivitäten nicht fortsetzen kann und der Prozess über einen anderen Client erneut gestartet wird, verwenden Sie die Option `-force`, um dem Client die Durchführung der Aktualisierung zu ermöglichen. Der Befehl **`osgiUpdate`** verhindert, dass mehrere Clients dasselbe MapSet gleichzeitig aktualisieren. Weitere Einzelheiten zum Befehl **`osgiUpdate`** finden Sie unter „OSGi-Services für eXtreme-Scale-Plug-ins mit **`xscmd`** aktualisieren".

OSGi-Services für eXtreme-Scale-Plug-ins mit `xscmd` aktualisieren

WebSphere eXtreme Scale unterstützt das Upgrade von Plug-in-Bundles für Container-Server bei aktivem Grid. Diese Unterstützung ermöglicht Administratoren, Anwendungsaktualisierungen und -erweiterungen durchzuführen, ohne Gridprozesse erneut starten zu müssen.

Vorbereitende Schritte

Führen Sie die folgenden Schritte aus, bevor Sie OSGi-Bundles von eXtreme Scale auf eine neue Version aktualisieren:

1. Starten Sie die Server von eXtreme Scale in einem unterstützten OSGi-Framework.
2. Packen Sie alle Plug-ins von eXtreme Scale in separate Bundles, die die Service-Rankings verwenden müssen, um die Version jedes Plug-ins zu identifizieren.
3. Geben Sie Cacheobjekte als primitive Java-Typen wie `byte[]`, Integer oder String an oder legen Sie fest, dass sie mit dem Plug-in MapSerializerPlugin gespeichert werden müssen. Die Datenobjekte werden im eXtreme-Scale-Bundle gespeichert und nicht aktualisiert. Nur die Plug-ins, die mit den Daten interagieren, werden aktualisiert.
4. Entwerfen Sie versionskompatible Cacheobjektdaten. Neue Plug-ins müssen in der Lage sein, mit Daten zu interagieren, die von älteren Plug-ins erstellt wurden.
5. Entwerfen Sie Plug-ins so, dass sie ObjectGridLifecycle- und BackingMapLifecycle-Ereignisse empfangen können und somit alle Referenzen auf andere Plug-ins oder die Metadaten, die die Plug-ins möglicherweise enthalten, aktualisiert werden, wenn sie geändert werden.
6. Der OSGi-Aktualisierungsprozess von eXtreme Scale betrifft nur Server. Sie müssen alle Clients, die Plug-ins verwenden, unabhängig voneinander aktualisieren.

Informationen zu diesem Vorgang

Wenn OSGi nicht aktiviert ist und ein Administrator die Anwendungs-Plug-ins oder Cacheobjekte aktualisieren muss, muss jeder Gridknoten einzeln aktualisiert werden, was zu einer Belastung des Netzes und zu einer höheren Speicher- und CPU-Auslastung führt. Dies ist erforderlich, da Plug-ins und Java-Cacheobjekte direkt im Grid gespeichert werden. Wenn Klassen ohne Neustart der Prozesse aktualisiert werden, treten in den Grid-Plug-ins Konflikte auf, weil jede Klasse einen anderen ClassLoader hat.

Das Produkt eXtreme Scale enthält das Dienstprogramm xscmd und MBeans, die Administratoren ermöglichen, alle Plug-in-Bundles anzuzeigen, die im übergeordneten OSGi-Framework jedes Grid-Containers installiert sind, und die zu ver-

wendende Überarbeitung auszuwählen. Wenn das Dienstprogramm "xscmd" für die Aktualisierung der Plug-ins auf ein neues Ranking verwendet wird, wird das Grid stillgelegt, alle Transaktionen werden bereinigt, die Plug-ins werden aktualisiert, und das Grid wird erneut aktiviert. Tritt während des Aktualisierungsprozesses ein Fehler auf, wird der Prozess rückgängig gemacht und das alte Ranking wiederhergestellt.

Vorgehensweise

1. Erstellen Sie eine Version des Bundles, und erhöhen Sie dabei die Rankingnummer im Bundlemanifest und das Ranking für jeden eXtreme-Scale-Plug-in-Service. Wenn die ursprüngliche Bundleversion `Bundle-Version: 1.0.0` ist, kann `Bundle-Version: 1.1.0` als nächste Version definiert werden.

   Wenn das ursprüngliche Service-Ranking `ranking="1"` ist, kann `ranking="2"` als nächstes Ranking definiert werden.

   Wichtig: Die Rankings von OSGi-Services müssen ganze Zahlen sein.
2. Kopieren Sie das neue Bundle auf jeden OSGi-Frameworkknoten, der einen Container-Server von eXtreme Scale hostet.
3. Installieren Sie das neue Bundle im OSGi-Framework. Dem Bundle wird eine Bundle-ID zugeordnet, z. B.:

   ```
   osgi> install <Bundle-URL>
   ```
4. Starten Sie das neue Bundle mit der zugeordneten Bundle-ID, z. B.:

   ```
   osgi> start <ID>
   ```

   Nach dem Start des neuen Bundles erkennt der OSGi-Service-Tracker von eXtreme Scale das Bundle und stellt es zur Aktualisierung bereit.
5. Verwenden Sie den Befehl **xscmd -c osgiAll**, um sicherzustellen, dass jeder Container-Server das neue Bundle sieht. Der Befehl **osgiAll** fragt alle Container im Grid nach allen Services ab, die in der ObjectGrid-XML-Deskriptordatei referenziert werden, und zeigt alle verfügbaren Rankings an, z. B.:

   ```
   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. Verwenden Sie den Befehl **xscmd -c osgiCheck**, um sicherzustellen, dass mindestens ein Service-Ranking ein gültiges Aktualisierungsziel ist, z. B.:

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```
7. Wenn der Befehl **osgiCheck** keine Fehler findet, setzen Sie die Ausführung der Balancer-Komponente des Verteilungsservice aus, um Shard-Verschiebungen zu vermeiden, falls während des Aktualisierungsprozesses ein Fehler auftritt. Zum Aussetzen der Verteilung verwenden Sie den Befehl **xscmd -c suspendBalancing** für jedes ObjectGrid und MapSet, das von der Aktualisierung betroffen ist, z. B.:

```
xscmd -c suspendBalancing -g MyGrid -ms MyMapSet
```

8. Nachdem die gleichmäßig Verteilung für jedes betroffene ObjectGrid und MapSet ausgesetzt wurde, verwenden Sie den Befehl **xscmd -c osgiCheck** erneut, um sicherzustellen, dass mindestens ein Service-Ranking ein gültiges Aktualisierungsziel ist, z. B.:

```
xscmd -c osgiCheck -sr
mySerializerServiceFactory;2,myLoaderServiceFactory;2

CWXSI0040I: The command osgiCheck has completed successfully.
```

9. Nachdem die gleichmäßig Verteilung für jedes betroffene ObjectGrid und MapSet ausgesetzt wurde, verwenden Sie den Befehl **osgiUpdate**, um den Service in allen Servern für ein ObjectGrid und MapSet zu aktualisieren, z. B.:

```
xscmd -c osgiUpdate  -sr
mySerializerServiceFactory;2,myLoaderServiceFactory;2 -g MyGrid -ms MyMapSet
```

10. Vergewissern Sie sich, dass das Upgrade erfolgreich war, z. B.:

```
Update succeeded for the following service rankings:
Service                    Ranking
-------                    -------
mySerializerServiceFactory  2
myLoaderServiceFactory      2
```

11. Nachdem Sie sichergestellt haben, dass das Ranking erfolgreich aktualisiert wurde, aktivieren Sie die gleichmäßige Verteilung mit dem Befehl **xscmd -c resumeBalancing** erneut, z. B.:

```
xscmd -c resumeBalancing -g MyGrid -ms MyMapSet
```

12. Stoppen und deinstallieren Sie das alte Bundle in jedem OSGi-Framework, das den Container von eXtreme Scale hostet. Geben Sie beispielsweise den folgenden Code in der Eclipse-Equinox-Konsole ein:

```
osgi> stop <ID>
osgi> uninstall <ID>
```

Ergebnisse

Das eXtreme-Scale-Bundle wurde auf eine neue Version aktualisiert.

Verteilung steuern

Sie können verschiedene Optionen verwenden, um die Verteilung von Shards auf die verschiedenen Container-Server in der Konfiguration zu steuern. Während des Starts können Sie festlegen, dass die Shard-Verteilung verzögert wird. Wenn Sie alle Container-Server ausführen, müssen Sie die Verteilung unter Umständen aussetzen, fortsetzen oder ändern, während Sie die Server verwalten.

Vorgehensweise

Verteilung während des Starts steuern
Sie können steuern, wann mit der Verteilung der Shards während des Umgebungsstarts begonnen wird. Eine gewisse Steuerung ist standardmäßig etabliert. Wenn Sie keine Aktionen zur Steuerung der Verteilung ausführen, wird unverzüglich mit der Shard-Verteilung begonnen. Bei unverzüglicher Shard-Verteilung werden die Shards möglicherweise nicht gleichmäßig verteilt, wenn weitere Container-Server gestartet werden, und es werden weitere Verteilungsoperationen ausgeführt, um die Shards gleichmäßig zu verteilen.

- Gleichmäßig Verteilung der Shards vorübergehend aussetzen, um eine unverzügliche Shard-Verteilung zu verhindern, wenn die Container-Server gestartet werden.

Bevor Sie Ihre Container-Server starten, verwenden Sie den Befehl **xscmd -c suspendBalancing**, um die gleichmäßige Verteilung der Shards für ein bestimmtes Datengrid und MapSet zu stoppen. Nachd dem Start der Container-Server können Sie den Befehl **xscmd -c resumeBalancing** verwenden, um di Verteilung der Shards auf die Container-Server zu starten.

- 7.1.1+ Eigenschaft **placementDeferralInterval** konfigurieren.

  Die Eigenschaft **placementDeferralInterval** minimiert die Anzahl der Shard-Verteilungszyklen in den Container-Servern. Die Shard-Verteilung wird im definierten Zeitintervall ausgelöst.

  Setzen Sie die Eigenschaft **placementDeferralInterval** in der Servereigenschaftendatei für den Katalogserver. Wenn sie die integrierte Server-API verwenden, verwenden Sie die Methode setPlacementDeferralInterval in der Schnittstelle CatalogServerProperties. Diese Eigenschaft legt die Wartezeit in Millisekunden fest, bevor mit der Verteilung der Shards auf die Container-Server begonnen wird. Der Standardwert für diese Eigenschaft sind 15 Sekunden. Wenn der Standardwert verwendet wird, wird die Verteilung beim Start eines Container-Servers erst gestartet, wenn die mit der Eigenschaft angegebene Zeit abgelaufen ist. Wenn mehrere Container-Server nacheinander gestartet werden, wird der Zeitgeber für das Verzögerungsintervall zurückgesetzt, wenn ein neuer Container-Server im angegebenen Intervall gestartet wird. Angenommen, ein zweiter Container-Server wird 10 Sekunden nach dem ersten Container-Server gestartet. In diesem fall wird die Verteilung erst 15 Sekunden nach dem Start des zweiten Container-Servers gestartet. Wird jedoch ein dritter Container-Server 20 Sekunden nach dem zweiten Container-Server gestartet, hat die Verteilung für die ersten beiden Container-Server bereits begonnen.

  Wenn Container-Server nicht verfügbar sind, wird die Verteilung ausgelöst, sobald der Katalogserver Kenntnis von diesem Ereignis erhält, so dass eine Wiederherstellung so schnell wie möglich stattfinden kann.

  Mithilfe der folgenden Tipps können Sie feststellen, ob die Verteilungsverzögerung auf den richtigen Wert gesetzt ist:

  – Sehen Sie sich während des gleichzeitigen Startens der Container-Server die CWOBJ1001-Nachrichten in der Datei `SystemOut.log` für jeden Container-Server an. Die Zeitmarke dieser Nachrichten in der Protokolldatei jedes Container-Servers zeigt die tatsächliche Startzeit des Container-Servers an. Sie können die Eigenschaft **placementDeferralInterval** anpassen, um weitere Container-Server-Starts einzuschließlen. Wird der erste Container-Server beispielsweise 90 Sekunden vor dem letzten Container-Server gestartet, können Sie die Eigenschaft auf 90 Sekunden setzen.
  – Achten Sie darauf, wie lange die CWOBJ1511-Nachrichten nach den CWOBJ1001-Nachrichten ausgegeben werden. Diese Zeit kann anzeigen, ob die Verzögerung erfolgreich war.
  – In einer Entwicklungsumgebung sollten Sie die beim Testen Ihrer Anwendung auch die Länge des Intervalls berücksichtigen.

- Attribut **numInitialContainers** konfigurieren.

  Wenn Sie zuvor das Attribut **numInitialContainers** verwendet haben, können Sie das Attribut weiter verwenden. Es wird jedoch empfohlen, anstelle des Attributs **numInitialContainers** die Befehle **xscmd -c suspendBalancing** und **xscmd -c resumeBalancing**, gefolgt von **placementDeferralInterval**, für die Steuerung der Verteilung zu verwenden. Das Attribut **numInitialContainers** gibt die Anzahl der Container-Server an, die erforderlich sind, bevor die anfängliche Verteilung für die Shards in diesem Element "mapSet" stattfindet. Das Attribut **numInitialContainers** wird in der XML-Deskriptordatei der Implementierungsrichtlinie definiert. Wenn Sie **numInitialContainers** und

placementDeferralInterval definiert haben, findet unabhängig vom Wert der Eigenschaft **placementDeferralInterval** eine Verteilung erst statt, wenn der Wert von **numInitialContainers** erreicht ist.

Verteilung nach dem Anfangsstart steuern.

- Verteilung erzwingen.

  Sie können den Befehl **xscmd -c triggerPlacement -g my_OG -ms my_Map_Set** verwenden, in dem *my_OG* und *my_Map_Set* auf die Werte für Ihr Datengrid und Ihr MapSet gesetzt sind, um die Verteilung zu einem bestimmten Zeitpunkt, zu dem die Verteilung sonst nicht stattfindet, zu erzwingen. Sie können diesen Befehl beispielsweise ausführen, wenn die mit der Eigenschaft **placementDeferralInterval** angegebene Zeit noch nicht abgelaufen ist oder wenn eine gleichmäßige Verteilung ausgesetzt ist.

- Primäres Shard erneut zuordnen.

  Verwenden Sie den Befehl **xscmd -c swapShardWithPrimary**, um ein Replikat-Shard als neues primäres Shard zuzuordnen. Das vorherige primäre Shard wird zu einem Replikat.

- Primäre Shards und Replikat-Shards neu verteilen.

  Verwenden Sie den Befehl **xscmd -c balanceShardTypes**, um das Verhältnis zwischen primären Shards und Replikat-Shards so anzupassen, dass diese in ausgeglichener Anzahl auf die aktiven Container-Server in der Konfiguration verteilt werden. Das Verhältnis ist innerhalb eines Shards in jedem Container-Server konsistent.

- Verteilung aussetzen oder fortsetzen.

  Verwenden Sie den Befehl **xscmd -c suspendBalancing** oder den Befehl **xscmd -c resumeBalancing**, um die gleichmäßige Verteilung von Shards für ein bestimmtes Datengrid und ein bestimmtes MapSet zu stoppen und zu starten. Wenn die gleichmäßig Verteilung ausgesetzt wurde, können die folgnden Verteilungsaktionen trotzdem ausgeführt werden:

  – Es kann eine Shard-Umstufung durchgeführt werden, wenn Container-Server ausfallen.

  – Mit dem Befehl **xscmd -c swapShardWithPrimary** kann ein Shard-Rollentausch durchgeführt werden.

  – Mit dem Befehl **xscmd -c triggerPlacement -g myOG -ms myMapSet** kann eine gleichmäßige Shard-Verteilung eingeleitet werden.

Nächste Schritte

Sie können die Verteilung in der Umgebung mit dem Befehl **xscmd -c placementServiceStatus** überwachen.

ObjectGrid-Verfügbarkeit verwalten

Der Verfügbarkeitsstatus einer ObjectGrids-Instanz bestimmt, welche Anforderungen zu einer bestimmten Zeit verarbeitet werden können. sie können die Schnittstelle StateManager verwenden, um den Status einer ObjectGrid-Instanz zu definieren und abzurufen.

Informationen zu diesem Vorgang

Es sind vier Verfügbarkeitsstatus für eine bestimmte ObjectGrid-Instanz vorhanden.

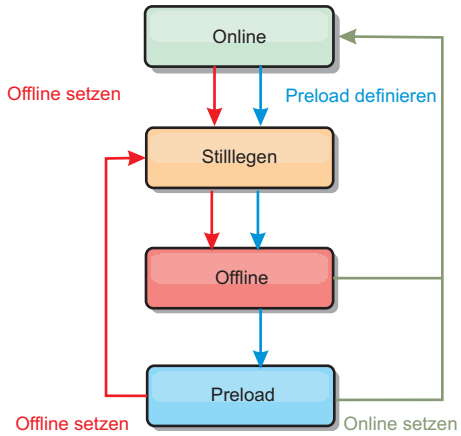


Abbildung 51. Verfügbarkeitsstatus einer ObjectGrid-Instanz

ONLINE
Der Status ONLINE ist der Standardverfügbarkeitsstatus für ein ObjectGrid. Ein ObjectGrid mit dem Verfügbarkeitsstatus ONLINE kann alle Anforderungen eines typischen eXtreme-Scale-Clients verarbeiten. Anforderungen von Preload-Clients werden jedoch zurückgewiesen, wenn das ObjectGrid den Status ONLINE hat.

QUIESCE
Der Status QUIESCE ist ein temporärer Status. Ein ObjectGrid mit dem Verfügbarkeitsstatus QUIESCE wird bald to den Status OFFLINE versetzt. Wenn ein ObjectGrid den Status QUIESCE hat, können ausstehende Transaktionen verarbeitet werden. Neue Transaktionen werden jedoch zurückgewiesen. Ein ObjectGrid kann bis zu 30 Sekunden im Status QUIESCE verbleiben. Danach wird der Verfügbarkeitsstatus in OFFLINE geändert.

OFFLINE
Der Status OFFLINE führt zur Zurückweisung aller Transaktionen, die an das ObjectGrid gesendet werden.

PRELOAD
Der Status PRELOAD kann verwendet werden, um Daten von einem Preload-Client in ein ObjectGrid zu laden. Während das ObjectGrid den Status PRELOAD hat, kann nur ein Preload-Client Transaktionen im ObjectGrid festschreiben. Alle anderen Transaktionen werden zurückgewiesen.

Eine Anforderung wird zurückgewiesen, wenn ein ObjectGrid nicht den erforderlichen Verfügbarkeitsstatus hat, der diese Anforderung unterstützt. Es wird eine Ausnahme des Typs "AvailabilityException" ausgegeben, wenn eine Anforderung zurückgewiesen wird.

Vorgehensweise

1. Anfangsstatus eines ObjectGrids mit der ObjectGrid-XML-Konfigurationsdatei festlegen.

   Sie können das Attribut **initialState** in einem ObjectGrid verwenden, um dessen Anfangsstatus anzugeben. Nach Abschluss der Initialisierung ist ein ObjectGrid normalerweise für Routing bereit. Der Status kann später geändert werden, um zu verhindern, dass Datenverkehr an ein ObjectGrid weitergeleitet wird. Wenn das ObjectGrid initialisiert werden muss, aber nicht sofort verfügbar ist, können Sie das Attribut **initialState** verwenden.

Das Attribut "initialState" wird in der XML-Konfigurationsdatei des ObjectGrids definiert. Der Standardstatus ist ONLINE. Die gültigen Werte sind:

- ONLINE (Standardeinstellung)
- PRELOAD
- OFFLINE

Weitere Informationen zum Attribut **initialState** finden Sie unter ObjectGrid-XML-Deskriptordatei.

Wenn das Attribut "initialState" in einem ObjectGrid definiert wird, muss der Status explizit auf ONLINE zurückgesetzt werden, oder das ObjectGrid bleibt nicht verfügbar. Es tritt eine Ausnahme des Typs "AvailabilityException" ein, wenn das ObjectGrid nicht den Status ONLINE hat.

Weitere Informationen finden Sie in der Dokumentation zur API AvailabilityState.

Attribut "initialState" für das vorherige Laden verwenden

Wenn das ObjectGrid vorher mit Daten geladen wird (Preload), kann es einen Zeitraum zwischen Verfügbarkeit des ObjectGrids und Wechseln in einen Preload-Status geben, in dem Clientdatenverkehr blockiert werden kann. Um diesen Zeitraum zu verhindern, kann der Anfangsstatus eines ObjectGrids auf PRELOAD gesetzt werden. Das ObjectGrid setzt zwar die erforderlichen Initialisierungsprozesse fort, blockiert den Datenverkehr aber so lange, bis sich der Status ändert und der Preload-Prozess durchgeführt werden kann.

Die Status PRELOAD und OFFLINE blockieren zwar beide den Datenverkehr, aber Sie müssen den Status PRELOAD verwenden, wenn Sie einen Preload-Prozess einleiten möchten.

Verhalten beim Failover und Lastausgleich

Wenn ein Replikatdatengrid in ein primäres Datengrid hochgestuft wird, verwende das Replikat nicht die **initialState**-Einstellung. Wenn das primäre Shard zur Neuverteilung verschoben wird, wird die **initialState**-Einstellung nicht verwendet, weil die Daten an die neue primäre Position kopiert werden, bevor der Verschiebevorgang durchgeführt wird. Wenn keine Replikation konfiguriert ist, übernimmt das primäre Shard den **initialState**-Wert, wenn ein Failover stattfindet und ein neues primäres Shards verteilt werden muss.

2. Verfügbarkeitsstatus mit der Schnittstelle StateManager ändern.

   Verwenden Sie die Schnittstelle "StateManager", um den Verfügbarkeitsstatus eines ObjectGrids festzulegen. Zum Festlegen des Verfügbarkeitsstatus eines auf den Servern ausgeführten ObjectGrids übergeben Sie einen entsprechenden ObjectGrid-Client an die Schnittstelle "StateManager". Der folgende Code veranschaulicht, wie der Verfügbarkeitsstatus eines ObjectGrids geändert wird.

   ```
   ClientClusterContext client = ogManager.connect("localhost:2809", null, null);
   ObjectGrid myObjectGrid = ogManager.getObjectGrid(client, "myObjectGrid");
   StateManager stateManager = StateManagerFactory.getStateManager();
   stateManager.setObjectGridState(AvailabilityState.OFFLINE, myObjectGrid);
   ```

   Jedes Shard des ObjectGrids nimmt den gewünschten Status an, wenn die Methode "setObjectGridState" in der Schnittstelle "StateManager" aufgerufen wird. Wenn die Methode zurückkehrt, sollten alle Shards im ObjectGrid den richtigen Status haben.

   Verwenden Sie ein ObjectGridEventListener-Plug-in, um den Verfügbarkeitsstatus eines serverseitigen ObjectGrids zu ändern. Ändern Sie den Verfügbarkeitsstatus eines serverseitigen ObjectGrids nur, wenn das ObjectGrid eine einzige Partition hat. Falls das ObjectGrid mehrere Partitionen hat, wird die Methode "shardActivated" für jedes primäre Shard aufgerufen, was zu überflüssigen Aufrufen zum Ändern des ObjectGrid-Status führt.

```
public class OGListener implements ObjectGridEventListener,
   ObjectGridEventGroup.ShardEvents {
    public void shardActivated(ObjectGrid grid) {
            StateManager stateManager = StateManagerFactory.getStateManager();
            stateManager.setObjectGridState(AvailabilityState.PRELOAD, grid);
    }
}
```

Da der Status QUIESCE ein Übergangsstatus ist, können Sie die Schnittstelle "StateManager" nicht verwenden, um ein ObjectGrid in den Status QUIESCE zu versetzen. Ein ObjectGrid nimmt diesen Status auf seinem Weg zum Status OFFLINE vorübergehend an.

3. Verfügbarkeitsstatus abrufen.

   Verwenden Sie die Methode "getObjectGridState" der Schnittstelle "StateManager", um den Verfügbarkeitsstatus eines bestimmten ObjectGrids abzurufen.

   ```
   StateManager stateManager = StateManagerFactory.getStateManager();
   AvailabilityState state = stateManager.getObjectGridState(inventoryGrid);
   ```

   Die Methode "getObjectGridState" wählt ein zufälliges primäres Shard im ObjectGrid aus und gibt dessen Verfügbarkeitsstatus zurück. Da alle Shards eines ObjectGrids denselben Verfügbarkeitsstatus haben bzw. auf dem Übergang zu demselben Verfügbarkeitsstatus sein sollten, gibt diese Methode ein akzeptables Ergebnis für den aktuellen Verfügbarkeitsstatus des ObjectGrids zurück.

Ausfälle in Rechenzentren verwalten

Bei einem Ausfall im Rechenzentrum können Sie das Quorum überschreiben, so dass Ereignisse der Container-Server nicht ignoriert werden. Sie können das Dienstprogramm **xscmd** verwenden, um Informationen zu Quorum-Tasks, wie z. B. den Quorumstatus, abzufragen und Quorum-Tasks auszuführen, z. B. das Quorum überschreiben.

Vorbereitende Schritte

- Konfigurieren Sie den Quorummechanismus so, dass er in allen Katalogservern dieselbe Einstellung hat. Weitere Informationen finden Sie unter Quorummechanismus konfigurieren.
- Das Quorum ist die minimale Anzahl an Katalogservern, die erforderlich ist, um Verteilungsoperationen für das Datengrid durchzuführen, und entspricht der vollständigen Gruppe an Katalogservern, sofern Sie keine niedrigere Zahl konfigurieren. WebSphere eXtreme Scale erwartet in den folgenden Fällen einen Quorumverlust:
  - Ein JVM-Member des Katalogservice fällt aus.
  - Es tritt ein Brownout im Netz ein.
  - Es tritt ein Ausfall des Rechenzentrums ein.

  Die folgende Nachricht zeigt einen Quorum-Verlust an. Suchen Sie nach dieser Nachricht in Ihren Katalogserviceprotokollen.

  ```
  CWOBJ1254W: Der Katalogservice wartet auf das Quorum.
  ```

Informationen zu diesem Vorgang

Überschreiben Sie das Quorum nur bei einem Ausfall im Rechenzentrum. Wenn Sie das Quorum überschreiben, kann jede noch aktive Katalogserviceinstanz verwendet werden. Alle noch aktiven Instanzen werden benachrichtigt, wenn eine Instanz angewiesen wird, das Quorum zu überschreiben.

Vorgehensweise

- Quorumstatus mit dem Dienstprogramm **xscmd** abfragen.

  ```
  xscmd -c showQuorumStatus -cep cathost:2809
  ```

  Verwenden Sie diese Option, um den Quorumstatus der Katalogserviceinstanz anzuzeigen. Es wird eines der folgenden Ergebnisse angezeigt:

  - `Quorum ist inaktiviert`: Die Katalogserver werden in einem Modus ausgeführt, in dem das Quorum inaktiviert ist. Der Modus mit inaktiviertem Quorum ist der Modus für Entwicklungsumgebungen oder Umgebungen mit einem einzigen Rechenzentrum. Verwenden Sie diesen Modus nicht für Konfigurationen mit mehreren Rechenzentren.
  - `Quorum ist aktiviert, und der Katalogserver hat das Quorum`: Das Quorum ist aktiviert, und das System arbeitet normal.
  - `Quorum ist aktiviert, aber der Katalogserver wartet auf das Quorum:`: Das Quorum ist aktiviert, und das Quorum ist verloren gegangen.
  - `Quorum ist aktiviert, und das Quorum wurde überschrieben`: Das Quorum ist aktiviert, und das Quorum wurde überschrieben.
  - `Quorumstatus ist unzulässig`: Wenn ein Brownout auftritt, wird der Katalogservice in zwei Partitionen, A und B, aufgeteilt. Der Katalogserver A hat das Quorum überschrieben. Die Netzpartition wird aufgelöst. Der Status des Servers in der Partition B ist unzulässig, und es ist ein Neustart der JVM erforderlich. Dieser Fall tritt auch ein, wenn die Katalog-JVM in Partition B wegen des Brownouts erneut gestartet wird und die Brownoutbedingung anschließend behoben wird.

- Quorum mit dem Dienstprogramm **xscmd** überschreiben.

  ```
  xscmd -c overrideQuorum -cep cathost:2809
  ```

  Wenn Sie diesen Befehl ausführen, werden die noch aktiven Katalogserver gezwungen, ein neues Quorum einzurichten.

- Quorum mit dem Dienstprogramm **xscmd** diagnostizieren.

  - **Liste der Stammgruppen anzeigen:**

    Verwenden Sie die Option **-c listCoreGroups**, um eine Liste aller Stammgruppen für den Katalogserver anzuzeigen.

    ```
    xscmd -c listCoreGroups -cep cathost:2809
    ```

  - **Server entfernen:**

    Verwenden Sie die Option **–c teardown**, um einen Server manuell aus dem Datengrid zu entfernen. Das Entfernen eines Servers aus dem Grid ist gewöhnlich nicht erforderlich. Server werden automatisch entfernt, wenn sie als ausgefallen erkannt werden, aber der Befehl wird bereitgestellt, damit Sie ihn unter Anleitung des IBM Support ausführen können. Weitere Informationen zur Verwendung dieses Befehls finden Sie unter „Server mit dem Dienstprogramm **xscmd** normal stoppen" auf Seite 417.

    ```
    xscmd –c teardown server1,server2,server3 -cep cathost:2809 –g Grid
    ```

  - **Routentabelle anzeigen:**

    Verwenden Sie die Option **-c routetable**, um die aktuelle Routentabelle durch Simulation einer neuen Clientverbindung zum Datengrid anzuzeigen. Außerdem validiert der Befehl die Routentabelle, indem er prüft, ob alle Container-Server ihre Rolle in der Routentabelle kennen, z. B. welcher Typ von Shard für welche Partition bestimmt ist.

    ```
    xscmd -c routetable -cep cathost:2809 –g myGrid
    ```

  - **Map-Größen überprüfen:**

Verwenden Sie die Option **-c showMapSizes**, um sicherzustellen, dass die Schlüsselverteilung auf die Shards im Schlüssel einheitlich erfolgt. Wenn einige Container-Server mehr Schlüssel haben als andere, ist es wahrscheinlich, dass die Hashfunktion in den Schlüsselobjekten eine schlechte Verteilung verwendet.

```
xscmd -c showMapSizes -cep cathost:2809 -g myGrid -ms myMapSet
```

– **Tracezeichenfolgen festlegen:**

Verwenden Sie die Option **-c setTraceSpec** , um die Traceeinstellungen für alle JVMs festzulegen, die dem für den Befehl **xscmd** angegebenen Filter entsprechen. Diese Einstellung ändert die Traceeinstellungen nur so lange, bis ein anderer Befehl verwendet wird bzw. die geänderten JVMs ausfallen oder gestoppt werden.

```
xscmd -c setTraceSpec -spec ObjectGrid*=event=enabled -cep cathost:1099
–g myGrid –hf host1
```

Diese Zeichenfolge aktiviert die Traceerstellung für alle JVMs in dem Server mit dem angegebenen Hostnamen, in diesem Fall `host1`.

– **Nicht zugeordnete Shards anzeigen:**

Verwenden Sie die Option **-c showPlacement -sf U**, um die Liste der Shards anzuzeigen, die nicht im Datengrid verteilt werden können. Shards können nicht verteilt werden, wenn der Verteilungsservice eine Einschränkung aufweist, die die Verteilung verhindert. Wenn Sie beispielsweise JVMs auf einem einzelnen physischen Server starten, das im Produktionsmodus arbeitet, können nur primäre Shards verteilt werden. Es werden keine Replikate zugeordnet, solange keine JVMs auf einem zweiten physischen Server gestartet werden. Der Verteilungsservice verteilt Replikate nur an JVMs, die andere Adressen haben als die JVMs, in denen sich die primären Shards befinden. Wenn eine Zone keine JVMs enthält, kann dies auch dazu führen, dass Shards nicht zugeordnet werden.

```
xscmd -c showPlacement -sf U  -cep cathost:2809 –g myGrid
```

Verwaltung mit Managed Beans (MBeans)

Sie können verschiedene Typen von JMX-Beans (Java Management Extensions) verwenden, um Implementierungen zu verwalten und zu überwachen. Jede MBean verweist auf eine bestimmte Entität, wie z. B. eine Map, ein Datengrid, einen Server oder einen Service.

JMX-MBean-Schnittstellen und WebSphere eXtreme Scale

Jede MBean hat get-Methoden, die Attributwerte darstellen. Diese get-Methoden können nicht direkt über Ihr Programm aufgerufen werden. Die JMX-Spezifikation behandelt Attribute anders als Operationen. Sie können Attribute über die JMX-Konsole eines anderen Anbieters anzeigen, und Sie können Operationen in Ihrem Programm oder über eine JMX-Konsole eines anderen Anbieters durchführen.

Paket "com.ibm.websphere.objectgrid.management"

Eine Übersicht und detaillierte Programmierspezifikatinen für alle verfügbaren MBeans finden Sie in der API-Dokumentation zum Paket com.ibm.websphere.objectgrid.management.

Mit dem Tool "wsadmin" auf Managed Bean (MBeans) zugreifen

Sie können das in WebSphere Application Server bereitgestellte Dienstprogramm "wsadmin" verwenden, um auf MBean-Informationen zuzugreifen.

Vorgehensweise

Führen Sie das Tool "wsadmin" im Verzeichnis `bin` Ihrer Installation von WebSphere Application Server aus. Im folgenden Beispiel wird eine Sicht der aktuellen Shard-Verteilung in einer dynamischen eXtreme-Scale-Umgebung abgerufen. Sie können das Tool "wsadmin" in jeder Installation ausführen, in der eXtreme Scale ausgeführt wird. Das Tool "wsadmin" muss nicht im Katalogservice ausgeführt werden.

```
$ 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>
```

Über das Programm auf Managed Beans (MBeans) zugreifen

Sie können mit Java-Anwendungen eine Verbindung zu MBeans herstellen. Diese Anwendungen verwenden die Schnittstellen im Paket "com.ibm.websphere.objectgrid.management".

Informationen zu diesem Vorgang

Die programmgesteuerten Methoden für den Zugriff auf MBeans variieren je nach Typ des Servers, zu dem eine Verbindung hergestellt wird.

- Verbindung zu einem eigenständigen MBean-Server des Katalogservice herstellen
- Verbindung zu einem MBean-Server eines Containers herstellen
- Verbindung zu einem MBean-Server des Katalogservice herstellen, der in WebSphere Application Server ausgeführt wird
- Verbindung zu einem MBean-Server des Katalogservice mit aktivierter Sicherheit herstellen

Vorgehensweise

- **Verbindung zu einem eigenständigen MBean-Server des Katalogservice herstellen:**

Das folgende Beispielprogramm stellt eine Verbindung zu einem eigenständigen MBean-Server des Katalogservice her und gibt eine XML-formatierte Zeichenfolge zurück, in der jeder Container-Server zusammen mit den ihm zugeordneten Shards für ein bestimmtes ObjectGrid und ein bestimmtes MapSet aufgelistet sind.

```
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;

/**
 * Erfasst die Verteilungsinformationen vom Katalogserver für ein bestimmtes ObjectGrid.
 */
public final class CollectPlacementPlan {
    private static String hostName = "localhost";

    private static int port = 1099;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Stellt eine Verbindung zum ObjectGrid-Katalogservice her, um Verteilungsinformationen
     * abzurufen, und gibt die Informationen aus.
     *
     * @param args
     * @throws Exception
     *             Wenn ein Problem beim Herstellen der Verbindung zum MBean-Server des Katalogservice auftritt.
     */
    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();
            }
        }
    }

}
```

Abbildung 52. CollectPlacementPlan.java

Im Folgenden finden Sie einige Anmerkungen zum Beispielprogramm:

– Der **JMXServiceURL**-Wert für den Katalogservice hat immer das folgende Format: `service:jmx:rmi:///jndi/rmi://<Host>:<Port>/objectgrid/MBeanServer`, wobei `<Host>` für den Host steht, auf dem der Katalogservice ausgeführt wird, und `<Port>` für den JMX-Service-Port, der mit der Option **-JMXServicePort** beim Starten des Katalogservice angegeben wird. Wenn kein Port angegeben wird, ist der Standardport `1099`.
– Für die Aktivierung der ObjectGrid- bzw. Map-Statistiken müssen Sie die folgende Eigenschaft in der Servereigenschaftendatei angeben, wenn Sie einen ObjectGrid-Container starten: `statsSpec=all=enabled`.
– Zum Inaktivieren der MBeans, die in den Container-Servern ausgeführt werden, geben Sie die folgende Eigenschaft in der Servereigenschaftendatei an: `enableMBeans=false`.

Es folgt ein Beispiel für die Ausgabe. Diese Ausgabe zeigt an, dass zwei Container-Server aktiv sind. Der Container-Server `Container-0` hostet vier primäre Shards. Der Container-Server `Container-1` hostet ein synchrones Replikat für jedes der primären Shards im Container-Server `Container-0`. In dieser Konfiguration sind zwei synchrones Replikate und ein asynchrones Replikat konfiguriert. Deshalb bleiben die restlichen Shards im Container-Server `Unassigned`. Wenn zwei weitere Container-Server gestartet werden, wird der Container-Server `Unassigned` nicht angezeigt.

```
<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>
```

- **Verbindung zum MBean-Server eines Containers herstellen:**

  Container-Server hosten MBeans, um Informationen zu den einzelnen Maps- und ObjectGrid-Instanzen abzufragen, die im Container-Server ausgeführt werden. Das folgende Beispielprogramm gibt den Status jedes Container-Servers aus, der im Katalogserver mit der JMX-Adresse `localhost:1099` ausgeführt wird:

```
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;

/**
 * Erfasst den Verteilungsstatus jedes verfügbaren Containers direkt.
 */
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();
            }
        }
    }
}
```

Abbildung 53. CollectContainerStatus.java

Das Beispielprogramm gibt den Container-Server-Status für jeden Container aus. Es folgt ein Beispiel für die Ausgabe:

```
<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>
```

- **Verbindung zum MBean-Server des Katalogservice herstellen, der in WebSphere Application Server ausgeführt wird:**

  Die Methode für den programmgesteuerten Zugriff auf MBeans in WebSphere Application Server weicht geringfügig von dem Zugriff auf MBeans in einer eigenständigen Konfiguration ab.

1. Erstellen und kompilieren Sie ein Java-Programm, um die Verbindung zum MBean-Server herzustellen. Es folgt ein Beispielprogramm:

```
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;

/**
 * Erfasst die Verteilungsinformationen von dem Katalogserver, der in einem Deployment Manager für ein bestimmtes ObjectGrid ausgeführt wird.
 */
public final class CollectPlacementPlanWAS {
    private static String hostName = "localhost";

    private static int port = 9809;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Stellt eine Verbindung zum Katalogservice her, um Verteilungsinformationen abzurufen, und gibt diese Informationen aus.
     *
     * @param args
     * @throws Exception
     *             Wenn ein Problem beim Herstellen der Verbindung zum MBean-Server des Katalogservice auftritt.
     */
    public static void main(String[] args) throws Exception {

        // Verbindung zum Bootstrap-Port des Deployment Manager herstellen.
        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();
            }
        }
    }

}
```

Abbildung 54. CollectPlacementPlan.java

2. Führen Sie den folgenden Befehl aus.

```
"$JAVA_HOME/bin/java" "$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:$WAS_HOME/lib/WMQ/java/lib" \
 -Djava.naming.provider.url=<IIOP-URL_ode_Corbaloc-URL_zur_Anwendungsservermaschine> \
 -Djava.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory \
 -Dserver.root="$WAS_HOME" "$CLIENTSAS" "$CLIENTSSL" $USER_INSTALL_PROP \
 -classpath "$WAS_CLASSPATH":<Liste_der_Anwendungs-JARs_und_Klassen> \
 <vollständig_qualifizierter_Name_der_auszuführenden_Klasse> <Anwendungsparameter>
```

Dieser Befehl setzt voraus, dass das Script *WAS-Stammverzeichnis*/bin/setupCmdLine.sh zur ordnungsgemäßen Definition der Variablen ausgeführt

wurde. Ein Beispiel für das Format des Werts der Eigenschaft "java.naming-.provider.url" ist `corbaloc:iiop:1.0@<Host>:<Port>/NameService`.

- **Verbindung zum MBean-Server des Katalogservice mit aktivierter Sicherheit herstellen:**

  Weitere Informationen zum Herstellen der Verbindung zur Katalogservice-MBean mit aktivierter Sicherheit finden Sie unter „JMX-Sicherheit (Java Management Extensions)" auf Seite 526.

Nächste Schritte

Weitere Beispiele für das Anzeigen von Statistiken und zum Durchführen von Verwaltungsoperationen mit MBeans finden Sie in der Beispielanwendung **xsadmin**. Sie finden Quellcode der Beispielanwendung "xsadmin" in der Datei *WXS-Ausgangsverzeichnis*`/samples/xsadmin.jar` in einer eigenständigen Installation bzw. in der Datei *WXS-Ausgangsverzeichnis*`/xsadmin.jar` in einer Installation mit WebSphere Application Server. Weitere Informationen zu den Operationen, die Sie mit der Beispielanwendung **xsAdmin** ausführen können, finden Sie unter Beispiel: Dienstprogramm **xsadmin**.

Weitere Informationen zu MBeans finden Sie auch im Paket com.ibm.websphere-.objectgrid.management.

Kapitel 8. Überwachung

Sie können die Überwachungskonsole, APIs, MBeans, Protokolle und Dienstprogramme, die im Produkt enthalten sind, verwenden, um die Leistung Ihrer Anwendungsumgebung zu überwachen.

Übersicht über Statistiken

Statistiken in WebSphere eXtreme Scale werden über eine interne Statistikstruktur erstellt. Die API "StatsAccessor", die PMI-Module (Performance Monitoring Infrastructure) und die MBean-API werden aus der internen Struktur erstellt.

Die folgende Abbildung zeigt eine allgemeine Konfiguration von Statistiken für WebSphere eXtreme Scale.

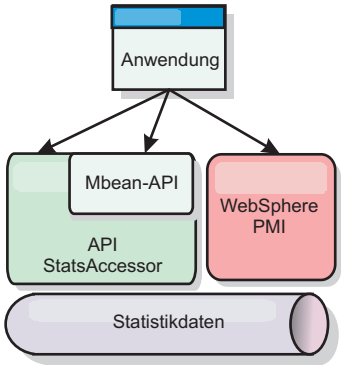


Abbildung 55. Übersicht über Statistiken

Jede dieser APIs bietet eine Sicht auf die Statistikstruktur, wird aber aus jeweils anderen Gründen verwendet:

- **Statistik-API**: Mit der Statistik-API können Entwickler direkt auf Statistiken zugreifen, was flexible und anpassbare Lösungen für die Integration von Statistiken wie angepasste MBeans oder Protokollierung ermöglicht.
- **API MBean**: Die API "MBean" ist ein spezifikationsbasierter Mechanismus für die Überwachung. Die API "MBean" verwendet die Statistik-API und wird lokal in der JVM des Servers ausgeführt. Die API- und MBean-Strukturen sind so konzipiert, dass sie problemlos in die Dienstprogramme anderer Anbieter integriert werden können. Verwenden Sie die API "MBean", wenn Sie mit einem verteilten ObjectGrid arbeiten.
- **PMI-Module (Performance Monitoring Infrastructure) von WebSphere Application Server:** Verwenden Sie PMI, wenn Sie WebSphere eXtreme Scale in WebSphere Application Server ausführen. Diese Module liefern eine Sicht der internen Statistikstruktur.

Statistik-API

Wie bei einer Baumstruktur-Map gibt es einen entsprechenden Pfad und einen Schlüssel, um ein bestimmtes Modul abzurufen, bzw. in diesem Fall eine Differenzierungs- oder Aggregationsstufe. Angenommen, es gibt bereits einen Stammknoten in der Baumstruktur und die Statistiken werden für eine Map mit dem Namen

"payroll" erfasst, die zu einem ObjectGrid mit dem Namen "accounting" gehört. Um beispielsweise auf das Modul für die Aggregations- bzw. Differenzierungsstufe einer Map zuzugreifen, können Sie einen Zeichenfolgebereich (String[]) der Pfade übergeben. In diesem Fall würde dieser "String[] {root, "accounting", "payroll"}" lauten, da jede Zeichenfolge den Pfad des Knotens darstellt. Der Vorteil dieser Struktur ist der, dass ein Benutzer den Bereich für jeden Knoten im Pfad angeben und die Aggregationsstufe für diesen Knoten abrufen kann. Wenn Sie also "String[] {root, "accounting"}" übergeben, erhalten Sie zwar Map-Statistiken, allerdings für das gesamte Grid "accounting." Damit kann der Benutzer sowohl die Typen der zu überwachenden Statistiken als auch die für die Anwendung erforderliche Aggregationsstufe angeben.

PMI-Module von WebSphere Application Server

WebSphere eXtreme Scale enthält Statistikmodule, die mit WebSphere Application Server PMI verwendet werden können. Wenn ein Profil von WebSphere Application Server mit WebSphere eXtreme Scale erweitert wird, integrierten die Erweiterungsscripts die Module von WebSphere eXtreme Scale automatisch in die Konfigurationsdateien von WebSphere Application Server. Mit PMI können Sie Statistikmodule aktivieren und inaktivieren, Statistiken automatisch mit verschiedenen Differenzierungsstufen automatisch zusammenfassen und die Daten über den integrierten Tivoli Performance Viewer selbst in einem Graphen darstellen. Weitere Informationen hierzu finden Sie im Abschnitt „Überwachung mit WebSphere Application Server PMI” auf Seite 471.

Integration von Managed Beans (MBean) in Produkte anderer Anbieter

Die APIs und Managed Beans von eXtreme Scale sind so konzipiert, dass sie problemlos in Überwachungsanwendungen anderer Anbieter integriert werden können. JConsole und MC4J sind zwei Beispiele für schlanke JMX-Konsolen (Java Management Extensions), die zum Analysieren von Informationen über eine eXtreme-Scale-Topologie verwendet werden können. Sie können auch die programmgesteuerten APIs verwenden, um Adapterimplementierungen zu schreiben, mit denen Momentaufnahmen der eXtreme-Scale-Leistung erstellt und die eXtreme-Scale-Leistung überwacht werden können. WebSphere eXtreme Scale enthält eine Beispielüberwachungsanwendung, die sofort einsatzfähige Funktionen bietet und die als Schablone für das Schreiben erweiterter angepasster Überwachungsdienstprogramme verwendet werden kann.

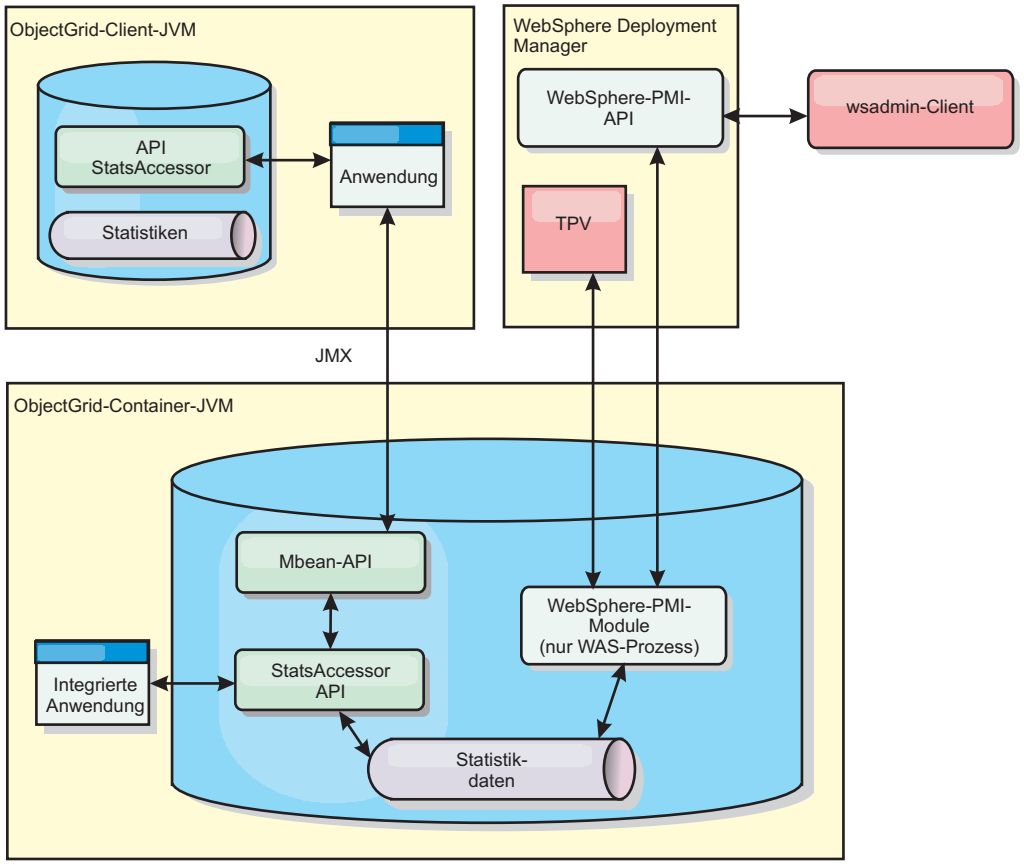


Abbildung 56. Übersicht über MBeans

Weitere Informationen hierzu finden Sie im Abschnitt Beispiel: Dienstprogramm **xsadmin**. Weitere Informationen zur Integration bestimmter Anwendungen anderer Anbieter finden Sie im folgenden Abschnitt:

- eXtreme Scale mit IBM Tivoli Monitoring Agent überwachen
- „eXtreme Scale mit Hyperic HQ überwachen" auf Seite 494
- „eXtreme-Scale-Anwendungen mit CA Wily Introscope überwachen" auf Seite 491

Überwachung mit der Webkonsole

Mit der Webkonsole können Sie aktuelle Statistiken und Protokollstatistiken in einem Diagramm darstellen. Diese Konsole enthält einige vorkonfigurierte Diagramme für allgemeine Übersichten und eine angepasste Berichtseite, die Sie verwenden können, um aus den verfügbaren Statistiken Diagramme zu erstellen. Sie können die Diagrammfunktionen in der Überwachungskonsole von WebSphere eXtreme Scale verwenden, um die allgemeine Leistung der Datengrids in Ihrer Umgebung anzuzeigen.

Webkonsole starten und anmelden

Starten Sie den Konsolserver mit dem Befehl **startConsoleServer**, und melden Sie sich mit der Standardbenutzer-ID und dem zugehörigen Kennwort am Server an.

Vorbereitende Schritte

- **Systemvoraussetzungen**

- Verwenden Sie ein AIX-, Linux- oder Windows-System, um die Webkonsole auszuführen.
- Installieren Sie einen eigenständigen Server von WebSphere eXtreme Scale auf eine System, der den Konsolserver hostet. Weitere Informationen finden Sie unter „WebSphere eXtreme Scale oder WebSphere eXtreme Scale Client standalone installieren” auf Seite 199.
- Das System mit dem Konsolserver muss in der Lage sein, eine Verbindung zu Ihrem Katalogservice herzustellen. Außerdem muss der Katalogservice eine Verbindung zum Server mit der Webkonsole herstellen können.

- **Web-Browser-Voraussetzungen:**

  Verwenden Sie einen der folgenden Browser für die Webkonsole:
  - Mozilla Firefox Version 3.5.x und höher
  - Mozilla Firefox Version 3.6.x und höher
  - Microsoft Internet Explorer Version 7 oder 8

Vorgehensweise

1. Optional: Wenn Sie Ihren Konsolserver an einem anderen als dem Standardport ausführen möchten, editieren Sie die Datei *WXS-Installationsstammverzeichnis*/`ObjectGrid/console/config/zero.config`. Der Standardport für den Konsolserver ist `7080` für HTTP und `7443` für HTTPS. Sie können die folgenden Eigenschaften bearbeiten, um die Standardwerte zu ändern:

   ```
   /config/http/port = 7080
   /config/https/port = 7443
   ```

   Wenn Sie diese Werte bearbeiten, nachdem der Konsolserver gestartet wurde, starten Sie den Server erneut, damit die neuen Portnummern verwendet werden.
2. Starten Sie den Konsolserver. Das Script **`startConsoleServer.bat|sh`** zum Starten des Konsolservers befindet sich im Verzeichnis *WXS-Installationsstammverzeichnis*/`ObjectGrid/bin` Ihrer Installation.
3. Melden Sie sich an der Konsole an.
   a. Rufen Sie im Webbrowser den URL `https://Ihr.Konsol.Host:7443` auf. Ersetzen Sie `Ihr.Konsol.Host` durch den Hostnamen des Servers, auf dem Sie die Konsole installiert haben.
   b. Melden Sie sich an der Konsole an.
      - **Benutzer-ID:** `admin`
      - **Kennwort:** `admin`

      Die Begrüßungsseite der Konsole erscheint.
4. Bearbeiten Sie die Konsolkonfiguration. Klicken Sie auf **Einstellungen** > **Konfiguration**, um die Konsolkonfiguration zu überprüfen. Die Konsolkonfiguration enthält Informationen wie die folgenden:
   - Tracezeichenfolge für den eXtreme-Scale-Client, z. B. `*=all=disabled`
   - Administratorname und -kennwort
   - E-Mail-Adresse des Administrators

Nächste Schritte

- Verbinden Sie Ihre Katalogserver mit der Webkonsole, um mit der Statistiküberwachung zu beginne. Weitere Informationen finden Sie unter „Webkonsole mit Katalogservern verbinden” auf Seite 453.

- Wenn Sie den Server der Webkonsole stoppen müssen, führen Sie das Script **stopConsoleServer.bat|sh** aus. Das Script befindet sich im Verzeichnis *WXS-Installationsstammverzeichnis*/ObjectGrid/bin Ihrer Installation.

Webkonsole mit Katalogservern verbinden

Wenn Sie Statistiken in der Webkonsole anzeigen möchten, müssen Sie zuerst eine Verbindung zu den Katalogservern herstellen, die Sie überwachen möchten. Es sind weitere Schritte erforderlich, wenn in Ihren Katalogservern die Sicherheit aktiviert ist.

Vorbereitende Schritte

- Der Server der Webkonsole muss aktiv sein. Weitere Informationen finden Sie unter „Webkonsole starten und anmelden” auf Seite 451.
- Es muss mindestens ein Katalogserver aktiv sein, zu dem Sie eine Verbindung herstellen möchten. Weitere Informationen hierzu finden Sie im Abschnitt „Eigenständigen Katalogservice starten” auf Seite 403.

Vorgehensweise

1. Wenn in Ihren Katalogservern Secure Sockets Layer (SSL) aktiviert ist, müssen Sie einen Keystore, einen Truststore und eine Clienteigenschaften konfigurieren. Sie aktivieren SSL für einen Katalogserver, indem Sie das Attribut "transportType" in der Servereigenschaftendatei auf SSL-Required setzen.
   a. Keystore und Truststore konfigurieren und dann die öffentlichen Zertifikate austauschen bzw. gegenseitig importieren. Sie können den Truststore und den Keystore beispielsweise an eine Position auf dem Server kopieren, auf dem die Webkonsole ausgeführt wird.
   b. Editieren Sie die Clienteigenschaftendatei auf dem Server der Webkonsole, um die Eigenschaften für die SSL-Konfiguration einzufügen. Sie können beispielsweise die Datei *WXS-Installationsstammverzeichnis*/ObjectGridProperties/sampleclient.properties editieren. Die folgenden Eigenschaften sind für abgehende SSL-Verbindungen über die Webkonsole erforderlich:

```
#--------------------------------------------------------------------------------
# SSL-Konfiguration
#
# - contextProvider     (IBMJSSE2, IBMJSSE, IBMJSSEFIPS usw.)
# - protocol            (SSL, SSLv2, SSLv3, TLS, TLSv1 usw.)
# - keyStoreType        (JKS, JCEK, PKCS12 usw.)
# - trustStoreType      (JKS, JCEK, PKCS12 usw.)
# - keyStore            (vollständig qualifizierter Pfad zur Keystore-Datei)
# - trustStore          (vollständig qualifizierter Pfad zur Truststore-Datei)
# - alias               (Zeichenfolge, die den zu verwendenden SSL-Zertifikatsalias aus dem Keystore angibt)
# - keyStorePassword    (Zeichenfolge, die das Kennwort für den Keystore angibt, verschlüsselt oder unverschlüsselt)
# - trustStorePassword  (Zeichenfolge, die das Kennwort für den Truststore angibt, verschlüsselt oder unverschlüsselt)
#
# Entfernen Sie die Kommentarzeichen für die folgenden Eigenschaften, um die
# SSL-Konfiguration zu definieren.
#--------------------------------------------------------------------------------
#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
```

Wichtig: Windows Wenn Sie Windows verwenden, müssen Sie ein Escapezeichen für alle Backslashs (\ ) im Pfad verwenden. Geben Sie beispielsweise zur Verwendung von C:\opt\ibm in der Eigenschaftendatei Folgendes ein:C:\\opt\\ibm.

2. Stellen Sie Verbindungen zu Katalogservern her, die Sie überwachen möchten, und verwalten Sie diese. Wiederholen Sie die folgenden Schritte, um die einzelnen Katalogserver zur Konfiguration hinzuzufügen.
   a. Klicken Sie auf **Einstellungen** > **eXtreme-Scale-Katalogserver**.
   b. Fügen Sie einen neuen Katalogserver hinzu.

      1) Klicken Sie auf das Symbol "Hinzufügen" ( ), um einen vorhandenen Katalogserver zu registrieren.
      2) Geben Sie Informationen wie den Hostnamen und den Listener-Port an. Weitere Informationen zur Portkonfiguration und zu den Portstandardwerten finden Sie unter „Netzports planen” auf Seite 65.
      3) Klicken Sie auf **OK**.
      4) Vergewissern Sie sich, dass der Katalogserver der Navigationsstruktur hinzugefügt wurde.
3. Gruppieren Sie die erstellten Katalogserver in einer Katalogservicedomäne. Sie müssen eine Katalogservicedomäne erstellen, wenn die Sicherheit in Ihren Katalogservern aktiviert ist, weil Sicherheitseinstellungen in der Katalogservicedomäne konfiguriert werden.
   a. Klicken Sie auf **Einstellungen** > **eXtreme-Scale-Domänen**.
   b. Fügen Sie eine neue Katalogservicedomäne hinzu.

      1) Klicken Sie auf das Symbol "Hinzufügen" ( ), um eine Katalogservicedomäne zu registrieren. Geben Sie einen Namen für die Katalogservicedomäne ein.
      2) Nach der Erstellung der Katalogservicedomäne können Sie die Eigenschaften bearbeiten. Im Folgenden sind die Eigenschaften der Katalogservicedomäne beschrieben:

         Name Gibt den vom Administrator zugeordneten Hostnamen der Domäne an.

         Katalogserver
         Listet die Katalogserver her, die zur ausgewählten Domäne gehören. Sie können die im vorherigen Schritt erstellten Katalogserver hinzufügen.

         Generatorklasse
         gibt den Namen der Klasse an, die die Schnittstelle CredentialGenerator implementiert. Diese Klasse wird verwendet, um Berechtigungsnachweise für Clients abzurufen. Wenn Sie einen Wert in diesem Feld angeben, überschreibt der Wert die Eigenschaft **credentialGeneratorClass** in der Datei `client.properties`.

         Generatoreigenschaften
         Gibt die Eigenschaften für die Implementierungsklasse CredentialGenerator an. Die Eigenschaften werden mit der Methode "setProperties(String)" auf das Objekt gesetzt. Der Wert "credentialGeneratorprops" wird nur verwendet, wenn der Wert der Eigenschaft "credentialGeneratorClass" ungleich null ist. Wenn Sie einen Wert in diesem Feld angeben, überschreibt der Wert die Eigenschaft **credentialGeneratorProps** in der Datei `client.properties`.

Pfad der eXtreme-Scale-Clienteigenschaften
Gibt den Pfad der Clienteigenschaftendatei an, in der Sie in einem vorherigen Schritt SSL-Eigenschaften eingefügt haben. Sie können beispielsweise die Datei `c:\ObjectGridProperties\sampleclient.properties` angeben. Wenn Sie die Verwendung von SSL-Verbindungen in der Konsole stoppen möchten, können Sie den Wert in diese Feld löschen. Nach dem Festlegen des Pfads verwendet die Konsole eine nicht gesicherte Verbindung.

3) Klicken Sie auf **OK**.
4) Vergewissern Sie sich, dass die Domäne der Navigationsstruktur hinzugefügt wurde.

Wenn Sie Informationen zu einer vorhandenen Katalogservicedomäne anzeigen möchten, klicken Sie in der Navigationsstruktur unter **Einstellungen** > **eXtreme-Scale-Domänen** auf den Namen der Katalogservicedomäne.

4. Zeigen Sie den Verbindungsstatus an. Im Feld **Aktuelle Domäne** wird der Name der Katalogservicedomäne angezeigt, die momentan verwendet wird, um Informationen in der Webkonsole anzuzeigen. Der Verbindungsstatus wird neben dem Namen der Katalogservicedomäne angezeigt.

Statistiken mit der Webkonsole anzeigen

Sie können Statistiken und andere Leistungsinformationen mit der Webkonsole überwachen.

Vorbereitende Schritte

Bevor Sie Statistiken mit der Webkonsole anzeigen können, müssen Sie die folgenden Aufgaben ausführen:

1. Starten Sie den Server der Webkonsole. Weitere Informationen finden Sie unter „Webkonsole starten und anmelden" auf Seite 451.
2. Verbinden Sie Ihre Katalogserver mit dem Server der Webkonsole. Weitere Informationen finden Sie unter „Webkonsole mit Katalogservern verbinden" auf Seite 453.
3. Führen Sie aktive Datengrids und Anwendungen in den Servern aus, die von Ihrer Katalogservicedomäne verwaltet werden.

Informationen zu diesem Vorgang

Nachdem Sie Ihre Datengrids erstellt und Ihre Anwendungen für die Verwendung der Datengrids konfiguriert haben, müssen Sie eine Weile warten, bis die Statistiken verfügbar sind. Wenn Sie beispielsweise ein Datengrid für einen dynamischen Cache haben, sind die Statistiken erst verfügbar, wenn ein WebSphere Application Server mit einem dynamischen Cache eine Verbindung zum dynamischen Cache herstellt. Im Allgemeinen müssen Sie nach einer größeren Konfigurationsänderung bis zu einer Minute warten, bis Sie die Änderungen in Ihren Statistiken sehen.

Tipp: Wenn Sie speziellere Informationen zu einem Datenpunkt im Diagramm anzeigen möchten, können Sie den Mauszeiger über diesen Datenpunkt bewegen.

Vorgehensweise

- Zum Anzeigen der aktuellen Serverstatistiken klicken Sie auf **Überwachen** > **Serverübersicht**.
- Zum Anzeigen der Leistung aller Datengrids klicken Sie auf **Überwachen** > **Übersicht über die Datengriddomänen**.

- Zum Anzeigen einzelner Datengrids klicken Sie auf **Überwachen** > **Übersicht über Datengrids** > *Name_des_Datengrids*. Auf dieser Seite wird eine Zusammenfassung angezeigt, die die Anzahl der Cacheeinträge, die durchschnittliche Transaktionsdauer und den durchschnittlichen Durchsatz enthält.
- Wenn Sie weitere Details eines bestimmten Datengrids anzeigen möchten, klicken Sie auf **Überwachen** > **Details des Datengrids**. Es wird eine Baumstruktur mit allen Datengrids in Ihrer Konfiguration angezeigt. Sie können Die Detailabfrage/-analyse auf ein bestimmtes Datengrid beschränken, um nur die Maps dieses Datengrids anzuzeigen. Sie können auf den Namen eines Datengrids oder auf eine Map klicken, um weitere Informationen anzuzeigen:
- Zum Auswählen der Statistiken, die Sie in den angepassten Bericht aufnehmen möchten, klicken Sie auf **Überwachen** > **Angepasste Berichte**.

  Verwenden Sie diese Sicht, um detaillierte Datendiagramme der verschiedenen Statistiken zu erstellen. Verwenden Sie die Baumstruktur, um die verfügbaren Datengrids und Server sowie die zugehörigen Statistiken zu untersuchen. Wenn Sie auf einen Knoten klicken bzw. die Eingabetaste drücken, wenn sich der Cursor auf einem Knoten befindet, der grafisch darstellbare Daten referenziert, erscheint ein Menü. Erstellen Sie ein neues Diagramm, das die Statistiken enthält, oder fügen Sie die Statistiken einem vorhandenen Diagramm mit kompatiblen Statistiken hinzu. Weitere Informationen finden Sie unter „Überwachung mit angepassten Berichten” auf Seite 461.

Webkonsolstatistiken

Ja nach Ansicht, die Sie in der Webkonsole verwenden, können Sie verschiedene Statistiken zu Ihrer Konfiguration anzeigen. Zu diesen Statistiken gehören Statistiken zur Speicherbelegung, zu den am häufigsten verwendeten Datengrids und zur Anzahl der Cacheeinträge.

- „Übersicht über die Datengriddomäne”
- „Übersicht über das Datengrid” auf Seite 457
- „Details des Datengrids” auf Seite 457
- „Serverübersicht” auf Seite 458
- „Angepasste Berichte: Statistiken zu Katalogservicedomänen” auf Seite 458
  - „Angepasste Berichte: Container-Server-Berichte” auf Seite 458
  - „Angepasste Berichte: Datengridstatistiken” auf Seite 460
  - „Angepasste Berichte: Map-Statistiken” auf Seite 460

Übersicht über die Datengriddomäne

Übersichtsstatistiken zur Datengriddomäne werden auf der Seite **Überwachen** > **Übersicht über die Datengriddomäne** angezeigt. Klicken Sie auf eine der folgenden Registerkarten, um weitere Informationen zur Datengriddomäne anzuzeigen:

Registerkarte "Genutzte Kapazität"
: Im Diagramm **Aktuelle Verteilung der genutzten Kapazität an Datengrids** werden Bilder der Gesamtpoolkapazität und der Konsumenten mit der höchsten Kapazitätsnutzung angezeigt. Es werden nur die Top 25 Datengrids angezeigt. Im Diagramm **Used Capacity Over Time** wird die Anzahl der vom Datengrid belegten Bytes angezeigt.

Registerkarte "Durchschnittlicher Durchsatz"
: Das Diagramm **5 Most Active Data Grids by Average Transaction Time in Milliseconds** enthält eine Liste der fünf aktivsten Datencaches, organisiert nach durchschnittlicher Transaktionsdauer. Im Diagramm **Average**

Throughput Over time werden der durchschnittliche, der maximale und der minimale Durchsatz in der letzten Stunde, am letzten Tag und in der letzten Woche angezeigt.

Registerkarte "Durchschnittliche Transaktionsdauer"
Im Diagramm **5 Slowest Data Grids** werden Daten zu den fünf langsamsten Datengrids angezeigt. Im Diagramm **Average Transaction Time Over Time** werden die durchschnittliche, die maximale und die minimale Transaktionsdauer in der letzten Stunde, am letzten Tag und in der letzten Woche angezeigt.

Übersicht über das Datengrid

Zum Anzeigen der Statistiken für ein einzelnes Datengrid klicken Sie auf **Überwachen** > **Datengridübersicht** > *Name_des_Datengrids*.

Current summary over last 30 seconds
Zeigt die aktuelle Anzahl an Cacheeinträgen, die durchschnittliche Transaktionsdauer, den durchschnittlichen Durchsatz und die Cachetrefferrate für das ausgewählte Datengrid an.

Registerkarte "Genutzte Kapazität"
Im Diagramm **Current summary over last 30 seconds** werden die Anzahl der Cacheeinträge und die genutzte Kapazität in Bytes im angegebenen Zeitraum angezeigt.

Registerkarte "Cache Usage"
Das Diagramm **Cache Usage** visualisiert die Anzahl erfolgreicher Abfragen im Cache und zeigt Cacheversuche, Cachetreffer und die Cachetrefferrate in einem bestimmten Zeitraum an.

Registerkarte "Durchschnittlicher Durchsatz"
Im Diagramm **Average Throughput vs. Average Transaction Time** werden die Transaktionsdauer und der Durchsatz in einem bestimmten Zeitraum angezeigt.

Details des Datengrids

Datengridstatistiken werden auf der Seite **Überwachen** > **Data Grid Details** angezeigt. Sie können sich die Daten für ein ausgewähltes Grid und Maps in diesem Grid anzeigen.

Current summary over last 30 seconds
Zeigt die momentan genutzte Kapazität, die Anzahl der Cacheeinträge, den durchschnittlichen Durchsatz und die durchschnittliche Transaktionsdauer für das ausgewählte Datengrid an.

Aktuelle Verteilung der genutzten Kapazität in den eXtreme-Scale-ObjectGrids-Maps
Zeigt einen Gesamtpool an, der die Kapazität nach Zone und die Gesamtkapazität in jeder Zone zeigt. Es werden nur die Top 25 ObjectGrid-Maps angezeigt. Außerdem können Sie die Konsumenten für jede Map anzeigen, die am meisten Kapazität nutzen.

Aktuelle Verteilung der genutzten Kapazität in den Zonen
Zeigt einen Gesamtpool an, einschließlich des Gesamtpools und der Konsumenten,die am meisten Kapazität nutzen, in der Zone des ausgewählten Datengrids an. Außerdem können Sie die Konsumenten für jede Zone anzeigen, die am meisten Kapazität nutzen.

Map-Statistiken:

Current summary over last 30 seconds
Zeigt die momentan genutzte Kapazität, die Anzahl der Cacheeinträge, den durchschnittlichen Durchsatz und die durchschnittliche Transaktionsdauer für die ausgewählte Map an.

Aktuelle Verteilung der genutzten Kapazität in den Partitionen
Zeigt eine Partition an, die den Gesamtpool und die Hauptkonsumenten der Kapazität enthält. Es werden nur die Top 25 Partitionen angezeigt. Außerdem können Sie die Konsumenten für jede Partition anzeigen, die am meisten Kapazität nutzen.

Serverübersicht

Serverstatistiken werden auf der Seite **Überwachen** > **Server Overview** angezeigt.

Current Server Used Memory Distribution
Dieses Diagramm setzt sich aus zwei Ansichten zusammen. In der Ansicht **Total Pool** wird der momentan genutzte (reale) Speicher in der Serverlaufzeitumgebung angezeigt. In der Ansicht **Largest Used memory Consumers** ist der genutzte Speicher in Server aufgegliedert. Es werden jedoch nur die Top 25 Server angezeigt, die am meisten Speicher nutzen.

Total Memory Over Time
Zeigt die wirkliche Speichernutzung in der Laufzeitumgebung des Servers an.

Used Memory Over Time
Zeigt die Speichernutzung in der Laufzeitumgebung des Servers an.

Angepasste Berichte: Statistiken zu Katalogservicedomänen

Sie können Statistiken zu Katalogservicedomänen durch Erstellen eines angepassten Berichts anzeigen. Klicken Sie auf **Überwachen** > **Angepasste Berichte**.

Durchschnittliche Transaktionsdauer (ms)
Zeigt die durchschnittlich erforderliche Zeit für die Fertigstellung einer Transaktion in dieser Domäne an.

Durchschnittlicher Transaktionsdurchsatz (Trans/Sek)
Zeigt die durchschnittliche Anzahl an Transaktionen pro Sekunde in dieser Domäne an.

Maximale Transaktionsdauer (ms)
Zeigt die Dauer der längsten Transaktion in dieser Domäne an.

Minimale Transaktionsdauer (ms)
Zeigt die Dauer der kürzesten Transaktion in dieser Domäne an.

Gesamttransaktionsdauer (ms)
Zeigt die Gesamtdauer der Transaktionen in dieser Domäne seit der Initialisierung der Domäne an.

Angepasste Berichte: Container-Server-Berichte

Sie können Statistiken zu Container-Servern anzeigen, indem Sie einen angepassten Bericht erstellen. Klicken Sie auf **Überwachen** > **Angepasste Berichte**.

Durchschnittliche Transaktionsdauer (ms)
Zeigt die durchschnittlich erforderliche Zeit für die Fertigstellung einer Transaktion für diesen Katalogserver an.

Durchschnittlicher Transaktionsdurchsatz (Trans/Sek)
Zeigt die durchschnittliche Anzahl an Transaktionen pro Sekunde für diesen Katalogserver an.

Maximale Transaktionsdauer (ms)
Zeigt die Dauer der längsten Transaktion für diesen Katalogserver an.

Minimale Transaktionsdauer (ms)
Zeigt die Dauer der kürzesten Transaktion für diesen Katalogserver an.

Gesamttransaktionsdauer (ms)
Zeigt die Gesamtzeit an, die für Transaktionen für diesen Katalogserver seit dessen Initialisierung aufgebracht wurde.

Gesamtanzahl der Einträge im Cache
Zeigt die aktuelle Anzahl der in den Grids zwischengespeicherten Objekte an, die von diesem Katalogserver überwacht werden.

Trefferrate (Prozent)
Zeigt die Trefferrate (Trefferquote) für das ausgewählte Datengrid an. Eine hohe Trefferrate ist wünschenswert. Die Trefferrate zeigt an, wie dienlich das Grid bei der Vermeidung von Zugriffen auf den persistenten Speicher ist.

Belegung in Bytes
Zeigt die Speicherbelegung durch diese Map an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

Minimale Belegung in Bytes
Zeigt die niedrigste Speicherbelegung durch diesen Katalogservice und die zugehörigen Maps an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

Maximale Belegung in Bytes
Zeigt die höchste Speicherbelegung durch diesen Katalogservice und die zugehörigen Maps an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

Gesamtanzahl der Treffer
Zeigt an, wie oft die angeforderten Daten in der Map gefunden wurden und somit kein Zugriff auf den persistenten Speicher stattfinden musste.

Gesamtanzahl der Get-Anforderungen
Zeigt an, wie oft die Map auf den persistenten Speicher zugreifen musste, um Daten abzurufen.

Freier Heapspeicher (MB)
Zeigt die tatsächliche Heapspeicherkapazität an, die der vom Katalogserver verwendeten JVM zur Verfügung steht.

Gesamtheapspeicher
Zeigt die Heapspeicherkapazität an, die der von diesem Katalogserver verwendeten JVM zur Verfügung steht.

Anzahl verfügbarer Prozessoren
Zeigt die Anzahl der Prozessoren an, die diesem Katalogservice und dessen Maps zur Verfügung steht. Die höchste Stabilität erzielen Sie, wenn Sie Ihre Server mit einer Prozessorauslastung von 60 % und Ihre JVMs mit einer Heapspeicherauslastung von 60 % betreiben. So können Lastspitzen die

Prozessorauslastung auf 80–90 % hochtreiben. Ein dauerhafter Betrieb der Server mit diesen Ständen oder höher sollte aber vermieden werden.

Maximale Größe des Heapspeichers (MB)
Zeigt die maximale Heapspeicherkapazität an, die der von diesem Katalogserver verwendeten JVM zur Verfügung steht.

Belegter Speicher
Zeigt die Speicherbelegung in der von diesem Katalogserver verwendeten JVM an.

Angepasste Berichte: Datengridstatistiken

Sie können Datengridstatistiken anzeigen, indem Sie einen angepassten Bericht erstellen. Klicken Sie auf **Überwachen** > **Angepasste Berichte**.

Durchschnittliche Transaktionsdauer (ms)
zeigt die durchschnittlich erforderliche Zeit für die Fertigstellung von Transaktionen an, an denen dieses Grid beteiligt ist.

Durchschnittlicher Transaktionsdurchsatz (Trans/Sek)
Zeigt die durchschnittliche Anzahl der von diesem Grid pro Sekunde ausgeführten Transaktionen an.

Maximale Transaktionsdauer (ms)
Zeigt die Dauer der längsten von diesem Grid ausgeführten Transaktion an.

Minimale Transaktionsdauer (ms)
Zeigt die Dauer der kürzesten von diesem Grid ausgeführten Transaktion an.

Gesamttransaktionsdauer (ms)
Zeigt die Gesamttransaktionsverarbeitungszeit für dieses Grid an.

Angepasste Berichte: Map-Statistiken

Sie können Map-Statistiken anzeigen, indem Sie einen angepassten Bericht erstellen. Klicken Sie auf **Überwachen** > **Angepasste Berichte**.

Gesamtanzahl der Einträge im Cache
Zeigt die aktuelle Anzahl der in dieser Map zwischengespeicherten Objekte an.

Trefferrate (Prozent)
Zeigt die Trefferrate für die ausgewählte Map an. Eine hohe Trefferrate ist wünschenswert. Die Trefferrate zeigt an, wie dienlich die Map bei der Vermeidung von Zugriffen auf den persistenten Speicher ist.

Belegung in Bytes
Zeigt die Speicherbelegung durch diese Map an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

Minimale Belegung in Bytes
Zeigt die minimale Belegung (in Bytes) für diese Map an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

Maximale Belegung in Bytes
Zeigt die maximale Belegung (in Bytes) für diese Map an. Die Statistiken

zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

Gesamtanzahl der Treffer
Zeigt an, wie oft die angeforderten Daten in der Map gefunden wurden und somit kein Zugriff auf den persistenten Speicher stattfinden musste.

Gesamtanzahl der Get-Anforderungen
Zeigt an, wie oft die Map auf den persistenten Speicher zugreifen musste, um Daten abzurufen.

Freier Heapspeicher (MB)
Zeigt die aktuell verfügbare Heapspeicherkapazität für diese Map in der vom Katalogserver verwendeten JVM an.

Gesamtheapspeicher (MB)
Zeigt die insgesamt verfügbare Heapspeicherkapazität für diese Map in der vom Katalogserver verwendeten JVM an. Die höchste Stabilität erzielen Sie, wenn Sie Ihre Server mit einer Prozessorauslastung von 60 % und Ihre JVMs mit einer Heapspeicherauslastung von 60 % betreiben. So können Lastspitzen die Prozessorauslastung auf 80–90 % hochtreiben. Ein dauerhafter Betrieb der Server mit diesen Ständen oder höher sollte aber vermieden werden.

Anzahl verfügbarer Prozessoren
Zeigt die Anzahl der Prozessoren an, die dieser Map zur Verfügung steht. Die höchste Stabilität erzielen Sie, wenn Sie Ihre Server mit einer Prozessorauslastung von 60 % und Ihre JVMs mit einer Heapspeicherauslastung von 60 % betreiben. So können Lastspitzen die Prozessorauslastung auf 80–90 % hochtreiben. Ein dauerhafter Betrieb der Server mit diesen Ständen oder höher sollte aber vermieden werden.

Maximale Größe des Heapspeichers (MB)
Zeigt die maximal verfügbare Heapspeicherkapazität für diese Map in der vom Katalogserver verwendeten JVM an.

Genutzter Speicher (MB)
Zeigt den genutzten Speicher in dieser Map an.

Überwachung mit angepassten Berichten

Sie können angepasste Berichte erstellen, um verschiedene Diagramme zu speichern, die Statistiken über die Katalogservicedomänen, Datengrids und Container-Server in Ihre Umgebung enthalten. Sie können die angepassten Berichte speichern und später erneut laden, um sie anzuzeigen.

Vorbereitende Schritte

Bevor Sie Statistiken mit der Webkonsole anzeigen können, müssen Sie die folgenden Aufgaben ausführen:

1. Starten Sie den Server der Webkonsole. Weitere Informationen finden Sie unter „Webkonsole starten und anmelden" auf Seite 451.
2. Verbinden Sie Ihre Katalogserver mit dem Server der Webkonsole. Weitere Informationen finden Sie unter „Webkonsole mit Katalogservern verbinden" auf Seite 453.
3. Führen Sie aktive Datengrids und Anwendungen in den Servern aus, die von Ihrer Katalogservicedomäne verwaltet werden.

Vorgehensweise

- Angepassten Bericht erstellen.
  1. Klicken Sie auf **Überwachen** > **Angepasste Berichte**. Die eXtreme-Scale-Domänen, die Sie definiert haben, werden in einem Baumstrukturformat aufgelistet. Sie können jede dieser Domänen erweitern, um die verfügbaren Statistiken anzuzeigen, die Sie dem angepassten Bericht hinzufügen können.
  2. Fügen Sie Diagramme mit den Statistiken hinzu, die Sie verfolgen möchten.

     Verfügbare Statistiken sind mit dem Diagrammsymbol () markiert. Klicken Sie auf eine der Statistiken, die Sie verfolgen möchten. Wählen Sie **Add to new chart** oder **Add to existing chart** aus. Je nach Auswahl wird die ausgewählte Statistik auf einer neuen Diagrammregisterkarte oder in einem ausgewählten vorhandenen Diagramm angezeigt. Sie können eine Metrik einem vorhandenen Diagramm nur dann hinzufügen, wenn die bereits im Diagramm enthaltenen Metriken und die neue Metrik dieselben Einheiten verwenden.
- Angepassten Bericht speichern. Wenn Sie den angepassten Bericht speichern, werden die Statistiken auf allen Registerkarten gespeichert, die Sie erstellt haben. Zum Speichern des Berichts klicken Sie auf **Speichern**.
- Angepassten Bericht laden. Klicken Sie auf **Laden**, und wählen Sie den gespeicherten angepassten Bericht aus, den Sie anzeigen möchten.

Überwachung mit CSV-Dateien

Sie können festlegen, dass Überwachungsdaten in CSV-Dateien geschrieben werden. Diese CSV-Dateien können Informationen zur JVM, zur Map oder zur ObjectGrid-Instanz enthalten.

Informationen zu diesem Vorgang

Wenn Sie festlegen, dass Überwachungsdaten in CSV-Dateien geschrieben werden, können Sie Langzeitdaten für einen einzelnen Container- oder Katalogserver herunterladen und analysieren. Die Datenerfassung beginnt, wenn Sie den Server mit den Servereigenschaften starten, die die CSV-Dateien aktivieren. Anschließend können Sie die CSV-Dateien jederzeit herunterladen und die Dateien beliebig verwenden.

Vorgehensweise

1. Aktualisieren Sie die Servereigenschaftendatei mit den folgenden Eigenschaften, die sich auf die Aktivierung der CSV-Dateien beziehen.

```
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
```

Weitere Informationen zu diesen Eigenschaften finden Sie unter Servereigenschaftendatei.

2. Starten Sie den Server erneut, damit die in der Servereigenschaftendatei vorgenommenen Änderungen wirksam werden.
3. Laden Sie die CSV-Datei herunter. Die CSV-Datei wird in das Verzeichnis *Servername*/logs geschrieben.
4. Importieren Sie die CSV-Datei in das Programm, das Sie für die Verarbeitung der Daten verwenden, z. B. ein Spreadsheet.

Nächste Schritte

Weitere Informationen zu den Daten, die in den CSV-Dateien enthalten sind, finden Sie unter „Statistikdefinitionen in CSV-Dateien".

Statistikdefinitionen in CSV-Dateien

Die CSV-Dateien, die Sie für einen Server herunterladen können, enthalten Statistiken, die Sie verwenden können, um Langzeitdiagramme oder andere Informationen zu erstellen.

JVM-Statistikprotokoll

TimeStamp
Gibt das Datum und die Uhrzeit an, zu der die Statistikmomentaufnahme für die JVM erstellt wurde.

ServerName
Gibt den Servernamen der JVM an.

Hostname
Gibt den Hostnamen der JVM an.

DomainName
Gibt die Katalogservicedomäne an, zu der die JVM gehört.

FreeMemory
Gibt die Anzahl verfügbarer Bytes für die JVM an.

MaxMemory
Gibt die maximale Anzahl an Bytes an, die für die JVM reserviert werden kann.

TotalMemory
Zeigt die wirkliche Speichernutzung in der Laufzeitumgebung des Servers an.

AvailProcs
Zeigt die Anzahl der Prozessoren an, die diesem Katalogservice und dessen Maps zur Verfügung steht. Die höchste Stabilität erzielen Sie, wenn Sie Ihre Server mit einer Prozessorauslastung von 60 % und Ihre JVMs mit einer Heapspeicherauslastung von 60 % betreiben. So können Lastspitzen die Prozessorauslastung auf 80–90 % hochtreiben. Ein dauerhafter Betrieb der Server mit diesen Ständen oder höher sollte aber vermieden werden.

Map-Statistikprotokoll

TimeStamp
Gibt das Datum und die Uhrzeit an, zu der die Statistikmomentaufnahme für die Map erstellt wurde.

MapName
Gibt den Namen der Map an.

OgName
Gibt den Namen des Datengrids an, zu dem diese Map gehört.

PartitionId
Gibt die ID der Partition an.

MapSetName
Gibt das MapSet an, zu dem diese Map gehört.

HitRate
Zeigt die Trefferrate für die ausgewählte Map an. Eine hohe Trefferrate ist wünschenswert. Die Trefferrate zeigt an, wie dienlich das Datengrid bei der Vermeidung von Zugriffen auf den persistenten Speicher ist.

Count Gibt die Anzahl der Datenstichproben an, die seit dem Serverstart erfasst wurden. Der Wert 100 gibt beispielsweise an, dass der Eintrag der hundertste Stichprobeneintrag ist, der seit dem Serverstart erfasst wurde.

TotalGetCount
Zeigt an, wie oft die Map auf den persistenten Speicher zugreifen musste, um Daten abzurufen.

TotalHitCount
Zeigt an, wie oft die angeforderten Daten in der Map gefunden wurden und somit kein Zugriff auf den persistenten Speicher stattfinden musste.

StartTime
Gibt die Zeit des letzten reset-Aufrufs an, bei dem die Zähler zurückgesetzt wurden. Die Zähler werden zurückgesetzt, wenn der Server gestartet bzw. erneut gestartet wird.

LastCount
Gibt die Zeit seit der letzten Datenstichprobenentnahme an.

LastTotalGetCount
Gibt die aktuelle Gesamtanzahl der Get-Operationen im Cache abzüglich der Anzahl der Get-Operationen im vorherigen Zeitraum an.

LastTotalHitCount
Gibt die aktuelle Gesamtanzahl der Treffer im Cache abzüglich der Anzahl der Treffer im vorherigen Zeitraum an.

UsedBytes
Zeigt die Speicherbelegung durch diese Map an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

MinUsedBytes
Zeigt die niedrigste Speicherbelegung durch diesen Katalogservice und die zugehörigen Maps an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

MaxUsedBytes
Zeigt die höchste Speicherbelegung durch diesen Katalogservice und die

zugehörigen Maps an. Die Statistiken zur Belegung in Bytes sind nur genau, wenn Sie einfache Objekte oder den Kopiermodus COPY_TO_BYTES verwenden.

LastUsedBytes
Gibt den aktuellen UsedBytes-Wert abzüglich des UsedBytes-Werts aus dem vorherigen Statistikerfassungszeitraum an.

SampleLen
Gibt die Länge (in Millisekunden) des Zeitraums für die Datenstichprobenentnahme an.

ObjectGrid-Statistikprotokoll

Count Gibt die Anzahl der Datenstichproben an, die seit dem Serverstart erfasst wurden. Der Wert 100 gibt beispielsweise an, dass der Eintrag der hundertste Stichprobeneintrag ist, der seit dem Serverstart erfasst wurde.

TimeStamp
Gibt das Datum und die Uhrzeit an, zu der die Statistikmomentaufnahme für das Datengrid erstellt wurde.

OgName
Gibt den Namen des Datengrids an.

PartitionId
Gibt die Partitions-ID an.

Hostname
Gibt den Hostnamen an.

DomainName
Gibt die Katalogservicedomäne an, zu der dieses Datengrid gehört.

MaxTime
Zeigt die Dauer der längsten Transaktion für diesen Katalogserver an.

MinTime
Zeigt die Dauer der kürzesten Transaktion für diesen Katalogserver an.

MeanTime
Gibt die durchschnittliche Dauer einer Transaktion an.

TotalTime
Zeigt die Gesamtzeit an, die für Transaktionen für diesen Katalogserver seit dessen Initialisierung aufgebracht wurde.

AvgTransTime
Zeigt die durchschnittlich erforderliche Zeit für die Fertigstellung einer Transaktion für diesen Katalogserver an.

AvgThroughPut
Zeigt die durchschnittliche Anzahl an Transaktionen pro Sekunde für diesen Katalogserver an.

SumOfSquares
Gibt die Quadratsumme für die Transaktionsdauer an. Mit diesem Wert wird die Abweichung vom Durchschnittswert zu einem bestimmten Zeitpunkt gemessen.

SampleLen
Gibt die Länge (in Millisekunden) des Zeitraums für die Datenstichprobenentnahme an.

LastCount
Gibt die Zeit seit der letzten Datenstichprobenentnahme an.

LastTotalTime
Gibt die aktuelle Gesamtzeit abzüglich der vorherigen Gesamtzeit für die Datenstichprobe an.

StartTime
Gibt die Startzeit der Statistikerfassung seit der letzten Zurücksetzung der Daten an. Die Daten werden zurückgesetzt, wenn der Server erneut gestartet wird.

Überwachung mit der Statistik-API

Die Statistik-API ist die direkte Schnittstelle zur internen Statistikstruktur. Statistiken sind standardmäßig inaktiviert, können aber über die Definition einer Schnittstelle "StatsSpec" aktiviert werden. Eine Schnittstelle "StatsSpec" definiert, wie WebSphere eXtreme Scale Statistiken überwachen soll.

Informationen zu diesem Vorgang

Sie können die lokale API "StatsAccessor" verwenden, um Daten abzufragen und auf Statistiken zu jeder ObjectGrid-Instanz zuzugreifen, die in derselben Java Virtual Machine (JVM) wie der aktive Code ausgeführt wird. Weitere Informationen zu den einzelnen Schnittstellen finden Sie in der API-Dokumentation. Verwenden Sie die folgenden Schritte, um die Überwachung der internen Statistikstruktur zu aktivieren.

Vorgehensweise

1. Rufen Sie das StatsAccessor-Objekt ab. Die Schnittstelle "StatsAccessor" folgt dem Singleton-Muster. Abgesehen von Problemen mit dem Klassenladeprogramm sollte deshalb nur eine einzige StatsAccessor-Instanz für jede JVM vorhanden sein. Diese Klasse dient als Hauptschnittstelle für alle lokalen Statistikoperationen. Der folgende Beispielcode veranschaulicht, wie die Klasse "accessor" abgerufen wird. Rufen Sie diese Operation auf, bevor Sie andere ObjectGrid-Aufrufe absetzen.

```
public class LocalClient {

   public static void main(String[] args) {

      // Handle für StatsAccessor abrufen
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

   }

}
```

2. Definieren Sie die Schnittstelle StatsSpec für das Datengrid. Definieren Sie diese JVM so, dass alle Statistiken nur auf ObjectGrid-Ebene erfasst werden. Sie müssen sicherstellen, dass eine Anwendung alle Statistiken aktiviert, die möglicherweise erforderlich sind, bevor Sie Transaktionen starten. Im folgenden Beispiel wird die Schnittstelle StatsSpec mit einem statischen Konstantenfeld und einer Spezifikationszeichenfolge definiert. Die Verwendung eines statischen Konstantenfelds ist einfacher, weil das Feld bereits die Spezifikation definiert hat. Wenn Sie jedoch eine Spezifikationszeichenfolge verwenden, können Sie jede erforderliche Kombination von Statistiken aktivieren.

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

      // Handle für StatsAccessor abrufen
```

```
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Spezifikation über das statische Feld definieren.
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      // Spezifikation über die Spezifikationszeichenfolge definieren.
      StatsSpec spec = new StatsSpec("og.all=enabled");
      accessor.setStatsSpec(spec);

}
```

3. Senden Sie Transaktionen an das Grid, damit Daten für die Überwachung erfasst werden. Zum Erfassen hilfreicher Daten für Statistiken müssen Sie Transaktionen an das Datengrid senden. Der folgende Codeauszug fügt einen Datensatz in MapA ein, die in ObjectGridA enthalten ist. Da die Statistiken auf ObjectGrid-Ebene erfasst werden, liefert jede Map im ObjectGrid dieselben Ergebnisse.

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

      // Handle für StatsAccessor abrufen
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Spezifikation über das statische Feld definieren.
      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");

      // Einfügenvorgang starten.
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();
}
```

4. Fragen Sie eine StatsFact mit der API "StatsAccessor" ab. Jedem Statistikpfad wird eine Schnittstelle "StatsFact" zugeordnet. Die Schnittstelle StatsFact ist ein generischer Platzhalter, der verwendet wird, um ein StatsModule-Objekt zu organisieren und aufzunehmen. Bevor Sie auf das eigentliche Statistikmodul zugreifen können, muss das StatsFact-Objekt abgerufen werden.

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

      // Handle für StatsAccessor abrufen
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Spezifikation über das statische Feld definieren.
      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");

      // Einfügenvorgang starten.
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // StatsFact abrufen
```

```
      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

}
```

5. Interagieren Sie mit dem StatsModule-Objekt. Das StatsModule-Objekt ist in der Schnittstelle StatsFact enthalten. Sie können eine Referenz auf das Modul über die Schnittstelle StatsFact anfordern. Da die Schnittstelle StatsFact eine generische Schnittstelle ist, müssen Sie das zurückgegebene Modul in den erwarteten StatsModule-Typ umsetzen. Da diese Task eXtreme-Scale-Statistiken erfasst, wird das zurückgegebene StatsModule-Objekt in den Typ "OGStatsModule" umgesetzt. Nach der Umsetzung des Moduls haben Sie Zugriff auf alle verfügbaren Statistiken.

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

      // Handle für StatsAccessor abrufen
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Spezifikation über das statische Feld definieren.
      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");

      // Einfügenvorgang starten.
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // StatsFact abrufen
      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

      // Modul und Zeit abrufen
      OGStatsModule module = (OGStatsModule)fact.getStatsModule();
      ActiveTimeStatistic timeStat =
    module.getTransactionTime("Default", true);
      double time = timeStat.getMeanTime();

}
```

Statistikmodule

WebSphere eXtreme Scale verwendet ein internes Statistikmodell, um Daten zu überwachen und zu filtern. Dieses Modell ist die grundlegende Struktur, die alle Datensichten verwenden, um Momentaufnahmen von Statistiken zu erfassen. Sie können verschiedene Methoden verwenden, um die Informationen von den Statistikmodulen abzurufen.

Übersicht

Statistiken in WebSphere eXtreme Scale werden überwacht und sind in StatsModules-Komponenten enthalten. Im Statistikmodell sind mehrere Typen von Statistikmodulen enthalten:

OGStatsModule
Liefert eine Statistik für eine ObjectGrid-Instanz, einschließlich der Transaktionsantwortzeiten.

MapStatsModule
Liefert eine Statistik für eine einzelne Map, einschließlich der Anzahl an Einträgen und der Trefferrate.

QueryStatsModule
Liefert eine Statistik zu Abfragen, einschließlich der Planerstellung und der Ausführungszeiten.

AgentStatsModule
Liefert eine Statistik zu DataGrid-API-Agenten, einschließlich Serialisierungs- und Ausführungszeiten.

HashIndexStatsModule
Liefert eine Statistik zu den Ausführungszeiten von HashIndex-Abfragen und Wartung.

SessionStatsModule
Liefert eine Statistik zum Plug-in für den HTTP-Sitzungsmanager.

Einzelheiten zu den Statistikmodulen finden Sie unter dem Paket com.ibm.websphere.objectgrid.stats in der API-Dokumentation.

Statistiken in einer lokalen Umgebung

Das Modell ist wie eine n-stufige Baumstruktur organisiert, die sich aus allen in der vorherigen Liste erwähnten StatsModule-Typen zusammensetzt. Aufgrund dieser Organisationsstruktur wird jeder Knoten in der Baumstruktur durch die Schnittstelle "StatsFact" dargestellt. Die Schnittstelle "StatsFact" kann ein einzelnes Modul oder zu Aggregationszwecken eine Gruppe von Modulen darstellen. Wenn beispielsweise mehrere Blattknoten in der Baumstruktur bestimmte MapStatsModule-Objekte darstellen, enthält der übergeordnete StatsFact-Knoten dieser Knoten zusammengefasste Statistiken für alle untergeordneten Module. Nach dem Abruf eines StatsFact-Objekts können Sie die Schnittstelle zum Abrufen des entsprechenden StatsModule verwenden.

Wie in einer Baumstruktur-Map verwenden Sie einen entsprechenden Pfad oder einen Schlüssel, um ein bestimmtes StatsFact abzurufen. Der Pfad ist ein String[]-Wert, der sich aus allen Knoten im Pfad zum angeforderten Fakt zusammengesetzt. Beispiel: Sie haben ein ObjectGrid mit dem Namen "ObjectGridA" erstellt, das zwei Maps enthält: MapA und MapB. Der Pfad zum StatsModule für Map A ist `[ObjectGridA, MapA]`. Der Pfad zu den zusammengefassten Statistiken für beide Maps ist `[ObjectGridA]`.

Statistiken in einer verteilten Umgebung

In einer verteilten Umgebung werden die Statistikmodule über einen anderen Pfad abgerufen. Da ein Server mehrere Partitionen enthalten kann, muss die Statistikbaumstruktur die Partitionen überwachen, zu denen die einzelnen Module gehören. Deshalb ist der Suchpfad für ein bestimmtes StatsFact-Objekt anders. Wenn Sie das vorherige Beispiel erweitern und hinzufügen, dass sich die Maps in Partition 1 befinden, lautet der Pfad zum Abrufen des StatsFact-Objekts für Map A `[1, ObjectGridA, MapA]`.

Überwachung mit dem Dienstprogramm xscmd

Das Dienstprogramm **xscmd** ersetzt das Beispieldienstprogramm **xsadmin** als vollständig unterstütztes Überwachungs- und Verwaltungstool. Mit dem Dienstprogramm **xscmd** können Sie Textinformationen zu Ihrer Topologie von WebSphere eXtreme Scale anzeigen.

Vorbereitende Schritte

- Damit das Dienstprogramm **xscmd** Ergebnisse anzeigt, müssen Sie Ihre Datengridtopologie erstellt haben. Ihre Katalogserver und Container-Server müssen gestartet sein. Weitere Informationen finden Sie unter „Eigenständige Server starten und stoppen" auf Seite 403.
- Weitere Informationen zum Starten des Dienstprogramms **xscmd** finden Sie unter „Verwaltung mit dem Dienstprogramm **xscmd**" auf Seite 423.

Informationen zu diesem Vorgang

Sie können das Dienstprogramm **xscmd** verwenden, um das aktuelle Layout und den jeweiligen Status des Datengrids, z. B. den Map-Inhalt, anzuzeigen. In diesem Beispiel setzt sich das Layout des Datengrids in dieser Aufgabe aus einem einzigen Datengrid (*ObjectGridA*) mit einer einzigen Map (*MapA*) zusammen, die zum Map-Set *MapSetA* gehört. Dieses Beispiel veranschaulicht, wie Sie alle aktiven Container in einem Datengrid anzeigen und gefilterte Metriken zur Map-Größe der Map *MapA* ausgeben. Zum Anzeigen aller möglichen Befehlsoptionen führen Sie das Dienstprogramm **xscmd** ohne Argumente oder mit der Option **-help** aus.

Vorgehensweise

Umgebung mit dem Dienstprogramm **xscmd** überwachen.

- Führen Sie den folgenden Befehl aus, um Statistiken für alle Server zu aktivieren:
  - UNIX `./xscmd.sh -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
  - Windows `xscmd.bat -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
- Führen Sie den folgenden Befehl aus, um alle Online-Container-Server für ein Datengrid anzuzeigen:
  - UNIX `./xscmd.sh -c showPlacement -g ObjectGridA -ms MapSetA`
  - Windows `xscmd.bat -c showPlacement -g ObjectGridA -ms MapSetA`

  Alle Containerinformationen werden angezeigt.

  Achtung: Zum Abrufen dieser Informationen, wenn Transport Layer Security/Secure Sockets Layer (TLS/SSL) aktiviert ist, müssen Sie die Katalog- und Container-Server mit definiertem JMX-Service-Port starten. Für die Definition des JMX-Service-Ports können Sie die Option **-JMXServicePort** im Script **startOgServer** verwenden oder die Methode setJMXServicePort in der Schnittstelle ServerProperties aufrufen.
- Führen Sie den folgenden Befehl aus, um Informationen zu den Maps für das Datengrid "ObjectGridA" anzuzeigen:
  - UNIX `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA`
  - Windows `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA`
- Führen Sie den folgenden Befehl aus, um eine Verbindung zum Katalogservice herzustellen und Informationen zur Map "MapA" für die gesamte Katalogservicedomäne anzuzeigen.

– 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`

Das Dienstprogramm **xscmd** stellt eine Verbindung zum MBean-Server her, der in einem Katalogserver ausgeführt wird. Durch die Herstellung der Verbindung zu einem einzigen Katalogserver können Sie Informationen zur gesamten Katalogservicedomäne abrufen. Ein Katalogserver kann als eigenständiger Prozess, als Prozess von WebSphere Application Server oder integriert in einem angepassten Anwendungsprozess ausgeführt werden. Verwenden Sie die Option **-cep**, um den Hostnamen und den Port des Katalagservers anzugeben. Wenn Sie eine Liste von Katalogservern für die Option **-cep** angeben, müssen die Katalogserver in derselben Katalogservicedomäne enthalten sein. Sie können Statistiken für jeweils eine Katalogservicedomäne abrufen.

- Führen Sie einen der folgenden Befehle aus, um die konfigurierte und die Laufzeitverteilung Ihrer Konfiguration anzuzeigen:
  – `xscmd -c placementServiceStatus`
  – `xscmd -c placementServiceStatus -g ObjectGridA -ms MapSetA`
  – `xscmd -c placementServiceStatus -ms MapSetA`
  – `xscmd -c placementServiceStatus -g ObjectGridA`

  Sie können den Befehl so ausführen, dass Verteilungsinformationen für die gesamte Konfiguration, für ein einzelnes Datengrid, für ein einzelnes MapSet oder für eine Kombination aus einem Datengrid und einem MapSet angezeigt werden.

Überwachung mit WebSphere Application Server PMI

WebSphere eXtreme Scale unterstützt Performance Monitoring Infrastructure (PMI), wenn Sie mit einem Anwendungsserver von WebSphere Application Server oder WebSphere Extended Deployment arbeiten. PMI erfasst Leistungsdaten zu Laufzeitanwendungen und stellt Schnittstellen bereit, über die externe Anwendungen für die Überwachung von Leistungsdaten unterstützt werden. Sie können die Administrationskonsole oder das Tool "wsadmin" verwenden, um auf Überwachungsdaten zuzugreifen.

Vorbereitende Schritte

Sie können PMI verwenden, um Ihre Umgebung zu überwachen, wenn Sie WebSphere eXtreme Scale in Kombination mit WebSphere Application Server verwenden.

Informationen zu diesem Vorgang

WebSphere eXtreme Scale verwendet das angepasste PMI-Feature von WebSphere Application Server, um eine eigene PMI-Instrumentierung hinzuzufügen. Über diesen Ansatz können Sie WebSphere eXtreme Scale PMI mit der Administrationskonsole oder mit JMX-Schnittstellen (Java Management Extensions) im Tool "wsadmin" aktivieren oder inaktivieren. Außerdem können Sie über die Standard-PMI- und -JMX-Schnittstellen, die von Überwachungsstools wie Tivoli Performance Viewer verwendet werden, auf Statistiken von WebSphere eXtreme Scale zugreifen.

Vorgehensweise

1. Aktivieren Sie PMI in eXtreme Scale. Sie müssen PMI aktivieren, um die PMI-Statistiken anzeigen zu können. Weitere Informationen hierzu finden Sie im Abschnitt „PMI aktivieren”.
2. Rufen Sie PMI-Statistiken von eXtreme Scale ab. Zeigen Sie die Leistung Ihrer eXtreme-Scale-Anwendungen mit Tivoli Performance Viewer an. Weitere Informationen hierzu finden Sie im Abschnitt „PMI-Statistiken abrufen” auf Seite 474.

Nächste Schritte

Weitere Informationen zum Tool "wsadmin" finden Sie im Abschnitt „Mit dem Tool "wsadmin" auf Managed Bean (MBeans) zugreifen” auf Seite 443.

PMI aktivieren

Sie können WebSphere Application Server Performance Monitoring Infrastructure (PMI) verwenden, um Statistiken auf jeder Stufe zu aktivieren und zu inaktivieren. So können Sie beispielsweise die Statistik für die Cachetrefferrate einer bestimmten Map aktivieren, und die Statistik für die Eintragsanzahl oder die Statistik für die Loader-Aktualisierungszeiten im Stapelbetrieb inaktivieren. PMI kann über die Administrationskonsole oder mit Scripting aktiviert werden.

Vorbereitende Schritte

Ihr Anwendungsserver muss gestartet sein und eine installierte Anwendung haben, die für eXtreme Scale aktiviert ist. Zum Aktivieren von PMI mit Scripting müssen Sie sich anmelden und das Tool "wsadmin" verwenden können. Weitere Informationen zum Tool "wsadmin" finden Sie im Artikel "Tool 'wsadmin'" im Information Center von WebSphere Application Server.

Informationen zu diesem Vorgang

Verwenden Sie WebSphere Application Server PMI, um einen differenzierten Mechanismus bereitzustellen, mit dem Sie Statistiken auf jeder Stufe aktivieren und inaktivieren können. So können Sie beispielsweise die Statistik für die Cachetreffertrate einer bestimmten Map aktivieren, und die Statistik für die Eintragsanzahl oder die Statistik für die Loader-Aktualisierungszeiten im Stapelbetrieb inaktivieren. In diesem Abschnitt wird beschrieben, wie Sie PMI über die Administrationskonsole und mit wsadmin-Scripts für ObjectGrid aktivieren können.

Vorgehensweise

- **PMI über die Administrationskonsole aktivieren**
  1. Klicken Sie in der Administrationskonsole auf **Überwachung und Optimierung** > **Performance Monitoring Infrastructure** > *Servername*.
  2. Stellen Sie sicher, dass "Performance Monitoring Infrastructure (PMI) aktivieren" ausgewählt ist. Diese Einstellung ist standardmäßig aktiviert. Wenn die Einstellung nicht aktiviert ist, wählen Sie das Kontrollkästchen aus, und starten Sie anschließend den Server erneut.
  3. Klicken Sie auf **Angepasst**. Wählen Sie in der Konfigurationsstruktur das ObjectGrid und das ObjectGrid-Modul "Maps" aus. Aktivieren Sie die Statistik für jedes Modul.

Die Transaktionstypkategorie für ObjectGrid-Statistiken wird zur Laufzeit erstellt. Die Unterkategorien der ObjectGrid- und Map-Statistiken sind nur auf der Registerkarte **Laufzeit** sichtbar.

- **PMI mit Scripting aktivieren**
  1. Öffnen Sie eine Befehlszeile. Navigieren Sie zum Verzeichnis *WAS-Stammverzeichnis*/bin. Geben Sie **wsadmin** ein, um das Befehlszeilentool "wsadmin" zu starten.
  2. Ändern Sie die PMI-Laufzeitkonfiguration für eXtreme Scale. Stellen Sie mit den folgenden Befehlen sicher, dass PMI für den Server aktiviert ist:

     ```
     wsadmin>set s1 [$AdminConfig getid /Cell:ZELLENNAME/Node:KNOTENNAME/
         Server:ANWENDUNGSSERVERNAME/]
     wsadmin>set pmi [$AdminConfig list PMIService $s1]
     wsadmin>$AdminConfig show $pmi.
     ```

     Wenn PMI nicht aktiviert ist, führen Sie die folgenden Befehle aus, um PMI zu aktivieren:

     ```
     wsadmin>$AdminConfig modify $pmi {{enable true}}
     wsadmin>$AdminConfig save
     ```

     Wenn Sie PMI aktivieren müssen, starten Sie den Server erneut.
  3. Setzen Sie unter Verwendung der folgenden Befehle Variablen, um die Statistikgruppe in eine angepasste Gruppe zu ändern:

     ```
     wsadmin>set perfName [$AdminControl completeObjectName type=Perf,
     process=ANWENDUNGSSERVERNAME,*]
     wsadmin>set perfOName [$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. Setzen Sie die Statistikgruppe mit dem folgenden Befehl auf "custom" (Angepasst):

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName setStatisticSet $params $sigs
     ```
  5. Setzen Sie mit den folgenden Befehlen Variablen, um die PMI-Statistik "objectGridModule" zu aktivieren:

     ```
     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. Setzen Sie die Statistikzeichenfolge mit dem folgenden Befehl:

     ```
     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. Setzen Sie die Statistikzeichenfolge mit dem folgenden Befehl:

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName setCustomSetString $params2 $sigs2
     ```

  Mit diesen Schritten haben Sie PMI für die Laufzeitumgebung von eXtreme Scale aktiviert, aber die PMI-Konfiguration nicht geändert. Wenn Sie den Anwendungsserver erneut starten, gehen die PMI-Einstellungen bis auf die eigentliche Aktivierung von PMI verloren.

Beispiel

Sie können die folgenden Schritte ausführen, um die PMI-Statistiken für die Beispielanwendung zu aktivieren:

1. Starten Sie die Anwendung mit der Webadresse `http://Host:Port/ObjectGridSample`, wobei "Host" und "Port" für den Hostnamen und die HTTP-Portnummer des Servers stehen, in dem die Beispielanwendung installiert ist.
2. Klicken Sie in der Beispielanwendung auf "ObjectGridCreationServlet" und anschließend auf die Aktionsschaltflächen 1, 2, 3, 4 und 5, um Aktionen für das ObjectGrid und die Maps zu generieren. Schließen Sie diese Servlet-Seite noch nicht.
3. Klicken Sie in der Administrationskonsole auf **Überwachung und Optimierung** > **Performance Monitoring Infrastructure** > *Servername*. Klicken Sie auf das Register **Laufzeit**.
4. Klicken Sie auf das Optionsfeld **Angepasst**.
5. Erweitern Sie das ObjectGrid-Modul "Maps" in der Laufzeitstruktur, und klicken Sie anschließend auf den Link "clusterObjectGrid". In der ObjectGrid-Gruppe "Maps" gibt es eine ObjectGrid-Instanz "clusterObjectGrid", und in der Gruppe "clusterObjectGrid" gibt es vier Maps: counters, employees, offices und sites. Die ObjectGrid-Instanz enthält die clusterObjectGrid-Instanz, und diese Instanz hat den Transaktionstyp DEFAULT.
6. Sie können die gewünschten Statistiken aktivieren. Sie können beispielsweise die Statistik für die Anzahl der Einträge in der Map "employees" und die Statistik für die Transaktionsantwortzeiten für den Transaktionstyp DEFAULT aktivieren.

Nächste Schritte

Nach der Aktivierung von PMI können Sie die PMI-Statistiken über die Administrationskonsole oder mit Scripting anzeigen.

PMI-Statistiken abrufen

Wenn Sie PMI-Statistiken abrufen, können Sie sich einen Eindruck über die Leistung Ihrer eXtreme-Scale-Anwendungen verschaffen.

Vorbereitende Schritte

- Aktivieren Sie die Verfolgung von PMI-Statistiken für Ihre Umgebung. Weitere Informationen hierzu finden Sie im Abschnitt „PMI aktivieren” auf Seite 472.
- Bei den Pfadangaben in dieser Task wird davon ausgegangen, dass Sie Statistiken für die Beispielanwendung abrufen, aber Sie können diese Statistiken mit ähnlichen Schritten für jede andere Anwendung verwenden.
- Wenn Sie die Administrationskonsole verwenden, um Statistiken abzurufen, müssen Sie sich an der Administrationskonsole anmelden können. Wenn Sie Scripting verwenden, müssen Sie sich bei wsadmin anmelden können.

Informationen zu diesem Vorgang

Sie können PMI-Statistiken abrufen und diese in Tivoli Performance Viewer anzeigen, indem Sie Schritte in der Administrationskonsole oder mit Scripting ausführen.

- Schritte für die Administrationskonsole
- Schritte für Scripting

Weitere Informationen zu den Statistiken, die abgerufen werden können, finden Sie im Abschnitt „PMI-Module” auf Seite 476.

Vorgehensweise

- Rufen Sie PMI-Statistiken in der Administrationskonsole ab.
  1. Klicken Sie in der Administrationskonsole auf **Überwachung und Optimierung** > **Performance Viewer** > **Aktuelle Aktivität**.
  2. Wählen Sie den Server, den Sie mit Tivoli Performance Viewer überwachen möchten, aus, und aktivieren Sie anschließend die Überwachung.
  3. Klicken Sie auf den Server, um die Seite "Performance Viewer" anzuzeigen.
  4. Erweitern Sie die Konfigurationsstruktur. Klicken Sie auf **ObjectGrid-Maps** > **clusterObjectGrid**, und wählen Sie **employees** aus. Klicken Sie auf **ObjectGrids** > **clusterObjectGrid**, und wählen Sie **DEFAULT** aus.
  5. Navigieren Sie in der ObjectGrid-Beispielanwendung zum Servlet "ObjectGridCreationServlet", klicken Sie auf Schaltfläche 1, und füllen Sie die Maps mit Daten. Sie können die Statistiken im Viewer anzeigen.
- Rufen Sie PMI-Statistiken mit Scripting ab.
  1. Navigieren Sie in einer Befehlszeile zum Verzeichnis *WAS-Stammverzeichnis*/`bin`. Geben Sie `wsadmin` ein, um das Tool "wsadmin" zu starten.
  2. Setzen Sie mit den folgenden Befehlen Variablen für die Umgebung:

     ```
     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. Setzen Sie mit den folgenden Befehlen Variablen, um die Statistik "mapModule" abzurufen:

     ```
     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. Rufen Sie die Statistik "mapModule" mit dem folgenden Befehl ab:

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params $sigs
     ```

  5. Setzen Sie mit den folgenden Befehlen Variablen, um die Statistik "objectGridModule" abzurufen:

     ```
     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. Rufen Sie die Statistik "objectGridModule" mit dem folgenden Befehl ab:

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params2 $sigs2
     ```

Ergebnisse

Sie können Statistiken in Tivoli Performance Viewer anzeigen.

PMI-Module

Sie können die Leistung Ihrer Anwendungen mit den PMI-Modulen (Performance Monitoring Infrastructure) überwachen.

objectGridModule

Das Modul "objectGridModule" enthält eine Zeitstatistik: Antwortzeit der Transaktion. Eine Transaktion ist wie folgt definiert: Dauer zwischen dem Aufruf der Methode "Session.begin" und dem Aufruf der Methode "Session.commit". Diese Dauer wird als Antwortzeit der Transaktion verfolgt. Das Stammelement ("root") des Moduls "the objectGridModule" dient als Einstiegspunkt für die Statistiken von WebSphere eXtreme Scale. Dieses Stammelement hat ObjectGrids als untergeordnete Elemente, die dieselben Transaktionstypen wie deren untergeordnete Elemente haben. Die Antwortzeitstatistik wird jedem Transaktionstyp zugeordnet.

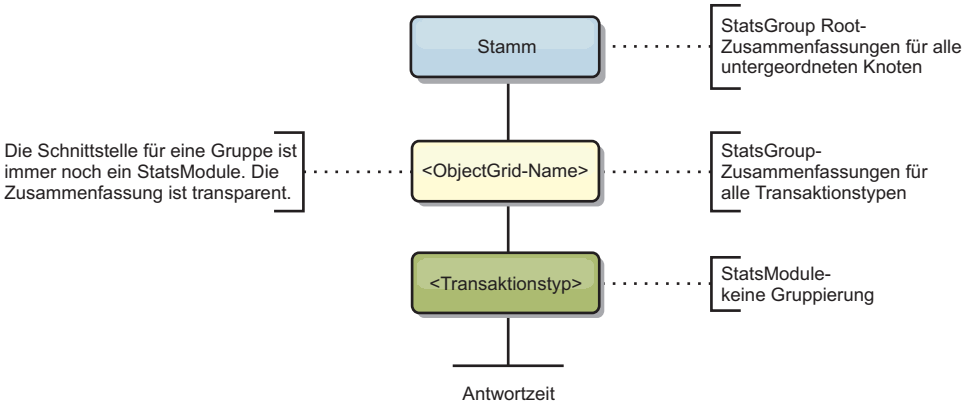


Abbildung 57. Struktur des Moduls "ObjectGridModule"

Die folgende Abbildung zeigt eine ObjectGridModule-Beispielstruktur. In diesem Beispiel sind zwei ObjectGrid-Instanzen im System vorhanden: ObjectGrid A und ObjectGrid B. Die ObjectGrid-Instanz A hat zwei Typen von Transaktionen: A und Standard. Die ObjectGrid-Instanz B hat nur den Transaktionstyp "Standard".

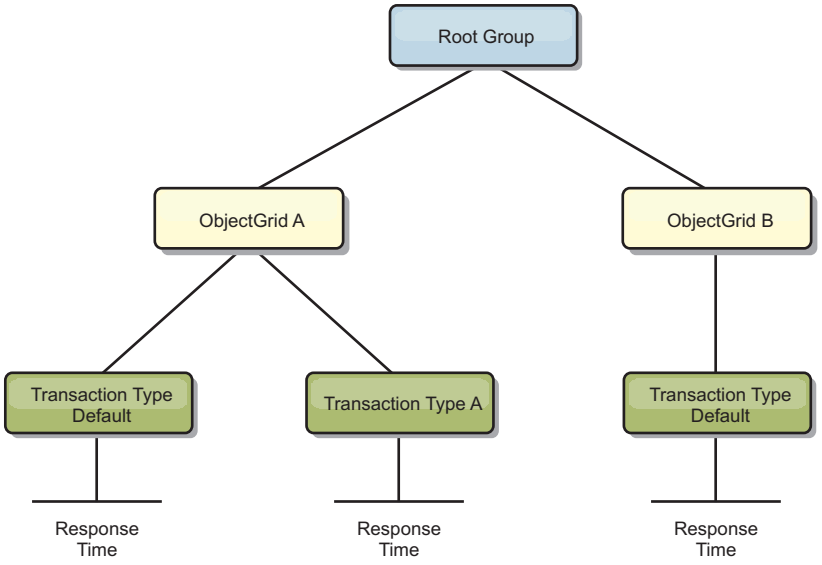


Abbildung 58. Beispielstruktur für das Modul "ObjectGridModule"

Transaktionstypen werden von Anwendungsentwicklern definiert, weil sie wissen, welche Typen von Transaktionen in ihren Anwendungen verwendet werden. Der Transaktionstyp wird mit der folgenden Methode "Session.setTransactionType(String)" gesetzt:

```
/**
* Legt den Transaktionstyp für künftige Transaktionen fest.
*
* Nach dem Aufruf dieser Methode haben alle künftigen Transaktionen denselben
* Typ, bis ein anderer Transaktionstyp festgelegt wird. Wenn Sie keinen
* Transaktionstyp festlegen, wird der Standardtransaktionstyp TRANSACTION_TYPE_DEFAULT
* verwendet.
*
* Transaktionstypen werden hauptsächlich für die Verfolgung statistischer Daten
* verwendet. Benutzer können Typen von Transaktionen, die in einer Anwendung
* ausgeführt werden, vordefinieren. Die Idee ist, Transaktionen mit denselben
* Merkmalen in einer Kategorie (Typ) zusammenzufassen, so dass jeweils eine
* einzige Statistik zur Transaktionsantwortzeit verwendet werden kann, um den
* jeweiligen Transaktionstyp zu verfolgen.
*
* Diese Verfolgung ist hilfreich, wenn Ihre Anwendung unterschiedliche
* Transaktionstypen hat.
* Einige Typen von Transaktionen, wie z. B. Aktualisierungstransaktionen,
* haben eine längere Verarbeitungszeit als andere Transaktionen, wie z. B.
* Lesetransaktionen. Wenn Sie den Transaktionstyp verwenden, können
* unterschiedliche Transaktionen über unterschiedliche Statistiken verfolgt
* werden, so dass die Statistiken hilfreicher sind.
*
* @param tranType Der Transaktionstyp für künftige Transaktionen.
*/
void setTransactionType(String tranType);
```

Im folgenden Beispiel wird der Transaktionstyp auf `updatePrice` gesetzt:

```
// Transaktionstyp auf "updatePrice" setzen.
// Die Zeit zwischen session.begin() und session.commit() wird in der
// Zeitstatistik für "updatePrice" verfolgt.
session.setTransactionType("updatePrice");
session.begin();
map.update(stockId, new Integer(100));
session.commit();
```

Die erste Zeile gibt an, dass der folgende Transaktionstyp "updatePrice" ist. In dem Beispiel ist eine Statistik "updatePrice" in der ObjectGrid-Instanz vorhanden, die dem Session-Objekt entspricht. Mit JMX-Schnittstellen (Java Management Extensions) können Sie die Transaktionsantwortzeit für Transaktionen des Typs "updatePrice" abrufen. Sie können auch die zusammengefasste Statistik für alle Transaktionstypen in der angegebenen ObjectGrid-Instanz abrufen.

mapModule

Das Modul "mapModule" enthält drei Statistiken, die sich auf eXtreme-Scale-Maps beziehen:

- **Map hit rate** - *BoundedRangeStatistic*: Verfolgt die Trefferrate einer Map. Die Trefferrate ist ein variabler Wert zwischen 0 und 100 einschließlich, der den Prozentsatz der Map-Treffer in Relation zu den get-Operationen für die Map darstellt.
- **Number of entries**-*CountStatistic*: Verfolgt die Anzahl der Einträge in der Map.
- **Loader batch update response time**-*TimeStatistic*: Verfolgt die Antwortzeit für Aktualisierungsoperationen im Stapelbetrieb des Loaders.

Das Stammelement ("root") des Moduls "mapModule" dient als Einstiegspunkt für die ObjectGrid-Map-Statistiken. Dieses Stammelement hat Maps als untergeordnete Elemente, die dieselben Transaktionstypen wie deren untergeordnete Elemente haben. Jede Map-Instanz hat die drei aufgelisteten Statistiken. Die Struktur des Moduls "mapModule" ist in der folgenden Abbildung dargestellt:

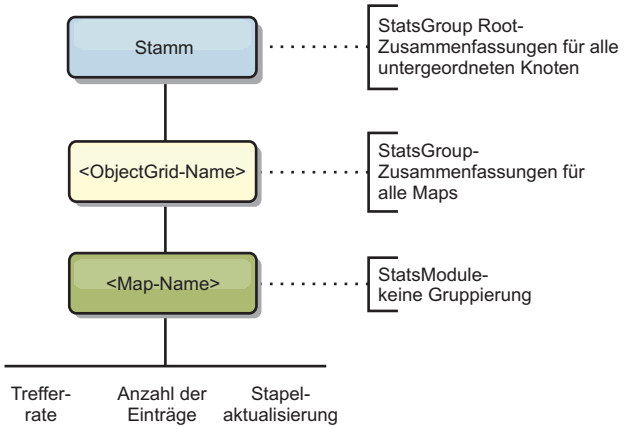
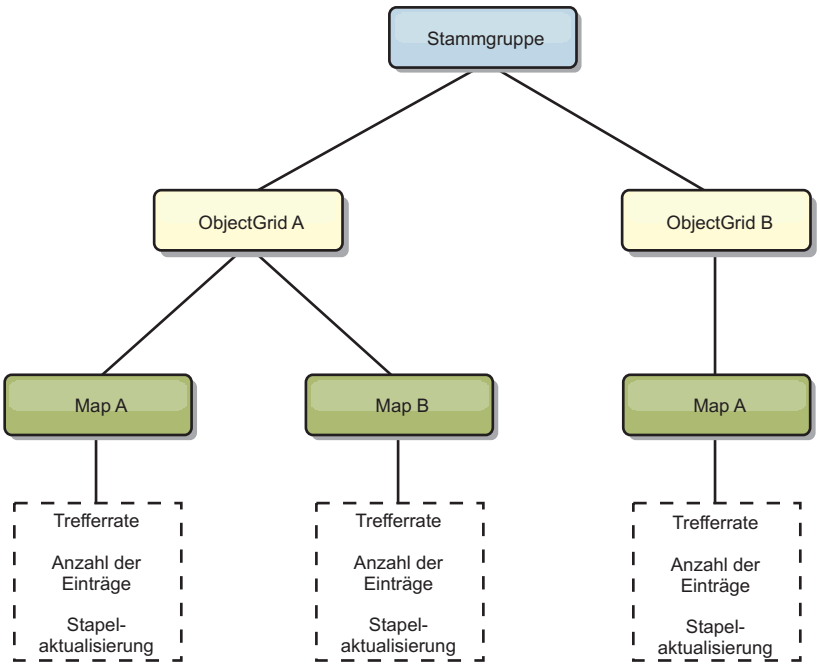


Abbildung 59. Struktur des Moduls "mapModule"

Die folgende Abbildung zeigt eine Beispielstruktur für das Modul "mapModule":

Abbildung 60. Beispielstruktur für das Modul "mapModule"



Modul "hashIndexModule"

Das Modul "hashIndexModule" enthält die folgenden Statistiken, die sich auf Indizes auf Map-Ebene beziehen:

- **Find Count**-*CountStatistic*: Die Anzahl der Aufrufe für die Indexoperation "find".
- **Collision Count**-*CountStatistic*: Die Anzahl der Kollisionen für die Operation "find".
- **Failure Count**-*CountStatistic*: Die Anzahl der Fehler für die Operation "find".
- **Result Count**-*CountStatistic*: Die Anzahl der von der Operation "find" zurückgegebenen Schlüssel.
- **BatchUpdate Count**-*CountStatistic*: Die Anzahl der für diesen Index ausgeführten Aktualisierungen im Stapelbetrieb. Wenn die entsprechende Map geändert wird, wird die Methode "doBatchUpdate()" des Index aufgerufen. Diese Statistik teilt Ihnen mit, wie oft sich der Index ändert bzw. aktualisiert wird.
- **Find Operation Duration Time**-*TimeStatistic*: Ausführungsdauer der Operation "find".

Das Stammelement ("root") des Moduls "hashIndexModule" "root" dient als Einstiegspunkt für die HashIndex-Statistik. Das Stammelement hat ObjectGrids als untergeordnete Elemente, ObjectGrids haben Maps als untergeordnete Elemente, die wiederum HashIndex-Instanzen als untergeordnete Elemente und Blattknoten der Baumstruktur haben. Jede HashIndex-Instanz hat die drei aufgelisteten Statistiken. Die Struktur des Moduls "hashIndexModule" wird in der folgenden Abbildung gezeigt:

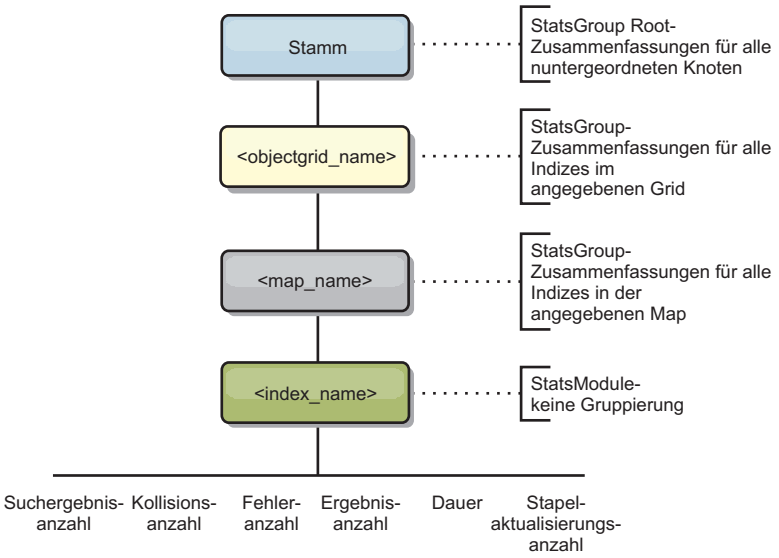


Abbildung 61. Struktur des Moduls "hashIndexModule"

Die folgende Abbildung zeigt eine Beispielstruktur für das Modul "hashIndexModule":

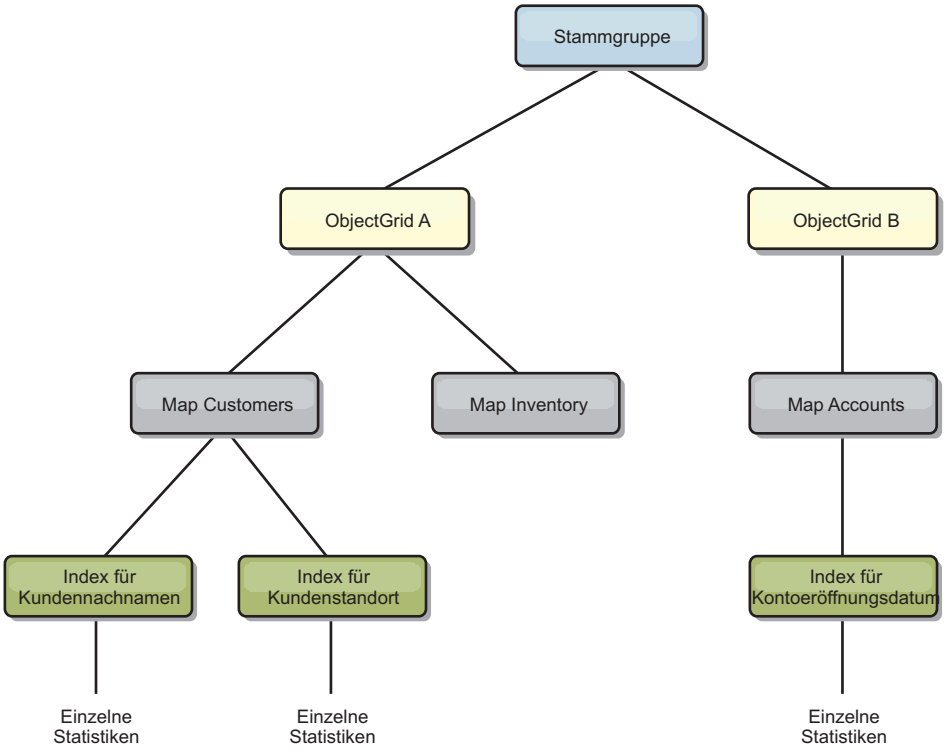


Abbildung 62. Beispielstruktur für das Modul "hashIndexModule"

Modul "agentManagerModule"

Das Modul "agentManagerModule" enthält Statistiken, die sich auf die Agenten auf Map-Ebene beziehen:

- **Reduce Time**: *TimeStatistic* - Die Zeit, die der Agent für die Ausführung der Operation "reduce" benötigt.

- **Total Duration Time**: *TimeStatistic* - Die Gesamtzeit, die der Agent für die Ausführung aller Operationen benötigt.
- **Agent Serialization Time**: *TimeStatistic* - Die Zeit für die Serialisierung des Agenten.
- **Agent Inflation Time**: *TimeStatistic* - Die Zeit, die benötigt wird, um den Agenten im Server zu dekomprimieren.
- **Result Serialization Time**: *TimeStatistic* - Die Zeit für die Serialisierung der Ergebnisse des Agenten.
- **Result Inflation Time**: *TimeStatistic* - Die Zeit für die Dekomprimierung der Ergebnisse des Agenten.
- **Failure Count**: *CountStatistic* - Die Anzahl der Fehler des Agenten.
- **Invocation Count**: *CountStatistic* - Die Anzahl der AgentManager-Aufrufe.
- **Partition Count**: *CountStatistic* - Die Anzahl der Partitionen, an die der Agent gesendet wird.

Das Stammelement ("root" des Moduls "agentManagerModule") dient als Einstiegspunkt für die AgentManager-Statistiken. Dieses Stammelement hat ObjectGrids als untergeordnete Elemente, die ObjectGrids haben Maps als untergeordnete Elemente, die wiederum AgentManager-Instanzen als untergeordnete Elemente und Blattknoten der Baumstruktur haben. Jede AgentManager-Instanz hat Statistiken.

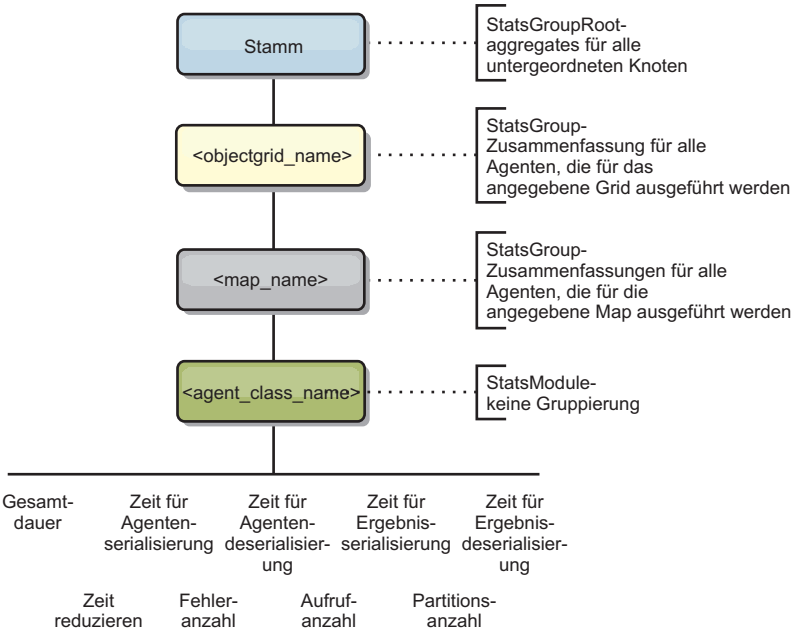


Abbildung 63. Struktur des Moduls "agentManagerModule"

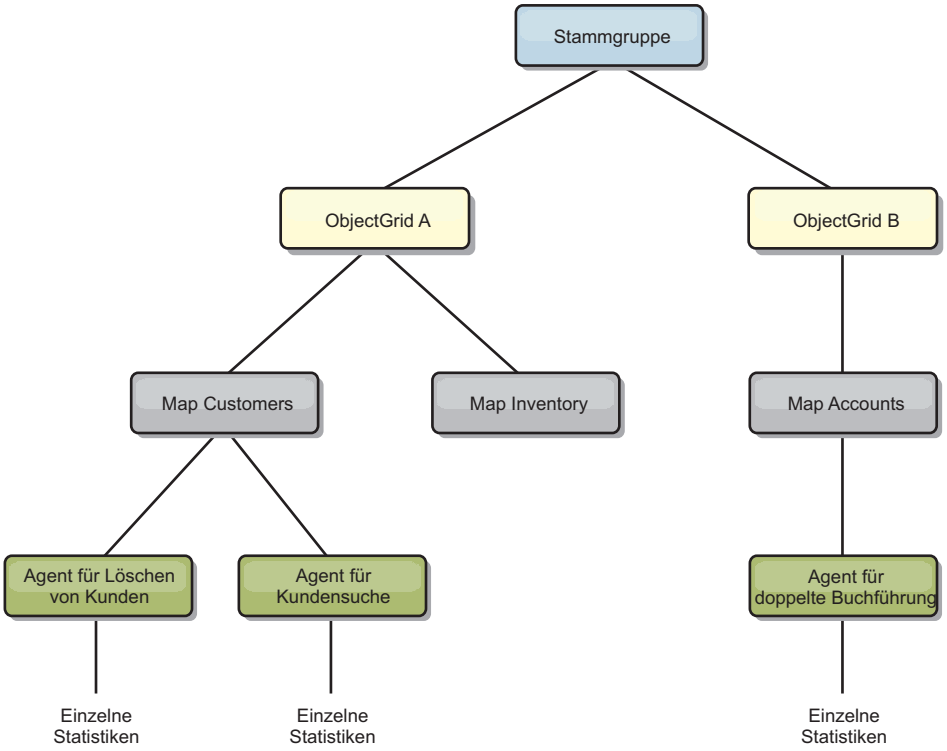


Abbildung 64. Beispielstruktur für das Modul "agentManagerModule"

Modul "queryModule"

Das Modul "queryModule" enthält Statistiken, die sich auf eXtreme-Scale-Abfragen beziehen:

- **Plan Creation Time**: *TimeStatistic* - Die Zeit für die Erstellung des Abfrageplans.
- **Execution Time**: *TimeStatistic* - Die Zeit für die Ausführung der Abfrage.
- **Execution Count**: *CountStatistic* - Die Anzahl der Abfrageläufe.
- **Result Count**: *CountStatistic* - Der Zähler für jede Ergebnismenge jedes Abfragelaufs.
- **FailureCount**: *CountStatistic* - Die Anzahl der Abfragefehler.

Das Stammelement ("root") des Moduls "queryModule" dient als Einstiegspunkt für die Abfragestatistiken. Dieses Stammelement hat ObjectGrids als untergeordnete Elemente, die Query-Objekt als untergeordnete Elemente und Blattknoten der Baumstruktur haben. Jede Query-Instanz hat die drei aufgelisteten Statistiken.

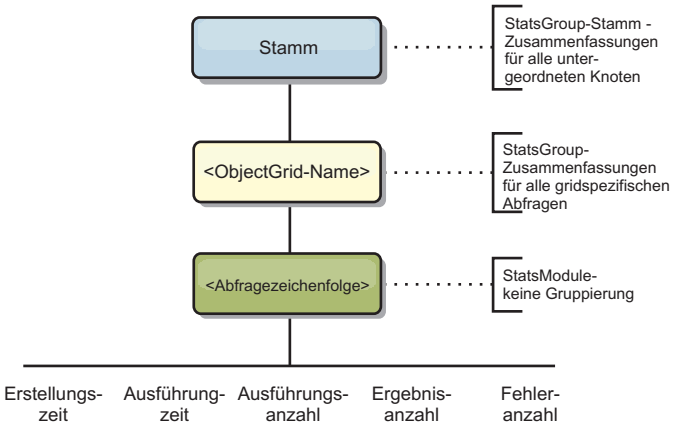


Abbildung 65. Struktur des Moduls "queryModule"

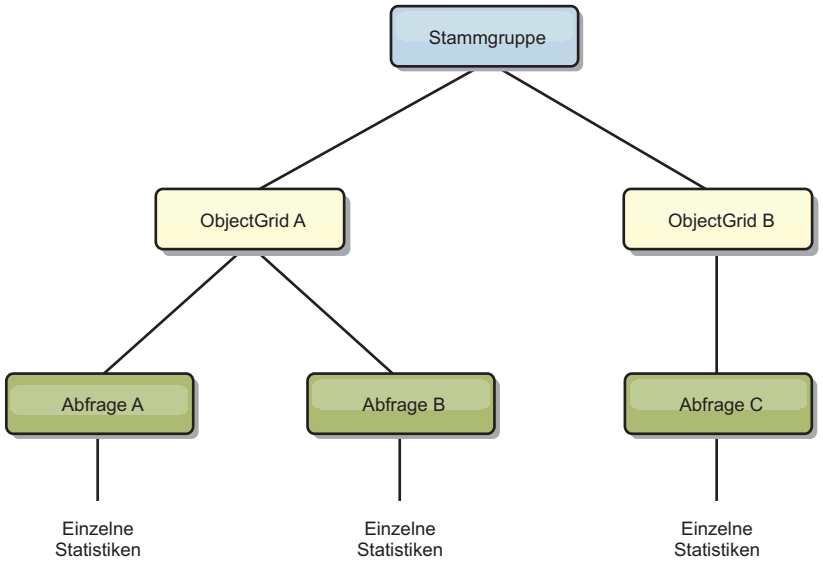


Abbildung 66. Beispielstruktur für das Modul "queryModule"

Mit dem Tool "wsadmin" auf Managed Bean (MBeans) zugreifen

Sie können das in WebSphere Application Server bereitgestellte Dienstprogramm "wsadmin" verwenden, um auf MBean-Informationen zuzugreifen.

Vorgehensweise

Führen Sie das Tool "wsadmin" im Verzeichnis `bin` Ihrer Installation von WebSphere Application Server aus. Im folgenden Beispiel wird eine Sicht der aktuellen Shard-Verteilung in einer dynamischen eXtreme-Scale-Umgebung abgerufen. Sie können das Tool "wsadmin" in jeder Installation ausführen, in der eXtreme Scale ausgeführt wird. Das Tool "wsadmin" muss nicht im Katalogservice ausgeführt werden.

```
$ 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>
```

Überwachung mit Managed Beans (MBeans)

Sie können Managed Beans (MBeans) verwenden, um Statistiken in Ihrer Umgebung zu verfolgen.

Vorbereitende Schritte

Für die aufzuzeichnenden Attribute müssen Sie die Statistiken aktivieren. Sie können Statistiken mit den folgenden Methoden aktivieren:

- **Über die Servereigenschaftendatei:**

  Sie können Statistiken in der Servereigenschaftendatei mit einem Schlüssel/Wert-Eintrag im Format `statsSpec=<StatsSpec>` aktivieren. Es folgenden verschiedene Beispiele für mögliche Einstellungen:

  – Zum Aktivieren aller Statistiken verwenden Sie `statsSpec=all=enabled`.

  – Wenn Sie nur ObjectGrid-Statistiken aktivieren möchten, verwenden Sie `statsSpec=og.all=enabled`. Eine Beschreibung aller möglichen Statistikspezifikationen finden Sie in der API-Dokumentation zur APi "StatsSpec".

  Weitere Informationen zur Servereigenschaftendatei finden Sie im Abschnitt Servereigenschaftendatei.

- **Mit einer Managed Bean:**

  Sie können Statistiken mit dem Attribut StatsSpec in der ObjectGrid-MBean aktivieren. Weitere Informationen finden Sie in der API-Dokumentation zur API StatsSpec.

- **Programmgesteuert:**

  Sie können Statistiken auch programmgesteuert mit der Schnittstelle "StatsAccessor" aktivieren, die über die Klasse " StatsAccessorFactory" abgerufen wird. Verwenden Sie diese Schnittstelle in einer Clientumgebung, oder wenn Sie ein Datengrid überwachen müssen, das im aktuellen Prozess ausgeführt wird.

Vorgehensweise

- **Zugriff auf MBean-Statistiken mit dem Tool "wsadmin".**

  Weitere Informationen finden Sie unter „Mit dem Tool "wsadmin" auf Managed Bean (MBeans) zugreifen” auf Seite 443.

- **Programmgesteuerter Zugriff auf MBean-Statistiken.**

  Weitere Informationen finden Sie unter „Über das Programm auf Managed Beans (MBeans) zugreifen” auf Seite 443.

Beispiel

Ein Beispiel für die Verwendung von Managed Beans finden Sie im Abschnitt Beispiel: Dienstprogramm **xsadmin**.

Überwachung mit Tools eines anderen Anbieters

WebSphere eXtreme Scale kann mit verschiedenen gängigen Lösungen für die Unternehmensüberwachung überwacht werden. Es werden Plug-in-Agenten für IBM Tivoli Monitoring and Hyperic HQ bereitgestellt, die WebSphere eXtreme Scale mit Hilfe öffentlich zugänglicher Management-Beans überwachen. CA Wily Introscope verwendet die Java-Methodeninstrumentierung, um Statistiken zu erfassen.

Überwachung mit IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale

IBM Tivoli Enterprise Monitoring Agent ist eine mit vielen Funktionen ausgestattete Überwachungslösung, die Sie verwenden können, um Datenbanken, Betriebssysteme und Server in verteilten Umgebungen und in Hostumgebungen zu überwachen. WebSphere eXtreme Scale enthält einen angepassten Agenten, den Sie verwenden können, um eXtreme-Scale-Management-Beans selbst zu überwachen. Diese Lösung funktioniert effizient in eigenständigen Implementierungen von eXtreme Scale und in Implementierungen von eXtreme Scale mit WebSphere Application Server.

Vorbereitende Schritte

- Installieren Sie WebSphere eXtreme Scale Version 7.0.0 oder höher.

  Außerdem müssen Statistiken aktiviert werden, um statistische Daten von eXtreme-Scale-Servern zu erfassen. Verschiedene Optionen zum Aktivieren von Statistiken sind in „Überwachung mit Managed Beans (MBeans)" auf Seite 484 und Beispiel: Dienstprogramm **xsadmin** beschrieben.
- Installieren Sie IBM Tivoli Monitoring Version 6.2.1 mit Fixpack 2 oder höher.
- Installieren Sie den Tivoli-Betriebssystemagenten auf jedem Server oder Host, auf dem eXtreme-Scale-Server ausgeführt werden.
- Installieren Sie den WebSphere eXtreme Scale-Agenten, den Sie kostenlos von der Website von IBM Open Process Automation Library (OPAL) herunterladen können.

Führen Sie die folgenden Schritte aus, um Tivoli Monitoring Agent zu installieren und zu konfigurieren:

Vorgehensweise

1. Installieren Sie Tivoli Monitoring Agent for WebSphere eXtreme Scale.

   Laden Sie das Tivoli-Installations-Image herunter, und entpacken Sie die Dateien in einem temporären Verzeichnis.
2. Installieren Sie die Unterstützungsdateien für eXtreme-Scale-Anwendungen.

   Installieren Sie die eXtreme-Scale-Anwendungsunterstützung in jeder der folgenden Implementierungen:
   - Tivoli Enterprise Portal Server (TEPS)
   - Enterprise Desktop Client (TEPD)
   - Tivoli Enterprise Monitoring Server (TEMS)

   a. Starten Sie in dem erstellten temporären Verzeichnis ein neues Befehlsfenster, und führen Sie die entsprechende ausführbare Datei für Ihre Plattform

aus. Das Installationsscript erkennt den Typ Ihrer Tivoli-Implementierung (TEMS, TEPD oder TEPS) automatisch. Sie können jeden beliebigen Typ auf einem einzelnen oder auf mehreren Hosts installieren, und alle drei Implementierungstypen erfordern die Installation der Unterstützungsdateien für die eXtreme-Scale-Agentenanwendung.

b. Vergewissern Sie sich, dass die ausgewählten Optionen für die implementierten Tivoli-Komponenten im Fenster des Installationsprogramms korrekt sind. Klicken Sie auf **Next**.

c. Geben Sie auf Anforderung Ihren Hostnamen und Ihre Verwaltungsberechtigungsnachweise an. Klicken Sie auf **Next**.

d. Wählen Sie **Monitoring Agent for WebSphere eXtreme Scale** aus. Klicken Sie auf **Next**.

e. Sie werden über die auszuführenden Installationsaktionen benachrichtigt. Klicken Sie auf **Next**. Der Fortschritt der Installation wird bis zum Ende hin angezeigt.

Nach der Ausführung dieser Prozedur sind alle erforderlichen Anwendungsunterstützungsdateien für den eXtreme-Scale-Agenten installiert.

3. Installieren Sie den Agenten auf jedem eXtreme-Scale-Knoten.

   Sie installieren einen Tivoli-Betriebssystemagenten auf jedem Computer. Sie müssen diesen Agenten nicht konfigurieren oder starten. Verwenden Sie dasselbe Installations-Image, das Sie bereits im vorherigen Schritt verwendet haben, um die plattformspezifische ausführbare Datei auszuführen.

   Als Richtlinie können Sie verwenden, dass nur ein einziger Agent pro Host installiert werden muss. Jeder Agent kann mehrere Instanzen von eXtreme-Scale-Servern unterstützen. Um die beste Leistung zu erzielen, verwenden Sie eine Agenteninstanz für ungefähr 50 eXtreme-Scale-Server.

   a. Klicken Sie in der Eingangsanzeige des Installationsassistenten auf **Next**, um die Anzeige zu öffnen, in der Sie Informationen zum Installationspfad angeben.

   b. Geben Sie im Feld **Tivoli Monitoring installation directory** einen Wert ein, oder navigieren Sie zum Verzeichnis `C:\IBM\ITM` (oder `/opt/IBM/ITM`). Vergewissern Sie sich, dass der im Feld **Location for installable media** angezeigte Wert korrekt ist, und klicken Sie auf **Next**.

   c. Wählen Sie die Komponenten aus, die Sie hinzufügen möchten, z. B. **Perform a local install of the solution**, und klicken Sie auf **Next**.

   d. Wählen Sie die Anwendungen aus, für die Sie die Unterstützung hinzufügen möchten, z. B. **Monitoring Agent for WebSphere eXtreme Scale**, und klicken Sie auf **Next**.

   e. Der Fortschritt der Installation wird angezeigt, bis alle Anwendungsunterstützungsdateien erfolgreich hinzugefügt wurden.

   Anmerkung: Wiederholen Sie diese Schritte auf jedem eXtreme-Scale-Knoten. Sie können auch eine unbeaufsichtigte Installation durchführen. Weitere Informationen zur unbeaufsichtigten Installation finden Sie im Information Center von IBM Tivoli Monitoring.

4. Konfigurieren Sie den WebSphere eXtreme Scale-Agenten.

   Jeder installierte Agent muss für die Überwachung eines Katalogservers und oder eXtreme-Scale-Servers konfiguriert werden.

   Die Schritte zum Konfigurieren von Windows- und UNIX-Plattformen sind verschieden. Die Konfiguration für die Windows-Plattform erfolgt über die Benutzerschnittstelle **Manage Tivoli Monitoring Services**. Die Konfiguration für UNIX-Plattformen ist befehlszeilenbasiert.

Windows Verwenden Sie die folgenden Schritte, um den Agenten anfänglich unter Windows zu konfigurieren.

a. Klicken Sie im Fenster **Manage Tivoli Enterprise Monitoring Services** auf **Start** > **All Programs** > **IBM Tivoli Monitoring** > **Manage Tivoli Monitoring Services**.
b. Klicken Sie mit der rechten Maustaste auf **Monitoring Agent for WebSphere eXtreme Scale**, und wählen Sie die Option **Configure using default** aus, woraufhin ein Fenster erscheint, in dem Sie eine eindeutige Instanz des Agenten erstellen können.
c. Wählen Sie einen eindeutigen Namen aus, z. B. `instance1`, und klicken Sie auf **Next**.

- Wenn Sie eigenständige eXtreme-Scale-Server überwachen möchten, führen Sie die folgenden Schritte aus:
  a. Aktualisieren Sie die Java-Parameter, und stellen Sie sicher, dass der Wert für **Java Home** korrekt ist. JVM-Argumente können leer bleiben. Klicken Sie auf **Next**.
  b. Wählen Sie für **MBean server connection type** einen Typ aus. Verwenden Sie `JSR-160-Complaint Server` für eigenständige eXtreme-Scale-Server. Klicken Sie auf **Next**.
  c. Wenn die Sicherheit aktiviert ist, aktualisieren Sie die Werte in den Feldern **User ID** und **Password**. Übernehmen Sie den Wert im Feld **JMX service URL**. Sie überschreiben diesen Wert später. Übernehmen Sie den Wert im Feld **JMX Class Path Information**. Klicken Sie auf **Next**.

  Zum Konfigurieren der Server für den Agenten unter Windows führen Sie die folgenden Schritte aus:
  a. Konfigurieren Sie Unterknoteninstanzen von eXtreme-Scale-Servern im Teilfenster **WebSphere eXtreme Scale Grid Servers**. Wenn keine Container-Server auf Ihrem Computer vorhanden sind, klicken Sie auf **Next**, um das Teilfenster für den Katalogservice aufzurufen.
  b. Wenn mehrere eXtreme-Scale-Container-Server auf Ihrem Computer vorhanden sind, konfigurieren Sie den Agenten so, dass jeder einzelne Server überwacht wird.
  c. Sie können so viele eXtreme-Scale-Server hinzufügen, wie Sie benötigen, sofern ihre Namen und Ports eindeutig sind. Klicken Sie dazu auf **New**. (Wenn ein eXtreme-Scale-Server gestartet wird, muss ein Wert für den JMX-Port angegeben werden.)
  d. Nach der Konfiguration der Container-Server klicken Sie auf **Next**. Daraufhin wird das Teilfenster **WebSphere eXtreme Scale Catalog Servers** angezeigt.
  e. Wenn Sie keine Katalogserver haben, klicken Sie auf **OK**. Wenn Sie Katalogserver haben, fügen Sie eine neue Konfiguration für jeden Server hinzu, so wie Sie es für die Container-Server getan haben. Wählen Sie auch hier einen eindeutigen Namen aus, vorzugsweise denselben Namen, den Sie auch beim Starten des Katalogservers verwendet haben. Klicken Sie auf **OK**, um die Konfiguration zu beenden.
- Wenn Sie Server für den Agenten in eXtreme-Scale-Servern überwachen möchten, die in einen Prozess von WebSphere Application Server integriert sind, führen Sie die folgenden Schritte aus:
  a. Aktualisieren Sie die Java-Parameter, und stellen Sie sicher, dass der Wert für **Java Home** korrekt ist. JVM-Argumente können leer bleiben. Klicken Sie auf **Next**.

b. Wählen Sie im Feld **MBean server connection type** einen Typ aus. Wählen Sie die Version von WebSphere Application Server für Ihre Umgebung aus. Klicken Sie auf **Next**.

c. Stellen Sie sicher, dass die Informationen zu WebSphere Application Server in dieser Anzeige korrekt sind. Klicken Sie auf **Next**.

d. Fügen Sie nur eine einzige Unterknotendefinition hinzu. Legen Sie einen Namen für die Unterknotendefinition fest, aber aktualisieren Sie die Portdefinition nicht. In einer Umgebung mit WebSphere Application Server können Daten von allen Anwendungsserverprozessen erfasst werden, die vom Node Agent verwaltet werden, der auf dem Computer ausgeführt wird. Klicken Sie auf **Next**.

e. Wenn keine Katalogserver in der Umgebung vorhanden sind, klicken Sie auf **OK**. Wenn Sie Katalogserver haben, fügen Sie eine neue Konfiguration für jeden Katalogserver hinzu, ähnlich wie bei den Container-Servern. Wählen Sie einen eindeutigen Namen für den Katalogservice aus, vorzugsweise den Namen, den Sie auch beim Starten des Katalogservice verwendet haben. Klicken Sie auf **OK**, um die Konfiguration zu beenden.

Anmerkung: Die Container-Server müssen nicht zusammen mit dem Katalogservice in einem Prozess ausgeführt werden.

Jetzt sind der Agent und die Server konfiguriert und betriebsbereit. Klicken Sie im nächsten Fenster mit der rechten Maustaste auf `instance1`, um den Agenten zu starten.

UNIX Zum Konfigurieren des Agenten auf der UNIX-Plattform über die Befehlszeile, führen Sie die folgenden Schritte aus:

Es folgt ein Beispiel für eigenständige Server, die einen JSR160-konformen Verbindungstyp verwenden. Das Beispiel zeigt drei eXtreme-Scale-Container auf dem einzelnen Host (rhea00b02), und die JMX-Listener-Adressen sind 15000,15001 und 15002. Es gibt keine Katalogserver.

Die Ausgabe des Konfigurationsdienstprogramms wird in *`Kursivschrift mit fester Breite`* angezeigt und die Benutzeraktion in **`Fettschrift mit fester Breite`**. (Wenn keine Benutzeraktion erforderlich ist, wird der Standardwert durch Drücken der Eingabetaste ausgewählt.)

```
rhea00b02 # ./itmcmd config -A xt
Agent configuration started...
Enter instance name (default is: ): inst1
Edit "Monitoring Agent for WebSphere eXtreme Scale" settings? [ 1=Yes, 2=No ] (default is: 1):
Edit 'Java' settings? [ 1=Yes, 2=No ] (default is: 1):
Java home (default is: C:\Program Files\IBM\Java50): /opt/OG61/java
Java trace level [ 1=Error, 2=Warning, 3=Information, 4=Minimum Debug, 5=Medium Debug, 6=Maximum Debug,
    7=All ] (default is: 1):
JVM arguments (default is: ):
Edit 'Connection' settings? [ 1=Yes, 2=No ] (default is: 1):
MBean server connection type [ 1=JSR-160-Compliant Server, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ] (default is: 1): 1
Edit 'JSR-160-Compliant Server' settings? [ 1=Yes, 2=No ] (default is: 1):
JMX user ID (default is: ):
Enter JMX password (default is: ):
Re-type : JMX password (default is: ):
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:port/objectgrid/MBeanServer):
----------------------------------------
JMX Class Path Information
JMX base paths (default is: ):
JMX class path (default is: ):
JMX JAR directories (default is: ):
Edit 'WebSphere eXtreme Scale Catalog Service' settings? [ 1=Yes, 2=No ] (default is: 1): 2
Edit 'WebSphere eXtreme Scale Grid Servers' settings? [ 1=Yes, 2=No ] (default is: 1): 1
No 'WebSphere eXtreme Scale Grid Servers' settings available?
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02_c0
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15000/objectgrid/MBeanServer
```

```
'WebSphere eXtreme Scale Grid Servers' settings:  WebSphere eXtreme Scale Grid Servers=ogx
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02_c1
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer

'WebSphere eXtreme Scale Grid Servers' settings:  WebSphere eXtreme Scale Grid Servers= rhea00b02_c1
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02_c2
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer

'WebSphere eXtreme Scale Grid Servers' settings:  WebSphere eXtreme Scale Grid Servers= rhea00b02_c2
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 5

Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
TEMS Host Name (Default is: rhea00b00):

Network Protocol [ip, sna, ip.pipe or ip.spipe] (Default is: ip.pipe):

     Now choose the next protocol number from one of these:
     - ip
     - sna
     - ip.spipe
     - 0 for none
Network Protocol 2 (Default is: 0):
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):

Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):
Enter Optional Primary Network Name or 0 for "none" (Default is: 0):
Agent configuration completed...
```

Der vorherige Beispielcode erstellt eine Agenteninstanz mit dem Namen "inst1" und aktualisiert die Einstellungen für das Java-Ausgangsverzeichnis. Die eXtreme-Scale-Container-Server sind konfiguriert, aber der Katalogserver ist nicht konfiguriert.

Anmerkung: Die vorherige Prozedur erstellt eine Textdatei im folgenden Format im Verzeichnis `<ITM-Installation>/config/<Host>_xt_<Instanzname>.cfg`.

Beispiel: `rhea00b02_xt_inst1.cfg`

Es empfiehlt sich, diese Datei mit einem Texteditor Ihrer Wahl zu editieren. Ein Beispiel für den Inhalt einer solchen Datei folgt:

```
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 }  ]]
```

Im Folgenden sehen Sie ein Beispiel, das die Konfiguration einer Implementierung von WebSphere Application Server zeigt:

```
rhea00b02 # ./itmcmd config -A xt
Agent configuration started...
Enter instance name (default is: ): inst1
Edit "Monitoring Agent for WebSphere eXtreme Scale" settings? [ 1=Yes, 2=No ] (default is: 1): 1
Edit 'Java' settings? [ 1=Yes, 2=No ] (default is: 1): 1
Java home (default is: C:\Program Files\IBM\Java50): /opt/WAS61/java
Java trace level [ 1=Error, 2=Warning, 3=Information, 4=Minimum Debug, 5=Medium Debug, 6=Maximum Debug,
    7=All ] (default is: 1):
JVM arguments (default is: ):
Edit 'Connection' settings? [ 1=Yes, 2=No ] (default is: 1):
MBean server connection type [ 1=JSR-160-Compliant Server, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ] (default is: 1): 4
Edit 'WebSphere Application Server version 7.0' settings? [ 1=Yes, 2=No ] (default is: 1):WAS user ID (default is: ):
Enter WAS password (default is: ):
Re-type : WAS password (default is: ):
```

```
WAS host name (default is: localhost): rhea00b02
WAS port (default is: 2809):
WAS connector protocol [ 1=rmi, 2=soap ] (default is: 1):
WAS profile name (default is: ): default
---------------------------------------
WAS Class Path Information
WAS base paths (default is: C:\Program Files\IBM\WebSphere\AppServer;/opt/IBM/WebSphere/AppServer): /opt/WAS61
WAS class path (default is: runtimes/com.ibm.ws.admin.client_6.1.0.jar;runtimes/com.ibm.ws.ejb.thinclient_7.0.0.jar):
WAS JAR directories (default is: lib;plugins):
Edit 'WebSphere eXtreme Scale Grid Servers' settings? [ 1=Yes, 2=No ] (default is: 1):
No 'WebSphere eXtreme Scale Grid Servers' settings available?
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):

'WebSphere eXtreme Scale Grid Servers' settings:  WebSphere eXtreme Scale Grid Servers=rhea00b02
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 5
Edit 'WebSphere eXtreme Scale Catalog Service' settings? [ 1=Yes, 2=No ] (default is: 1): 2
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
TEMS Host Name (Default is: rhea00b02):

Network Protocol [ip, sna, ip.pipe or ip.spipe] (Default is: ip.pipe):

     Now choose the next protocol number from one of these:
     - ip
     - sna
     - ip.spipe
     - 0 for none
Network Protocol 2 (Default is: 0):
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):

Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):
Enter Optional Primary Network Name or 0 for "none" (Default is: 0):
Agent configuration completed...
rhea00b02 #
```

Für Implementierungen von WebSphere Application Server müssen Sie nicht mehrere Unterknoten erstellen. Der eXtreme-Scale-Agent stellt die Verbindung zum Node Agent her, um alle Informationen von den Anwendungsservern zu erfassen, für die er zuständig ist.

`SECTION=CAT` bezeichnet eine Katalogservicezeile, wohingegen `SECTION=OGS` eine Konfigurationszeile für einen eXtreme-Scale-Server bezeichnet.

5. Konfigurieren Sie den JMX-Port für alle Container-Server von eXtreme Scale.

   Wenn eXtreme-Scale-Container-Server ohne das Argument **-JMXServicePort** gestartet werden, wird einem MBean-Server ein dynamischer Port zugeordnet. Der Agent muss im Voraus wissen, mit welchem Port er zu kommunizieren hat. Der Agent funktioniert nicht mit dynamischen Ports.

   Wenn Sie die Server starten, müssen Sie das Argument `-JMXServicePort <Portnummer>` angeben, wenn Sie den eXtreme-Scale-Server mit dem Befehl `startOgServer.sh | .bat` starten. Die Ausführung dieses Befehls gewährleistet, dass der JMX-Server im Prozess an einem statischen vordefinierten Port empfangsbereit ist.

   Für die vorherigen Beispiele müssen in einer UNIX-Installation zwei eXtreme-Scale-Server mit definierten Ports gestartet werden:

   a. "-JMXServicePort" "15000" (für rhea00b02_c0)
   b. "-JMXServicePort" "15001" (für rhea00b02_c1)
   a. Starten Sie den eXtreme Scale-Agenten.

      Davon ausgehend, dass wie im vorherigen Beispiel die Instanz `inst1` erstellt wurde, setzen Sie die folgenden Befehle ab:

      1) `cd <ITM-Installation>/bin`
      2) `itmcmd agent –o inst1 start xt`

   b. Stoppen Sie den eXtreme Scale-Agenten.

Davon ausgehend, dass wie im vorherigen Beispiel die Instanz `inst1` erstellt wurde, setzen Sie die folgenden Befehle ab:

1) `cd <ITM-Installation>/bin`
2) `itmcmd agent –o inst1 stop xt`

6. Aktivieren Sie Statistiken für alle Container-Server von eXtreme Scale.

   Der Agent verwendet die Statistik-MBeans von eXtreme Scale, um Statistiken aufzuzeichnen. Die Statistikspezifikation von eXtreme Scale muss mit einer der folgenden Methoden aktiviert werden.

   - Konfiguration von Servereigenschaften für die Aktivierung aller Statistiken, wenn die Container-Server gestartet werden: all=enabled.
   - Verwendung des Beispieldienstprogramms xsadmin für die Aktivierung von Statistiken für alle aktiven Container mit den Parametern "-setstatsspec all=enabled"

Ergebnisse

Nachdem alle Server konfiguriert und gestartet wurden, werden MBean-Daten in der Konsole von IBM Tivoli Portal angezeigt. In vordefinierten Arbeitsbereichen werden Graphen und Datenmetriken auf jeder Knotenebene angezeigt.

Die folgenden Arbeitsbereiche sind definiert: **eXtreme Scale Grid Servers** - für alle überwachten Knoten

- eXtreme Scale Transactions View
- eXtreme Scale Primary Shard View
- eXtreme Scale Memory View
- eXtreme Scale ObjectMap View

Sie können auch einen eigenen Arbeitsbereich konfigurieren. Weitere Einzelheiten finden Sie in den Informationen zum Anpassen von Arbeitsbereichen im Information Center von IBM Tivoli Monitoring.

eXtreme-Scale-Anwendungen mit CA Wily Introscope überwachen

CA Wily Introscope ist ein Managementprodukt eines anderen Anbieters, das Sie verwenden können, um Leistungsprobleme in Unternehmensanwendungsumgebungen zu erkennen und zu diagnostizieren. eXtreme Scale enthält Einstellungen für die Konfiguration von CA Wily Introscope für die Introspektion ausgewählter Komponenten der Laufzeitumgebung von eXtreme Scale, um eXtreme-Scale-Anwendungen schnell anzeigen und validieren zu können. CA Wily Introscope funktioniert effizient für eigenständige Implementierungen und Implementierungen von WebSphere Application Server.

Übersicht

Wenn Sie eXtreme-Scale-Anwendungen mit CA Wily Introscope überwachen möchten, müssen Sie Einstellungen in den PBD-Dateien (ProbeBuilderDirective) festlegen, die Ihnen Zugriff auf die Überwachungsinformationen für eXtreme Scale geben.

Achtung: Die Instrumentierungspunkte für Introscope können sich mit jedem Fixpack oder Release ändern. Wenn Sie ein neues Fixpack oder Release installieren, suchen Sie in der Dokumentation nach Hinweisen zu Änderungen bezüglich der Instrumentierungspunkte.

Sie können PBD-Dateien (ProbeBuilderDirective) von CA Wily Introscope konfigurieren, um Ihre eXtreme-Scale-Anwendungen zu überwachen. CA Wily Introscope ist ein Anwendungsmanagementprodukt, mit dem Sie Leistungsprobleme in komplexen, Verbund- und Webanwendungsumgebungen proaktiv erkennen, sichten und diagnostizieren können.

Einstellungen in der PBD-Datei für die Überwachung des Katalogservice

Sie können eine oder mehrere der folgenden Einstellungen in Ihrer PBD-Datei verwenden, um den Katalogservice zu überwachen:

```
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}"
```

Klassen für die Überwachung des Katalogservice

HAControllerImpl
Die Klasse "HAControllerImpl" verarbeitet Lebenszyklus- und Feedback-Ereignisse für die Stammgruppe. Sie können diese Klasse überwachen, um einen Hinweis auf die Struktur und Änderungen der Stammgruppe zu erhalten.

ServerAgent
Die Klasse "ServerAgent" ist für die Kommunikation von Stammgruppenereignissen an den Katalogservice zuständig. Sie können die verschiedenen Aufrufe für den Austausch von Überwachungssignalen überwachen, um wichtige Ereignisse zu erkennen.

PlacementServiceImpl
Die Klasse "PlacementServiceImpl" koordiniert die Container. Sie können die Methoden in dieser Klasse verwenden, um das Beitreten von Servern zur Stammgruppe und Verteilungsereignisse zu überwachen.

BalanceGridEventListener
Die Klasse "BalanceGridEventListener" steuert die Leitung des Katalogs. Sie können diese Klasse überwachen, um einen Hinweis auf den Katalogservice zu erhalten, der momentan als leitender Server agiert.

Einstellungen in der PBD-Datei für die Überwachung der Container

Sie können eine oder mehrere der folgenden Einstellungen in Ihrer PBD-Datei verwenden, um die Container zu überwachen:

```
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}"
```

Klassen für die Überwachung der Container

ShardImpl

Die Klasse "ShardImpl" hat die Methode "processMessage. Die Methode "processMessage" ist die Methode für Clientanforderungen. Mit dieser Methode können Sie serverseitige Antwortzeiten und Anforderungszähler abrufen. Indem Sie die Zähler für alle Server und die Auslastung des Heapspeichers überwachen, können Sie feststellen, ob das Grid ausgeglichen ist.

CheckpointIterator

Die Klasse "CheckpointIterator" hat die Methode "activateListener", die primäre Shards in den Peermodus versetzt. Wenn die primären Shards in den Peermodus versetzt werden, ist das Replikat nach Abschluss der Methode auf demselben Stand wie das primäre Shard. Wenn ein Replikat über ein vollständiges primäres Shard neu generiert wird, kann diese Operation längere Zeit dauern. Das System ist erst dann vollständig wiederhergestellt, wenn diese Operation abgeschlossen ist. Sie können diese Klasse verwenden, um den Fortschritt der Operation zu überwachen.

CommittedLogSequenceListenerProxy

Die Klasse "CommittedLogSequenceListenerProxy" hat zwei Methoden, die von Interesse sind. Die Methode "applyCommitted" wird für jede Transaktion ausgeführt, und die Methode "sendApplyCommitted" wird ausgeführt, wenn das Replikat Informationen extrahiert. Das Verhältnis, in dem die beiden Methoden ausgeführt werden, kann Ihnen einen Hinweis darauf geben, inwieweit das Replikat in der Lage ist, mit dem primären Shard Schritt zu halten.

Einstellungen in der PBD-Datei für die Überwachung der Clients

Sie können eine oder mehrere der folgenden Einstellungen in Ihrer PBD-Datei verwenden, um die Clients zu überwachen:

```
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}"
```

Klassen für die Überwachung der Clients

ORBClientCoreMessageHandler
Die Klasse "ORBClientCoreMessageHandler" ist für das Senden von Anwendungsanforderungen an die Container zuständig. Sie können die Methode "sendMessage" überwachen, um Clientantwortzeiten und Anforderungsanzahl zu erhalten.

ClusterStore
Die Klasse "ClusterStore" enthält Routing-Informationen auf der Clientseite.

BaseMap
Die Klasse "BaseMap" hat die Methode "evictMapEntries", die aufgerufen wird, wenn der Evictor Einträge aus der Map entfernen möchte.

SelectionServiceImpl
Die Klasse "SelectionServiceImpl" trifft Routing-Entscheidungen. Wenn der Client Failover-Entscheidungen trifft, können Sie diese Klasse verwenden, um die aus den Entscheidungen resultierenden Aktionen zu überwachen.

ObjectGridImpl
Die Klasse "ObjectGridImpl" hat die Methode "getSession", die Sie überwachen können, um die Anzahl der Anforderungen an diese Methode zu erhalten.

eXtreme Scale mit Hyperic HQ überwachen

Hyperic HQ ist eine Überwachungslösung eines anderen Anbieters, die kostenlos als Open-Source-Lösung oder als Unternehmensprodukt verfügbar ist. WebSphere eXtreme Scale enthält ein Plug-in, mit dem Hyperic-HQ-Agenten eXtreme-Scale-Container-Server erkennen und Statistiken mit Hilfe von eXtreme-Scale-Management-Beans berichten und zusammenfassen können. Sie können Hyperic HQ verwenden, um eigenständige Implementierungen von eXtreme Scale zu überwachen.

Vorbereitende Schritte

- Die folgenden Anweisungen gelten für Hyperic Version 4.0. Wenn Sie eine neuere Version von Hyperic haben, schlagen Sie in der Hyperic-Dokumentation Informationen wie Pfadnamen und Informationen zum Starten von Agenten und Servern nach.
- Laden Sie Hyperic-Server- und -Agenteninstallationen herunter. Eine Serverinstallation muss aktiv sein. Um alle eXtreme-Scale-Server zu erkennen, muss ein Hyperic-Agent auf jeder Maschine ausgeführt werden, auf dem ein eXtreme-Scale-Server ausgeführt wird. Downloadinformationen und Dokumentationsunterstützung finden Sie auf der Website von Hyperic.

- Sie müssen Zugriff auf die Dateien `objectgrid-plugin.xml` und `hqplugin.jar` haben. Diese Dateien befinden sich im Verzeichnis *WXS-Installationsstammverzeichnis*`/hyperic/etc`.

Informationen zu diesem Vorgang

Durch die Integration der Überwachungssoftware Hyperic HQ in eXtreme Scale können Sie Metriken zur Leistung Ihrer Umgebung grafisch überwachen und anzeigen. Sie konfigurieren diese Integration über eine Plug-in-Implementierung in jedem Agenten.

Vorgehensweise

1. Starten Sie Ihre eXtreme-Scale-Server. Das Hyperic-Plug-in prüft die lokalen Prozesse, um eine Verbindung zu den Java Virtual Machines herzustellen, in denen eXtreme Scale ausgeführt wird. Für eine ordnungsgemäße Verbindungsherstellung zu den Java Virtual Machines muss jeder Server mit der Option **-jmxServicePort** gestartet werden. Informationen zum Starten von Servern mit der Option **-jmxServicePort** finden Sie im Abschnitt „Script **startOgServer**" auf Seite 409.
2. Kopieren Sie die Datei `extremescale-plugin.xml` und die Datei `wxshyperic.jar` in die entsprechenden Server- und Agenten-Plug-in-Verzeichnisse in Ihrer Hyperic-Konfiguration. Zum Integrieren von Hyperic müssen die Agenten- und die Serverinstallationen Zugriff auf das Plug-in und die JAR-Dateien haben. Obwohl der Server Konfigurationen dynamisch wechseln kann, müssen Sie die Integration vor dem Starten von Agenten durchführen.
   a. Kopieren Sie die Datei `extremescale-plugin.xml` in das Serververzeichnis `plugin` an der folgenden Position:

      hyperic_home`/server_home/hq-engine/server/default/deploy/hq.ear/hq-plugins`
   b. Kopieren Sie die Datei `extremescale-plugin.xml` in das Agentenverzeichnis `plugin` an der folgenden Position:

      agent_home`/bundles/gent-4.0.2-939/pdk/plugins`
   c. Kopieren Sie die Datei `wshyperic.jar` in das Agentenverzeichnis `lib` an der folgenden Position:

      agent_home`/bundles/gent-4.0.2-939/pdk/lib`
3. Konfigurieren Sie den Agenten. Die Datei `agent.properties` dient als Konfigurationspunkt für die Agentenlaufzeitumgebung. Diese Datei ist im Verzeichnis *agent_home*`/conf` enthalten. Die folgenden Schlüssel sind optional, aber wichtig für das eXtreme-Scale-Plug-in:
   - `autoinventory.defaultScan.interval.millis=<`*Millisekunden*`>`

     Legt das Intervall (in Millisekunden) fest, in dem die Erkennungsoperationen des Agenten ausgeführt werden.
   - `log4j.logger.org.hyperic.hq.plugin.extremescale.XSServerDetector=DEBUG`

     Aktiviert ausführliche Debug-Anweisungen vom eXtreme-Scale-Plug-in.
   - `username=<`*Benutzername*`>`: Legt den JMX-Benutzernamen (Java Management Extensions) fest, wenn die Sicherheit aktiviert ist.
   - `password=<`*Kennwort*`>`: Legt das JMX-Kennwort fest, wenn die Sicherheit aktiviert ist.
   - `sslEnabled=<`*true*`|`*false*`>`: Teilt dem Plug-in mit, ob Secure Sockets Layer (SSL) verwendet werden soll. Der Standardwert ist `false`.

- `trustPath=<Pfad>`: Legt den Sicherheitspfad für die SSL-Verbindung fest.
- `trustType=<Typ>`: Legt den Sicherheitstyp für die SSL-Verbindung fest.
- `trustPass=<Kennwort>`: Legt das Sicherheitskennwort für die SSL-Verbindung fest.

4. Starten Sie die Agentenerkennung. Die Hyperic-Agenten senden Erkennungs- und Metrikinformationen an den Server. Verwenden Sie den Server, um Datensichten anzupassen und logische Bestandsobjekt zu gruppieren, um hilfreiche Informationen zu erhalten. Wenn der Server verfügbar ist, müssen Sie das Startscript ausführen oder den Windows-Dienst für den Agenten starten:
   - Linux *agent_home*`/bin/hq-agent.sh start`
   - Windows Starten Sie den Agenten mit dem Windows-Dienst.

   Nachdem Sie den Agenten gestartet haben, werden die Server erkannt und Gruppen konfiguriert. Sie können sich an der Serverkonsole anmelden und die Ressourcen auswählen, die der Bestandsdatenbank für den Server hinzugefügt werden sollen. Die Serverkonsole ist standardmäßig über den folgenden URL erreichbar: `http://<Hostname_des_Servers>:7080/`
5. Statistiken müssen aktiviert werden, damit Hyperic statistische Daten erfasst.

   Verwenden Sie die Steueraktion **SetStatsSpec** in der Hyperic-Konsole für eXtreme Scale. Navigieren Sie zur Ressource, und verwenden Sie anschließend das Dropdown-Menü **Control Action** auf der Registerkarte **Control**, um `SetStatsSpec` mit der Einstellung `ALL=enabled` im Textfeld **Control Arguments** einzugeben.

   Katalogserver werden von dem Filtersatz in der Hyperic-Konsole nicht erkannt. Lesen Sie die Informationen zur Eigenschaft **statsSpec** im Abschnitt Servereigenschaftendatei, mit der Statistiken beim Containerstart aktiviert werden. Verschiedene Optionen zum Aktivieren von Statistiken sind in „Überwachung mit Managed Beans (MBeans)” auf Seite 484 und Beispiel: Dienstprogramm **xsadmin** beschrieben.
6. Überwachen Sie Server mit der Hyperic-Konsole. Nachdem die Server dem Bestandsmodell hinzugefügt wurden, sind ihre Services nicht mehr erforderlich.
   - **Dashboard-Sicht**: Wenn Sie die Ressourcenerkennungsereignisse anzeigen möchten, müssen Sie sich bei der Haupt-Dashboard-Sicht anmelden. Die Dashboard-Sicht ist eine generische Sicht, die als Message Center dient, das Sie anpassen können. Sie können Graphen oder Bestandsobjekte in dieses Haupt-Dashboard exportieren.
   - **Ressourcensicht:** Sie können das gesamte Bestandsmodell über diese Seite abfragen und anzeigen. Nachdem Sie die Server hinzugefügt haben, wird jeder eXtreme-Scale-Server ordnungsgemäß beschriftet im Abschnitt mit den Servern aufgelistet. Sie können auf die einzelnen Server klicken, um die Basismetriken anzuzeigen.
7. Zeigen Sie den gesamten Serverbestand auf der Seite mit der Ressourcensicht an. Auf dieser Seite können Sie mehrere ObjectGrid-Server auswählen und gruppieren. Nach der Gruppierung von Ressourcen können die gemeinsamen Metriken der jeweiligen Gruppe in Graphen dargestellt werden, um Überschneidungen und Unterschiede zwischen den Gruppen-Membern anzuzeigen. Um eine Überschneidung anzuzeigen, wählen Sie die Metrik in der Anzeige Ihrer Servergruppe aus. Die Metrik wird dann im Diagrammbereich angezeigt. Zum Anzeigen einer Überschneidung für alle Gruppen-Member klicken Sie auf den unterstrichenen Metriknamen. Sie können alle Diagramme, Knotensichten und Vergleichsüberschneidungen über das Menü **Tools** in das Haupt-Dashboard exportieren.

Informationen von eXtreme Scale in DB2 überwachen

Wenn der JPALoader oder JPAEntityLoader mit DB2 als Back-End-Datenbank verwendet wird, können eXtreme-Scale-spezifische Informationen an DB2 übergeben werden. Sie können diese Informationen mit einem Leistungsüberwachungstool wie DB2 Performance Expert überwachen, um die Anwendungen von eXtreme Scale zu überwachen, die auf die Datenbank zugreifen.

Vorbereitende Schritte

Weitere Informationen zu den verschiedenen Methoden, mit denen Sie den zu verwendenden Trace festlegen, finden Sie unter „Trace erfassen" auf Seite 544.

Informationen zu diesem Vorgang

Wenn der Loader für die Verwendung von DB2 als Back-End-Datenbank konfiguriert ist, können die folgenden Informationen zu eXtreme Scale für Überwachungszwecke an DB2 übergeben werden:

- **Benutzer:** Gibt den Namen des Benutzers an, der sich bei eXtreme Scale authentifiziert. Wenn keine Basisauthentifizierung verwendet wird, werden die Principals aus der Authentifizierung verwendet.
- **Workstationname:** Gibt den Hostnamen oder die IP-Adresse des Container-Servers von eXtreme Scale an.
- **Anwendungsname:** Gibt den Namen des ObjectGrids bzw. den Name der Persistenzeinheit (sofern definiert) an.
- **Abrechnungsinformationen:** Gibt die Thread-ID, den Transaktionstyp, die Transaktions-ID und die Verbindungszeichenfolge an.

Lesen Sie die Dokumentation zu DB2 Performance Expert, um mehr über die Überwachung des Datenbankzugriffs zu erfahren.

Vorgehensweise

- Wenn Sie alle Clientinformationen für eXtreme Scale aktivieren möchten, setzen Sie die folgenden Tracezeichenfolgen:

  ```
  ObjectGridClientInfo*=event=enabled
  ```

- Verwenden Sie eine der folgenden Einstellungen, um alle Informationen bis zu die Benutzerinformationen zu aktivieren:
  -
    ```
    ObjectGridClientInfo*=event=enabled,ObjectGridClientInfoUser=event=disabled
    ```
    oder
  -
    ```
    ObjectGridClientInfo=event=enabled
    ```

Ergebnisse

Nach der Aktivierung der Tracefunktion werden Daten im Leistungsüberwachungstool wie DB2 Performance Expert angezeigt.

Beispiel

Im folgenden Beispiel wird der Benutzer `bob` als Benutzer von eXtreme Scale authentifiziert. Die Anwendung greift über die Persistenzeinheit `DB2Hibernate` auf das Datengrid `mygrid` zu. Der Container-Server hat den Namen `XS_Server1`. Die generierten Informationen sind folgende:

- **Benutzer**=`bob`

- **WorkstationName**=XS_Server1,192.168.1.101
- **Anwendungsname**=mygrid,DB2Hibernate
- **Abrechnungsinformationen**=1, DEFAULT,FE7954BD-0126-4000-E000-2298094151DB,com.ibm.db2.jcc.t4.b@71787178

Im folgenden Beispiel wird der Benutzer bob mithilfe eines Tokens von WebSphere Application Server authentifiziert. Die Anwendung greift über die Persistenzeinheit DB2OpenJPA auf das Datengrid mygrid zu. Der Container-Server hat den Namen XS_Server2. Die generierten Informationen sind folgende:

- **Benutzer**

  =acme.principal.UserPrincipal[Bob],acme.principal.GroupPrincipal[admin]
- **Workstationname**=XS_Server2,192.168.1.102
- **Anwendungsname**=mygrid,DB2OpenJPA
- **Abrechnungsinformationen**=188,DEFAULT,FE72BC63-0126-4000-E000-851C092A4E33,com.ibm.ws.rsadapter.jdbc.WSJccSQLJConnection@2b432b43

Kapitel 9. Leistung optimieren

Sie können Einstellungen in Ihrer Umgebung optimieren, um die Gesamtleistung Ihrer Umgebung von WebSphere eXtreme Scale zu erhöhen.

Betriebssystem- und Netzeinstellungen optimieren

Eine Netzoptimierung kann durch Änderung der Verbindungseinstellung Verzögerungen im TCP-Stack (Transmission Control Protocol) reduzieren und den Durchsatz durch Änderung der TCP-Puffer verbessern.

Betriebssysteme

Ein Windows-System erfordert die geringste Optimierung, ein Solaris-System die meiste. Die folgenden Informationen gelten für jedes angegebene System und können die Leistung von WebSphere eXtreme Scale verbessern. Sie müssen die Optimierung Ihrem Netz und Ihrer Anwendungslast entsprechend optimieren.

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
```

ORB-Eigenschaften

ORB-Eigenschaften (Object Request Broker) ändern das Transportverhalten des Datengrids. Diese Eigenschaften können über eine Datei `orb.properties`, als Einstellungen in der Administrationskonsole von WebSphere Application Server oder als angepasste Eigenschaften im ORB in der Administrationskonsole von WebSphere Application Server definiert werden.

orb.properties

Sie finden die Datei `orb.properties` im Verzeichnis `java/jre/lib`. Wenn Sie die Datei `orb.properties` im Verzeichnis `java/jre/lib` von WebSphere Application Server ändern, werden die ORB-Eigenschaften im Node Agent und in allen anderen Java Virtual Machines (JVM), die die Java Runtime Environment (JRE) verwenden, aktualisiert. Wenn Sie dieses Verhalten nicht wünschen, verwenden Sie angepasste Eigenschaften oder die ORB-Einstellungen in der Administrationskonsole von WebSphere Application Server.

Standardeinstellungen von WebSphere Application Server

In WebSphere Application Server werden einige Eigenschaften im ORB standardmäßig definiert. Diese Einstellungen sind in den Container-Services des Anwendungsanwendungsservers und im Deployment Manager enthalten. Diese Standardeinstellungen überschreiben alle Einstellungen, die Sie in der Datei `orb.properties` erstellen. Für jede beschriebene Eigenschaft können Sie im Abschnitt **Definitionsposition** nachlesen, wo der empfohlene Wert zu definieren ist.

Dateideskriptoreinstellungen

Für UNIX- und Linux-Systeme gibt es einen Grenzwert für die zulässige Anzahl offener Dateien pro Prozess. Das Betriebssystem gibt die zulässige Anzahl offener Dateien an. Wenn dieser Wert zu klein ist, tritt ein Fehler bei der Speicherzuordnung unter AIX auf, und es wird protokolliert, dass zu viele Dateien offen sind.

Setzen Sie diese Einstellung im Terminalfenster auf dem UNIX-System auf einen höheren Wert als den Systemstandardwert. Für große SMP-Maschinen mit Klonen legen Sie den Wert für eine uneingeschränkte Anzahl offener Dateien fest.

Für AIX-Konfiguration setzen Sie diese Einstellung mit dem Befehl `ulimit -n -1` auf den Wert `-1` (uneingeschränkt).

Für Solaris-Konfigurationen setzen Sie diese Einstellung mit dem Befehl `ulimit -n 16384` auf den Wert `16384`.

Zum Anzeigen des aktuellen Werts verwenden Sie den Befehl `ulimit –a`.

Basiseinstellungen

Die folgenden Einstellungen bilden eine gute Grundlage, sind aber nicht unbedingt die besten Einstellungen für jede Umgebung. Die Einstellungen helfen Ihnen dabei, eine gute Entscheidung bezüglich der Werte zu treffen, die in Ihrer Umgebung angemessen sind:

```
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=0com.ibm.CORBA.iiop.NoLocalCopies=true
com.ibm.CORBA.NoLocalInterceptors=true
```

Beschreibungen der Eigenschaften

Zeitlimiteinstellungen

Die folgenden Einstellungen beziehen sich auf die Zeit, die der ORB wartet, bevor er Anforderungsoperationen aufgibt. Verwenden Sie diese Einstellungen, um zu verhindern, dass in einer abnormalen Situation zu viele Threads erstellt werden.

Anforderungszeitlimit

Eigenschaftsname: com.ibm.CORBA.RequestTimeout

Gültiger Wert: Ganzzahliger Wert (in Sekunden)

Empfohlener Wert: 30

Definitionsposition: Administrationskonsole von WebSphere Application Server

Beschreibung: Gibt an, wie lange (in Sekunden) eine Anforderung auf eine Antwort warten soll, bevor sie aufgibt. Diese Eigenschaft beeinflusst die Zeit, die ein Client für das Failover benötigt, wenn ein Netzausfall eintritt. Wenn Sie einen zu niedrigen Wert für diese Eigenschaft wählen, können nicht beabsichtigte Zeitlimitüberschreitungen bei Anforderungen auftreten. Sie können dies verhindern, indem Sie den Wert für diese Eigenschaft sorgfältig auswählen.

Verbindungszeitlimit

Eigenschaftsname: com.ibm.CORBA.ConnectTimeout

Gültiger Wert: Ganzzahliger Wert (in Sekunden)

Empfohlener Wert: 10

Definitionsposition: Datei `orb.properties`

Beschreibung: Gibt an, wie viele Sekunden bei einem Versuch, eine Socket-Verbindung herzustellen, gewartet werden soll, bis der Versuch eingestellt wird. Diese Eigenschaft kann wie das Anforderungszeitlimit die Zeit beeinflussen, die ein Client für das Failover benötigt, wenn ein Netzausfall eintritt. Im Allgemeinen setzen Sie diese Eigenschaft auf einen kleineren Wert als das Anforderungszeitlimit, weil die Zeit zum Herstellen einer Verbindung relativ konstant sein sollte.

Fragmentzeitlimit

Eigenschaftsname: com.ibm.CORBA.FragmentTimeout

Gültiger Wert: Ganzzahliger Wert (in Sekunden)

Empfohlener Wert: 30

Definitionsposition: Datei `orb.properties`

Beschreibung: Gibt an, wie lange (in Sekunden) eine Fragmentanforderung auf eine Antwort warten soll, bevor sie aufgibt. Diese Eigenschaft ist der Eigenschaft "Anforderungszeitlimit" ähnlich.

Einstellungen des Thread-Pools

Diese Eigenschaften beschränken die Größe des Thread-Pools auf eine bestimmte Anzahl an Threads. Die Threads werden vom ORB verwendet, um die Serveranforderungen auszugliedern, nachdem sie am Socket empfangen wurden. Wenn Sie zu niedrige Werte für diese Eigenschaften wählen, kann dies zu einer erhöhten Socket-Warteschlangenlänge und zu möglichen Zeitlimitüberschreitungen führen.

Verbindungsmultiplizität

Eigenschaftsname: com.ibm.CORBA.ConnectionMultiplicity

Gültiger Wert: Ganzzahliger Wert für die Anzahl der Verbindungen zwischen dem Client und dem Server. Der Standardwert ist 1. Die Festlegung eines höheren Werts definiert das Multiplexing für mehrere Verbindungen.

Empfohlener Wert: 1

Definitionsposition: Datei `orb.properties`**Beschreibung:** Ermöglicht dem ORB, mehrere Verbindungen zu jedem Server herzustellen. In der Theorie sollte die Definition dieses Werts die Parallelität der Verbindungen fördern. In der Praxis profitiert die Leistung nicht von der Festlegung der Verbindungsmultiplizität. Setzen Sie diesen Parameter nicht.

Offene Verbindungen

Eigenschaftsnamen: com.ibm.CORBA.MinOpenConnections, com.ibm-.CORBA.MaxOpenConnections

Gültiger Wert: Ein ganzzahliger Wert für die Anzahl der Verbindungen.

Empfohlener Wert: 1024

Definitionsposition: Administrationskonsole von WebSphere Application Server**Beschreibung:** Gibt eine Mindest- und eine maximale Anzahl offener Verbindungen an. Der ORB verwaltet einen Cache mit den Verbindungen, die mit Clients hergestellt wurden. Diese Verbindungen werden nach Ablauf dieses Werts bereinigt. Das Löschen von Verbindungen kann zu einem mangelhaften Verhalten im Datengrid führen.

Ist erweiterbar

Eigenschaftsname: com.ibm.CORBA.ThreadPool.IsGrowable

Gültiger Wert: Die gültigen Werte sind `true` und `false` (Boolean).

Empfohlener Wert: `false`

Definitionsposition: Datei `orb.properties`**Beschreibung:** Wenn Sie diese Einstellung auf `true` setzen, kann der vom ORB für eingehende Anforderungen verwendete Thread-Pool über die vom Pool unterstützte Größe hinweg anwachsen. Wenn die Poolgröße überschritten wird, werden neue Threads für die Bearbeitung der Anforderungen erstellt, aber die Threads werden nicht in den Pool gestellt. Sie können das Wachstum des Thread-Pools verhindern, indem Sie die Einstellung auf `false` setzen.

Länge der Server-Socket-Warteschlange

Eigenschaftsname: com.ibm.CORBA.ServerSocketQueueDepth

Gültiger Wert: Ein ganzzahliger Wert für die Anzahl der Verbindungen.

Empfohlener Wert: `1024`

Definitionsposition: Datei `orb.properties`**Beschreibung:** Gibt die Länge der Warteschlange für eingehende Verbindungen von Clients an. Der ORB reiht eingehende Verbindungen von Clients in Warteschlangen ein. Wenn die Warteschlange voll ist, werden Verbindungen zurückgewiesen. Das Zurückweisen von Verbindungen kann zu einem mangelhaften Verhalten im Datengrid führen.

Fragmentgröße

Eigenschaftsname: com.ibm.CORBA.FragmentSize

Gültiger Wert: Eine ganzzahliger Wert, der die Anzahl der Bytes angibt. Der Standardwert ist 1024.

Empfohlener Wert: `0`

Definitionsposition: Datei `orb.properties`**Beschreibung:** Gibt die maximale Paketgröße an, die der ORB verwendet, wenn er eine Anforderung sendet. Wenn eine Anforderung den Grenzwert für die Fragmentgröße überschreitet, wird diese Anforderung in Anforderungsfragmente aufgeteilt, die jeweils separat gesendet und dann im Server erneut assembliert werden. Das Fragmentieren von Anforderungen ist in unzuverlässigen Netzen hilfreich, in denen Paket unter Umständen erneut gesendet werden müssen. Wenn das Netz jedoch zuverlässig ist, kann das Aufteilen von Anforderungen in Fragmente zu einer unnötigen Verarbeitung führen.

Keine lokalen Kopien

Eigenschaftsname: com.ibm.CORBA.iiop.NoLocalCopies

Gültiger Wert: Die gültigen Werte sind `true` und `false` (Boolean).

Empfohlener Wert: `true`

Definitionsposition: Administrationskonsole von WebSphere Application Server, Einstellung **Durch Referenz übergeben** (Pass-by-Reference) **Beschreibung:** Gibt an, ob der ORB nach Referenz übergibt. Der ORB verwendet standardmäßig Aufrufe des Typs "pass by value" (Übergeben nach Wert). Aufrufe des Typs "pass by value" verursachen zusätzliche Kosten für Garbage-Collection und Serialisierung im Pfad, wenn die Schnittstelle lokal gestartet wird. Wenn Sie diese Einstellung auf "true" setzen, verwendet der ORB die Methode "pass by reference" (Übergeben nach Referenz), die effizienter ist als der Aufruf des Typs "pass by value".

Keine lokalen Interceptor

Eigenschaftsname: com.ibm.CORBA.NoLocalInterceptors

Gültiger Wert: Die gültigen Werte sind `true` und `false` (Boolean).

Empfohlener Wert: `true`

Definitionsposition: Datei `orb.properties` **Beschreibung:** Gibt an, ob der ORB Anforderungsinterceptor auch dann startet, wenn lokale Anforderungen (prozessintern) gestellt werden. Die von WebSphere eXtreme Scale verwendeten Interceptor sind für die Sicherheit und Routenverarbeitung bestimmt und nicht erforderlich, wenn die Anforderung innerhalb des Prozesses verarbeitet wird. Interceptor zwischen Prozessen sind nur für RPC-Operationen (Remote Procedure Call) erforderlich. Wenn Sie die Eigenschaft "Keine lokalen Interceptor" aktivieren, können Sie die zusätzliche Verarbeitung vermeiden, den lokale Interceptor mit sich bringen.

Achtung: Wenn Sie die Sicherheit von WebSphere eXtreme Scale verwenden, setzen Sie die Eigenschaft "com.ibm.CORBA.NoLocalInterceptors" auf den Wert `false`. Die Sicherheitsinfrastruktur verwendet Interceptor für die Authentifizierung.

Java Virtual Machines optimieren

Sie müssen bestimmte Aspekte der JVM-Optimierung (Java Virtual Machine) berücksichtigen, um die beste Leistung mit WebSphere eXtreme Scale zu erzielen. In den meisten Fällen sind nur wenige bzw. gar keine speziellen JVM-Einstellungen erforderlich. Wenn viele Objekte im Datengrid gespeichert werden, passen Sie die Größe des Heapspeichers entsprechend an, um Speicherengpässe zu vermeiden.

7.1.1+ Wenn Sie eXtremeMemory konfigurieren, können Sie Objekte im nativen Speicher anstelle des Java-Heapspeichers speichern. Die Konfiguration von eXtremeMemory aktiviert eXtremeIO, einen neuen Transportmechanismus. Durch die Auslagerung von Objekten aus dem Java-Heapspeicher können Sie Garbage-Collection-Pausen vermeiden und damit eine konstantere Leistung und vorhersehbare Antwortzeiten erzielen. Weitere Informationen finden Sie im Abschnitt „IBM eXtremeMemory und IBM eXtremeIO konfigurieren" auf Seite 285.

Getestete Plattformen

Leistungstests wurden hauptsächlich auf AIX- (32 Wege), Linux- (four Wege) und Windows-Computern (acht Wege) durchgeführt. Mit High-End-AIX-Computern können Sie Szenarien mit sehr vielen Threads testen, um Konfliktpunkte zu identifizieren und zu beheben.

Garbage-Collection

WebSphere eXtreme Scale erstellt temporäre Objekte für jede Transaktion, wie z. B. Anforderungs- und Antwortobjekte und eine Protokollfolge. Da sich diese Objekte auf die Effizienz der Garbage-Collection auswirken, ist die Optimierung der Garbage-Collection kritisch.

Alle modernen JVMs verwenden Algorithmen für parallele Garbage-Collection, d. h., durch den Einsatz weiterer Kerne können die Pausen zwischen den Garbage-Collections reduziert werden. Ein physischer Server mit acht Kernen hat eine schnellere Garbage-Collection als ein physischer Server mit vier Kernen.

Wenn die Anwendung ein großes Datenvolumen für jede Partition verwalten muss, kann die Garbage-Collection ein entscheidender Faktor sein. In einem Szenario, in dem hauptsächlich Leseoperationen durchgeführt werden, ist die Leistung selbst bei sehr großen Heapspeichern (20 GB und mehr) in Ordnung, wenn eine Garbage-Collection nach Objektalter verwendet wird. Wenn der Heapspeicher für die permanenten Objekte gefüllt ist, ist jedoch eine Pause zu bemerken, die proportional zur Größe des Live-Heapspeichers und der Anzahl der Prozessoren im Computer ist. Diese Pause kann auf kleineren Computern mit großen Heapspeichern sehr lang sein.

Garbage-Collection von IBM Virtual Machine for Java

Für IBM Virtual Machine for Java verwenden Sie den Collector **`optavgpause`** in Szenarien mit einer hohene Aktualisierungsrate (100 % der Transaktionen ändern Einträge). Der Collector **`gencon`** funktioniert in Szenarien, in denen die Daten nur relativ selten aktualisiert werden (10 % der Zeit oder weniger), viel besser als der

Collector **optavgpause**. Experimentieren Sie mit beiden Collector, um festzustellen, welcher sich besser in Ihrem Szenario eignet. Führen Sie die Collector mit aktivierter ausführlicher Garbage-Collection aus, um zu prüfen, wie viel Zeit für die Garbage-Collection aufgebracht wird. Es wurden Szenarien gefunden, in denen 80 % der Zeit für die Garbage-Collection aufgebracht wurde, bis das Problem durch Optimierung behoben wurde.

Verwenden Sie den Parameter **-Xgcpolicy**, um den Garbage-Collection-Mechanismus zu ändern. Der Wert des Parameters **-Xgcpolicy** kann je nach verwendetem Garbage-Collector au f **-Xgcpolicy:gencon** oder **-Xgcpolicy:optavgpause** gesetzt werden.

- In einer Konfiguration mit WebSphere Application Server setzen Sie den Parameter **-Xgcpolicy** parameter über die Administrationskonsole. Klicken Sie auf **Server** > **Anwendungsserver** > **Servername** > **Prozessdefinition** > **Java Virtual Machine**. Fügen Sie den Parameter im Feld **Generische JVM-Argumente** hinzu.
- In eine eigenständigen Konfiguration übergeben Sie den Parameter **-jvmArgs** an das Script **startOgServer**, um den Garbage-Collector anzugeben. Der Parameter **-jvmArgs** muss der letzte Parameter sein, der an das Script übergeben wird.

Weitere Garbage-Collection-Optionen

Achtung: Wenn Sie eine JVM von Sun verwenden, müssen Sie möglicherweise Anpassungen an deren JVM, adjustments to the default Standard-Garbage-Collection und der Optimierungsrichtlinie vornehmen.

WebSphere eXtreme Scale unterstützt WebSphere Real Time Java. Bei WebSphere Real Time Java sind die Antwortzeiten der Transaktionsverarbeitung für WebSphere eXtreme Scale konsitenter und vorhersehbar. Deshalb sind die Auswirkungen der Garbage-Collection und der Threadplanung weitgehend minimal. Die Auswirkungen werden so weit reduziert, dass die Standardabweichung bei den Antwortzeiten weniger als 10 % als beim regulären Java beträgt.

JVM-Leistung

WebSphere eXtreme Scale kann in verschiedenen Versionen von Java Platform, Standard Edition ausgeführt werden. WebSphere eXtreme Scale unterstützt Java SE Version 5 und höher. Zur Steigerung der Entwicklerproduktivität und der Leistung verwenden Sie Java SE 5 oder höher, um die Vorteile von Annotationen und verbesserter Garbage-Collection zu nutzen. WebSphere eXtreme Scale kann in 32-Bit- und 64-Bit-JVMs ausgeführt werden.

WebSphere eXtreme Scale wurde mit einem Teil der verfügbaren virtuellen Maschinen getestet, aber die Liste der unterstützten JVMs ist nicht exklusiv. Sie können WebSphere eXtreme Scale in jeder JVM der Edition 5 oder höher jedes Anbieters ausführen. Wenn jedoch ein Problem mit einer JVM eines Anbieters auftritt, müssen Sie sich an den JVM-Anbieter wenden, um Unterstützung zu erhalten. Verwenden Sie, wenn möglich, auf jeder Plattfom, die von WebSphere Application Server unterstützt wird, die JVM aus der Laufzeitumgebung von WebSphere.

In den meisten Szenarien, in denen WebSphere eXtreme Scale verwendet wird, bietet Java SE Version 6 der JVM eine bessere Leistung als Edition 5. Im Allgemeinen erzielen Sie mit der aktuellsten verfügbaren Version von Java Platform, Standard Edition die beste Leistung.

Größe des Heapspeichers

Es wird empfohlen, Heapspeicher mit einer Größe von 1 bis 2 GB für eine JVM mit vier Kernen zu verwenden. Die optimale Größe des Heapspeichers ist von den folgenden Faktoren abhängig:

- Anzahl der Liveobjekte im Heapspeicher
- Komplexität der Liveobjekte im Heapspeicher
- Anzahl verfügbarer Kerne für die JVM

Eine Anwendung, die beispielsweise 10-K-Byte-Arrays speichert, kann einen viel größeren Heapspeicher haben als eine Anwendung, die komplexe POJO-Graphen verwendet.

Threadanzahl

Die Threadanzahl ist von verschiedenen Faktoren abhängig. Die Anzahl der Threads, die von einem einzelnen Shard verwaltet werden können, ist begrenzt. Ein Shard ist eine Instanz einer Partition. Ein Shard kann ein primäres Shard oder ein Replikat-Shard sein. Je mehr Shards jede JVM enthält, desto mehr Threads und mehr gemeinsame Datenzugriffspfade sind möglich. Jedes Shard ist so nebenläufig wie möglich, aber nichtsdestotrotz gibt es einen Grenzwert.

ORB-Anforderungen (Object Request Broker)

IBM SDK enthält eine IBM ORB-Implementierung, die mit WebSphere Application Server und WebSphere eXtreme Scale getestet wurde. Zur Vereinfachung des Unterstützungsprozesses sollten Sie eine von IBM bereitgestellte JVM verwenden. Andere JVM-Implementierungen verwenden einen anderen ORB. Der IBM ORB wird nur mit IBM Java Virtual Machines bereitgestellt. WebSphere eXtreme Scale erfordert für den Betrieb einen funktionierenden ORB. Sie können WebSphere eXtreme Scale mit ORBs anderer Anbieter verwenden. Tritt jedoch ein Problem mit diesem ORB auf, müssen Sie sich an den Anbieter dieses ORB wenden, um Unterstützung zu erhalten. Die IBM ORB-Implementierung ist mit JVMs anderer Anbieter kompatibel und kann bei Bedarf ersetzt werden.

Optimierung der Datei "orb.properties"

IBM hat für Tests die folgende Datei für Datengrids mit bis zu 1500 JVMs verwendet. Die Datei `orb.properties` befindet sich im Ordner `lib` der Laufzeitumgebung.

```
# Eigenschaften des IBM JDK für den ORB
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton

# WS-Interceptor
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer

# Eigenschaften des WS-ORB und der Plug-ins
com.ibm.CORBA.ForceTunnel=never
com.ibm.CORBA.RequestTimeout=10
com.ibm.CORBA.ConnectTimeout=10

# Erforderlich, wenn sehr viele JVMs gleichzeitig eine Verbindung zum Katalog herstellen
com.ibm.CORBA.ServerSocketQueueDepth=2048

# Clients und Katalogserver können offene Sockets zu allen JVMs haben
com.ibm.CORBA.MaxOpenConnections=1016

# Thread-Pool für die Verarbeitung eingehender Anforderungen, hier 200 Threads
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ThreadPool.MaximumSize=200
com.ibm.CORBA.ThreadPool.MinimumSize=200
com.ibm.CORBA.ThreadPool.InactivityTimeout=180000

# Keine Aufteilung großer Anforderungen/Antworten in kleinere Blöcke
com.ibm.CORBA.FragmentSize=0
```

Einstellung für das Intervall der Überwachungssignale für Failover-Erkennung optimieren

Sie können das Intervall, in dem das System nach ausgefallenen Servern sucht, mit der Einstellung für das Intervall der Überwachungssignale konfigurieren.

Informationen zu diesem Vorgang

Die Konfiguration des Failovers variiert je nach Typ der verwendeten Umgebung. Wenn Sie eine eigenständige Umgebung verwenden, können Sie das Failover über die Befehlszeile konfigurieren. Wenn Sie eine Umgebung mit WebSphere Application Server Network Deployment verwenden, müssen Sie das Failover über die Administrationskonsole von WebSphere Application Server Network Deployment konfigurieren.

Vorgehensweise

- Failover für eigenständige Umgebungen konfigurieren

  Sie können die Intervalle der Überwachungssignale über die Befehlszeile mit dem Parameter **-heartbeat** in der Scriptdatei **startOgServer** konfigurieren. Setzen Sie den Parameter auf einen der folgenden Werte:

  Tabelle 30. Intervall der Überwachungssignale

Wert	Aktion	Beschreibung
0	Typisch (Standardeinstellung)	Failover werden gewöhnlich innerhalb von 30 Sekunden erkannt.
-1	Aggressiv	Failover werden gewöhnlich innerhalb von 5 Sekunden erkannt.
1	Gelockert	Failover werden gewöhnlich innerhalb von 180 Sekunden erkannt.

  Ein aggressives Intervall der Überwachungssignale kann hilfreich sein, wenn die Prozesse und das Netz stabil sind. Wenn das Netz oder die Prozesse nicht optimal konfiguriert sind, können Überwachungssignale verpasst werden, was zu einer falschen Fehlererkennung führen kann.

- Failover für Umgebungen mit WebSphere Application Server konfigurieren

  Sie können WebSphere Application Server Network Deployment Version 6.0.2 und höher so konfigurieren, dass ein schnelles Failover von WebSphere eXtreme Scale unterstützt wird. Die Standard-Failover-Zeit für permanente Fehler sind 200 Sekunden. Ein permanenter Fehler ist ein physischer Computer- oder Serverabsturz, das Ziehen des Netzkabels oder ein Betriebssystemfehler. Bei Fehlern aufgrund von Prozessabstürzen oder temporären Fehlern findet das Failover gewöhnlich in weniger als einer Sekunde statt. Die Fehlererkennung für temporäre Fehler findet statt, wenn die Netz-Sockets des inaktiven Prozesses für den Server, in dem der Prozess ausgeführt wird, automatisch vom Betriebssystem geschlossen werden.

  Überwachungssignalkonfiguration für Stammgruppen

  Wenn WebSphere eXtreme Scale in einem Prozess von WebSphere Application Server ausgeführt wird, werden die Failover-Merkmale aus den Stammgruppeneinstellungen des Anwendungsservers übernommen. In den folgenden Abschnitten wird beschrieben, wie Sie die Überwachungssignaleinstellungen der Stammgruppe für verschiedene Versionen von WebSphere Application Server Network Deployment konfigurieren:

– **Stammgruppeneinstellungen für WebSphere Application Server Network Deployment Version 6.x und 7.x aktualisieren**

  Geben Sie das Intervall der Überwachungssignale in WebSphere Application Server Version 6.0 bis Version 6.1.0.12 in Sekunden und ab Version 6.1.0.13 in Millisekunden an. Außerdem müssen Sie die Anzahl verpasster Überwachungssignale angeben. Dieser Wert gibt an, wie viele Überwachungssignale verpasst werden können, bevor eine Peer-JVM als ausgefallen betrachtet wird. Die Erkennungszeit für permanente Fehler entspricht in etwa dem Produkt aus Intervall der Überwachungssignale und Anzahl verpasster Überwachungssignale.

  Diese Eigenschaften werden mit Hilfe von angepassten Eigenschaften in der Stammgruppe über die WebSphere-Administrationskonsole angegeben. Einzelheiten zur Konfiguration finden Sie im Abschnitt Angepasste Eigenschaften der Stammgruppe. Diese Eigenschaften müssen für alle Stammgruppen angegeben werden, die von Anwendungen verwendet werden:

  - Das Intervall der Überwachungssignale wird mit der angepassten Eigenschaft IBM_CS_FD_PERIOD_SEC (in Sekunden) bzw. der angepassten Eigenschaft IBM_CS_FD_PERIOD_MILLIS (in Millisekunden) (erfordert Version 6.1.0.13 oder höher) angegeben.
  - Die Anzahl verpasster Überwachungssignale wird mit der angepassten Eigenschaft IBM_CS_FD_CONSECUTIVE_MISSED angegeben.

  Der Standardwert für die Eigenschaft IBM_CS_FD_PERIOD_SEC ist 20, und der Standardwert für die Eigenschaft IBM_CS_FD_CONSECUTIVE_MISSED ist 10. Wenn Sie die Eigenschaft IBM_CS_FD_PERIOD_MILLIS angeben, überschreibt diese jede definierte angepasste Eigenschaft IBM_CS_FD_PERIOD_SEC. Die Werte dieser Eigenschaften sind positive ganze Zahlen.

  Verwenden Sie die folgenden Einstellungen, um eine Erkennungszeit von 1500 ms für Server der WebSphere Application Server Network Deployment Version 6.x zu erzielen:

  - Setzen Sie `IBM_CS_FD_PERIOD_MILLIS = 750` (WebSphere Application Server Network Deployment Version 6.1.0.13 und höher).
  - Setzen Sie `IBM_CS_FD_CONSECUTIVE_MISSED = 2`.

– **Stammgruppeneinstellungen für WebSphere Application Server Network Deployment Version 7.0 aktualisieren**

  WebSphere Application Server Network Deployment Version 7.0 stellt zwei Stammgruppeneinstellungen bereit, die angepasst werden können, um die Failover-Erkennungszeit zu erhöhen oder zu verringern:

  - **Übertragungsintervall für Überwachungssignale.** Der Standardwert sind 30.000 Millisekunden.
  - **Überwachungssignalzeitlimit.** Der Standardwert sind 180.000 Millisekunden.

  Weitere Einzelheiten zum Ändern dieser Einstellungen finden Sie im Information Center von WebSphere Application Server Network Deployment unter "Einstellungen für die Erkennung und Fehlererkennung".

  Verwenden Sie die folgenden Einstellungen, um eine Fehlererkennungszeit von 1500 ms für Server der WebSphere Application Server Network Deployment Version 7 zu erzielen:

  - Setzen Sie das Übertragungsintervall für Überwachungssignale auf 750 Millisekunden.
  - Setzen Sie das Überwachungssignalzeitlimit auf 1500 Millisekunden.

Nächste Schritte

Wenn Sie diese Einstellungen ändern, um kürzere Failover-Zeiten anzugeben, müssen verschiedene Probleme bei der Systemoptimierung beachtet werden. Java ist keine Echtzeitumgebung. Es ist möglich, dass Threads verzögert werden, wenn die JVM lange Garbage-Collection-Zeiten verzeichnet. Threads können auch verzögert werden, wenn die Maschine, auf der die JVM ausgeführt wird, unter hoher Last steht (durch die JVM selbst oder durch andere Prozesse, die auf der Maschine ausgeführt werden). Wenn Threads verzögert werden, werden Überwachungssignale möglicherweise nicht rechtzeitig gesendet. Im schlimmsten Fall werden die durch die erforderliche Failover-Zeit verzögert. Wenn Threads verzögert werden, treten falsche Fehlererkennungen auf. Das System muss optimiert und dimensioniert werden, um sicherzustellen, dass falsche Fehlererkennungen in der Produktionsumgebung nicht auftreten. Dies kann am Zuverlässigsten durch angemessene Lasttests sichergestellt werden.

Anmerkung: Die aktuelle Version von eXtreme Scale unterstützt WebSphere Real Time.

Garbage-Collection mit WebSphere Real Time optimieren

Die Verwendung von WebSphere eXtreme Scale mit WebSphere Real Time erhöht die Konsistenz und die Voraussagbarkeit, verringert jedoch den Leistungsdurchsatz im Vergleich mit der Standard-Garbage-Collection-Richtlinie, die in der standardmäßig verwendeten IBM Java™ SE Runtime Environment (JRE) genutzt wird. Die Kosten/Nutzen-Aussage kann variieren. WebSphere eXtreme Scale erstellt zahlreiche temporäre Objekte für jede Transaktion. Die temporären Objekte beziehen sich auf Anforderungen, Antworten, Protokollfolgen und Sitzungen. Wenn Sie WebSphere Real Time nicht verwenden, kann die Antwortzeit auf mehrere Hundert Millisekunden ansteigen. Der Einsatz von WebSphere Real Time mit WebSphere eXtreme Scale kann die Effizienz der Garbage-Collection steigern und die Antwortzeit auf 10 % der Antwortzeit in der eigenständigen Konfiguration verringern.

WebSphere Real Time in einer eigenständigen Umgebung

Sie können WebSphere Real Time mit WebSphere eXtreme Scale verwenden. Durch die Aktivierung von WebSphere Real Time erreichen Sie eine vorhersehbarere Garbage-Collection mit stabilen, konsistenten Antwortzeiten und Transaktionsdurchsätzen in einer eigenständigen eXteme-Scale-Umgebung.

Vorteile von WebSphere Real Time

WebSphere eXtreme Scale erstellt zahlreiche temporäre Objekte für jede Transaktion. Die temporären Objekte beziehen sich auf Anforderungen, Antworten, Protokollfolgen und Sitzungen. Wenn Sie WebSphere Real Time nicht verwenden, kann die Antwortzeit auf mehrere Hundert Millisekunden ansteigen. Der Einsatz von WebSphere Real Time mit WebSphere eXtreme Scale kann die Effizienz der Garbage-Collection steigern und die Antwortzeit auf 10 % der Antwortzeit in der eigenständigen Konfiguration verringern.

WebSphere Real Time aktivieren

Installieren Sie WebSphere Real Time und die eigenständige Konfiguration von WebSphere eXtreme Scale auf den Computern, auf denen Sie eXtreme Scale ausführen möchten. Setzen Sie die Umgebungsvariable JAVA_HOME so, dass sie auf eine Standard-JRE (Java SE Runtime Environment) zeigt.

Setzen Sie die Umgebungsvariable JAVA_HOME so, dass sie auf das installierte Produkt WebSphere Real Time zeigt. Aktivieren Sie WebSphere Real Time anschließend wie folgt.

1. Editieren Sie die Datei `objectgridRoot/bin/setupCmdLine.sh | .bat` der eigenständigen Installation, indem Sie das Kommentarzeichen aus der folgenden Zeile entfernen:

   ```
   WXS_REAL_TIME_JAVA="-Xrealtime -Xgcpolicy:metronome
   -Xgc:targetUtilization=80"
   ```

2. Speichern Sie die Datei.

Jetzt haben Sie WebSphere Real Time aktiviert. Wenn Sie WebSphere Real Time inaktivieren möchten, können Sie der Zeile das Kommentarzeichen wieder hinzufügen.

Bewährte Verfahren

Wenn Sie WebSphere Real Time einsetzen, sind die Antwortzeiten von eXtreme-Scale-Transaktionen vorhersehbarer. Die Ergebnisse zeigen, dass sich die Abweichung der Antwortzeit einer eXtreme-Scale-Transaktion mit WebSphere Real Time im Vergleich zum Standard-Java mit dem Standard-Garbage-Collector erheblich verbessert. Die Aktivierung von WebSphere Real Time mit eXtreme Scale ist optimal, wenn Stabilität und Antwortzeiten Ihrer Anwendung von entscheidender Bedeutung sind.

Die in diesem Abschnitt beschriebenen bewährten Verfahren verdeutlichen, wie WebSphere eXtreme Scale durch Optimierung und Codeverfahren für die erwartete Last effizienter gemacht werden kann.

- Legen Sie die richtige Prozessorbelegungsstufe für Ihre Anwendung und den Garbage-Collector fest.

  WebSphere Real Time bietet die Möglichkeit, die Prozessorbelegung zu steuern, so dass die Auswirkungen der Garbage-Collection auf Ihre Anwendung kontrolliert und minimiert werden. Verwenden Sie den Parameter `-Xgc:targetUtilization=NN`, um `NN` Prozent der Prozessorkapazität festzulegen, die alle 20 Sekunden von Ihrer Anwendung belegt werden. Der Standardwert für WebSphere eXtreme Scale ist 80 %, aber Sie können das Script in der Datei `objectgridRoot/bin/setupCmdLine.sh` ändern und eine andere Zahl festlegen, wie z. B. `70`, womit Sie dem Garbage-Collector mehr Prozessorkapazität zuweisen. Implementieren Sie genügend Server, um die Prozessorbelegung für Ihre Anwendungen unter 80 % zu halten.
- Legen Sie einen höheren Wert für die Heapspeichergröße fest.

  WebSphere Real Time belegt mehr Hauptspeicher als reguläres Java. Planen Sie WebSphere eXtreme Scale deshalb mit einem größeren Heapspeicher, und legen Sie die Heapspeichergröße beim Start der Katalogserver und Container mit dem Parameter `–jvmArgs –XmxNNNM` im Befehl **ogStartServer** fest. Sie können den Parameter `–jvmArgs –Xmx500M` beispielsweise verwenden, um Katalogserver zu starten und eine entsprechende Hauptspeichergröße zum Starten der Container zu verwenden. Sie können die Hauptspeichergröße auf 60-70 % der erwarteten Datenmenge pro JVM setzen. Wenn Sie diesen Wert nicht festlegen, kann ein Fehler des Typs "OutOfMemoryError" auftreten. Optional können Sie auch den Parameter `–jvmArgs –Xgc:noSynchronousGCOnOOM` verwenden, um ein nicht deterministisches Verhalten zu verhindern, wenn in der JVM eine abnormale Speicherbedingung auftritt.
- Threads für die Garbage-Collection anpassen.

WebSphere eXtreme Scale erstellt eine Vielzahl temporärer Objekte für jede Transaktion und jeden RPC-Thread (Remote Procedure Call). Die Garbage-Collection bietet eine bessere bessere Leistung, wenn Ihr Computer genügtend Prozessorzyklen besitzt. Die Standardanzahl der Threads ist 1. Sie können die Anzahl der Threads mit dem Argument `–Xgcthreads n` anpassen. Der vorgeschlagene Wert für dieses Argument ist die Anzahl der verfügbaren Kerne unter Berücksichtigung der Anzahl der Java Virtual Machines pro Computer.

- Leistung für Anwendungen mit kurzer Laufzeit mit WebSphere eXtreme Scale anpassen.

  WebSphere Real Time ist für Anwendungen mit langer Laufzeit optimiert. Gewöhnlich müssen Transaktionen von WebSphere eXtreme Scale kontinuierlich über einen Zeitraum von two Stunden hinweg ausgeführt werden, um zuverlässige Leistungsdaten zu erhalten. Sie können den Parameter `–Xquickstart` verwenden, um die Leistung Ihrer Anwendungen mit kurzer Laufzeit zu verbessern. Dieser Parameter weist den JIT-Compiler an, die untere Stufe der Optimierung zu verwenden.

- Clientwarteschlange von WebSphere eXtreme Scale und Clientvermittlung von WebSphere eXtreme Scale minimieren

  Der Hauptvorteil der Verwendung von WebSphere eXtreme Scale mit WebSphere Real Time ist eine hoch zuverlässige Transaktionsantwortzeit, die gewöhnlich erhebliche Verbesserungen bei der Abweichung der Transaktionsantwortzeiten zur Folge hat. Alle in die Warteschlange eingereihten Clientanforderungen und Clientanforderungsvermittlungen über andere Software wirken sich auf die Antwortzeit aus, die außerhalb der Kontrolle von WebSphere Real Time und WebSphere eXtreme Scale liegt. Sie sollten müssen Ihre Thread- und Socket-Parameter ändern, um eine stabile und gleichmäßige Last ohne größere Verzögerungen zu erzielen und die Länge der Warteschlangen zu verringern.

- Anwendungen von WebSphere eXtreme Scale so schreiben, dass sie das Threading von WebSphere Real Time verwenden

  Sie können ohne Änderung Ihrer Anwendung hoch zuverlässige Transaktionsantwortzeiten in WebSphere eXtreme Scale mit erheblichen Verbesserungen bei der Antwortzeitabweichung erreichen. Außerdem können Sie Threading für Ihre transaktionsorientierten Anwendungen (von regulären Java-Threads zu Realtime-Threads) nutzen, das Ihnen eine bessere Steuerung der Thread-Prioritäten und der Planung bietet.

  Ihre Anwendung enthält derzeit den folgenden Code:

  ```
  public class WXSCacheAppImpl extends Thread implements WXSCacheAppIF
  ```

  Optional können Sie diesen Code durch Folgenden ersetzen.

  ```
  public class WXSCacheAppImpl extends RealtimeThread implements
  WXSCacheAppIF
  ```

WebSphere Real Time in WebSphere Application Server

Sie können WebSphere® Real Time mit eXtreme Scale in einer Umgebung von WebSphere Application Server Network Deployment Version 7.0 verwenden. Durch die Aktivierung von WebSphere Real Time erreichen Sie eine vorhersehbarere Garbage-Collection mit stabilen, konsistenten Antwortzeiten und Transaktionsdurchsätzen.

Vorteile

Die Verwendung von WebSphere eXtreme Scale mit WebSphere Real Time erhöht die Konsistenz und die Vorhersagbarkeit, verringert jedoch den Leistungsdurchsatz im Vergleich mit der Standard-Garbage-Collection-Richtlinie, die in der standard-

mäßig verwendeten IBM Java™ SE Runtime Environment (JRE) genutzt wird. Die Kosten/Nutzen-Aussage kann je nach Kriterien variieren. Im Folgenden sind einige der Hauptkriterien aufgeführt:

- Serverkapazitäten - Verfügbarer Hauptspeicher, CPU-Geschwindigkeit und -kapazität, Netzgeschwindigkeit und -belegung
- Serverlast – CPU-Dauerlast, CPU-Spitzenlast
- Java-Konfiguration – Größe des Heapspeichers, Zielbelegung, Garbage-Collection-Threads
- Kopiermoduskonfiguration von WebSphere eXtreme Scale – Bytefeldgruppe vs. POJO-Speicher
- Anwendungsspezifikationen – Thread-Belegung, Antwortzeitvoraussetzungen und -toleranz, Objektgröße usw.

Zusätzlich zu der in WebSphere Real Time verfügbaren Metronom-Garbage-Collection-Richtlinie gibt es optionale Garbage-Collection-Richtlinien in der Standard-JRE von IBM (Java™ SE Runtime Environment). Diese Richtlinien, optthruput (Standard), gencon, optavgpause und subpool, sind speziell für die verschiedenartigen Anwendungsanforderungen und -umgebungen konzipiert. Weitere Informationen zu diesen Richtlinien finden Sie unter „Java Virtual Machines optimieren" auf Seite 504. Je nach Anwendungs- und Umgebungsanforderungen, Ressourcen und Einschränkungen, können Sie durch die Prototyperstellung einer oder mehrerer dieser Garbage-Collection-Richtlinien sicherstellen, dass die Anforderungen erfüllt werden, und eine optimale Richtlinie festlegen.

Möglichkeiten mit WebSphere Application Server Network Deployment

1. Im Folgenden sind einige der unterstützten Versionen aufgelistet:
   - WebSphere Application Server Network Deployment Version 7.0.0.5 und höher
   - WebSphere Real Time V2 SR2 for Linux und höher. Weitere Informationen finden Sie auf der Website von IBM WebSphere Real Time V2 for Linux.
   - WebSphere eXtreme Scale Version 7.0.0.0 und höher
   - 32- und 64-Bit-Linux-Betriebssysteme
2. WebSphere-eXtreme-Scale-Server können nicht mit einem Deployment Manager von WebSphere Application Server kollokiert werden.
3. Real Time unterstützt keine Deployment Manager.
4. Real Time unterstützt keine WebSphere-Node-Agents.

WebSphere Real Time aktivieren

Installieren Sie WebSphere Real Time und WebSphere eXtreme Scale auf den Computern, auf denen Sie eXtreme Scale ausführen möchten. Aktualisieren Sie WebSphere Real Time Java auf SR2.

Sie können die JVM-Einstellungen für jeden Server über die Konsole von WebSphere Application Server Version 7.0 wie folgt festlegen.

Wählen Sie **Server** > **Servertypen** > **WebSphere-Anwendungsserver** > **<erforderlicher installierter Server>** aus.

Wählen Sie auf der daraufhin angezeigten Seite "Prozessdefinition" aus.

Klicken Sie auf der nächsten Seite oben in der rechten Spalte auf Java Virtual Machine. (Hier können Sie die Größe des Heapspeicher, die Garbage-Collection und weitere Flags für jeden Server definieren.)

Setzen Sie die folgenden Flags im Feld "Generische JVM-Argumente":

```
-Xrealtime -Xgcpolicy:metronome  -Xnocompressedrefs -Xgc:targetUtilization=80
```

Wenden Sie die Änderungen an, und speichern Sie sie.

Wenn Sie Real Time in WebSphere Application Server 7.0 mit eXtreme-Scale-Servern, einschließlich der zuvor genannten JVM-Flags, verwenden möchten, müssen Sie eine Umgebungsvariable JAVA_HOME erstellen.

Setzen Sie JAVA_HOME wie folgt.

1. Klicken Sie auf "Umgebung".
2. Wählen Sie "WebSphere-Variablen" aus.
3. Stellen Sie sicher, dass das Feld "Alle Geltungsbereiche" unter "Geltungsbereich anzeigen" ausgewählt ist.
4. Wählen Sie den erforderlichen Server in der Dropdown-Liste aus. (Wählen Sie weder Deployment-Manager- noch Node-Agent-Server aus.)
5. Wenn die Umgebungsvariable JAVA_HOME nicht aufgelistet ist, wählen Sie "Neu" aus, und geben Sie JAVA_HOME als Variablennamen an. Geben Sie im Feld "Wert" den vollständig qualifizierten Pfadnamen von Real Time an.
6. Wenden Sie Ihre Änderung an, und speichern Sie sie.

Bewährte Verfahren

Ein Satz bewährter Verfahren ist im Abschnitt "Bewährte Verfahren" unter „Garbage-Collection mit WebSphere Real Time optimieren” auf Seite 509 beschrieben. In dieser Liste bewährter Verfahren sind verschiedene wichtige Änderungen für eine eigenständige Umgebung von WebSphere eXtreme Scale zu beachten, wenn das Produkt in einer Umgebung von WebSphere Application Server Network Deployment installiert wird.

Sie müssen alle zusätzlichen JVM-Befehlszeilenparameter an derselben Position wie die im vorherigen Abschnitt beschriebenen Parameter für die Garbage-Collection-Richtlinie platzieren.

Ein annehmbarer Anfangszielwert für die Dauerlast der Prozessoren ist 50 % mit kurzfristigen Spitzenlasten bis zu 75 %. Wenn die Last diese Werte überschreitet, müssen Sie zusätzliche Kapazitäten hinzufügen, bevor Sie feststellen, dass Vorhersagbarkeit und Konsistenz messbar nachlassen. Sie können die Leistung geringfügig erhöhen, wenn Sie längere Antwortzeiten tolerieren. Das Überschreiten eines Schwellenwerts von 80 % führt häufig zu einer signifikanten Verschlechterung von Konsistenz und Vorhersagbarkeit.

Dynamischen Cache-Provider optimieren

Der dynamische Cache-Provider von WebSphere eXtreme Scale unterstützt die folgenden Konfigurationsparameter für die Leistungsoptimierung.

Informationen zu diesem Vorgang

- **com.ibm.websphere.xs.dynacache.ignore_value_in_change_event**: Wenn Sie einen Listener für Änderungsereignisse beim dynamischen Cache-Provider regist-

rieren und eine ChangeEvent-Instanz generieren, entstehen Kosten für die Entserialisierung des Cacheeintrags, so dass der Wert in das ChangeEvent gestellt werden kann. Die Entserialisierung des Cacheeintrags wird beim Generieren von CacheEvents übersprungen, wenn Sie diesen optionalen Parameter in der Cacheinstanz auf `true` setzen. Der zurückgegebene Wert ist null, wenn eine Entfernungsoperation durchgeführt wird, bzw. eine Bytefeldgruppe, die die serialisierte Form des Objekts enthält. InvalidationEvent-Instanzen bringen ähnliche Leistungseinbußen mit sich, die Sie verhindern können, indem Sie "com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent" auf `true` setzen.

- **com.ibm.websphere.xs.dynacache.enable_compression**: Standardmäßig komprimiert der dynamische Cache-Provider von eXtreme Scale die Cacheeinträge im Hauptspeicher, um die Cachedichte zu erhöhen. Auf diese Weise können Sie den Speicherbedarf für Anwendungen wie Servlet-Caching erheblich reduzieren. Wenn Sie wissen, dass die meisten Cachedaten nicht komprimierbar sind, sollten Sie diese Einstellung auf `false` setzen.

Kapitel 10. Sicherheit

WebSphere eXtreme Scale kann den Datenzugriff sichern, unter anderem durch Integration mit externen Sicherheitsprovidern. Zu den Aspekten der Sicherheit gehören Authentifizierung, Berechtigung, Transportsicherheit, Datengridsicherheit, lokale Sicherheit und JMX-Sicherheit (MBean).

Anwendungsclientauthentifizierung

Die Anwendungsclientauthentifizierung setzt sich aus der Aktivierung der Client/Server-Sicherheit, der Authentifizierung des Berechtigungsnachweises und der Konfiguration eines Authentifikators und eines Generators für Systemberechtigungsnachweise zusammen.

Client/Server-Sicherheit aktivieren

Für eine erfolgreiche Authentifizierung bei ObjectGrid müssen Sie die Sicherheit im Client und im Server aktivieren.

Clientsicherheit aktivieren

WebSphere eXtreme Scale stellt eine Beispielclienteigenschaftendatei mit dem Namen `sampleClient.properties` im Verzeichnis *WAS-Stammverzeichnis*`/optionalLibraries/ObjectGrid/properties` für eine Installation von WebSphere Application Server bzw. im Verzeichnis `/ObjectGrid/properties` für eine heterogene Serverinstallation bereit. Sie können diese Schablonendatei mit entsprechenden Werten anpassen. Setzen Sie die Eigenschaft "securityEnabled" in der Datei `objectgridClient.properties` auf `true`. Die Eigenschaft "securityEnabled" gibt an, ob die Sicherheit aktiviert ist. Wenn ein Client eine Verbindung zu einem Server herstellt, muss die Eigenschaft "securityEnabled" auf der Client- und auf der Serverseite denselben Wert haben: `true` oder `false`. Ist die Sicherheit beispielsweise im verbindungsherstellenden Server aktiviert, muss die Eigenschaft im Client auf `true` gesetzt werden, damit die Verbindung zum Server hergestellt werden kann.

Die Schnittstelle "com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration" stellt die Datei `security.ogclient.props` dar. Sie können die allgemein zugängliche Anwendungsprogrammierschnittstelle "com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory" verwenden, um eine Instanz dieser Schnittstelle mit Standardwerten zu erstellen. Sie können aber auch eine Instanz erstellen, indem Sie die Datei mit den Sicherheitseigenschaften des ObjectGrid-Clients übergeben. Die Datei `security.ogclient.props` enthält weitere Eigenschaften. Weitere Einzelheiten finden Sie in der Dokumentation zur API "ClientSecurityConfiguration" und in der Dokumentation zur API "ClientSecurityConfigurationFactory".

Serversicherheit aktivieren

Zum Aktivieren der Sicherheit auf der Serverseite können Sie die Eigenschaft **`securityEnabled`** in der Datei `security.xml` auf `true` setzen. Verwenden Sie eine XML-Sicherheitsdeskriptordatei, um die Sicherheit des Datengrids so zu definieren, das die gridweite Sicherheitskonfiguration von der Konfiguration, die sich nicht auf die Sicherheit bezieht, isoliert wird.

Authentifizierung des Berechtigungsnachweises

Nachdem der eXtreme-Scale-Client das Credential-Objekt mit dem CredentialGenerator-Objekt abgerufen hat, wird das Credential-Objekt zusammen mit der Clientanforderung an den eXtreme-Scale-Server gesendet. Der Server authentifiziert das Credential-Objekt, bevor er die Anforderung verarbeitet. Bei erfolgreicher Authentifizierung des Credential-Objekts wird ein Subject-Objekt zurückgegeben, dass dieses Credential-Objekt repräsentiert. Dieses Subject-Objekt wird anschließend für die Berechtigung der Anforderung verwendet.

Setzen Sie die Eigenschaft **credentialAuthentication** in den Client- und Servereigenschaftendateien so, dass die Authentifizierung des Berechtigungsnachweises aktiviert wird. Weitere Informationen finden Sie in den Abschnitten Clienteigenschaftendatei und Servereigenschaftendatei.

Die folgende Tabelle enthält eine Übersicht über die für verschiedene Einstellungen zu verwendenden Authentifizierungsverfahren.

Tabelle 31. Authentifizierung des Berechtigungsnachweises bei Client- und Servereinstellungen

Authentifizierung des Clientberechtigungsnachweises	Authentifizierung des Serverberechtigungsnachweises	Ergebnis
Nein	Nie	Inaktiviert
Nein	Unterstützt	Inaktiviert
Nein	Erforderlich	Fehlersituation
Unterstützt	Nie	Inaktiviert
Unterstützt	Unterstützt	Aktiviert
Unterstützt	Erforderlich	Aktiviert
Erforderlich	Nie	Fehlersituation
Erforderlich	Unterstützt	Aktiviert
Erforderlich	Erforderlich	Aktiviert

Authentifikator konfigurieren

Der eXtreme-Scale-Server verwendet das Authenticator-Plug-in, um das Credential-Objekt zu authentifizieren. Eine Implementierung der Schnittstelle "Authenticator" ruft das Credential-Objekt ab und authentifiziert es dann bei einer Benutzerregistry, z. B. einem LDAP-Server (Lightweight Directory Access Protocol) usw. eXtreme Scale stellt keine Registrykonfiguration bereit. Die Herstellung einer Verbindung zu und die Authentifizierung bei einer Benutzerregistry muss in diesem Plug-in implementiert werden.

Eine Authenticator-Implementierung extrahiert beispielsweise die Benutzer-ID und das Kennwort aus dem Berechtigungsnachweis, verwendet diese Informationen für die Herstellung einer Verbindung zu einem und Validierung bei einem LDAP-Server und erstellt als Ergebnis der Authentifizierung ein Subject-Objekt. Die Implementierung kann JAAS-Anmeldemodule (Java Authentication and Authorization Service) verwenden. Als Ergebnis der Authentifizierung wird ein Subject-Objekt zurückgegeben.

Sie können den Authentifikator, wie im folgenden Beispiel gezeigt, in der XML-Sicherheitsdeskriptordatei konfigurieren:

```
<?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>
```

Verwenden Sie die Option **-clusterSecurityFile** beim Starten eines sicheren Servers, um die XML-Sicherheitsdatei festzulegen. Weitere Informationen finden Sie im Lernprogramm zur Java-SE-Sicherheit in der Veröffentlichung *Produktübersicht*.

Generator für Systemberechtigungsnachweise konfigurieren

Der Generator für Systemberechtigungsnachweise wird für die Darstellung einer Factory für die Systemberechtigungsnachweise verwendet. Ein Systemberechtigungsnachweis gleicht einem Administratorberechtigungsnachweis. Sie können das Element SystemCredentialGenerator, wie im folgenden Beispiel gezeigt, in der XML-Katalogsicherheitsdatei konfigurieren:

```
<systemCredentialGenerator className ="com.ibm.websphere.objectgrid.security.plugins.
 builtins.UserPasswordCredentialGenerator">
                    <property name="properties" type="java.lang.String" value="manager manager1"
      description="username password" />
  </systemCredentialGenerator>
```

Zu Demonstrationszwecken werden Benutzername und Kennwort in Klartext gespeichert. Speichern Sie den Benutzernamen und das Kennwort in einer Produktionsumgebung nicht in Klartext.

WebSphere eXtreme Scale stellt einen Standardgenerator für Systemberechtigungsnachweise bereit, der die Serverberechtigungsnachweise verwendet. Wenn Sie den Generator für Systemberechtigungsnachweise nicht explizit angeben, wird dieser Standardgenerator für Systemberechtigungsnachweise verwendet.

Anwendungsclientberechtigung

Die Anwendungsclientberechtigung setzt sich aus ObjectGrid-Berechtigungsklassen, Berechtigungsmechanismen, einem Berechtigungsprüfintervall und dem Feature "Zugriff nur durch Ersteller" zusammen.

Bei eXtreme Scale basiert die Berechtigung auf dem Subject-Objekt und auf Berechtigungen. Das Produkt unterstützt zwei Arten von Berechtigungsmechanismen: Java Authentication and Authorization Service (JAAS) und angepasste Berechtigungen.

ObjectGrid-Berechtigungsklassen

Die Berechtigung basiert auf Berechtigungen. Es gibt die folgenden vier verschiedenen Typen von Berechtigungsklassen:

- Die Klasse MapPermission stellt Berechtigungen für den Zugriff auf die Daten in ObjectGrid-Maps dar.
- Die Klasse ObjectGridPermission stellt Berechtigungen für den Zugriff auf ObjectGrid dar.

- Die Klasse ServerMapPermission stellt Berechtigungen für den Zugriff auf ObjectGrid-Maps auf der Serverseite über einen Client dar.
- Die Klasse AgentPermission stellt Berechtigungen zum Starten eines Agenten auf der Serverseite dar.

Weitere Informationen zu APIs und den zugehörigen Berechtigungen finden Sie im Artikel zur Programmierung der Clientberechtigung in der Veröffentlichung *Programmierung*.

Berechtigungsprüfintervall

eXtreme Scale unterstützt das Caching der Berechtigungsprüfergebnisse für Leistungszwecke. Wenn dieser Mechanismus nicht verwendet wird und eine Methode, die in der Liste der Methoden für Ihre spezielle Berechtigungsklasse aufgeführt ist, aufgerufen wird, ruft die Laufzeitumgebung den konfigurierten Berechtigungsmechanismus für die Berechtigung des Zugriffs auf. Wenn das Berechtigungsprüfintervall definiert ist, wird der Berechtigungsmechanismus auf der Basis des festgelegten Berechtigungsprüfintervalls in regelmäßigen Abständen aufgerufen. Eine Liste der Methoden für jede Berechtigungsklasse finden Sie im Artikel zur Programmierung der Clientberechtigung in der Veröffentlichung *Programmierung*.

Die Berechtigungsinformationen basieren auf dem Subject-Objekt. Wenn ein Client versucht, auf die Methoden zuzugreifen, durchsucht die Laufzeitumgebung von eXtreme Scale den Cache nach dem Subject-Objekt. Wird das Objekt nicht im Cache gefunden, prüft die Laufzeitumgebung die für dieses Subject-Objekt erteilten Berechtigungen und speichert dann die Berechtigungen in einem Cache.

Das Berechtigungsprüfintervall muss vor der Initialisierung von ObjectGrid definiert werden. Sie können das Berechtigungsprüfintervall auf zwei Arten konfigurieren:

Sie können die ObjectGrid-XML-Datei verwenden, um ein ObjectGrid zu definieren und das Berechtigungsprüfintervall festzulegen. Im folgenden Beispiel wird das Berechtigungsprüfintervall auf 45 Sekunden gesetzt:

```
<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>
```

Wenn Sie ein ObjectGrid über APIs erstellen möchten, rufen Sie die folgende Methode auf, um das Berechtigungsprüfintervall festzulegen. Diese Methode kann nur vor der Initialisierung der ObjectGrid-Instanz aufgerufen werden. Sie gilt nur dann für das lokale eXtreme-Scale-Programmiermodell, wenn Sie die ObjectGrid-Instanz direkt instanziieren.

```
/**
 * Diese Methode akzeptiert einen einzigen Parameter, der angibt, wie oft
 * die Berechtigung geprüft werden soll, die verwendet wird, um einem
 * Clientzugriff zuzulassen. Wenn der Parameter den Wert 0 hat, wird
 * der Berechtigungsmechanismus (JAAS-Berechtigung oder angepasste
 * Berechtigung) bei jedem Aufruf von get/put/update/remove/evict
 * aufgefordert zu prüfen, ob das aktuelle Subject-Objekt die erforderlichen
 * Berechtigungen besitzt. Dies kann je nach Berechtigungsimplementierung
 * vom Leistungsstandpunkt aus gesehen untragbar kostenintensiv sein, aber
 * sollten Sie den Berechtigungsmechanismus jemals aufrufen müssen, dann
 * setzen Sie den Parameter auf 0.
 * Wenn Sie den Parameter auf einen Wert > 0 setzen, gibt er an, wie lange
 * (in Sekunden) ein Berechtigungssatz zwischengespeichert werden soll, bevor
 * sie zur Aktualisierung an den Berechtigungsmechanismus zurückgegeben werden.
 * Dieser Wert liefert eine sehr viel bessere Leistung, aber wenn die
 * Back-End-Berechtigungen in dieser Zeit geändert werden, kann das
 * ObjectGrid den Zugriff zulassen oder verweigern, obwohl der
```

```
 * Back-End-Sicherheitsprovider geändert wurde.
 *
 * @param Zeitraum - Das Berechtigungsprüfintervall in Sekunden
 */
void setPermissionCheckPeriod(int period);
```

Berechtigung "Zugriff nur durch Ersteller"

Die Berechtigung "Zugriff nur durch Ersteller" gewährleistet, dass ausschließlich der Benutzer (dargestellt durch die zugeordneten Principal-Objekte), der den Eintrag in die ObjectGrid-Map einfügt, auf diesen Eintrag zugreifen (lesen, aktualisieren, ungültig machen und entfernen) kann.

Das vorhandene Berechtigungsmodell für ObjectGrid-Maps basiert auf dem Zugriffstyp, aber nicht auf Dateneinträgen. Anders ausgedrückt, ein Benutzer kann mit einem bestimmten Zugriffstyp (z. B. lesen, schreiben, einfügen, löschen oder ungültig machen) entweder auf alle Daten in der Map oder auf keine Daten in der Map zugreifen. eXtreme Scale berechtigt Benutzer jedoch nicht für den Zugriff auf einzelne Dateneinträge. Dieses Feature bietet eine neue Methode für die Berechtigung von Benutzern für den Zugriff auf Dateneinträge.

In einem Szenario, in dem verschiedene Benutzer auf verschiedene Datengruppen zugreifen, kann dieses Modell hilfreich sein. Wenn der Benutzer Daten aus dem persistenten Speicher in die ObjectGrid-Maps lädt, kann der Zugriff vom persistenten Speicher berechtigt werden. In diesem Fall muss keine weitere Berechtigung auf der Ebene der ObjectGrid-Maps erfolgen. Sie müssen lediglich sicherstellen, dass die Person, die die Daten in die Map lädt, auf die Map zugreifen kann, indem Sie das Feature "Zugriff nur durch Ersteller" aktivieren.

Werte für das Attribut "accessByCreatorOnlyMode":

`disabled`
: Das Feature "Zugriff nur durch Ersteller" ist inaktiviert.

`complement`
: Das Feature "Zugriff nur durch Ersteller" ist aktiviert, um die Map-Berechtigung zu ergänzen. In anderen Worten, die Map-Berechtigung und das Feature "Zugriff nur durch Ersteller" sind wirksam. Deshalb können Sie die Datenoperationen weiter einschränken. Der Ersteller kann die Daten beispielsweise nicht ungültig machen.

`supersede`
: Das Feature "Zugriff nur durch Ersteller" ist aktiviert, um die Map-Berechtigung außer Kraft zu setzen. In anderen Worten, das Feature "Zugriff nur durch Ersteller" setzt die Map-Berechtigung außer Kraft, d. h., es wird keine Map-Berechtigung durchgeführt.

Sie können den Modus für das Feature "Zugriff nur durch Ersteller" auf zwei Arten konfigurieren:

Mit einer XML-Datei:

Sie können die ObjectGrid-XML-Datei verwenden, um ein ObjectGrid zu definieren und den Modus für das Feature "Zugriff nur durch Ersteller" auf `disabled` (Inaktiviert), `complement` (Ergänzung) oder `supersede` (Überlagern) zu setzen,

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
         accessByCreatorOnlyMode="supersede"
        <bean id="TransactionCallback"
              classname="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
    ...
</objectGrids>
```

Programmgesteuert:

Wenn Sie ein ObjectGrid über das Programm erstellen möchten, können Sie die folgende Methode aufrufen, um den Modus für das Feature "Zugriff nur durch Ersteller" festzulegen. Der Aufruf dieser Methode gilt nur dann für das lokale eXtreme-Scale-Programmiermodell, wenn Sie die ObjectGrid-Instanz direkt instanziieren:

```
/**
 * Legen Sie den Modus für das Feature "Zugriff nur durch Ersteller"
 * (ACCESS_BY_CREATOR_ONLY) fest.
 * Wenn Sie das Feature "Zugriff nur durch Ersteller" aktivieren, kann nur
 * der Benutzer (dargestellt durch die zugeordneten Principals), der den
 * Datensatz in die Map einfügt, auf den Datensatz zugreifen (lesen,
 * aktualisieren, ungültig machen und entfernen).
 * Das Feature "Zugriff nur durch Ersteller"  kann inaktiviert werden
 * oder das ObjectGrid-Berechtigungsmodell ergänzen oder sogar außer
 * Kraft setzen. Standardmäßig ist das Feature inaktiviert:
 * {@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);
```

Zur weiteren Veranschaulichung stellen Sie sich ein Szenario vor, in dem eine ObjectGrid-Map "account" ein Banken-Grid ist und Manager1 und Employee1 zwei Benutzer sind. Die eXtreme-Scale-Berechtigungsrichtlinie erteilt "Manager1" alle Zugriffsberechtigungen, "Employee1" aber nur Lesezugriff. Die JAAS-Richtlinie für die ObjectGrid-Map-Berechtigung ist im folgenden Beispiel gezeigt:

```
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"
};
```

Machen Sie sich Gedanken darüber, wie sich das Feature "Zugriff nur durch Ersteller" auf die Berechtigung auswirkt:

- **Inaktiviert:** Wenn das Feature "Zugriff nur durch Ersteller" inaktiviert ist, sind keine Auswirkungen auf die Map-Berechtigung zu verzeichnen. Der Benutzer "Manager1" kann auf alle Daten in der Map "account" zugreifen. Der Benutzer "Employee1" kann alle Daten in der Map lesen, aber keine Daten in der Map aktualisieren, ungültig machen oder entfernen.
- **Ergänzung:** Wenn Sie das Feature "Zugriff nur durch Ersteller" mit der Option "complement" (Ergänzung) aktivieren, sind die Map-Berechtigung und die Berechtigung über das Feature "Zugriff nur durch Ersteller" wirksam. Der Benutzer "Manager1" kann auf die Daten in der Map "account" zugreifen, aber nur dann, wenn ausschließlich er sie in die Map geladen hat. Der Benutzer "Employee1" kann die Daten in der Map "account" lesen, aber nur dann, wenn auschließlich er sie in die Map geladen hat. (Dieser Benutzer kann jedoch keine Daten in der Map aktualisieren, ungültig machen oder entfernen.)
- **Überlagern:** Wenn Sie das Feature "Zugriff nur durch Ersteller" mit der Option "supersede" (Überlagern) aktivieren, wird die Map-Berechtigung nicht umgesetzt. Die Berechtigung über das Feature "Zugriff nur durch Ersteller" ist die einzige Berechtigungsrichtlinie. Der Benutzer "Manager1" hat dieselben Privilegien wie im Modus "Ergänzung". Er kann nur auf die Daten in der Map "account"

zugreifen, wenn er selbst die Daten in die Map geladen hat. Der Benutzer "Employee1" hat jedoch vollständigen Zugriff auf die Daten in der Map "account", wenn er selbst die Daten in die Map geladen hat. Anders ausgedrückt, die in der JAAS-Richtlinie definierte Berechtigungsrichtlinie wird nicht umgesetzt.

Datengridauthentifizierung

Sie können das sichere Token-Manager-Plug-in verwenden, um die Authentifizierung zwischen Servern zu aktivieren. Hierfür müssen Sie die Schnittstelle "SecureTokenManager" implementieren.

Die Methode "generateToken(Object)" verwendet ein Objekt und generiert anschließend ein Token, das von anderen nicht interpretiert werden kann. Die Methode "verifyTokens(byte[])" führt den umgekehrten Prozess aus. Sie konvertiert das Token zurück in das ursprüngliche Objekt.

Eine einfache SecureTokenManager-Implementierung verwendet einen einfachen Verschlüsselungsalgorithmus, wie z. B. einen XOR-Algorithmus (exklusives Oder), um das Objekt in serialisierter Form zu verschlüsseln, und anschließend den entsprechenden Entschlüsselungsalgorithmus, um das Token zu entschlüsseln. Diese Implementierung ist nicht sicher und anfällig für Attacken.

Standardimplementierung von WebSphere eXtreme Scale

WebSphere eXtreme Scale stellt eine sofort verfügbare Implementierung für diese Schnittstelle bereit. Diese Standardimplementierung verwendet ein Schlüsselpaar, um die Signatur zu signieren und zu prüfen, und einen geheimen Schlüssel, um den Inhalt zu verschlüsseln. Jeder Server hat einen JCKES-Keystore, in dem das Schlüsselpaar (privater und öffentlicher Schlüssel) und der geheime Schlüssel gespeichert werden. Der Keystore muss ein JCKES-Keystore sein, damit geheime Schlüssel gespeichert werden können. Diese Schlüssel werden verwendet, um die Shared-Secret-Zeichenfolge auf Senderseite zu verschlüsseln und zu signieren bzw. zu prüfen. Außerdem wird dem Token eine Verfallszeit zugeordnet. Auf Empfängerseite werden die Daten geprüft, entschlüsselt und mit der Shared-Secret-Zeichenfolge des Empfängers verglichen. Es sind keine SSL-Kommunikationsprotokolle (Secure Sockets Layer) zwischen einem Serverpaar für die Authentifizierung erforderlich, weil die privaten und öffentlichen Schlüssel demselben Zweck dienen. Wenn die Serverkommunikation jedoch nicht verschlüsselt ist, können die Daten einfach durch Ansicht der Kommunikation gestohlen werden. Da das Token relativ bald verfällt, ist das Sicherheitsrisiko durch Attacken durch Nachrichtenaufzeichnung und -wiederholung minimal. Das Risiko ist erheblich geringer, wenn alle Server hinter einer Firewall implementiert werden.

Dieser Ansatz hat den Nachteil, dass die Administratoren von WebSphere eXtreme Scale Schlüssel generieren und an alle Server übermitteln müssen, was während des Transports der Schlüssel zu Sicherheitsverletzungen führen kann.

Datengridsicherheit

Die Datengridsicherheitstellt sicher, dass ein in das Datengrid aufgenommener Server die erforderlichen Berechtigungsnachweise besitzt, so dass zerstörerische Server abgewehrt werden. Die Datengridsicherheit verwendet einen Mechanismus mit Shared-Secret-Zeichenfolgen.

Alle Server von WebSphere eXtreme Scale, einschließlich der Katalogserver, einigen sich auf eine Shared-Secret-Zeichenfolge. Wenn ein Server dem Datengrid beitritt,

wird er aufgefordert, diese Shared-Secret-Zeichenfolge vorzulegen. Wenn die Shared-Secret-Zeichenfolge des beitretenden Servers mit der Zeichenfolge im führenden Server oder Katalogserver übereinstimmt, wird der beitretende Server akzeptiert, wenn nicht, wird die Join-Anforderung zurückgewiesen.

Das Senden einer Shared-Secret-Zeichenfolge als Klartext ist nicht sicher. Die Sicherheitsinfrastruktur von WebSphere eXtreme Scale stellt ein Manager-Plug-in für sichere Token bereit, damit der Server dieses Shared Secret vor dem Senden sichern kann. Sie müssen festlegen, wie die Sicherungsoperation implementiert wird. WebSphere eXtreme Scale stellt eine vordefinierte Implementierung bereit, in der die Sicherungsoperation so implementiert ist, dass das Shared Secret verschlüsselt und signiert wird.

Die Shared-Secret-Zeichenfolge wird in der Datei `server.properties` festgelegt. Weitere Informationen zur Eigenschaft "authenticationSecret" finden Sie im Abschnitt Servereigenschaftendatei.

SecureTokenManager-Plug-in

Ein Manager-Plug-in für sichere Token wird über die Schnittstelle "com.ibm.websphere.objectgrid.security.plugins.SecureTokenManager" bereitgestellt.

Weitere Informationen zum SecureTokenManager-Plug-in finden Sie in der Dokumentation zur API 'SecureTokenManager'.

Die Methode "generateToken(Object)" verwendet ein Objekt und generiert anschließend ein Token, das von anderen nicht interpretiert werden kann. Die Methode "verifyTokens(byte[])" führt den umgekehrten Prozess aus. Sie konvertiert das Token zurück in das ursprüngliche Objekt.

Eine einfache SecureTokenManager-Implementierung verwendet einen einfachen Verschlüsselungsalgorithmus, wie z. B. einen XOR-Algorithmus (exklusives Oder), um das Objekt in serialisierter Form zu verschlüsseln, und anschließend den entsprechenden Entschlüsselungsalgorithmus, um das Token zu entschlüsseln. Diese Implementierung ist nicht sicher.

WebSphere eXtreme Scale stellt eine sofort verfügbare Implementierung für diese Schnittstelle bereit.

Die Standardimplementierung verwendet ein Schlüsselpaar, um die Signatur zu signieren und zu prüfen, und einen geheimen Schlüssel, um den Inhalt zu verschlüsseln. Jeder Server hat einen JCKES-Keystore, in dem das Schlüsselpaar (privater und öffentlicher Schlüssel) und der geheime Schlüssel gespeichert werden. Der Keystore muss ein JCKES-Keystore sein, damit geheime Schlüssel gespeichert werden können.

Diese Schlüssel werden verwendet, um die Shared-Secret-Zeichenfolge auf Senderseite zu verschlüsseln und zu signieren bzw. zu prüfen. Außerdem wird dem Token eine Verfallszeit zugeordnet. Auf Empfängerseite werden die Daten geprüft, entschlüsselt und mit der Shared-Secret-Zeichenfolge des Empfängers verglichen. Es sind keine SSL-Kommunikationsprotokolle (Secure Sockets Layer) zwischen einem Serverpaar für die Authentifizierung erforderlich, weil die privaten und öffentlichen Schlüssel demselben Zweck dienen. Wenn die Serverkommunikation jedoch nicht verschlüsselt ist, können die Daten einfach durch Ansicht der Kommunikation gestohlen werden. Da das Token relativ bald verfällt, ist das Si-

cherheitsrisiko durch Attacken durch Nachrichtenaufzeichnung und -wiederholung minimal. Das Risiko ist erheblich geringer, wenn alle Server hinter einer Firewall implementiert werden.

Dieser Ansatz hat den Nachteil, dass die eXtreme-Scale-Administratoren Schlüssel generieren und an alle Server übermitteln müssen, was während des Transports der Schlüssel zu Sicherheitsverletzungen führen kann.

Beispielscripts zum Erstellen der Standardeigenschaften eines sicheren Token-Managers

Wie bereits im vorherigen Abschnitt erwähnt, können Sie einen Keystore erstellen, der ein Schlüsselpaar für die Signatur und Prüfung der Signatur sowie einen geheimen Schlüssel für die Verschlüsselung des Inhalts enthält.

Sie können beispielsweise den Befehl "keytool" von JDK 6 wie folgt verwenden, um die Schlüssel zu erstellen:

```
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
```

Diese beiden Befehle erstellen ein Schlüsselpaar "keypair1" und einen geheimen Schlüssel "seckey1". Anschließend können Sie Folgendes in der Servereigenschaftendatei konfigurieren:

```
secureTokenKeyStore=key1.jck
secureTokenKeyStorePassword=key111
secureTokenKeyStoreType=JCEKS
secureTokenKeyPairAlias=keypair1
secureTokenKeyPairPassword=keypair1
secureTokenSecretKeyAlias=seckey1
secureTokenSecretKeyPassword=seckey1
secureTokenCipherAlgorithm=DES
secureTokenSignAlgorithm=RSA
```

Konfiguration

Weitere Informationen zu den Eigenschaften, mit denen Sie den Manager für sichere Token konfigurieren, finden Sie im Abschnitt Servereigenschaften.

Transport Layer Security und Secure Sockets Layer

WebSphere eXtreme Scale unterstützt TCP/IP und Transport Layer Security/Secure Sockets Layer (TLS/SSL) für die sichere Kommunikation zwischen Clients und Servern.

TLS/SSL in beide Richtungen aktivieren

TLS/SSL wird manchmal unidirektional aktiviert. Das öffentliche Zertifikat des Servers wird beispielsweise in den Client-Truststore importiert, aber das öffentliche Zertifikat des Clients wird nicht in den Server-Truststore importiert. WebSphere eXtreme Scale verwendet jedoch Datengridagenten ausgiebig. Ein Merkmal eines Datengridagenten ist Folgendes: Wenn der Server Antworten an den Client zurücksendet, wird eine neue Verbindung erstellt. Der eXtreme-Scale-Server agiert dann als Client. Deshalb müssen Sie das öffentliche des Clients in den Server-Truststore importieren.

Transportsicherheit für Sun JDK aktivieren

WebSphere eXtreme Scale setzt IBM Java Secure Sockets Extension (IBMJSSE) oder IBM Java Secure Sockets Extension 2 (IBMJSSE2) voraus. Die Provider IBMJSSE und IBMJSSE2 enthalten eine Referenzimplementierung, die die Protokolle SSL und TLS (Transport Layer Security) und ein API-Framework unterstützt.

Im Sun JDK werden die Provider IBM JSSE und IBM JSSE2 nicht mitgeliefert. Deshalb kann die Transportsicherheit mit einem Sun JDK nicht aktiviert werden. Für die Aktivierung der Transportsicherheit ist ein mit WebSphere Application Server geliefertes Sun JDK erforderlich. Das mit WebSphere Application Server gelieferte Sun JDK enthält die Provider IBM JSSE und IBM JSSE2.

Informationen zur Verwendung eines JDK eines anderen Anbieters für WebSphere eXtreme Scale finden Sie unter „Angepassten Object Request Broker konfigurieren" auf Seite 298. Wenn `-Djava.endorsed.dirs` konfiguriert ist, zeigt diese Eigenschaft auf die Verzeichnisse `objectgridRoot/lib/endorsed` und `JRE/lib/endorsed`. Das Verzeichnis `objectgridRoot/lib/endorsed` ist erforderlich, damit der IBM ORB verwendet wird, und das Verzeichnis `JRE/lib/endorsed` ist erforderlich, um die Provider IBM JSSE und IBM JSSE2 zu laden.

In Schritt 4 des Lernprogramms zur Sicherheit in der *Produktübersicht* finden Sie Informationen zur Konfiguration der erforderlichen SSL-Eigenschaften, zum Erstellen von Keystores und Truststores und zum Starten sicherer Server in WebSphere eXtreme Scale.

Sichere Transporttypen konfigurieren

Transport layer security (TLS) unterstützt die sichere Kommunikation zwischen Client und Server. Der verwendete Kommunikationsmechanismus richtet sich nach dem Wert des Parameters **`transportType`**, der in den Konfigurationsdateien von Client und Server angegeben ist.

Informationen zu diesem Vorgang

Wenn Secure Sockets Layer (SSL) verwendet wird, müssen die SSL-Konfigurationsparameter auf Client- und auf Serverseite angegeben werden. In einer Java-SE-Umgebung wird SSL in den Client- und Servereigenschaftendateien konfiguriert. Wenn der Client oder Server in WebSphere Application Server ausgeführt wird, können Sie die vorhandenen CSIv2-Transporteinstellungen von WebSphere Application Server für Ihre Container-Server und Clients verwenden. Weitere Informationen finden Sie unter „Integration der Sicherheit mit WebSphere Application Server" auf Seite 533.

Tabelle 32. Für bestimmte Clienttransport- und Servertransporteinstellungen zu verwendendes Protokoll.

Wenn die transportType-Einstellungen von Client und Server unterschiedlich sind, kann das verwendete Protokoll variieren, oder es kann ein Fehler auftreten.

Eigenschaft "transportType" des Clients	Eigenschaft "transportType" des Servers	Zu verwendendes Protokoll
TCP/IP	TCP/IP	TCP/IP
TCP/IP	SSL-supported	TCP/IP
TCP/IP	SSL-required	Fehler
SSL-supported	TCP/IP	TCP/IP
SSL-supported	SSL-supported	SSL (wenn SSL scheitert, dann TCP/IP)
SSL-supported	SSL-required	SSL
SSL-required	TCP/IP	Fehler

Tabelle 32. Für bestimmte Clienttransport- und Servertransporteinstellungen zu verwendendes Protokoll (Forts.).

Wenn die transportType-Einstellungen von Client und Server unterschiedlich sind, kann das verwendete Protokoll variieren, oder es kann ein Fehler auftreten.

Eigenschaft "transportType" des Clients	Eigenschaft "transportType" des Servers	Zu verwendendes Protokoll
SSL-required	SSL-supported	SSL
SSL-required	SSL-required	SSL

Vorgehensweise

1. Informationen zum Definieren der Eigenschaft **transportType** in der Clientsicherheitskonfiguration finden Sie im Abschnitt Clienteigenschaftendatei.
2. Informationen zum Definieren der Eigenschaft **transportType** in der Sicherheitskonfiguration der Container- und Katalogserver finden Sie unter Servereigenschaftendatei.

SSL-Parameter (Secure Sockets Layer) für Clients oder Server konfigurieren

Wie Sie SSL-Parameter konfigurieren, variiert zwischen Clients und Servern.

Informationen zu diesem Vorgang

TLS/SSL wird manchmal unidirektional aktiviert. Das öffentliche Zertifikat des Servers wird beispielsweise in den Client-Truststore importiert, aber das öffentliche Zertifikat des Clients wird nicht in den Server-Truststore importiert. WebSphere eXtreme Scale verwendet jedoch Datengridagenten ausgiebig. Ein Merkmal eines Datengridagenten ist Folgendes: Wenn der Server Antworten an den Client zurücksendet, wird eine Verbindung erstellt. Der eXtreme-Scale-Server agiert dann als Client. Deshalb müssen Sie das öffentliche des Clients in den Server-Truststore importieren.

Vorgehensweise

- Konfigurieren Sie die SSL-Parameter des Clients.

  Verwenden Sie eine der folgenden Optionen, um SSL-Parameter im Client zu konfigurieren:

  – Erstellen Sie ein Objekt "com.ibm.websphere.objectgrid.security.config.SSLConfiguration" mit der Factoryklasse com.ibm.websphere.objectgrid.security-.config. ClientSecurityConfigurationFactory.

  – Konfigurieren Sie die Parameter in der Datei `client.properties`. Anschließend können Sie die Eigenschaftendatei als JVM-Clienteigenschaft definieren oder die APIs von WebSphere eXtreme Scale verwenden. Übergeben Sie die Eigenschaftendatei an die Methode ClientSecurityConfigurationFactory.getClientSecurityConfiguration(String) für den Client, und verwenden Sie das zurückgegebene Objekt als Parameter an die Methode ObjectGridManager.connect(String, ClientSecurityConfiguration, URL).

- Konfigurieren Sie die SSL-Parameter des Servers.

  SSL-Parameter werden für Server mit der Datei `server.properties` konfiguriert. Zum Starten eines Container- oder Katalogservers mit einer bestimmten Eigenschaftendatei verwenden Sie den Parameter **-serverProps** im Script **startOgServer**. Weitere Informationen zu den SSL-Parametern, die Sie für Server von eXtreme Scale definieren können, finden Sie unter Sicherheitseigenschaften des Servers.

JMX-Sicherheit (Java Management Extensions)

Sie können MBean-Aufrufe (Managed Beans) in einer verteilten Umgebung sichern.

Weitere Informationen zu den verfügbaren MBeans finden Sie im Abschnitt „Verwaltung mit Managed Beans (MBeans)” auf Seite 442.

In der verteilten Implementierungstopologie befinden sich MBeans direkt in den Katalogservern und Container-Servern. Im Allgemeinen folgt die JMX-Sicherheit in einer verteilten Topologie der JMX-Sicherheitsspezifikation, die in der Spezifikation "Java Management Extensions" festgelegt ist. Sie setzt sich aus den folgenden drei Komponenten zusammen:

1. Authentifizierung: Der ferne Client muss im Connector-Server authentifiziert werden.
2. Zugriffssteuerung: Die MBean-Zugriffssteuerung legt fest, wer auf die MBean-Informationen zugreifen und wer die MBean-Operationen durchführen kann.
3. Sicherer Transport: Der Transport zwischen dem JMX-Client und dem Server kann mit TLS/SSL gesichert werden.

Authentifizierung

JMX stellt den Connector-Servern Methoden für die Authentifizierung der fernen Clients zur Verfügung. Für den RMI-Connector wird die Authentifizierung durchgeführt, indem beim Erstellen des Connector-Servers ein Objekt übergeben wird, das die Schnittstelle JMXAuthenticator implementiert. Deshalb implementiert eXtreme Scale die Schnittstelle JMXAuthenticator, um das ObjectGrid-Authenticator-Plug-in zu nutzen, um die fernen Clients zu authentifizieren. In „Lernprogramm zur Java-SE-Sicherheit - Schritt 2” auf Seite 75dem Lernprogramm zur Sicherheit in der *Produktübersicht* wird detailliert beschrieben, wie eXtreme Scale einen Client authentifiziert.

Der JMX-Client folgt den JMX-APIs, um die Berechtigungsnachweise für die Verbindungsherstellung zum Connector-Server bereitzustellen. Das JMX-Framework übergibt die Berechtigungsnachweise an den Connector-Server und ruft dann die JMXAuthenticator-Implementierung für die Authentifizierung auf. Wie zuvor beschrieben, delegiert die JMXAuthenticator-Implementierung die Authentifizierung an die ObjectGrid-Authenticator-Implementierung.

Sehen Sie sich das folgende Beispiel an, das veranschaulicht, wie mit einem Berechtigungsnachweis eine Verbindung zu einem Connector-Server hergestellt wird:

```
javax.management.remote.JMXServiceURL jmxUrl = new JMXServiceURL(
        "service:jmx:rmi:///jndi/rmi://localhost:1099/objectgrid/MBeanServer");

    environment.put(JMXConnector.CREDENTIALS, new UserPasswordCredential("admin", "xxxxxx"));

    // JMXConnectorServer erstellen
    JMXConnector cntor = JMXConnectorFactory.newJMXConnector(jmxUrl, null);

    // Verbindung herstellen und eine Operation im fernen MBeanServer aufrufen
    cntor.connect(environment);
```

Im vorherigen Beispiel wird ein UserPasswordCredential-Objekt mit der Benutzer-ID `admin` und dem Kennwort `xxxxx` bereitgestellt. Dieses UserPasswordCredential-Objekt wird in der Umgebungs-Map gesetzt, die von der Methode JMXConnector-.connect(Map) verwendet wird. Anschließend wird dieses UserPasswordCredential-Objekt über das JMX-Framework zunächst an den Server und schließlich zur Authentifizierung an das ObjectGrid-Authentifizierungs-Framework übergeben.

Das Clientprogrammiermodell folgt strikt der JMX-Spezifikation.

Zugriffssteuerung

Ein JMX-MBean-Server kann Zugriff auf sensible Informationen haben und deshalb in der Lage sein, sensible Operationen durchzuführen. JMX stellt die erforderliche Zugriffssteuerung bereit, die feststellt, welche Clients auf diese Informationen zugreifen und welche Clients diese Operationen durchführen dürfen. Die Zugriffssteuerung wird in das Java-Standardsicherheitsmodell integriert, indem Berechtigungen definiert werden, die den Zugriff auf den MBean-Server und seine Operationen steuern.

Bei der Zugriffssteuerung und -berechtigung für JMX-Operationen stützt sich eXtreme Scale auf die JAAS-Unterstützung, die die JMX-Implementierung bereitstellt. Zu jedem beliebigen Zeitpunkt während der Ausführung eines Programms besitzt ein Ausführungs-Thread einen aktuellen Satz an Berechtigungen. Wenn ein solcher Thread eine Operation der JMX-Spezifikation aufruft, handelt es sich um die so genannten "gehaltenen Berechtigungen". Bei der Durchführung einer JMX-Operation wird eine Sicherheitsprüfung durchgeführt, um festzustellen, ob die "gehaltenen Berechtigungen" die erforderliche Berechtigung abdecken.

Die MBean-Richtliniendefinition folgt dem Java-Richtlinienformat. Die folgende Richtlinie erteilt beispielsweise allen Unterzeichnern und allen Codebasen das Recht, die JMX-Adresse des Servers für die MBean "PlacementServiceMBean" abzurufen, allerdings mit Einschränkung auf die Domäne "com.ibm.websphere.objectgrid":

```
grant {
    permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
     [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

Sie können das folgende Richtlinienbeispiel verwenden, um die Berechtigung auf der Basis der Identität des fernen Clients durchzuführen. Die Richtlinie erteilt dieselben MBean-Berechtigungen wie im vorherigen Beispiel, aber nur den Benutzern mit dem X500Principal-Namen "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";
}
```

Java-Richtlinien werden nur geprüft, wenn der Sicherheitsmanager aktiviert ist. Starten Sie Katalogserver und Container-Server mit dem JVM-Argument "-Djava.security.manager", um die Zugriffssteuerung für MBean-Operationen umzusetzen.

Sicherer Transport

Der Transport zwischen dem JMX-Client und dem Server kann mit TLS/SSL gesichert werden. Wenn das Attribut "transportType" des Katalogserver oder Container-Servers auf "SSL_Required" oder "SSL_Supported" gesetzt ist, müssen Sie SSL verwenden, um die Verbindung zum JMX-Server herzustellen.

Zur Verwendung von SSL müssen Sie den Truststore, den Truststore-Typ und das Truststore-Kennwort im MBean-Client mit Systemeigenschaften konfigurieren, die mit "-D" beginnen:

1. -Djavax.net.ssl.trustStore=TRUST_STORE_LOCATION
2. -Djavax.net.ssl.trustStorePassword=TRUST_STORE_PASSWORD
3. -Djavax.net.ssl.trustStoreType=TRUST_STORE_TYPE

Wenn Sie "com.ibm.websphere.ssl.protocol.SSLSocketFactory" als SSL-Socket-Factory in der Datei *Java-Ausgangsverzeichnis*`/jre/lib/security/java.security` verwenden, definieren Sie die folgenden Eigenschaften:

1. -Dcom.ibm.ssl.trustStore=TRUST_STORE_LOCATION
2. -Dcom.ibm.ssl.trustStorePassword=TRUST_STORE_PASSWORD
3. -Dcom.ibm.ssl.trustStoreType=TRUST_STORE_TYPE

Zum Abrufen dieser Informationen, wenn Transport Layer Security/Secure Sockets Layer (TLS/SSL) aktiviert ist, müssen Sie die Katalog- und Container-Server mit definiertem JMX-Service-Port starten. Für die Definition des JMX-Service-Ports können Sie die Option **`-JMXServicePort`** im Script **`startOgServer`** verwenden oder die Methode setJMXServicePort in der Schnittstelle ServerProperties aufrufen.

Zum Aktivieren des sicheren JMX-Transports für den Container-Server müssen Sie den JMX-Service-Port definieren. Die Definition des JMX-Service-Port ist erforderlich, wenn Sie Transport Layer Security/Secure Sockets Layer (TLS/SSL) verwenden und Container-Server-Informationen aus dem Katalogserver anzeigen. Der Port ist beispielsweise erforderlich, wenn Sie den Befehl **`xscmd -c showMapSizes`** verwenden. Verwenden Sie eine der folgenden Methoden, um den JMX-Service-Port zu definieren:

- Verwenden Sie die Option **`-JMXServicePort`** im Script **`startOgServer`**.
- Wenn Sie einen integrierten Server, rufen Sie die Methode setJMXServicePort in der Schnittstelle ServerProperties auf, um den JMX-Service-Port zu definieren.

Sie müssen für jede JVM in Ihrer Konfiguration eine andere Portnummer verwenden. Wenn Sie JMX/RMI verwenden, geben Sie die Option **`-JMXServicePort`** und die Portnummer explizit an, selbst wenn Sie den Standardportwert verwenden möchten.

Sicherheitsintegration mit externen Providern

Zum Schutz Ihrer Daten kann das Produkt mit verschiedenen Sicherheitsprovidern integriert werden.

WebSphere eXtreme Scale kann mit einer externen Sicherheitsimplementierung integriert werden. Diese externe Implementierung muss Authentifizierungs- und Berechtigungsservices für WebSphere eXtreme Scale bereitstellen. WebSphere eXtreme Scale besitzt Plug-in-Punkte für die Integration einer Sicherheitsimplementierung. WebSphere eXtreme Scale wurde erfolgreich mit den folgenden Komponenten integriert:

- Lightweight Directory Access Protocol (LDAP)
- Kerberos
- ObjectGrid-Sicherheit
- Tivoli Access Manager
- Java Authentication and Authorization Service (JAAS)

eXtreme Scale verwendet den Sicherheitsprovider für die folgenden Tasks:

- Authentifizierung von Clients bei Servern,

- Berechtigung von Clients für den Zugriff auf bestimmte Artefakte von eXtreme Scale oder Festlegung der Verwendung von eXtreme-Scale-Artefakten.

eXtreme Scale hat die folgenden Typen von Berechtigungen:

Map-Berechtigung
Clients oder Gruppen können für die Durchführung von Einfüge-, Lese-, Aktualisierungs- oder Löschoperationen in Maps berechtigt werden.

ObjectGrid-Berechtigung
Clients oder Gruppen können für die Ausführung von Objekt- oder Entitätsabfragen in ObjectGrids berechtigt werden.

DataGrid-Agentenberechtigung
Clients oder Gruppen können für die Implementierung von DataGrid-Agenten in ein ObjectGrid berechtigt werden.

Serverseitige Map-Berechtigung
Clients oder Gruppen können für die Replikation einer Server-Map auf Clientseite oder die Erstellung eins dynamischen Index für die Server-Map berechtigt werden.

Verwaltungsberechtigung
Clients oder Gruppen können für die Ausführung von Verwaltungs-Tasks berechtigt werden.

Anmerkung: Wenn Sie die Sicherheit für Ihr Back-End bereits aktiviert haben, müssen Sie beachten, dass diese Sicherheitseinstellungen für den Schutz Ihrer Daten nicht mehr ausreichen. Die Sicherheitseinstellungen Ihrer Datenbank oder eines anderen Datenspeichers werden in keiner Weise auf Ihren Cache übertragen. Sie müssen die Daten, die jetzt zwischengespeichert werden, mit dem Sicherheitsmechanismus von eXtreme Scale schützen, der Authentifizierung, Berechtigung und Sicherheit auf Transportebene umfasst.

Einschränkung: Verwenden Sie kein Development Kit und keine Runtime Environment der Version 1.6 oder höher, wenn Sie gleichzeitig die SSL-Transportsicherheit für eine eigenständige Konfiguration von WebSphere eXtreme Scale verwenden. Die Anwendungsprogrammierschnittstelleen von WebSphere eXtreme Scale Version 7.1 werden in Version 1.6 und höher nicht unterstüztt. Verwenden Sie Version 1.5 oder früher für Konfigurationen, die die SSL-Transportsicherheit für eigenständige Installationen von eXtreme Scale erfordern. Diese Einschränkung gilt nur, wenn Sie die SSL-Sicherheit in einer eigenständigen Konfiguration von eXtreme Scale verwenden. Version 1.6 und höhere Versionen werden für Konfigurationen ohne SSL-Transport unterstützt.

REST-Datenservice sichern

Sie können diverse Aspekte des REST-Datenservice sichern. Der Zugriff auf den REST-Datenservice von eXtreme Scale kann mit Authentifizierung und Berechtigung gesichert werden. Außerdem kann der Zugriff durch servicebezogene Konfigurationsregel, so genannte Zugriffsregeln, gesteuert werden. Die Transportsicherheit ist der dritte Aspekt.

Informationen zu diesem Vorgang

Der Zugriff auf den REST-Datenservice von eXtreme Scale kann mit Authentifizierung und Berechtigung gesichert werden. Authentifizierung und Berechtigung werden durch die Integration mit der Sicherheit von eXtreme Scale erreicht.

Der Zugriff kann auch über servicebezogene Konfigurationsregeln, so genannte Zugriffsregeln, gesteuert werden. Es gibt zwei Typen von Zugriffsregeln: Serviceoperationsrechte, die die CRUD-Operationen steuern, die der Service zulässt, und Entitätszugriffsrechte, die die CRUD-Operationen steuern, die für einen bestimmten Entitätstyp zulässig sind.

Die Transportsicherheit wird durch das Hosten der Containerkonfiguration für Verbindungen zwischen dem Web-Client und dem REST-Service erreicht. Außerdem wird die Transportsicherheit von der eXtreme-Scale-Clientkonfiguration (für Verbindungen zwischen dem REST-Service und dem eXtreme-Scale-Datengrid) bereitgestellt.

Vorgehensweise

- Steuerung der Authentifizierung und der Berechtigung.

  Der Zugriff auf den REST-Datenservice von eXtreme Scale kann mit Authentifizierung und Berechtigung gesichert werden. Authentifizierung und Berechtigung werden durch die Integration mit der Sicherheit von eXtreme Scale erreicht.

  Der REST-Datenservice von eXtreme Scale verwendet die Sicherheit von eXtreme Scale für die Authentifizierung und die Berechtigung, um zu steuern, welche Benutzer auf den Service zugreifen, und welche Operationen ein Benutzer über den Service ausführen darf. Der REST-Datenservice von eXtreme Scale verwendet entweder einen konfigurierten globalen Berechtigungsnachweis mit Benutzer und Kennwort oder einen Berechtigungsnachweis, der aus einer HTTP-Basisanforderung abgeleitet wird, die mit jeder Transaktion an das eXtreme-Scale-Grid gesendet wird, in dem die Authentifizierung und Berechtigung durchgeführt werden.

  1. Konfigurieren Sie die eXtreme-Scale-Clientauthentifizierung und -berechtigung im Grid. Einzelheiten zum Konfigurieren der eXtreme-Scale-Clientauthentifizierung und -berechtigung finden Sie in „Sicherheitsintegration mit externen Providern” auf Seite 528.
  2. Konfigurieren Sie den vom REST-Service verwendeten eXtreme-Scale-Client für die Sicherheit.

     Der REST-Datenservice von eXtreme Scale ruft die eXtreme-Scale-Clientbibliothek auf, wenn er mit dem eXtreme-Scale-Grid kommuniziert. Deshalb muss der eXtreme-Scale-Client für die Sicherheit von eXtreme Scale konfiguriert werden.

     Die eXtreme-Scale-Clientauthentifizierung wird über Eigenschaften in der ObjectGrid-Clienteigenschaftendatei aktiviert. Wenn die Clientsicherheit mit den REST-Service verwendet wird, müssen mindestens die folgenden Attribute aktiviert werden:

     ```
     securityEnabled=true
     credentialAuthentication=Supported [-oder-] Required
     credentialGeneratorProps=user:pass [-oder-] {xor encoded user:pass}
     ```

     Hinweis: Die mit der Eigenschaft "credentialGeneratorProps" angegebene Benutzer/Kennwort-Kombination muss einer ID im Authentifizierungsregister entsprechen und genügend ObjectGrid-Richtlinienrechte haben, um eine Verbindung zu ObjectGrids herzustellen und ObjectGrids zu erstellen.

     Eine ObjectGrid-Beispielclientrichtliniendatei finden Sie unter *Ausgangsverzeichnis_des_REST-Service*`/security/security.ogclient.properties`. Lesen Sie auch den Abschnitt Clienteigenschaftendatei.
  3. Konfigurieren Sie den REST-Datenservice von eXtreme Scale für die Sicherheit.

Die Konfigurationseigenschaftendatei des REST-Datenservice von eXtreme Scale muss für die Integration mit der Sicherheit von eXtreme Scale die folgenden Einträge enthalten:

```
ogClientPropertyFile=Dateiname
```

ogClientPropertyFile gibt die Position der Eigenschaftendatei an, die die im vorherigen Schritt genannten ObjectGrid-Clienteigenschaften enthält. Der REST-Service verwendet diese Datei, um den eXtreme-Scale-Client für die Kommunikation mit dem Grid zu initialisieren, wenn die Sicherheit aktiviert ist.

```
loginType=basic [-oder-] none
```

Die Eigenschaft "loginType" konfiguriert den REST-Service für den Anmeldetyp. Wenn Sie den Wert `none` angeben, wird die mit "credentialGeneratorProps" definierte globale Benutzer-ID/Kennwort-Kombination für jede Transaktion an das Grid gesendet. Wenn Sie den Wert `basic` angeben, sendet der REST-Service eine HTTP-Basisanforderung an den Client und fordert diesen zur Bereitstellung von Berechtigungsnachweisen auf, die bei der Kommunikation mit dem Grid in jeder Transaktion gesendet werden.

Weitere Informationen zu den Eigenschaften "ogClientPropertyFile" und "loginType" finden Sie unter Eigenschaftendatei des REST-Datenservice.

- Wenden Sie Zugriffsregeln an.

  Der Zugriff kann auch über servicebezogene Konfigurationsregeln, so genannte Zugriffsregeln, gesteuert werden. Es gibt zwei Typen von Zugriffsregeln: Serviceoperationsrechte, die die CRUD-Operationen steuern, die der Service zulässt, und Entitätszugriffsrechte, die die CRUD-Operationen steuern, die für einen bestimmten Entitätstyp zulässig sind.

  Der REST-Datenservice von eXtreme Scale lässt optional Zugriffsregeln zu, die konfiguriert werden können, um den Zugriff auf den Service und die Entitäten im Service zu beschränken. Diese Zugriffsregeln werden in der Eigenschaftendatei mit den Zugriffsrechten für den REST-Datenservice angegeben. Der Name dieser Datei wird in der Eigenschaftendatei des REST-Datenservice mit der Eigenschaft "wxsRestAccessRightsFile" angegeben. Weitere Informationen zu dieser Eigenschaft finden Sie unter Eigenschaftendatei des REST-Datenservice. Diese Datei ist eine typische Java-Eigenschaftendatei mit Schlüssel/Wert-Paaren. Es gibt zwei Typen von Zugriffsregeln: Serviceoperationsrechte, die die CRUD-Operationen steuern, die der Service zulässt, und Entitätszugriffsrechte, die die CRUD-Operationen steuern, die für einen bestimmten Entitätstyp zulässig sind.

  1. Konfigurieren Sie die Rechte für die Serviceoperationen.

     Die Rechte für Serviceoperationen legen die Zugriffsrechte fest, die für alle ObjectGrids gelten, die über den REST-Service bereitgestellt werden, bzw. für alle Entitäten eines angegebenen ObjectGrids.

     Verwenden Sie die folgende Syntax.

     ```
     serviceOperationRights=Recht_für_Serviceoperation
     serviceOperationRights.Gridname -ODER- *=Recht_für_Serviceoperation
     ```

     Für diese Angaben gilt Folgendes:

     – Die gültigen Werte für "serviceOperationRights2 sind NONE, READSINGLE, READMULTIPLE, ALLREAD und ALL.

     – serviceOperationRights.*Gridname -ODER-* * impliziert, dass das Zugriffsrecht für alle ObjectGrids gilt, sofern nicht der Name eines bestimmten ObjectGrids angegeben wird.

     Beispiel:

```
serviceOperationsRights=ALL
serviceOperationsRights.*=NONE
serviceOperationsRights.EMPLOYEEGRID=READSINGLE
```

Das erste Beispiel gibt an, dass alle Serviceoperationen für alle ObjectGrids ausgeführt werden können, die von diesem REST-Service bereitgestellt werden. Das zweite Beispiel gleicht dem ersten Beispiel insofern, dass es für alle vom REST-Service bereitgestellten ObjectGrids gilt, definiert aber das Zugriffsrecht mit NONE, d. h., dass keine Serviceoperation für die ObjectGrids zulässig ist. Das letzte Beispiel gibt an, wie die Serviceoperationen für ein bestimmtes Grid gesteuert werden. In diesem Fall werden nur "Reads" (Leseoperationen), die einen einzigen Datensatz ergeben, für alle Entitäten des EMPLOYEEGRID zugelassen.

Standardmäßig geht der REST-Service von `serviceOperationsRights=ALL` aus, was bedeutet, dass alle Operationen für alle von diesem Service bereitgestellten ObjectGrids zugelassen werden. Dies unterscheidet sich von der Microsoft-Implementierung, die die Standardeinstellung `NONE` hat, so dass keine Operationen sind im REST-Service zugelassen werden.

Wichtig: Die Rechte für Serviceoperationen werden in der Reihenfolge ausgewertet, in der sie in dieser Datei angegeben sind, so dass das zuletzt angegebene Recht die vorherigen überschreibt.

2. Konfigurieren Sie die Zugriffsrechte für Entitäten.

   Entitätszugriffsrechte legen die Zugriffsrechte fest, die für bestimmte ObjectGrid-Entitäten gelten, die über den REST-Service bereitgestellt werden. Diese Rechte sind eine Methode für die Einrichtung einer strengeren und differenzierteren Zugriffssteuerung für einzelne ObjectGrid-Entitäten im Vergleich zu Rechten für Serviceoperationen.

   Verwenden Sie die folgende Syntax.

   `entitySetRights.`*`Grid.Name.Entitätsname=Recht_für_Entität`*

   Für diese Angaben gilt Folgendes:

   – Die gültigen Werte für *Recht_für_Entität* sind im Folgenden aufgeführt.

Tabelle 33. Entitätszugriffsrechte. Unterstützte Werte

Zugriffsrecht	Beschreibung
NONE	Weist alle Rechte für den Zugriff auf Daten zurück.
READSINGLE	Lässt das Lesen einzelner Datenelemente zu.
READMULTIPLE	Lässt das Lesen von Datengruppen zu.
ALLREAD	Lässt das Lesen einzelner oder mehrerer Datengruppen zu.
WRITEAPPEND	Lässt das Erstellen neuer Datenelemente in Datensätzen zu.
WRITEREPLACE	Lässt das Ersetzen von Daten zu.
WRITEDELETE	Lässt das Löschen von Datenelementen aus Datensätzen zu.
WRITEMERGE	Lässt das Zusammenführen von Daten zu.
ALLWRITE	Lässt das Schreiben von Daten zu (d. h. erstellen, ersetzen, zusammenfügen oder löschen).
ALL	Lässt das Erstellen, Lesen, Aktualisieren und Löschen von Daten zu.

   – *Gridname* steht für den Namen eines bestimmten ObjectGrids im REST-Service.

– *Entitätsname* steht für den Namen einer bestimmten Entität im angegebenen ObjectGrid.

Anmerkung: Wenn Rechte für Serviceoperationen und Entitätzugriffsrechte für ein ObjectGrid und dessen Entitäten festgelegt werden, werden die restriktiveren dieser Rechte umgesetzt, wie in den folgenden Beispielen veranschaulicht wird. Beachten Sie auch, dass die Entitätszugriffsrechte in der Reihenfolge ausgewertet werden, in der sie in der Datei angegeben sind. Das zuletzt angegebene Recht überschreibt die vorherigen Rechte.

Beispiel 1: Wenn "serviceOperationsRights.NorthwindGrid=READSINGLE" und "entitySetRights.NorthwindGrid.Customer=ALL" angegeben werden, wird READSINGLE für die Entität "Customer" umgesetzt.

Beispiel 2: Wenn "serviceOperationsRights.NorthwindGrid=ALLREAD" und "entitySetRights.NorthwindGrid.Customer=ALLWRITE" angegeben werden, werden nur Reads (Leseoperationen) für alle Entitäten von NorthwindGrid zugelassen. Für "Customer" verhindern die zugehörigen Entitätszugriffsrechte jedoch alle Leseoperationen (da ALLWRITE angegeben wurde), und somit gilt für die Entität "Customer" das Zugriffsrecht NONE.

- Sichern Sie die Transporte.

  Die Transportsicherheit wird durch das Hosten der Containerkonfiguration für Verbindungen zwischen dem Web-Client und dem REST-Service erreicht. Die Transportsicherheit wird für Verbindungen zwischen dem REST-Service und dem eXtreme-Scale-Grid über die eXtreme-Scale-Clientkonfiguration bereitgestellt.

  1. Sichern Sie die Verbindung von Client und REST-Service. Die Transportsicherheit für diese Verbindung wird von der Hostcontainerumgebung bereitgestellt und nicht in eXtreme Scale.
  2. Sichern Sie die Verbindung des REST-Service und des eXtreme-Scale-Grids. Die Transportsicherheit für diese Verbindung wird in eXtreme Scale konfiguriert. Weitere Informationen finden Sie unter „Transport Layer Security und Secure Sockets Layer” auf Seite 523.

Integration der Sicherheit mit WebSphere Application Server

Wenn WebSphere eXtreme Scale in einer Umgebung von WebSphere Application Server implementiert ist, können Sie die Konfiguration des Authentifizierungsablaufs und der Transportsicherheit über WebSphere Application Server vereinfachen.

Vereinfachter Authentifizierungsablauf

Wenn eXtreme-Scale-Clients und -Server in WebSphere Application Server und in derselben Sicherheitsdomäne ausgeführt werden, können Sie die Sicherheitsinfrastruktur von WebSphere Application Server verwenden, um die Berechtigungsnachweise für die Clientauthentifizierung an den eXtreme-Scale-Server weiterzugeben. Versucht ein Servlet beispielsweise als eXtreme-Scale-Client eine Verbindung zu einem eXtreme-Scale-Server in derselben Sicherheitsdomäne herzustellen und ist das Servlet bereits authentifiziert, kann das Authentifizierungstoken von Client (Servlet) an den Server weitergegeben und anschließend die Sicherheitsinfrastruktur von WebSphere Application Server verwendet werden, um das Authentifizierungstoken an die Clientberechtigungsnachweise zurückzugeben.

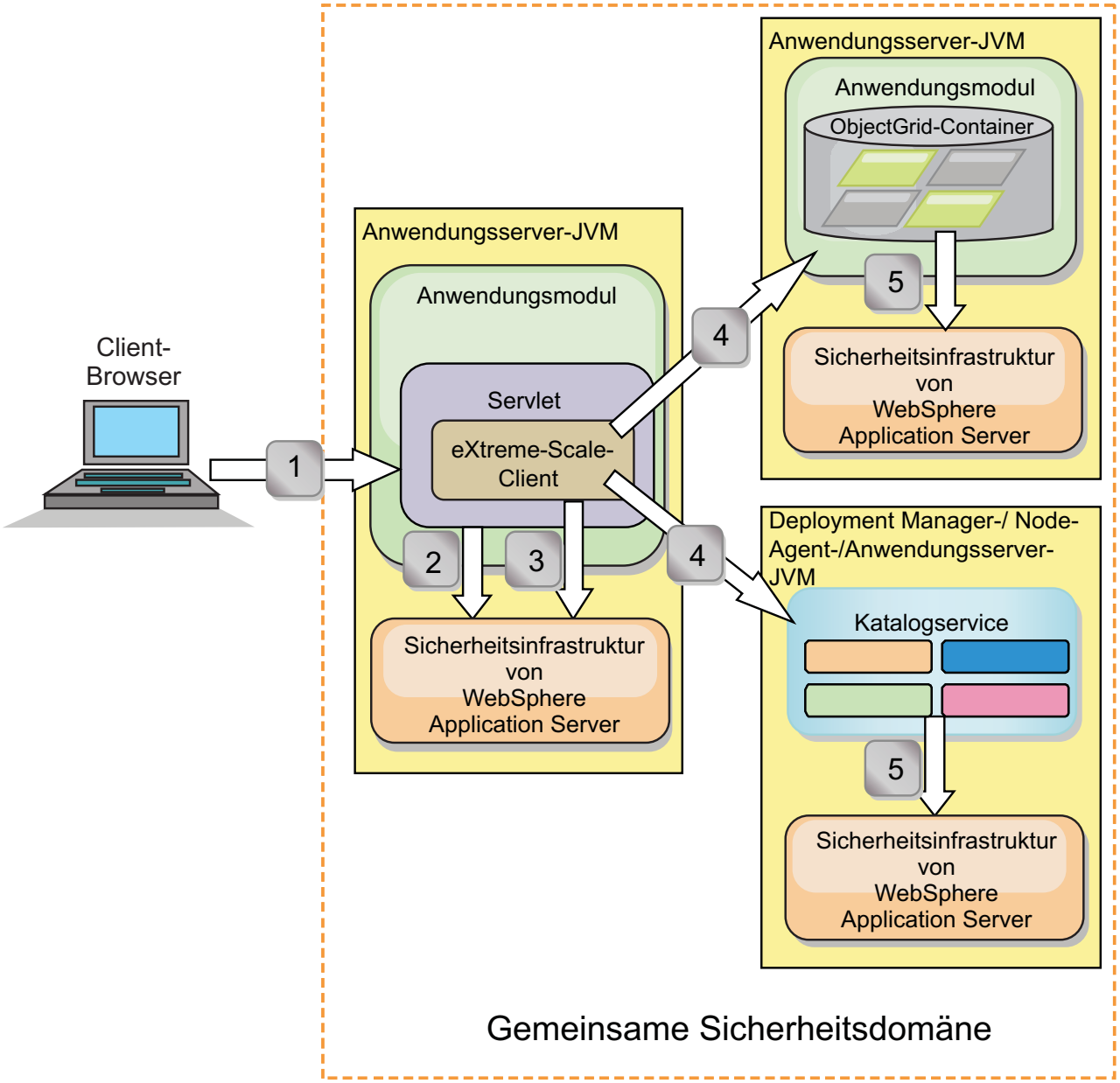


Abbildung 67. Authentifizierungsablauf für Server in derselben Sicherheitsdomäne

In der vorherigen Abbildung befinden sich die Anwendungsserver in derselben Sicherheitsdomäne. Ein Anwendungsserver hostet die Webanwendung, die auch ein eXtreme-Scale-Client ist. Der andere Anwendungsserver hostet den Container-Server. Die JVM des Deployment Manager bzw. Node Agent hostet den Katalogservice. Die Pfeile in der Abbildung zeigen den Ablauf des Authentifizierungsprozesses an:

1. Ein Unternehmensanwendungsbenutzer verwendet einen Web-Browser, um sich beim ersten Anwendungsserver mit einem Benutzernamen und einem Kennwort anzumelden.
2. Der erste Anwendungsserver sendet den Clientbenutzernamen und das Kennwort an die Sicherheitsinfrastruktur von WebSphere Application Server, um sich bei der Benutzerregistry zu authentifizieren. Diese Benutzerregistry kann beispielsweise ein LDAP-Server sein. Deshalb werden die Sicherheitsinformationen im Thread des Anwendungsservers gespeichert.

3. Die JSP-Datei (JavaServer Pages) dient als eXtreme-Scale-Client, um die Sicherheitsinformationen vom Server-Thread abzurufen. Die JSP-Datei ruft die Sicherheitsinfrastruktur von WebSphere Application Server auf, um die Sicherheitstoken abzurufen, die den Unternehmensanwendungsbenutzer darstellen.
4. Der eXtreme-Scale-Client bzw. die JSP-Datei sendet die Sicherheitstoken mit der Anforderung an den Container-Server und den Katalogservice, der in den anderen JVMs gehostet wird. Der Katalogserver und der Container-Server verwenden die Sicherheitstoken von WebSphere Application Server als eXtreme-Scale-Clientberechtigungsnachweis.
5. Die Katalog- und Container-Server senden die Sicherheitstoken an die Sicherheitsinfrastruktur von WebSphere Application Server, um die Sicherheitstoken in Benutzersicherheitsinformationen zu konvertieren. Diese Benutzersicherheitsinformationen werden durch ein Objekt Subject dargestellt, das die Principals, öffentlichen Berechtigungsnachweise und privaten Berechtigungsnachweise enthält. Diese Konvertierung kann stattfinden, weil die Anwendungsserver, die den eXtreme-Scale-Client, Katalogserver und Container-Server hosten, dieselben LDAP-Token (WebSphere Application Server Lightweight Third-Party Authentication) nutzen.

Integration der Authentifizierung

Verteilte Sicherheitsintegration mit WebSphere Application Server:

Verwenden Sie für das verteilte Modell die folgenden Klassen:

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenAuthenticator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredential

Beispiele für die Verwendung dieser Klassen finden Sie unter „Lernprogramm: Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server integrieren” auf Seite 88.

Verwenden Sie serverseitig den WSTokenAuthentication-Authentifikator, um das WSTokenCredential-Objekt zu authentifizieren.

Lokale Sicherheitsintegration mit WebSphere Application Server:

Verwenden Sie für das lokale ObjectGrid-Modell die folgenden Klassen:

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectSourceImpl
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectValidationImpl

Weitere Informationen zu diesen Klassen finden Sie unter Lokale Programmierung der Sicherheit. Sie können die Klasse "WSSubjectSourceImpl" als SubjectSource-Plug-in und die Klasse "WSSubjectValidationImpl" als SubjectValidation-Plug-in konfigurieren.

Unterstützung der Transportsicherheit in WebSphere Application Server

Wenn ein eXtreme-Scale-Client, -Container-Server oder -Katalogserver in einem Prozess von WebSphere Application Server ausgeführt wird, wird die Transportsicherheit von eXtreme Scale mit den CSIv2-Transporteinstellungen von WebSphere Application Server verwaltet. Für den eXtreme-Scale-Client oder -Container-Server

sollten Sie die SSL-Einstellungen nicht über die Eigenschaften des eXtreme-Scale-Clients oder -Servers konfigurieren. Alle SSL-Einstellungen müssen in der Konfiguration von WebSphere Application Server angegeben werden.

Der Katalogserver ist jedoch ein wenig anders. Der Katalogserver hat eigene proprietäre Transportpfade, die nicht mit den CSIv2-Transporteinstellungen von WebSphere Application Server verwaltet werden können. Deshalb müssen die SSL-Eigenschaften weiterhin in der Servereigenschaftendatei für den Katalogserver konfiguriert werden. Weitere Informationen finden Sie unter „Lernprogramm: Sicherheit von WebSphere eXtreme Scale mit WebSphere Application Server integrieren” auf Seite 88.

Clientsicherheit in einer Katalogservicedomäne konfigurieren

Wenn Sie die Clientsicherheit in einer Katalogservicedomäne konfigurieren, können Sie Standardkonfigurationseigenschaften für die Clientauthentifizierung definieren. Diese Eigenschaften werden verwendet, wenn keine Clienteigenschaftendatei in der Java Virtual Machine (JVM) gefunden wird, in der der Client ausgeführt wird, oder wenn der Client keine Sicherheitseigenschaften über das Programm angibt. Wenn eine Clienteigenschaftendatei vorhanden ist, überschreiben die Eigenschaften, die Sie in der Konsole angeben, die Werte in der Datei. Sie können diese Eigenschaften überschreiben, indem Sie eine Datei `splicer.properties` mit der angepassten Eigenschaft "com.ibm.websphere.xs.sessionFilterProps" angeben oder die EAR-Datei der Anwendung verbinden.

Vorbereitende Schritte

- Sie müssen die CredentialGenerator-Implementierung kennen, die Sie verwenden, um Clients beim fernen Datengrid zu authentifizieren. Sie können eine der Implementierungen verwenden, die von WebSphere eXtreme Scale: UserPasswordCredentialGenerator oder WSTokenCredentialGenerator bereitgestellt werden.

  Sie können auch eine angepasste Implementierung der Schnittstelle CredentialGenerator verwenden. Die angepasste Implementierung muss im Klassenpfad des Laufzeitclients und des Servers enthalten sein. Wenn Sie ein HTTP-Sitzungsszenario mit WebSphere Application Server konfigurieren, müssen Sie die Implementierung in den Klassenpfad des Deployment Manager und in den Klassenpfad des Anwendungsservers, in dem der Client ausgeführt wird, gespeichert werden.
- Sie müssen eine definierte Katalogservicedomäne haben. Weitere Informationen finden Sie unter „Katalogservicedomänen in WebSphere Application Server erstellen” auf Seite 264.

Informationen zu diesem Vorgang

Sie müssen die Clientsicherheit in der Katalogservicedomäne konfigurieren, wenn Sie die Authentifizierung mit Berechtigungsnachweisen auf der Serverseite aktiviert haben, indem Sie eines der folgenden Szenarien konfigurieren:

- In der serverseitigen Sicherheitsrichtlinie ist die Eigenschaft **`credentialAuthentication`** auf `Required` gesetzt.
- In der serverseitigen Sicherheitsrichtlinie ist die Eigenschaft **`credentialAuthentication`** auf `Supported` gesetzt und **`authorizationMechanism`** wurde in der ObjectGrid-XML-Datei angegeben.

In diesen Szenarien muss ein Berechtigungsnachweis vom Client übergeben werden. Der vom Client übergebene Berechtigungsnachweis wird von der Methode

getCredential in einer Klasse abgerufen, die die Schnittstelle CredentialGenerator implementiert. In einem Szenario mit einer HTTP-Sitzungskonfiguration muss die Laufzeitumgebung die CredentialGenerator-Implementierung kennen, die für die Generierung eines Berechtigungsnachweises zu verwenden ist, der an ein fernes Datengrid übergeben wird. Wenn Sie die zu verwendende CredentialGenerator-Implementierungsklasse nicht angeben, weist das ferne Datengrid Anforderungen vom Client zurück, weil der Client nicht authentifiziert werden kann.

Vorgehensweise

Definieren Sie die Clientsicherheitseigenschaften. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Systemverwaltung** > **WebSphere eXtreme Scale** > **Katalogservicedomänen** > *Name_der_Katalogservicedomäne* > **Clientsicherheitseigenschaften**. Geben Sie die Clientsicherheitseigenschaften auf der Seite an, und speichern Sie Ihre Änderungen. Eine Liste der Eigenschaften, die Sie definieren können, finden Sie unter „Clientsicherheitseigenschaften" auf Seite 280.

Ergebnisse

Die Clientsicherheitseigenschaften, die Sie in der Katalogservicedomäne konfiguriert haben, werden als Standardwerte verwendet. Die Werte, die Sie angeben, überschreiben alle Eigenschaften, die in den Dateien `client.properties` definiert sind.

Nächste Schritte

Konfigurieren Sie Ihre Anwendungen für die Verwendung von WebSphere eXtreme Scale für die Sitzungsverwaltung. Weitere Informationen finden Sie unter „Anwendungen für die HTTP-Sitzungsverwaltung in WebSphere Application Server" auf Seite 312.

Lokale Sicherheit aktivieren

WebSphere eXtreme Scale stellt mehrere Sicherheitsendpunkte für die Integration angepasster Mechanismen bereit. Im lokalen Programmiermodell ist die Hauptsicherheitsfunktion Berechtigung. Authentifizierung wird nicht unterstützt. Sie müssen die Authentifizierung unabhängig von der bereits vorhandenen Authentifizierung in WebSphere Application Server durchführen. Sie können die bereitgestellten Plug-ins jedoch verwenden, um Subject-Objekte abzurufen und zu validieren.

Informationen zu diesem Vorgang

Sie können die lokale Sicherheit mit der ObjectGrid-XML-Deskriptordatei oder über das Programm aktivieren.

Vorgehensweise

Lokale Sicherheit mit der ObjectGridXML-Deskriptordatei aktivieren.
Die Datei `secure-objectgrid-definition.xml`, die in der Beispielunternehmensanwendung "ObjectGridSample" verwendet wird, wird im folgenden Beispiel gezeigt. Setzen Sie das Attribut "securityEnabled" auf `true`, um die Sicherheit zu aktivieren.

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
        authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS">
    ...
</objectGrids>
```

Nächste Schritte

Starten Sie die Container- und Katalogserver mit aktivierter Sicherheit.

Sichere Server starten und stoppen

Die Sicherheit wird aktiviert, indem Sie sicherheitsspezifische Konfigurationen angeben, wenn Sie Server starten und stoppen.

Sichere Server in einer eigenständigen Umgebung starten

Zum Starten sicherer eigenständiger Server übergeben Sie die richtigen Konfigurationsdateien, indem Sie Parameter mit dem Befehl **startOgServer** angeben.

Vorbereitende Schritte

Wenn Sie einen externen Clientsicherheitsprovider für Authentifizierung und Berechtigung verwenden, definieren Sie die Umgebungsvariable *CLIENT_AUTH_LIB*. Öffnen Sie eine Befehlszeile oder ein Terminalfenster, und führen Sie den Befehl aus, der für Ihr Betriebssystem geeignet ist:

- Windows `set CLIENT_AUTH_LIB=<Pfad_zu_Sicherheits-JARs_oder_Klassen>`
- UNIX `set CLIENT_AUTH_LIB=<Pfad_zu_Sicherheits-JARs_oder_Klassen>`
  `export CLIENT_AUTH_LIB`

Wenn die Befehle **startOgServer** und **stopOgServer** ausgeführt werden, wird diese Variable an den Klassenpfad angehängt.

Vorgehensweise

- Sichere Container-Server starten.

  Für das Starten eines sicheren Container-Servers ist die folgende Sicherheitskonfigurationsdatei erforderlich:

  – **Servereigenschaftendatei:** Die Servereigenschaftendatei konfiguriert die für den Server spezifischen Sicherheitseigenschaften. Weitere Einzelheiten finden Sie im Abschnitt Servereigenschaftendatei.

  Geben Sie die Position dieser Konfigurationsdatei an, indem Sie das folgende Argument mit dem Script **startOgServer** angeben:

  -serverProps
  Gibt die Position der Servereigenschaftendatei an, die die serverspezifischen Sicherheitseigenschaften enthält. Der Dateiname für diese Eigenschaft kann im herkömmlichen Dateipfadformat angegeben werden, z. B. `../security/server.properties`.

- Sichere Katalogserver starten.

  Zum Starten eines sicheren Katalogservice müssen die folgenden Konfigurationsdateien verfügbar sein:

  – **XML-Sicherheitsdeskriptordatei:** Die XML-Sicherheitsdeskriptordatei beschreibt die Sicherheitseigenschaften, die für alle Server, einschließlich Katalogservern und Container-Servern, gelten. Ein Beispiel für eine solche Eigenschaft ist die Authentifikatorkonfiguration, die die Benutzerregistry und das Authentifizierungsverfahren darstellt.

  – **Servereigenschaftendatei:** Die Servereigenschaftendatei konfiguriert die für den Server spezifischen Sicherheitseigenschaften.

  Geben Sie die Position dieser Konfigurationsdateien an, indem Sie die folgenden Argumente mit dem Script **startOgServer** angeben:

-clusterSecurityFile und -clusterSecurityUrl
Diese Argumente geben die Position der XML-Sicherheitsdeskriptordatei an. Verwenden Sie den Parameter **-clusterSecurityFile**, um eine lokale Datei anzugeben, bzw. den Parameter **-clusterSecurityUrl**, um den URL der Datei `objectGridSecurity.xml` anzugeben.

-serverProps
Gibt die Position der Servereigenschaftendatei an, die die serverspezifischen Sicherheitseigenschaften enthält. Der Dateiname für diese Eigenschaft kann im herkömmlichen Dateipfadformat angegeben werden, z. B. `c:/tmp/og/catalogserver.props`.

Sichere Server in WebSphere Application Server starten

Zum Starten sicherer Server in WebSphere Application Server müssen Sie die Sicherheitskonfigurationsdateien in den generischen JVM-Argumenten (Java Virtual Machine) angeben.

Vorgehensweise

- Sicheren Katalogservice in WebSphere Application Server starten.

  Ein Katalogserver enthält zwei verschiedene Stufen von Sicherheitsinformationen:

  – `-Dobjectgrid.cluster.security.xml.url`: Gibt die Position der Datei `objectGridSecurity.xml` an, in der die allgemeinen Sicherheitseigenschaften für alle Server, einschließlich Katalogservern und Container-Servern, beschrieben werden. Ein Beispiel für die definierten Sicherheitseigenschaften ist die Authentifikatorkonfiguration, die die Benutzerregistry und den Authentifizierungsmechanismus darstellt. Der für diese Eigenschaft angegebene Dateiname muss ein URL-Format haben, z. B. `file:///tmp/og/objectGridSecurity.xml`.

  – `-Dobjectgrid.server.props`: Gibt die Servereigenschaftendatei an, die die serverspezifischen Sicherheitseigenschaften enthält. Der Dateiname für diese Eigenschaft kann im herkömmlichen Dateipfadformat angegeben werden, z. B. `c:/tmp/og/catalogserver.props`.

  1. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Systemverwaltung**. Klicken Sie auf den Prozess, in dem der Katalogserver implementiert ist, z. B. auf den Deployment Manager.
  2. Klicken Sie auf **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
  3. Geben Sie die Eigenschaften im Feld **Generische JVM-Argumente** ein. Im Folgenden sehen Sie ein Beispiel für die Werte, die Sie hinzufügen können:

     ```
     -Dobjectgrid.cluster.security.xml.url=file:///tmp/og/objectGridSecurity.xml
     -Dobjectgrid.server.props=/tmp/og/catalog.server.props
     ```

  4. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

- Sicheren Container-Server in WebSphere Application Server starten.

  Ein Container-Server übernimmt beim Herstellen einer Verbindung zum Katalogserver die Sicherheitskonfiguration, die in der Datei `objectGridSecurity.xml` enthalten ist, z. B. die Authentifikatorkonfiguration oder die Zeitlimiteinstellungen für Anmeldesitzungen. Außerdem müssen Sie serverspezifische Sicherheitseigenschaften für bestimmte Container-Server in der Eigenschaft `-Dobjectgrid.server.props` definieren.

  Der Dateiname für diese Eigenschaft kann im herkömmlichen Dateipfadformat angegeben werden, z. B. `c:/tmp/og/server.props`.

  Führen Sie dieselben Schritte wie zuvor aus, um die Sicherheitseigenschaft den generischen JVM-Argumenten hinzuzufügen.

1. Öffnen Sie sie Seite "Java Virtual Machine" für den Server. Klicken Sie in der Administrationskonsole von WebSphere Application Server auf **Server** > **Anwendungsserver** > *Servername* > **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine**.
2. Geben Sie die Eigenschaft im Feld **Generische JVM-Argumente** ein. Im Folgenden sehen Sie ein Beispiel für die Werte, die Sie hinzufügen können:
   ```
   -Dobjectgrid.server.props=/opt/wxs/security/server2.props
   ```
3. Klicken Sie auf **OK**, und speichern Sie Ihre Änderungen.

Sichere Server stoppen

Das Stoppen sicherer Katalogserver oder Container-Server erfordert eine einzeige Sicherheitskonfigurationsdatei.

Vorgehensweise

Sicheren Katalogserver oder Container-Server stoppen.
Das Stoppen eines sicheren Servers erfordert die folgende Sicherheitskonfigurationsdatei:

- **Clienteigenschaftendatei:** Die Clienteigenschaftendatei kann zum Konfigurieren der Clientsicherheitseigenschaften verwendet werden. Die Clientsicherheitseigenschaften sind erforderlich, damit ein Client eine Verbindung zu einem sicheren Server herstellen kann. Weitere Einzelheiten finden Sie im Abschnitt Clienteigenschaftendatei.

Geben Sie die Position dieser Konfigurationsddateien an, indem Sie das folgende Argument mit dem Script **stopOgServer** angeben:

-clientSecurityFile
Gibt den Pfad zu der Clienteigenschaftendatei an, die die Sicherheitseigenschaften für den Client definiert. Der Dateiname, den Sie für diese Eigenschaft angeben, muss im Dateipfadformat angegeben werden, z. B. `../security/objectGridClient.properties`.

Beispiel

```
stopOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602-clientSecurityFile ../security/objectGridClient.properties
```

Sicherheitsprofile für das Dienstprogramm `xscmd` konfigurieren

Wenn Sie ein Sicherheitsprofil erstellen, können Sie gespeicherte Sicherheitsparameter verwenden, um das Dienstprogramm **xscmd** mit sicheren Umgebungen zu verwenden.

Vorbereitende Schritte

Weitere Informationen zum Konfigurieren des Dienstprogramms **xscmd** finden Sie unter „Verwaltung mit dem Dienstprogramm **xscmd**" auf Seite 423.

Informationen zu diesem Vorgang

Sie können den Parameter **-ssp** *Profilname* oder **--saveSecProfile** *Profilname* mit dem Rest des Befehls **xscmd** angeben, um ein Sicherheitsprofil zu speichern. Das Profil kann Einstellungen für Benutzernamen und Kennwörter, Berechtigungsnachweisgeneratoren, Keystores, Truststores und Transporttypen enthalten.

Die Befehlsgruppe **ProfileManagement** im Dienstprogramm **xscmd** enthält Befehle für die Verwaltung Ihrer Sicherheitsprofile.

Vorgehensweise

- Sicherheitsprofil speichern.

  Zum Speichern eines Sicherheitsprofils verwenden Sie den Parameter **-ssp** *Profilname* oder **--saveSecProfile** *Profilname* mit dem Rest des Befehls. Durch das Hinzufügen dieses Parameters zu Ihrem Befehl werden die folgenden Parameter gespeichert:

  ```
  -al,--alias <Alias>
  -arc,--authRetryCount <Integer>
  -ca,--credAuth <Unterstützung>
  -cgc,--credGenClass <Klassenname>
  -cgp,--credGenProps <Eigenschaft>
  -cxpv,--contextProvider <Provider>
  -ks,--keyStore <Dateipfad>
  -ksp,--keyStorePassword <Kennwort>
  -kst,--keyStoreType <Typ>
  -prot,--protocol <Protokoll>
  -pwd,--password <Kennwort>
  -ts,--trustStore <Dateipfad>
  -tsp,--trustStorePassword <Kennwort>
  -tst,--trustStoreType <Typ>
  -tt,--transportType <Typ>
  -user,--username <Benutzername>
  ```

  Sicherheitsprofile werden im Verzeichnis *Benutzerausgangsverzeichnis*`\.xscmd\profiles\security\<`*Profilname*`>.properties` gespeichert.

- Gespeichertes Sicherheitsprofil verwenden.

  Zur Verwendung eines gespeicherten Sicherheitsprofils fügen Sie den Parameter **-sp** *Profilname* oder **--securityProfile** *Profilname* dem Befehl hinzu, den Sie ausführen. Befehlsbeispiel: `xscmd -c listHosts -cep myhost.mycompany.com -sp myprofile`

- Befehle in der Befehlsgruppe **ProfileManagement** auflisten.

  Führen Sie den folgenden Befehl aus: **xscmd -lc ProfileManagement**.

- Vorhandene Sicherheitsprofile auflisten.

  Führen Sie den folgenden Befehl aus: **xscmd -c listProfiles -v**.

- In einem Sicherheitsprofil gespeicherte Einstellungen anzeigen.

  Führen Sie den folgenden Befehl aus: **xscmd -c showProfile -pn** *Profilname*.

- Vorhandenes Sicherheitsprofil entfernen.

  Führen Sie den folgenden Befehl aus: **xscmd -c RemoveProfile -pn** *Profilname*.

Kapitel 11. Fehlerbehebung

Zusätzlich zu den in diesem Abschnitt beschriebenen Protokollen, Trace, Nachrichten und Releaseinformationen können Sie Überwachungstools verwenden, um Gegebenheiten zu verstehen, wie z. B. die Position der Daten in der Umgebung, die Verfügbarkeit der Server im Datengrid usw. Wenn Sie in einer Umgebung mit WebSphere Application Server arbeiten, können Sie Performance Monitoring Infrastructure (PMI) verwenden. Wenn Sie ein einer eigenständigen Umgebung arbeiten, können Sie Überwachungstools anderer Anbieter verwenden, wie z. B. CA Wily Introscope oder Hyperic HQ. Außerdem können Sie das Dienstprogramm **xscmd** verwenden und anpassen, um Textinformationen zu Ihrer Umgebung anzuzeigen.

Protokollierung aktivieren

Sie können Protokolle verwenden, um Ihre Umgebung zu überwachen und Fehler zu beheben.

Informationen zu diesem Vorgang

Protokolle werden je nach Konfiguration an unterschiedlichen Positionen und in unterschiedlichen Formaten gespeichert.

Vorgehensweise

- **Protokolle in einer eigenständigen Umgebung aktivieren.**

  Bei eigenständigen Katalogservern befinden sich die Protokolle an der Position, an der Sie den Befehl **startOgServer** ausführen. Bei Container-Servern können Sie die Standardposition verwenden oder eine angepasste Protokollposition festlegen:

  – **Standardprotokollposition:** Die Protokolle befinden sich in dem Verzeichnis, in dem der Serverbefehl ausgeführt wurde. Wenn Sie die Server im Verzeichnis *WXS-Ausgangsverzeichnis*/bin starten, werden die Protokoll- und Tracedateien in den Verzeichnissen logs/<*Servername*> des Verzeichnisses bin gespeichert.

  – **Angepasste Protokollposition:** Wenn Sie eine andere Position für die Container-Server-Protokolle festlegen möchten, erstellen Sie eine Eigenschaftendatei, z. B. server.properties, mit dem folgenden Inhalt:

    ```
    workingDirectory=<Verzeichnis>
    traceSpec=
    systemStreamToFileEnabled=true
    ```

    Die Eigenschaft **workingDirectory** ist das Stammverzeichnis für die Protokolle und die optionale Tracedatei. WebSphere eXtreme Scale erstellt ein Verzeichnis mit dem Namen des Container-Servers mit einer Datei SystemOut.log, einer Datei SystemErr.log und einer Tracedatei. Wenn eine Eigenschaftendatei während des Containerstarts verwendet werden soll, verwenden Sie die Option **-serverProps**, und geben Sie die Position der Servereigenschaftendatei an.

- **Protokolle in WebSphere Application Server aktivieren.**

  Weitere Informationen finden Sie unter WebSphere Application Server: Protokollierung aktivieren und inaktivieren.

- **FFDC-Dateien abrufen.**

  FFDC-Dateien sind als Debug-Hilfe für die IBM Unterstützungsfunktion bestimmt. Diese Dateien werden möglicherweise von der IBM Unterstützungsfunk-

tion angefordert, wenn ein Problem auftritt. Diese Dateien befinden sich in einem Verzeichnis mit dem Namen `ffdc`. Das Verzeichnis enthält Dateien wie die folgenden:

```
server2_exception.log
server2_20802080_07.03.05_10.52.18_0.txt
```

Nächste Schritte

Sehen Sie sich die Protokolldateien an den angegebenen Positionen an. Häufig ausgegebene Nachrichten, die in der Datei `SystemOut.log` aufgezeichnet werden, sind Bestätigungsnachrichten für den Start, z. B.:

```
CWOBJ1001I: Der ObjectGrid-Server catalogServer01 ist für die Verarbeitung
von Anforderungen bereit.
```

Weitere Informationen zu bestimmten Nachrichten in den Protokolldateien finden Sie im Abschnitt Nachrichten.

Trace erfassen

Sie können Traces verwenden, um Ihre Umgebung zu überwachen und Fehler zu beheben. Sie müssen den Trace für einen Server bereitstellen, wenn Sie mit dem IBM Support zusammenarbeiten.

Informationen zu diesem Vorgang

Die Erfassung eines Trace kann Ihnen bei der Überwachung und der Behebung von Fehlern in Ihrer Implementierung von WebSphere eXtreme Scale helfen. Wie Sie den Trace erfassen, richtet sich nach Ihrer Konfiguration. Eine Liste der verschiedenen Tracespezifikationen, die Sie erfassen können, finden Sie unter „Traceoptionen" auf Seite 546.

Vorgehensweise

- **Trace in einer Umgebung von WebSphere Application Server erfassen.**

  Wenn Ihre Katalog- und Container-Server in einer Umgebung von WebSphere Application Server ausgeführt werden, finden Sie unter WebSphere Application Server: Working with trace weitere Informationen.

- **Trace mit dem Startbefehl für eigenständige Katalog- oder Container-Server erfassen.**

  Sie können die Traceerstellung für einen Katalog-Service oder Container-Server mit den Parametern **`-traceSpec`** und **`-traceFile`** des Befehls **`startOgServer`** festlegen. Beispiel:

  ```
  startOgServer.sh catalogServer -traceSpec ObjectGridPlacement=all=enabled -traceFile  /home/user1/logs/trace.log
  ```

  Der Parameter **`-traceFile`** ist optional. Wenn Sie keine Position mit **`-traceFile`** angeben, wird die Tracedatei an derselben Position wie die Systemausgabeprotokolldateien abgelegt.Weitere Informationen zu diesen Parametern finden Sie in „Script **`startOgServer`**" auf Seite 409.

- **Trace mit einer Eigenschaftendatei für den eigenständigen Katalog- oder Container-Server erfassen.**

  Wenn Sie den Trace mithilfe einer Eigenschaftendatei erfassen möchten, erstellen sie eine Datei, z. B. `server.properties`, mit dem folgenden Inhalt:

  ```
  workingDirectory=<Verzeichnis>
  traceSpec=<Tracespezifikation>
  systemStreamToFileEnabled=true
  ```

Die Eigenschaft **workingDirectory** ist das Stammverzeichnis für die Protokolle und die optionale Tracedatei. Wenn Sie keinen Wert für die Eigenschaft **workingDirectory** festlegen, wird die zum Starten verwendete Position, z. B. *WXS-Ausgangsverzeichnis*/bin, standardmäßig als Arbeitsverzeichnis verwendet. Zur Verwendung einer Eigenschaftendatei während des Serverstarts verwenden Sie den Parameter **-serverProps** mit dem Befehl **startOgServer**, und geben Sie die Position der Servereigenschaftendatei an.Weitere Informationen zur Servereigenschaftendatei und zu deren Verwendung finden Sie in Servereigenschaftendatei.

- **Trace für einen eigenständigen Client erfassen.**

  Sie können die Traceerfassung für einen eigenständigen Client starten, indem Sie dem Startscript für die Clientanwendung Systemeigenschaften hinzufügen. Im folgenden Beispiel werden Traceeinstellungen für die Anwendung com.ibm.samples.MyClientProgram angegeben:

  ```
  java -DtraceSettingsFile=MyTraceSettings.properties
  -Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager
  -Djava.util.logging.configureByServer=true com.ibm.samples.MyClientProgram
  ```

  Weitere Informationen finden Sie unter WebSphere Application Server: Enabling trace on client and stand-alone applications.

- **Trace mit der Schnittstelle ObjectGridManager erfassen.**

  Sie können die Traceerstellung auch zur Laufzeit in einer ObjectGridManager-Schnittstelle definieren. Die Definition der Traceerstellung in einer ObjectGrid-Manager-Schnittstelle kann verwendet werden, um einen Trace für einen eXtreme-Scale-Client zu erstellen, wenn dieser eine Verbindung zu einer eXtreme-Scale-Instanz herstellt und Transaktionen festschreibt. Wenn Sie die Traceerstellung in einer ObjectGridManager-Schnittstelle festlegen möchten, geben Sie eine Tracespezifikation und ein Traceprotokoll an.

  ```
  ObjectGridManager manager = ObjectGridManagerFactory.getObjectGridManager();
  ...
  manager.setTraceEnabled(true);
  manager.setTraceFileName("logs/myClient.log");
  manager.setTraceSpecification("ObjectGridReplication=all=enabled");
  ```

  Weitere Informationen zur Schnittstelle ObjectGridManager finden Sie in den Informationen zur Interaktion mit ObjectGrid über die Schnittstelle ObjectGridManager in der Veröffentlichung *Programmierung*.

- **Trace für Container-Server mit dem Dienstprogramm xscmd erfassen.**

  Zum Erfassen des Trace mit dem Dienstprogramm **xscmd** verwenden Sie den Befehl **-c setTraceSpec**. Verwenden Sie das Dienstprogramm **xscmd**, um den Trace in einer eigenständigen Umgebung zur Laufzeit und nicht während des Starts zu erfassen. Sie können den Trace für alle Server und Katalogservices erfassen oder die Server basierend auf dem ObjectGrid-Namen und anderen Eigenschaften filtern. Führen Sie beispielsweise den folgenden befehl aus, um den ObjectGridReplication-Trace mit Zugriff auf die Server des Katalogservice zu erfassen:

  ```
  xscmd -c setTraceSpec "ObjectGridReplication=all=enabled"
  ```

  Sie können die Traceerstellung inaktivieren, indem Sie die Tracespezifikation auf *=all=disabled setzen.

Ergebnisse

Tracedateien werden an die angegebene Position geschrieben.

Traceoptionen

Sie können die Traceerstellung aktivieren, um der IBM Unterstützungsfunktion Informationen über Ihre Umgebung bereitzustellen.

Informationen zur Traceerstellung

Der Trace von WebSphere eXtreme Scale ist in mehrere Komponenten unterteilt. Sie können die zu verwendende Tracestufe angeben. Zu den gängigen Tracestufen gehören all, debug, entryExit und event.

Im Folgenden sehen Sie ein Beispiel für eine Tracezeichenfolge:

```
ObjectGridComponent=level=enabled
```

Sie können Tracezeichenfolgen verknüpfen. Verwenden Sie das Symbol `*` (Stern), um einen Platzhalterwert anzugeben, z. B. `ObjectGrid*=all=enabled`. Wenn Sie einen Trace für die IBM Unterstützungsfunktion bereitstellen müssen, ist eine bestimmte Tracezeichenfolge erforderlich. So kann beispielsweise die Tracezeichenfolge `ObjectGridReplication=debug=enabled` angefordert werden, wenn ein Problem mit der Replikation auftritt.

Tracespezifikation

ObjectGrid
Allgemeine Basiscachesteuerkomponente.

ObjectGridCatalogServer
Allgemeiner Katalogservice.

ObjectGridChannel
Statische Kommunikation in der Implementierungstopologie.

ObjectGridClientInfo
DB2-Clientinformationen.

ObjectGridClientInfoUser
DB2-Benutzerinformationen.

ObjectgridCORBA
Dynamische Kommunikation in der Implementierungstopologie.

ObjectGridDataGrid
Die API "AgentManager".

ObjectGridDynaCache
Der dynamische Cache-Provider von WebSphere eXtreme Scale.

ObjectGridEntityManager
Die API "EntityManager". Mit der Option "Projector" zu verwenden.

ObjectGridEvictors
Integrierte ObjectGrid-Evictor (Bereinigungsprogramme).

ObjectGridJPA
JPA-Loader (Java Persistence API).

ObjectGridJPACache
JPA-Cache-Plug-ins.

ObjectGridLocking
Sperrenmanager für ObjectGrid-Cacheeinträge.

ObjectGridMBean
Management-Beans.

ObjectGridMonitor
Infrastruktur für Langzeitüberwachung.

7.1.1+ **ObjectGridNative**
Traceerstellung für den nativen Code von WebSphere eXtreme Scale, einschließlich des nativen Codes von eXtremeMemory.

7.1.1+ **ObjectGridOSGi**
Die OSGi-Integrationskomponenten von WebSphere eXtreme Scale.

ObjectGridPlacement
Katalogserverservice für Shard-Verteilung.

ObjectGridQuery
ObjectGrid-Abfrage.

ObjectGridReplication
Replikationsservice.

ObjectGridRouting
Details zum Client/Server-Routing.

ObjectGridSecurity
Sicherheitstrace.

7.1.1+ **ObjectGridSerializer**
Die Infrastruktur des Plug-ins "DataSerializer".

ObjectGridStats
ObjectGrid-Statistiken.

ObjectGridStreamQuery
Die API "Stream Query".

7.1.1+ **ObjectGridTransactionManager**
Der Transaktionsmanager von WebSphere eXtreme Scale.

ObjectGridWriteBehind
ObjectGrid-Write-behind.

7.1.1+ **ObjectGridXM**
Allgemeiner Trace für IBM eXtremeMemory.

7.1.1+ **ObjectGridXMEviction**
Trace für eXtremeMemory-Bereinigung.

7.1.1+ **ObjectGridXMTransport**
Allgemeiner Trace für eXtremeMemory-Transporte.

7.1.1+ **ObjectGridXMTransportInbound**
Trace für eingehende eXtremeMemory-Transporte.

7.1.1+ **ObjectGridXMTransportOutbound**
Trace für abgehende eXtremeMemory-Transporte.

Projector
Die Engine in der API EntityManager.

QueryEngine
Die Abfrageengine für die API "Object Query" und die API "EntityManager Query".

QueryEnginePlan
Trace für Abfrageplan.

7.1.1+ **TCPChannel**
Der TCP/IP-Kanal von IBM eXtremeIO.

7.1.1+ **XsByteBuffer**
Trace für Bytepuffer von WebSphere eXtreme Scale.

Protokoll- und Tracedaten analysieren

Sie können die Protokollanalysetools verwenden, um zu analysieren, welche Leistung Ihre Laufzeitumgebung aufweist und wie diese Probleme behebt, die in der Umgebung auftreten.

Informationen zu diesem Vorgang

Sie können aus den vorhandenen Protokoll- und Tracedateien Berichte generieren. Diese visuellen Berichte können für die folgenden Zwecke verwendet werden:

- **Analyse des Zustands und der Leistung der Laufzeitumgebung:**
  - Konsistenz der Implementierungsumgebung
  - Protokollierungsintervall
  - Vergleich der aktiven Topologie und der konfigurierten Topologie
  - Nicht geplante Topologieänderungen
  - Quorumstatus
  - Partitionsreplikationsstatus
  - Statistiken zu Hauptspeicher, Durchsatz, Prozessorbelegung usw.
- **Behebung von Problemen in der Umgebung:**
  - Topologieansichten zu bestimmten Zeitpunkten
  - Statistiken zu Hauptspeicher, Durchsatz, Prozessorbelegung bei Clientfehlern
  - Aktuelle Fixpackstufen, Optimierungseinstellungen
  - Quorumstatus

Übersicht über die Protokollanalyse

Sie können das Tool **`xsLogAnalyzer`** zur Unterstützung der Fehlerbehebung in der Umgebung verwenden.

Alle Failover-Nachrichten

Zeigt die Gesamtanzahl der Failover-Nachrichten als Diagramm über der Zeit an. Zeigt außerdem eine Liste der Failover-Nachrichten, einschließlich der betroffenen Server, an.

Alle kritischen eXtreme-Scale-Nachrichten

Zeigt Nachrichten-IDs zusammen mit den zugehörigen Erläuterungen und Benutzeraktionen an. Mithilfe dieser Aufstellung können Sie Zeit bei Suchen von Nachrichten einsparen.

Alle Ausnahmen

Zeigt die fünf am häufigsten ausgegebenen Ausnahmen, einschließlich der Nachrichten und der Anzahl ihrer Vorkommen, sowie die von der Ausnahme betroffenen Server an.

Topologiezusammenfassung

Zeigt anhand der Protokolldateien ein Diagramm an, das veranschaulicht, wie Ihre Topologie konfiguriert ist. Sie können diese Zusammenfassung verwenden, um Ihre eigentliche Konfiguration zu vergleichen, und dabei mögliche Konfigurationsfehler erkennen.

Topologiekonsistenz: ORB-Vergleichstabelle (Object Request Broker)

Zeigt ORB-Einstellungen in der Umgebung an. Sie können diese Tabelle verwenden, um festzustellen ob die Einstellungen in Ihrer Umgebung konsistent sind.

Zeitachsensicht von Ereignissen

Zeigt ein Zeitachsendiagramm verschiedener Aktionen an die im Datengrid ausgeführt wurden, einschließlich Lebenszyklusereignissen, Ausnahmen, kritischer Nachrichten und FFDC-Ereignissen (First-Failure Data Capture).

Protokollanalyse durchführen

Sie können das Tool **xsLogAnalyzer** für eine Reihe von Protokoll- und Tracedateien auf jedem Computer ausführen.

Vorbereitende Schritte

- Aktivieren Sie Protokolle und Trace. Weitere Informationen finden Sie unter „Protokollierung aktivieren" auf Seite 543 und „Trace erfassen" auf Seite 544.
- Erfassen Sie Ihre Protokolldateien. Die Protokolldateien können sich je nach Konfiguration an verschiedenen Positionen befinden. Wenn Sie die Standardprotokolleinstellungen verwenden, können Sie die Protokolldateien von den folgenden Positionen abrufen:
  - In einer eigenständigen Installation: *WXS-Installationsstammverzeichnis*/bin/logs/<*Servername*>
  - In einer Installation, die mit WebSphere Application Server integriert ist: *WAS-Stammverzeichnis*/logs/<*Servername*>
- Erfassen Sie Ihre Tracedateien. Die Tracedateien können sich je nach Konfiguration an verschiedenen Positionen befinden. Wenn Sie die Standardtraceeinstellungen verwenden, können Sie die Tracedateien von den folgenden Positionen abrufen:
  - In einer eigenständigen Installation: Wenn kein bestimmter Tracewert gesetzt wird, werden die Tracedateien an dieselbe Position wie die Systemausgabeprotokolldateien geschrieben.
  - In einer Installation, die mit WebSphere Application Server integriert ist: *WAS-Stammverzeichnis*/profiles/*Servername*/logs.

  Kopieren Sie die Protokoll- und Tracedateien auf den Computer, über den Sie das Protokollanalysetool verwenden möchten.
- Wenn Sie angepasste Scanner in Ihrem generierten Bericht verwenden möchten, erstellen Sie eine Eigenschaftendatei und eine Konfigurationsdatei für die Scan-

nerspezifikationen, bevor Sie das Tool ausführen. Weitere Informationen finden Sie unter „Angepasste Scanner für die Protokollanalyse erstellen” auf Seite 551.

Vorgehensweise

1. Führen Sie das Tool **xsLogAnalyzer** aus.

   Das Script befindet sich an den folgenden Positionen:

   - In einer eigenständigen Installation: *WXS-Installationsstammverzeichnis*/ObjectGrid/bin
   - In einer Installation, die mit WebSphere Application Server integriert ist: *WAS-Stammverzeichnis*/bin

   Tipp: Wenn Ihre Protokolldateien groß sind, können Sie die Parameter **-startTime**, **-endTime** und **-maxRecords** bei der Ausführung des Berichts verwenden, um die Anzahl der gescannten Protokolleinträge zu beschränken. Wenn Sie diese Parameter bei der Ausführung des Berichts verwenden, lassen sich die Berichte einfacher lesen und effizienter ausführen. Sie können mehrere Berichte für dieselbe Gruppe von Protokolldateien ausführen.

   ```
   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
   Gibt den absoluten Pfad zum Protokollverzeichnis an, das Sie auswerten möchten (erforderlich).

   -outDir
   Gibt ein vorhandenes Verzeichnis an, in das die Berichtsausgabe geschrieben werden soll. Wenn Sie keinen Wert angeben, wird der Bericht an die Stammposition des Tools **xsLogAnalyzer** geschrieben.

   -startTime
   Gibt die Startzeit für die Auswertung in den Protokollen an. Das Startdatum hat das folgende Format:
   Jahr.Monat.Tag.Stunde.Minute.Sekunde.Millisekunde

   -endTime
   Gibt die Endzeit für die Auswertung in den Protokollen an. Das Startdatum hat das folgende Format:
   Jahr.Monat.Tag.Stunde.Minute.Sekunde.Millisekunde

   -trace Gibt die Tracezeichenfolge an, z. B. `ObjectGrid*=all=enabled`.

   -maxRecords
   Gibt die maximale Anzahl der im Bericht zu generierenden Datensätze an. Der Standardwert ist `100`. Wenn Sie 50 angeben, werden die ersten 50 Datensätze für den angegebenen Zeitraum generiert.

2. Öffnen Sie die generierten Dateien. Wenn Sie kein Ausgabeverzeichnis definiert haben, werden die Berichte in einem Ordner mit dem Namen `report_`*`Datum_Uhrzeit`* generiert. Zum Öffnen der Hauptseite der Berichte öffnen Sie die Datei `index.html`.
3. Verwenden Sie die Berichte, um die Protokolldaten zu analysieren. Verwenden Si die folgenden Tipps, um die Leistung der Berichtsanzeigen zu maximieren.
   - Zum Maximieren der Leistung von Protokolldatenabfragen verwenden Sie so spezifische Informationen wie möglich. Eine Abfrage von `server` dauert beispielsweise sehr viel länger und gibt mehr Ergebnisse zurück als eine Abfrage von `Hostname_des_Servers`.

- Einigen Ansichten haben eine beschränkte Anzahl an Datenpunkten, die gleichzeitig angezeigt werden. Sie können das Zeitsegment, das angezeigt wird, anpassen, indem Sie die aktuellen Daten, wie z. B. Start- und Endzeit, in der Ansicht ändern.

Nächste Schritte

Weitere Informationen zur Behebung von Fehlern im Tool **xsLogAnalyzer** und in den gerierten Berichten finden Sie unter „Fehlerbehebung bei der Protokollanalyse" auf Seite 552.

Angepasste Scanner für die Protokollanalyse erstellen

Sie können angepasste Scanner für die Protokollanalyse erstellen. Nach der Konfiguration des Scanners werden die Ergebnisse in den Berichten generiert, wenn Sie das Tool **xsLogAnalyzer** ausführen. Der angepasste Scanner scannt die Protokolle basierend auf den von Ihnen angegebenen regulären Ausdrücken nach Ereignisdatensätzen.

Vorgehensweise

1. Eigenschaftendatei für Scannerspezifikation erstellen, die die allgemeinen Ausdrücke enthält, die für den angepassten Scanner ausgeführt werden sollen.
   a. Erstellen und speichern Sie eine Eigenschaftendatei. Die Datei muss im Verzeichnis *Stammverzeichnis_für_Protokollanalyse*/config/custom gespeichert werden. Sie können einen beliebigen Namen für die Datei verwenden. Die Datei wird vom neuen Scanner verwendet. Deshalb ist die Nennung des Scanners in der Eigenschaftendatei sinnvoll, z. B.: *my_new_server_scanner_spec*.properties.
   b. Schließen Sie die folgenden Eigenschaften in die Datei *my_new_server_scanner_spec*.properties ein:
      ```
      include.regular_expression = REGULÄRER_AUSDRUCK_FÜR_SCAN
      ```

      Die Variable *REGULÄRER_AUSDRUCK_FÜR_SCAN* steht für einen regulären Ausdruck, auf dessen Basis die Protokolldateien gefiltert werden.

      Beispiel: Wenn Sie Instanzen von Zeilen suchen möchten, die die Zeichenfolge "xception" und die Zeichenfolge "rror" enthalten, unabhängig von der Reihenfolge, setzen Sie die Eigenschaft **include.regular_expression** auf den folgenden Wert:
      ```
      include.regular_expression = (xception.+rror)|(rror.+xception)
      ```

      Dieser reguläre Ausdruck bewirkt, dass Ereignisse aufgezeichnet werden, wenn die Zeichenfolge "rror" vor oder hinter der Zeichenfolge "xception" steht.

      Beispiel:Wenn Sie die gesamten Protokolle nach allen Instanzen von Zeilen durchsuchen möchten, die die Phrase "xception" oder die Phrase "rror" enthalten, unabhängig von der Reihenfolge, setzen Sie die Eigenschaft **include.regular_expression** auf den folgenden Wert:
      ```
      include.regular_expression = (xception)|(rror)
      ```

      Dieser reguläre Ausdruck bewirkt, dass Ereignisse aufgezeichnet werden, wenn die Zeichenfolge "rror" vor oder hinter der Zeichenfolge "xception" steht.
2. Konfigurationsdatei erstellen, die das Tool **xsLogAnalyer** zum Erstellen des Scanners verwendet.

a. Erstellen und speichern Sie eine Eigenschaftendatei. Die Datei muss im Verzeichnis *Stammverzeichnis_für_Protokollanalyse*/config/custom gespeichert werden. Sie können diese Datei *Scannername*Scanner.config nennen, wobei *Scannername* für einen eindeutigen Namen für den neuen Scanner steht. Sie können die Datei beispielsweise serverScanner.config nennen.

b. Schließen Sie die folgenden Eigenschaften in die Datei *Scannername*Scanner.config ein:

```
scannerSpecificationFiles = POSITION_DER_SCANNERSPEZIFIKATIONSDATEI
```

Die Variable *POSITION_DER_SCANNERSPEZIFIKATIONSDATEI* steht für den Pfad und die Position der Spezifikationsdatei, die Sie im vorherigen Schritte erstellt haben, z. B. *Stammverzeichnis_für_Protokollanalyse*/config/custom/my_new_scanner_spec.properties. Sie können auch mehrere Scannerspezifikationsdateien angeben, indem Sie eine durch Semikolons getrennte Liste verwenden:

```
scannerSpecificationFiles = POSITION_DER_SCANNERSPEZIFIKATIONSDATEI1;POSITION_DER_SCANNERSPEZIFIKATIONSDATEI2
```

3. Führen Sie das Tool **xsLogAnalyzer** aus. Weitere Informationen finden Sie im Abschnitt „Protokollanalyse durchführen” auf Seite 549.

Ergebnisse

Nach der Ausführung des Tools **xsLogAnalyzer** enthält der Bericht neue Registerkarten für die konfigurierten angepassten Scanner. Jede Registerkarte enthält die folgenden Ansichten.

Diagramme
Ein gezeichneter Graph, der die aufgezeichneten Ereignisse darstellt. Die Ereignisse werden in der Reihenfolge angezeigt, in der sie gefunden wurden.

Tabellen
Eine tabellarische Darstellung der aufgezeichneten Ereignisse.

Ergebnisberichte

Fehlerbehebung bei der Protokollanalyse

Verwenden Sie die folgenden Fehlerbehebungsinformationen, um Probleme mit dem Tool **xsLogAnalyzer** und den von diesem Tool generierten Berichten zu diagnostizieren und zu beheben.

Vorgehensweise

- **Problem:** Es treten abnormale Speicherbedingungen auf, wenn Sie das Tool **xsLogAnalyzer** zum Generieren von Berichten verwenden. Im Folgenden sehen Sie ein Beispiel für einen solchen Fehler: `java.lang.OutOfMemoryError: GC overhead limit exceeded`.

  Lösung: Das Tool **xsLogAnalyzer** wird in einer Java Virtual Machine (JVM) ausgeführt. Sie können die JVM konfigurieren, um die Größe des Heapspeichers zu erhöhen, bevor Sie das Tool **xsLogAnalyzer** ausführen, indem Sie beim Ausführen des Tools einige Einstellungen angeben. Wenn Sie die Größe des Heapspeichers erhöhen, können noch mehr Ereignisdatensätze im JVM-Speicher gespeichert werden. Beginnen Sie mit der Einstellung 2048M, sofern das Betriebssystem genügend Hauptspeicher besitzt. Setzen Sie in derselben Befehlszeileninstanz, in der Sie das Tool **xsLogAnalyzer** ausführen möchten, die maximale Größe des JVM-Heapspeichers:

  ```
  java -Xmx Größe_des_Heapspeichers m
  ```

Für *Größe_des_Heapspeichers* können Sie jede beliebige ganze Zahl angeben. Der Wert stellt die Anzahl der Megabyte dar, die für den JVM-Heapspeicher reserviert werden. Sie können beispielsweise `java -Xmx2048m` ausführen. Wenn weiterhin Nachrichten über abnormale Speicherbedingungen ausgegeben werden oder wenn die Ressourcen nicht ausreichen, um 2048m oder mehr Speicher zu reservieren, beschränken Sie die Anzahl der Ereignisse, die im Heapspeicher verwaltet werden. Sie können die Anzahl der Ereignisse im Heapspeicher beschränken, indem Sie den Parameter **-maxRecords** an den Befehl **xsLogAnalyzer** übergeben.

- **Problem:** Wenn Sie einen generierten Bericht über das Tool **xsLogAnalyzer** öffnen, blockiert der Browser oder lädt die Seite nicht.

  Ursache: Die generierten HTML-Dateien sind zu groß und können vom Browser nicht geladen werden. Diese Dateien sind so groß, weil der Geltungsbereich der Protokolldateien, die Sie analysieren, zu weitgefasst ist.

  Lösung: Sie können die Parameter **-startTime**, **-endTime** und **-maxRecords** bei der Ausführung des Tools **xsLogAnalyzer** verwenden, um die Anzahl der gescannten Protokolleinträge zu beschränken. Wenn Sie diese Parameter bei der Ausführung des Berichts verwenden, lassen sich die Berichte einfacher lesen und effizienter ausführen. Sie können mehrere Berichte für dieselbe Gruppe von Protokolldateien ausführen.

Fehlerbehebung bei der Installation

Verwenden Sie diese Informationen, um Probleme in Ihrer Installation zu beheben.

Vorgehensweise

- **Problem:** Wenn Sie den Installationsbefehl über einen fernen Computer ausführen, z. B. `\\mymachine\downloads\`, wird die folgende Nachricht angezeigt: `CMD.EXE was started with the above path as the current directory. UNC paths are not supported. Defaulting to Windows directory.` Deshalb wird die Installation nicht ordnungsgemäß durchgeführt.

  Lösung: Ordnen Sie den fernen Computer einem Netzlaufwerk zu. Unter Windows können Sie beispielsweise mit der rechten Maustaste auf **Arbeitsplatz** klicken, **Netzlaufwerk verbinden** auswählen und den UNC-Pfad (Uniform Naming Conventions) zum fernen Computer einfügen. Anschließend können Sie das Installationsscript erfolgreich über das Netzlaufwerk ausführen, z. B.: `y:\mymachine\downloads\WXS\install.bat`.

- **Problem:** Die Installation wird nicht erfolgreich durchgeführt.

  Lösung: Überprüfen Sie die Protokolldateien, um festzustellen, wo die Installation fehlgeschlagen ist. Wenn die Installation scheitert, befinden sich die Protokolle im Verzeichnis *WXS-Installationsstammverzeichnis*`/logs/wxs`.

- **Problem:** Während der Installation tritt ein katastrophaler Fehler auf.

  Lösung: Überprüfen Sie die Protokolldateien, um festzustellen, wo die Installation fehlgeschlagen ist. Wenn die Installation nach teilweiser Ausführung scheitert, befinden sich die Protokolle im Allgemeinen im Verzeichnis *Benutzerstammverzeichnis*`/wxs_install_logs/`.

- Windows **Problem:** Wenn Sie WebSphere eXtreme Scale Client unter Windows installieren, finden Sie in den Ergebnissen der Installation möglicherweise den folgenden Text:

  ```
  Erfolg: Die Installation des folgenden Produkts war erfolgreich:
  WebSphere eXtreme Scale Client. In einigen Konfigurationsschritten sind Fehler
  aufgetreten. Weitere Informationen finden Sie in der folgenden Protokolldatei:
  <WAS-Installationsstammverzeichnis>\logs\wxs_client\install\log.txt"
  Sehen Sie sich das Installationsprotokoll (log.txt) und das Erweiterungsprotokoll
  des Deployment Manager an.
  ```

Lösung: Wenn Sie einen Fehler sehen, der sich auf die Datei `iscdeploy.sh` bezieht, können Sie diesen ignorieren. Dieser Fehler verursacht keine Probleme.

Fehlerbehebung bei der Cacheintegration

Verwenden Sie diese Informationen, um Probleme mit der Ihrer Konfiguration der Cacheintegration, einschließlich HTTP-Sitzungs- und dynamischen Cachekonfigurationen, zu beheben.

Vorgehensweise

- 7.1.1+ **Problem:** HTTP-Sitzungs-IDs werden nicht wiederverwendet.

  Ursache: Sie können Sitzungs-IDs wiederverwenden. Wenn Sie ein Datengrid für die Sitzungspersistenz in Version 7.1.1 oder höher erstellen, ist die Wiederverwendung von Sitzungs-IDs automatisch aktiviert. Haben Sie jedoch frühere Konfigurationen erstellt wurde diese Einstellung unter Umständen mit dem falschen Wert gesetzt.

  Lösung: Überprüfen Sie die folgenden Einstellungen, um sicherzustellen, dass die Wiederverwendung der HTTP-Sitzungs-IDs aktiviert ist.

  – Die Eigenschaft `reuseSessionId` in der Datei `splicer.properties` muss auf `true` gesetzt sein.

  – Die angepasste Eigenschaft `HttpSessionIdReuse` muss auf `true` gesetzt sein. Diese angepasste Eigenschaft kann über einen der folgenden Pfade in der Administrationskonsole von WebSphere Application Server definiert werden:

    - Klicken Sie auf **Server** > *Servername* > **Sitzungsverwaltung** > **Angepasste Eigenschaften**.
    - Klicken Sie auf **Dynamische Cluster** > *Name_des_dynamischen_Clusters* > **Serverschablone** > **Sitzungsverwaltung** > **Angepasste Eigenschaften**.
    - Klicken Sie auf **Server** > **Servertypen** > **WebSphere-Anwendungsserver** > *Servername* und anschließend unter "Serverinfrastruktur" auf **Java- und Prozessverwaltung** > **Prozessdefinition** > **Java Virtual Machine** > **Angepasste Eigenschaften**.
    - Klicken Sie auf **Server** > **Servertypen** > **WebSphere-Anwendungsserver** > *Servername* > **Einstellungen des Webcontainers** > **Webcontainer**.

  Wenn Sie Werte angepasster Eigenschaften aktualisieren, müssen Sie die eXtreme-Scale-Sitzungsverwaltung so rekonfigurieren, dass die Datei `splicer.properties` von der Änderung Kenntnis erhält.

- **Problem:** wenn Sie ein Datengrid zum Speichern von HTTP-Sitzungen verwenden und die Transaktionslast hoch ist, wird eine Nachricht CWOBJ0006W in der Datei `SystemOut.log` angezeigt.

  ```
  CWOBJ0006W: Es ist eine Ausnahme eingetreten:
  com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
  java.util.ConcurrentModificationException
  ```

  Diese Nachricht wird nur angezeigt, wenn der Parameter **`replicationInterval`** in der Datei `splicer.properties` auf einen Wert größer als null gesetzt ist und die Webanwendung ein List-Objekt ändert, das als Attribut in HTTPSession definiert wurde.

  Lösung: Klonen Sie das Attribut, das das geänderte List-Objekt enthält, und fügen Sie das geklonte Attribut in das Sitzungsobjekt ein.

Fehlerbehebung beim JPA-Cache-Plug-in

Verwenden Sie diese Informationen, um Probleme mit Ihrer JPA-Cache-Plug-in-Konfiguration zu beheben. Diese Probleme können in Hibernate- und OpenJPA-Konfigurationen auftreten.

Vorgehensweise

- **Problem:** Die folgende Ausnahme wird angezeigt: `CacheException: Failed to get ObjectGrid server.`

  Mit dem **`ObjectGridType`**-Attributwert `EMBEDDED` oder `EMBEDDED_PARTITION` versucht der eXtreme-Scale-Cache, eine Serverinstanz von der Laufzeitumgebung abzurufen. In einer Java-SE-Umgebung wird ein Server von eXtreme Scale mit integriertem Katalogservice gestartet. Der integrierte Katalogservice versucht, an Port 2809 empfangsbereit zu sein. Wenn dieser Port von einem anderen Prozess verwendet wird, tritt dieser Fehler auf.

  Lösung: Wenn externe Katalogserviceendpunkte angegeben werden, z. B. in der Datei `objectGridServer.properties`, tritt dieser Fehler auf, wenn der Hostname oder Port falsch angegeben sind. Beheben Sie den Portkonflikt.

- **Problem:** Die folgende Ausnahme wird angezeigt: `CacheException: Failed to get REMOTE ObjectGrid for configured REMOTE ObjectGrid. objectGridName = [ObjectGridName], PU name = [persistenceUnitName]`

  Dieser Fehler tritt auf, weil der Cache die ObjectGrid-Instanz nicht von den bereitgestellten Endpunkten des Katalogservice abrufen kann.

  Lösung: Dieses Problem tritt gewöhnlich auf, weil ein ungültiger Hostname oder Port angegeben wurde.

- **Problem:** Die folgende Ausnahme wird angezeigt: `CacheException: Cannot have two PUs [persistenceUnitName_1, persistenceUnitName_2] configured with same ObjectGridName [ObjectGridName] of EMBEDDED ObjectGridType`

  Diese Ausnahme wird angezeigt, wenn viele Persistenzeinheiten konfiguriert sind und die eXtreme Scale-Caches dieser Einheiten mit demselben ObjectGrid-Namen und dem Wert `EMBEDDED` für das Attribut **`ObjectGridType`** konfiguriert sind. Diese Persistenzeinheitenkonfigurationen können in derselben oder in unterschiedlichen Dateien `persistence.xml` enthalten sein.

  Lösung: Sie müssen sicherstellen, dass der ObjectGrid-Name für jede Persistenzeinheit eindeutig ist, wenn das **`ObjectGridType`**-Attribut den Wert `EMBEDDED` hat.

- **Problem:** Die folgende Ausnahme wird angezeigt: `CacheException: REMOTE ObjectGrid [ObjectGridName] does not include required BackingMaps [mapName_1, mapName_2,...]`

  Wenn der ObjectGrid-Typ REMOTE verwendet wird und das abgerufene clientseitige ObjectGrid keine vollständigen Entitäts-BackingMaps für die Unterstützung des Caches der Persistenzeinheit hat, wird diese Ausnahme ausgelöst. Beispiel: Es sind fünf Entitätsklassen in der Konfiguration der Persistenzeinheit aufgelistet, aber das abgerufene ObjectGrid hat nur zwei BackingMaps. Diese Ausnahme wird auch dann ausgelöst, wenn das abgerufene ObjectGrid zehn BackingMaps enthält, aber eine der fünf erforderlichen Entitäts-BackingMaps nicht unter den zehn vorhandenen gefunden wird.

  Lösung: Stellen Sie sicher, dass Ihre BackingMap-Konfiguration den Persistenzeinheitencache unterstützt.

Fehlerbehebung bei der Verwaltung

Verwenden Sie die folgenden Informationen, um Fehler bei der Verwaltung, z. B. beim Stoppen und Starten von Servern, bei der Verwendung des Dienstprogramms **xscmd** usw., zu beheben.

Vorgehensweise

- **Problem:** Es fehlen Verwaltungsscripts im Verzeichnis *Profilstammverzeichnis*/bin einer Installation von WebSphere Application Server.

  Ursache: Wenn Sie die Installation aktualisieren, werden neue Scriptdateien nicht automatisch in den Profilen installiert.

  Lösung: Wenn Sie ein Script im Verzeichnis *Profilstammverzeichnis*/bin ausführen möchten, heben Sie die Erweiterung des Profils auf, und erweitern Sie das Profil dann erneut mit dem neuesten Release. Weitere Informationen finden Sie unter Erweiterung eines Profils über die Eingabeaufforderung aufheben und „Profile für WebSphere eXtreme Scale erstellen und erweitern" auf Seite 189.

- **Problem:** Wenn Sie einen **xscmd**-Befehl ausführen, wird die folgende Nachricht am Bildschirm ausgegeben:

  ```
  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
  ```

  Ursache: Es ist ein Fehler in der Verbindung mit dem Katalogserver aufgetreten.

  Lösung: Vergewissern Sie sich, dass Ihre Katalogserver aktiv und über das Netz verfügbar sind. Diese Nachricht kann auch ausgegeben werden, wenn Sie eine Katalogservicedomäne definiert haben, aber weniger als zwei Katalogserver aktiv sind. Die Umgebung ist erst verfügbar, wenn zwei Katalogserver gestartet sind.

Fehler in Konfigurationen mit mehreren Rechenzentren beheben

Verwenden Sie diese Informationen, um Fehler in Konfigurationen mit mehreren Rechenzentren, einschließlich Verbindungen zwischen Katalogservicedomänen, zu beheben.

Vorgehensweise

Problem: Es fehlen Daten in mindestens einer Katalogservicedomäne. Sie können beispielsweise den folgenden Befehl `xscmd -c establishLink` ausführen. Wenn Sie sich die Daten für jede verbundene Katalogservicedomäne ansehen, sehen die Daten anders aus, z. B. die Daten des Befehls `xscmd -c showMapSizes`.
Lösung: Sie können dieses Problem mit dem Befehl `xscmd -c showLinkedPrimaries` beheben. Dieser Befehl gibt jedes primäre Shard, einschließlich der Verbindungen mit fremden primären Shards.
Im beschriebenen Szenario stellen Sie beim Ausführen des Befehls `xscmd -c showLinkedPrimaries` möglicherweise fest, dass die primären Shards der ersten Katalogservicedomäne mit den primären Shards der zweiten Katalogservicedomäne verbunden sind, aber die zweite Katalogservicedomäne ist nicht mit der ersten Ka-

talogservicedomäne verbunden. Sie können den Befehl `xscmd -c establishLink` mit der zweiten Katalogservicedomäne als Quelle und der ersten Katalogservicedomäne als Ziel erneut ausführen.

Fehlerbehebung bei Loadern

Verwenden Sie die folgenden Optionen, um Probleme mit Ihren Datenbankladeprogrammen (Loader) zu beheben.

Vorgehensweise

- **Problem:** Wenn Sie einen OpenJPA-Loader mit DB2 in WebSphere Application Server verwenden, tritt eine Ausnahme wegen geschlossener Cursor auf.

  Die folgende Ausnahme in der org.apache.openjpa.persistence.PersistenceException-Protokolldatei stammt von DB2:

  ```
  [jcc][t4][10120][10898][3.57.82] Invalid operation: result set is closed.
  ```

  Lösung: Der Anwendungsserver konfiguriert die angepasste Eigenschaft "resultSetHoldability" standardmäßig mit dem Wert `2` (CLOSE_CURSORS_AT_COMMIT). Diese Eigenschaft bewirkt, dass DB2 seine Ergebnismenge bzw. seinen Cursor an Transaktionsgrenzen schließt. Zur Behebung der Ausnahme ändern Sie den Wert der angepassten Eigenschaft in 1 (HOLD_CURSORS_OVER_COMMIT). Definieren Sie die angepasste Eigenschaft "resultSetHoldability" über den folgenden Pfad in der Zelle von WebSphere Application Server: **Ressourcen** > **JDBC-Provider** > **DB2 Universal JDBC Driver Provider** > **Datenquellen** > *Name_der_Datenquelle* > **Angepasste Eigenschaften** > **Neu**.

- **Problem** DB2 zeigt eine Ausnahme an: `The current transaction has been rolled back because of a deadlock or timeout. Reason code "2".. SQLCODE=-911, SQLSTATE=40001, DRIVER=3.50.152`

  Diese Ausnahme tritt aufgrund eines Sperrenkonflikts ein, wenn Sie OpenJPA mit DB2 in WebSphere Application Server ausführen. Die Standardisolationsstufe für WebSphere Application Server ist "Repeatable Read (RR)" (wiederholbares Lesen), bei der lange Sperren bei DB2 angefordert werden.**Lösung:**

  Setzen Sie die Isolationsstufe auf "Read Committed" (Lesen mit COMMIT), um die Sperrenkonflikte zu reduzieren. Definieren Sie die angepasste Datenquelleneigenschaft "webSphereDefaultIsolationLevel", um die Isolationsstufe auf `2`(TRANSACTION_READ_COMMITTED) über den folgenden Pfad in der Zelle von WebSphere Application Server zu setzen: **Ressourcen** > **JDBC-Provider** > *JDBC-Provider* > **Datenquellen** > *Name_der_Datenquelle* > **Angepasste Eigenschaften** > **Neu**. Weitere Informationen zur angepassten Eigenschaft "webSphereDefaultIsolationLevel" und zu den Transaktionsisolationsstufen finden Sie unter Voraussetzungen für das Festlegen von Isolationsstufen für Datenzugriff.

- **Problem:** Wenn Sie die Funktion für vorheriges Laden (Preload) von JPALoader oder JPAEntityLoader verwenden, wird die folgende CWOBJ1511-Nachricht nicht für die Partition in einem Container-Server angezeigt: `CWOBJ1511I: GRID_NAME:MAPSET_NAME:PARTITION_ID (primär) ist für Business bereit..`

  Stattdessen tritt eine Ausnahme des Typs "TargetNotAvailableException" im Container-Server ein, der die Partition aktiviert, die mit der Eigenschaft "preloadPartition" angegeben wird.

  Lösung: Setzen Sie das Attribut "preloadMode" auf `true`, wenn Sie einen JPALoader oder JPAEntityLoader für das vorherige Laden von Daten in die Map verwenden. Wenn die Eigenschaft "preloadPartition" von JPALoader und JPAEntityLoader auf einen Wert zwischen `0` und `Gesamtpartitionsanzahl - 1` gesetzt ist, versuchen JPALoader und JPAEntityLoader, die Daten vorher aus der Back-

End-Datenbank in die Map zu laden. Das folgende Code-Snippet veranschaulicht, wie das Attribut "preloadMode" so gesetzt wird, dass das asynchrone vorherige Laden aktiviert wird:

```
BackingMap bm = og.defineMap( "map1" );
bm.setPreloadMode( true );
```

Sie können das Attribut "preloadMode" auch mithilfe einer XML-Datei definieren, wie im folgenden Beispiel veranschaulicht wird:

```
<backingMap name="map1" preloadMode="true" pluginCollectionRef="map1"
 lockStrategy="OPTIMISTIC" />
```

Fehler in der XML-Konfiguration beheben

Wenn Sie eXtreme Scale konfigurieren, kann ein unerwartetes Verhalten bei Ihren XML-Dateien auftreten. In den folgenden Abschnitten sind Probleme, die auftreten können, und deren Lösungen beschrieben.

Vorgehensweise

- **Problem:** Ihre Implementierungsrichtlinie und die ObjectGrid-XML-Dateien müssen übereinstimmen.

  Die XML-Datei für die Implementierungsrichtlinien und die ObjectGrid-XML-Datei müssen übereinstimmen. Es treten Fehler auf, wenn die ObjectGrid-Namen und Map-Namen in der XML-Datei für die Implementierungsrichtlinien und in der ObjectGrid-XML-Datei nicht identisch sind.

  Wenn die BackingMap-Liste in einer ObjectGrid-XML-Datei nicht mit der Liste der Map-Referenzen in der XML-Datei für Implementierungsrichtlinien übereinstimmt, tritt ein Fehler im Katalogserver auf.

  Die folgende ObjectGrid-XML-Datei und die folgende XML-Datei für Implementierungsrichtlinien werden beispielsweise zum Starten eines Containerprozesses verwendet. Die Datei für die Implementierungsrichtlinien enthält mehr Map-Referenzen, als in der ObjectGrid-XML-Datei aufgelistet sind.

  Beispiel für eine ungültige Datei "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="accounting">
               <backingMap name="payroll" readOnly="false" />
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

  Beispiel für eine ungültige Datei "deploymentPolicy.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="4" minSyncReplicas="1"
     maxSyncReplicas="2" maxAsyncReplicas="1">
            <map ref="payroll"/>
            <map ref="ledger"/>
        </mapSet>
    </objectgridDeployment>
</deploymentPolicy>
```

  Nachrichten: Es wird eine Fehlernachricht in der Datei `SystemOut.log` aufgezeichnet, wenn die Datei für die Implementierungsrichtlinien mit der ObjectGrid-XML-Datei nicht kompatibel ist. Für das vorherige Beispiel wird die folgende Nachricht aufgezeichnet:

```
CWOBJ3179E: Die Map ledger, die im MapSet mapSet1 der Implementierungsdeskriptordatei von
ObjectGrid accounting referenziert wird, referenziert keine gültige BackingMap aus der ObjectGrid-XML.
```

Wenn in der Implementierungsrichtlinie Map-Referenzen auf BackingMaps fehlen, die in der ObjectGrid-XML-Datei aufgelistet sind, wird eine Fehlernachricht in der Datei SystemOut.log aufgezeichnet. Beispiel:

```
CWOBJ3178E: Die Map ledger in ObjectGrid accounting, die in der ObjectGrid-XML-Datei
referenziert wird, wurde nicht in der Implementierungsdeskriptordatei gefunden.
```

Lösung: Bestimmen Sie, welche Datei die richtige Liste enthält, und ändern Sie den relevanten Code entsprechend.

- **Problem:** Ungültige ObjectGrid-Namen in XML-Dateien können ebenfalls Fehler verursachen.

  Der Name des ObjectGrids wird in der ObjectGrid-XML-Datei und in der XML-Dateif ür Implementierungsrichtlinien referenziert.

  Nachricht: Es tritt eine Ausnahme des Typs "ObjectGridException" ein, die durch eine Ausnahme des Typs "IncompatibleDeploymentPolicyException" verursacht wird. Es folgt ein Beispiel:

  Caused by: com.ibm.websphere.objectgrid.IncompatibleDeploymentPolicyException: The objectgridDeployment with objectGridName "accountin" does not have a corresponding objectGrid in the ObjectGrid XML.

  Die ObjectGrid-XML-Datei ist die Masterliste mit ObjectGrid-Namen. Wenn eine Implementierungsrichtlinie einen ObjectGrid-Namen enthält, der nicht in der ObjectGrid-XML-Datei enthalten ist, tritt ein Fehler auf.

  Lösung: Überprüfen Sie die Rechtschreibung des ObjectGrid-Namens. Entfernen Sie alle zusätzlichen Namen, bzw. fügen Sie fehlende ObjectGrid-Namen in der ObjectGrid-XML-Datei bzw. in der XML-Datei für Implementierungsrichtlinien hinzu. In der Beispielnachricht ist der ObjectGrid-Name falsch geschrieben: "accountin" anstatt "accounting".

- **Problem:** Einigen Attributen in der XML-Datei können nur bestimmte Werte zugeordnet werden. Diese Attribute haben gültige Werte, die nach Schema aufgelistet sind. Die folgende Liste enthält einige dieser Attribute:
  - Attribut "authorizationMechanism" im Element "objectGrid"
  - Attribut "copyMode" im Element "backingMap"
  - Attribut "lockStrategy" im Element "backingMap"
  - Attribut "ttlEvictorType" im Element "backingMap"
  - Attribut "type" im Element "property"
  - Attribut "initialState" im Element "objectGrid"
  - Attribut "evictionTriggers" im Element "backingMap"

  Wenn einem dieser Attribute ein ungültiger Wert zugeordnet wird, scheitert die XML-Validierung. In der folgenden XML-Beispieldatei wird der ungültige Wert INVALID_COPY_MODE verwendet:

  Beispiel für 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>
```

  Die folgende Nachricht wird im Protokoll aufgezeichnet:

```
CWOBJ2403E: Die XML-Datei ist ungültig. Es wurde ein Fehler in
< null > in Zeile 5 gefunden. Die Fehlernachricht ist "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]'".
It must be a value from the enumeration.
```

- **Problem:** Fehlende oder ungültige Attribute oder Tags in einer XML-Datei verursachen Fehler. Im folgenden Beispiel fehlt in der ObjectGrid-XML-Datei beispielsweise das abschließende Tag </objectGrid >:

  Fehlende Attribute - Beispiel-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>
  ```

  Nachricht:

  ```
  CWOBJ2403E: Die XML-Datei ist ungültig. Es wurde ein Fehler in
  < null > in Zeile 7 gefunden. Die Fehlernachricht ist "The
  end-tag for element type "objectGrid" must end with a '>' delimiter."
  ```

  Es wird eine Ausnahme des Typs "ObjectGridException" für die ungültige XML-Datei ausgelöst, die den Namen der XML-Datei enthält.

  Lösung: Stellen Sie sicher, dass die erforderlichen Tags und Attribute im richtigen Format in den XML-Dateien enthalten sind.

- **Problem:** Wenn in einer XML-Datei ein ungültiges Syntaxformat verwendet wird oder Syntax fehlt, wird eine Nachricht CWOBJ2403E im Protokoll angezeigt. Die folgende Nachricht wird beispielsweise angezeigt, wenn ein Anführungszeichen in einem der XML-Attribute fehlt:

  ```
  CWOBJ2403E: Die XML-Datei ist ungültig. Es wurde ein Fehler in < null > in
  Zeile 7 gefunden. Die Fehlernachricht ist "Open quote is expected for attribute
  "maxSyncReplicas" associated with an element type "mapSet"".
  ```

  Außerdem wird eine Ausnahme des Typs "ObjectGridException" für die ungültige XML-Datei ausgelöst.

  Lösung: Es können verschiedene Lösungen für einen bestimmten XML-Syntaxfehler verwendet werden. Die Syntax für das XML-Script können Sie der entsprechenden Dokumentation entnehmen.

- **Problem:** Die Referenzierung einer nicht vorhandenen Plug-in-Sammlung führt zu einer ungültigen XML-Datei. Wenn Sie beispielsweise XML für die Definition von BackingMap-Plug-ins verwenden, muss das Attribut "pluginCollectionRef" des Elements "backingMap" eine backingMapPluginCollection referenzieren. Das Attribut "pluginCollectionRef" muss den backingMapPluginCollection-Elementen entsprechen.

  Nachricht:

  Wenn das Attribut "pluginCollectionRef" mit keiner der IDs der backingMapPluginConfiguration-Elemente übereinstimmt, wird die folgende Nachricht oder eine ähnliche im Protokoll angezeigt:

  ```
  [7/14/05 14:02:01:971 CDT] 686c060e XmlErrorHandl E CWOBJ9002E:
  This is an English only Error message: Invalid XML file. Line: 14; URI:
  null; Message: Key 'pluginCollectionRef' with
  value 'bookPlugins' not found for identity constraint of
  element 'objectGridConfig'.
  ```

  Die folgende XML-Datei wird verwendet, um den Fehler zu produzieren. Beachten Sie, dass das Attribut "pluginCollectionRef" für den Namen der BackingMap "book" auf "bookPlugins" gesetzt ist und dass die einzige backingMapPluginCollection die ID "collection1" hat:

  Referenzierung eines nicht vorhandenen Attributs - XML-Beispiel

  ```
  <?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>
```

Lösung:

Zum Beheben des Problems müssen Sie sicherstellen, dass der Wert jedes pluginCollectionRef-Attributs mit der ID eines der backingMapPluginCollection-Elemente übereinstimmt. Ändern Sie einfach den Wert des Attributs "pluginCollectionRef" in "collection1", damit dieser Fehler nicht mehr auftritt. Alternativ können Sie die ID des vorhandenen backingMapPluginCollection-Elements so ändern, dass sie dem Wert des Attributs "pluginCollectionRef" entspricht, oder ein zusätzliches Element "backingMapPluginCollection" mit einer ID hinzufügen, die dem Wert des Attributs "pluginCollectionRef" entspricht.

- **Problem:** IBM Software Development Kit (SDK) Version 5 enthält Implementierungen einiger JAXP-Funktionen (Java API for XML Processing), mit denen Sie die XML anhand des Schemas validieren können. Wenn Sie ein SDK verwenden, das diese Implementierungen nicht enthält, können die Validierungsversuche scheitern.

  Wenn Sie versuchen, die XML mit einem SDK zu validieren, das die erforderlichen Implementierungen nicht enthält, enthält das Protokoll den folgenden Fehler:

```
XmlConfigBuild XML validation is enabled
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: No attributes are implemented
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)...
```

  Das verwendete SDK enthält nicht die Implementierung der JAXP-Funktion, die erforderlich ist, um XML-Dateien anhand des Schemas zu validieren.

  Lösung: Wenn Sie XML mit einem SDK validieren möchten, das keine JAXP-Implementierung enthält, laden Sie Apache Xerces herunter, und fügen Sie die JAR-Dateien von Apache Xerces in den Klassenpfad ein. Sie können dieses Problem vermeiden, indem Sie Apache Xerces herunterladen und die JAR-Dateien in den Klassenpfad einfügen. Anschließend können Sie die XML-Datei erfolgreich validieren.

Fehlerbehebung bei der Sicherheit

Verwenden Sie diese Informationen, um Probleme mit Ihrer Sicherheitskonfiguration zu beheben.

Vorgehensweise

- **Problem:** Die Clientseite der Verbindung setzt Secure Sockets Layer (SSL) mit dem Wert `SSL-Required` für die Einstellung "transportType" voraus. Die Serverseite der Verbindung unterstützt jedoch kein SSL und weist den Wert `TCP/IP` für die Einstellung "transportType" auf. Deshalb wird die folgende Ausnahme mit einer anderen Ausnahme in den Protokolldateien verkettet:

```
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)
```

Die Adresse in dieser Ausnahme kann ein Katalogserver, ein Container-Server oder ein Client sein.

Lösung: Eine Tabelle mit den gültigen Sicherheitskonfigurationen zwischen Clients und Servern finden Sie unter „Sichere Transporttypen konfigurieren" auf Seite 524.

- Wenn ein Agent verwendet wird, sendet der Client den Agentenaufruf an den Server, und der Server sendet die Antwort zur Bestätigung des Agentenaufrufs an den Client zurück. Wenn der Agent die Verarbeitung beendet, leitet der Server einen Verbindungsaufbau ein, um die Agentenergebnisse zu senden. Damit wird der Container-Server aus Verbindungssicht zu einem Client. Wenn TLS/SSL konfiguriert ist, stellen Sie deshalb sicher, dass das öffentliche Zertifikat des Clients in den Truststore des Servers importiert wird.

IBM Support Assistant für WebSphere eXtreme Scale

Sie können IBM Support Assistant verwenden, um Daten zu erfassen, Symptome zu analysieren und auf Produktinformationen zuzugreifen.

IBM Support Assistant Lite

IBM Support Assistant Lite for WebSphere eXtreme Scale unterstützt die automatische Datenerfassung und Symptomanalyse für Problembestimmungsszenarien.

Mit IBM Support Assistant Lite reduziert sich die Zeit, die erforderlich ist, um ein Problem mit den entsprechend definierten Tracestufen für Zuverlässigkeit, Verfügbarkeit und Servicefreundlichkeit Tracestufen werden automatisch vom Tool gesetzt) zu reproduzieren, um die Fehlerbestimmung zu optimieren. Wenn Sie zusätzliche Unterstützung benötigen, verringert IBM Support Assistant Lite auch den erforderlichen Aufwand für das Senden der entsprechenden Protokollinformationen an die IBM Unterstützungsfunktion.

IBM Support Assistant Lite ist in jeder Installation von WebSphere eXtreme Scale Version 7.1.0 enthalten.

IBM Support Assistant

IBM® Support Assistant (ISA) ermöglicht Ihnen den schnellen Zugriff auf Produkt-, Schulungs- und Unterstützungsressourcen, die Ihnen helfen können, eigenständig Antworten auf Fragen zu finden und Probleme mit IBM Softwareprodukten zu finden, ohne sich an die IBM Unterstützungsfunktion wenden zu müssen. Es werden verschiedene produktspezifische Plug-ins bereitstellt, mit denen Sie IBM Support Assistant für Ihre installierten Produkte anpassen können. IBM Support Assistant kann auch Systemdaten, Protokolldateien und andere Informationen erfassen, die der IBM Unterstützungsfunktion bei der Bestimmung der Ursache eines bestimmten Problems helfen.

IBM Support Assistant ist ein Dienstprogramm, das für die Installation auf der Workstation und nicht für die direkte Installation auf dem System mit dem eXtreme-Scale-Server bestimmt ist. Der Speicher- und Ressourcenbedarf für Assistant kann sich nachteilig auf die Leistung des Systems mit dem eXtreme-Scale-Server

auswirken. Die enthaltenen portierbaren Diagnosekomponenten sind so konzipiert, dass sie nur minimale Auswirkungen auf den normalen Betrieb eines Servers haben.

Sie können IBM Support Assistant für folgende Unterstützungszwecke einsetzen:

- Für die Suche von Informationen in Wissens- und Informationsquellen von IBM und anderen Anbietern zu mehreren IBM Produkten, um Antworten auf eine Frage zu finden oder um ein Problem zu lösen.
- Für die Suche zusätzlicher Informationen in produktspezifischen Webressourcen, einschließlich Produkt- und Unterstützungs-Homepages, Kunden-Newsgroups und -Foren, Wissens- und Schulungsressourcen sowie Informationen zur Fehlerbehebung und zu häufig gestellten Fragen.
- Zur Erweiterung Ihrer Möglichkeiten für die Diagnose produktspezifischer Probleme mit den in Support Assistant bereitgestellten zielspezifischen Diagnosetools.
- Für eine vereinfachte Erfassung von Diagnosedaten, die Ihnen und IBM helfen, Probleme zu beheben (Erfassung allgemeiner oder produkt- bzw. symptomspezifischer Daten).
- Zur Unterstützung beim Melden von Problemvorfällen an die IBM Unterstützungsfunktion über eine angepasste Onlineschnittstelle, einschließlich der Möglichkeit, zuvor referenzierte Diagnosedaten oder andere Informationen zu neuen oder vorhandenen Vorfällen anzuhängen.

Und dann können Sie noch das integrierte Updater-Tool verwenden, um Unterstützung für zusätzliche Softwareprodukte und Funktionen zu erhalten, sobald diese verfügbar sind. Zum Einrichten von IBM Support Assistant für WebSphere eXtreme Scale installieren Sie zuerst IBM Support Assistant unter Verwendung der Dateien, die in dem von der Webseite "IBM Support Overview" unter http://www-947.ibm.com/support/entry/portal/Overview/Software/Other_Software/IBM_Support_Assistant heruntergeladenen Image bereitgestellt werden. Anschließend verwenden Sie IBM Support Assistant, um Produktaktualisierungen zu suchen und zu installieren. Sie können auch neue Plug-ins installieren, die für andere IBM Software in Ihrer Umgebung verfügbar sind. Weitere Informationen und die aktuelle Version von IBM Support Assistant sind auf der Webseite von IBM Support Assistant unter http://www.ibm.com/software/support/isa/ verfügbar.

Bemerkungen

Hinweise auf IBM Produkte, Programme und Services in dieser Veröffentlichung bedeuten nicht, dass IBM diese in allen Ländern, in denen IBM vertreten ist, anbietet. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Fremdservices in Verbindung mit Fremdprodukten und Fremdservices liegt beim Kunden, soweit solche Verbindungen nicht ausdrücklich von IBM bestätigt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Requests

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Marken

Folgende Namen sind Marken der IBM Corporation in den USA und/oder anderen Ländern:

- AIX
- CICS
- Cloudscape
- DB2
- Domino
- IBM
- Lotus
- RACF
- Redbooks
- Tivoli
- WebSphere
- z/OS

Java und alle auf Java basierenden Marken und Logos sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

LINUX ist eine Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

Index

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z