IBM WebSphere eXtreme Scale Versión 7.1.1
Versión 7 Release 1

Guía de administración

21 de noviembre de 2011

Esta edición se aplica a la versión 7, release 1, modificación 1 de WebSphere eXtreme Scale y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Contenido

Figuras

Tablas

Acerca de la *Guía de administración*

El conjunto de documentación de WebSphere eXtreme Scale incluye tres volúmenes que proporcionan la información necesaria para utilizar, programar y administrar el producto WebSphere eXtreme Scale.

Biblioteca de WebSphere eXtreme Scale

La biblioteca de WebSphere eXtreme Scale contiene las siguientes publicaciones:

- El *Visión general del producto* contiene una vista de nivel superior de los conceptos de WebSphere eXtreme Scale, incluidos casos de ejemplo y guías de aprendizaje.
- *Guía de instalación* describe cómo instalar topologías comunes de WebSphere eXtreme Scale.
- La *Guía de administración* contiene la información necesaria para los administradores del sistema, incluido cómo planificar despliegues de aplicaciones, planificar la capacidad, instalar y configurar el producto, iniciar y detener servidores, supervisar el entorno y proteger el entorno.
- La *Guía de programación* contiene información dirigida a los desarrolladores de aplicaciones que indica cómo desarrollar aplicaciones para WebSphere eXtreme Scale utilizando la información de API incluida.

Para descargar las publicaciones, vaya a la página de la biblioteca WebSphere eXtreme Scale.

También puede acceder a la misma información en esta biblioteca en el Information Center de WebSphere eXtreme Scale Versión 7.1.1.

Utilización fuera de línea de los manuales

Todos los manuales de la biblioteca de WebSphere eXtreme Scale contienen enlaces al Information Center, con el siguiente URL raíz: `http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r1m1`. Estos enlaces le llevan directamente a la información relacionada. Sin embargo, si está trabajando fuera de línea y se encuentra con uno de estos enlaces, puede buscar el título del enlace en los otros manuales de la biblioteca. La documentación de la API, el glosario y los mensajes de referencia no están disponibles en los manuales en formato PDF.

Quién debe utilizar esta publicación

Esta publicación está especialmente indicada para administradores del sistema, administradores de seguridad y operadores del sistema.

Cómo obtener actualizaciones de esta publicación

Puede obtener actualizaciones para esta publicación descargando la versión más reciente desde la página de la biblioteca de WebSphere eXtreme Scale.

Envío de comentarios

Póngase en contacto con el equipo de documentación. ¿Ha encontrado lo que necesita? ¿Ha sido la información precisa y completa? Envíe sus comentarios sobre

esta documentación mediante correo electrónico a wasdoc@us.ibm.com.

Capítulo 1. Cómo empezar

Después de instalar el producto, puede utilizar el ejemplo de iniciación para probar la instalación y utilizar el producto por primera vez.

Guía de aprendizaje: Cómo empezar con WebSphere eXtreme Scale

Tras instalar WebSphere eXtreme Scale en un entorno autónomo, puede utilizar la aplicación de ejemplo de iniciación como una introducción sencilla a su capacidad como una cuadrícula de datos en memoria.

Objetivos del aprendizaje

- Obtener información sobre el archivo XML de descriptor de ObjectGrid y los archivos XML de descriptor de política de despliegue que utiliza para configurar el entorno
- Iniciar servidores de catálogo y contenedor mediante los archivos de configuración
- Obtener información sobre el desarrollo de una aplicación cliente
- Ejecutar la aplicación cliente para insertar datos en la cuadrícula de datos
- Supervisar las cuadrículas de datos con la consola web

Tiempo necesario

60 minutos

Lección 1 de la guía de aprendizaje de iniciación: Definición de cuadrículas de datos con archivos de configuración

Para configurar cuadrículas de datos simples, utilice los archivos `objectgrid.xml` y `deployment.xml` que se proporciona en el ejemplo de iniciación.

El ejemplo utiliza los archivos `objectgrid.xml` y `deployment.xml` que están en el directorio *raíz_intal_wxs*`/ObjectGrid/gettingstarted/xml`. Estos archivos se pasan a los mandatos de inicio para iniciar los servidores de contenedor y un servidor de catálogo. El archivo `objectgrid.xml` es el archivo XML de descriptor de ObjectGrid. El archivo `deployment.xml` es el archivo XML de política de descriptor de ObjectGrid. Estos archivos definen conjuntamente una topología distribuida.

Archivo XML de descriptor ObjectGrid

Se utiliza un archivo XML de descriptor de ObjectGrid para definir la estructura del ObjectGrid que es utilizado por la aplicación. Incluye una lista de configuraciones de correlación de respaldo. Estas correlaciones de respaldo almacenan los datos de memoria caché. El ejemplo siguiente es un archivo `objectgrid.xml` de ejemplo. Las primeras líneas del archivo incluyen la cabecera necesaria para cada archivo XML de ObjectGrid. Este archivo de ejemplo define el ObjectGrid `Grid` con las correlaciones de respaldo `Map1` y `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>
```

Archivo XML de descriptor de política de despliegue

Se proporciona un archivo XML de descriptor de política de despliegue al servidor de contenedor durante el inicio. Se debe utilizar una política de despliegue con un archivo XML de ObjectGrid y debe ser compatible con el XML de ObjectGrid que se utiliza con la misma. Para cada elemento objectgridDeployment de la política de despliegue, debe tener un elemento ObjectGrid correspondiente en el archivo XML de ObjectGrid. Los elementos backingMap que están definidos dentro del elemento objectgridDeployment deben ser coherentes con las backingMaps que se encuentran en el XML de ObjectGrid. Debe hacerse referencia a cada backingMap dentro de únicamente un mapSet.

El archivo XML de descriptor de política de despliegue intenta emparejarse con el XML correspondiente de ObjectGrid, el archivo `objectgrid.xml`. En el siguiente ejemplo, las primeras líneas del archivo `deployment.xml` incluyen la cabecera necesaria para cada archivo XML de política de despliegue. El archivo define el elemento objectgridDeployment para el ObjectGrid Grid que está definido en el archivo `objectgrid.xml`. Ambas BackingMaps, Map1 y Map2, que están definidas dentro del ObjectGrid Grid se incluyen en el mapSet mapSet que tiene los atributos numberOfPartitions, minSyncReplicas y maxSyncReplicas configurados.

```
<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>
```

El atributo numberOfPartitions del elemento mapSet especifica el número de particiones para el mapSet. Es un atributo opcional y el valor predeterminado es 1. El número debe ser adecuado para la capacidad prevista de la cuadrícula de datos.

El atributo minSyncReplicas de mapSet especifica el número mínimo de réplicas síncronas para cada partición del mapSet. Se trata de un atributo opcional y el valor predeterminado es 0. El primario y la réplica no se colocan hasta que el dominio pueda soportar el número mínimo de réplicas síncronas. Para dar soporte al valor minSyncReplicas, es necesario un contenedor más que el valor de minSyncReplicas. Si el número de réplicas síncronas cae por debajo del valor de minSyncReplicas, ya no se permiten transacciones de grabación para esa partición.

El atributo maxSyncReplicas de mapSet especifica el número máximo de réplicas síncronas para cada partición del mapSet. Se trata de un atributo opcional y el valor predeterminado es 0. No se coloca ninguna otra réplica síncrona para una

partición después de que un dominio alcance este número de réplicas síncronas para dicha partición específica. La adición de contenedores que puedan dar soporte a este ObjectGrid puede comportar un aumento en el número de réplicas síncronas si todavía no se ha alcanzado el valor de maxSyncReplicas. El ejemplo de valor maxSyncReplicas establecido en 1 significa que el dominio colocará, como mínimo, una réplica síncrona. Si inicia más de una instancia de servidor de contenedor, sólo habrá una réplica síncrona colocada en una de las instancias del servidor de contenedor.

Punto de comprobación de la lección

En esta lección, ha aprendido lo siguiente:

- Cómo definir correlaciones que almacenan datos en el archivo XML de descriptor de ObjectGrid.
- Cómo utilizar el archivo XML de descriptor de despliegue para definir el número de particiones y réplicas para la cuadrícula de datos.

Lección 2 de la guía de aprendizaje de iniciación: Creación de una aplicación cliente

Para insertar, suprimir, actualizar y recuperar datos de la cuadrícula de datos, debe escribir una aplicación de cliente. El ejemplo de iniciación incluye una aplicación cliente que puede utilizar para aprender a crear su propia aplicación cliente.

El archivo `Client.java` del directorio *raíz_intal_wxs*`/ObjectGrid/gettingstarted/client/src/` es el programa cliente que muestra cómo conectarse a un servidor de catálogo, obtener la instancia de ObjectGrid y utilizar la API ObjectMap. La API ObjectMap almacena datos como pares de clave-valor y es ideal para almacenar en memoria caché objetos que no tienen relaciones implicadas.

Si necesita almacenar en memoria caché objetos que tienen relaciones, utilice la API EntityManager.

1. Conéctese al servicio de catálogo obteniendo una instancia de ClientClusterContext.

   Para conectarse al servidor de catálogo, utilice el método connect de la API ObjectGridManager. El método connect que se utiliza sólo necesita el punto final de servidor de catálogo en el formato *`nombre_de_host:puerto`*. Pude indicar varios puntos finales de servidor de catálogo separando la lista de valores de *`nombre_de_host:puerto`* con comas. El fragmento de código siguiente muestra cómo conectar con un servidor de catálogo y obtener una instancia de ClientClusterContext:

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

   Si las conexiones con los servidores de catálogo son satisfactorias, el método connect devuelve una instancia ClientClusterContext. La instancia de ClientClusterContext es necesaria para obtener el ObjectGrid de la API ObjectGridManager.

2. Obtenga una instancia de ObjectGrid.

   Para obtener una instancia de ObjectGrid, utilice el método getObjectGrid de la API ObjectGridManager. El método getObjectGrid requiere tanto la instancia de ClientClusterContext, como el nombre de la instancia de cuadrícula de datos. La instancia de ClientClusterContext se obtiene durante la conexión con el servidor de catálogo. El nombre de la instancia de ObjectGrid es `Grid` (cuadrícula) que se especifica en el archivo `objectgrid.xml`. El siguiente

fragmento de código demuestra cómo obtener la cuadrícula de datos llamando al método getObjectGrid de la interfaz de programación de aplicaciones ObjectGridManager.

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

3. Obtenga una instancia de Session.

   Puede obtener una Session desde la instancia de ObjectGrid obtenida. Es necesaria una instancia de Session para obtener la instancia de ObjectMap, y realizar la demarcación de la transacción. En el siguiente fragmento de código se muestra cómo obtener una instancia de Session llamando al método getSession de la API ObjectGrid.

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

4. Obtenga una instancia de ObjectMap.

   Después de obtener una sesión, puede obtener una instancia de ObjectMap desde una sesión llamando al método getMap de la interfaz de programación de aplicaciones de la sesión en cuestión. Debe pasar el nombre de la correlación como correlación al métodogetMap para obtener la instancia de ObjectMap. El fragmento de código siguiente muestra cómo obtener ObjectMap llamando al método getMap de la API de sesión.

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

5. Utilice los métodos ObjectMap.

   Después de obtener una instancia de ObjectMap, puede utilizar la API de ObjectMap. Recuerde que la interfaz de ObjectMap es una correlación transaccional y requiere la demarcación de transacción utilizando los métodos begin y commit de la API Session. Si no hay ninguna demarcación de transacción explícita en la aplicación, las operaciones de ObjectMap se ejecutan con transacciones de confirmación automática.

   El siguiente fragmento de código demuestra cómo utilizar la API ObjectMap con una transacción de confirmación automática.

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

   El siguiente fragmento de código demuestra cómo utilizar la API ObjectMap con la demarcación de transacción explícita.

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

Punto de comprobación de la lección

En esta lección, ha aprendido cómo crear una única aplicación cliente para realizar operaciones de cuadrícula de datos.

Lección 3 de la guía de aprendizaje de iniciación: Ejecución de la aplicación cliente de ejemplo de iniciación

Utilice los pasos siguientes para iniciar la primera cuadrícula de datos y ejecutar un cliente para interactuar con la cuadrícula de datos.

Los otros scripts llaman al script `env.sh|bat` para establecer las variables de entorno necesarias. Normalmente, no necesita cambiar este script.

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

Para ejecutar la aplicación, en primer lugar inicie el proceso de servicio de catálogo. El servicio de catálogo es el centro de control de la cuadrícula de datos.

Realiza un seguimiento de las ubicaciones de servidores de contenedor y controla la colocación de los datos para alojar servidores de contenedor. Después de que se inicie el servicio de catálogo, puede iniciar los servidores de contenedor, que almacenan los datos de la aplicación para la cuadrícula de datos. Para almacenar varias copias de los datos, puede iniciar varios servidores de contenedor. Cuando se han iniciado todos los servidores, puede ejecutar la aplicación cliente para insertar, actualizar, eliminar y obtener datos de la cuadrícula de datos.

1. Abra una ventana de sesión de terminal o de línea de mandatos.
2. Utilice el siguiente mandato para ir hasta el directorio `gettingstarted`:

   `cd` *`raíz_intal_wxs`*`/ObjectGrid/gettingstarted`

   Sustituya *raíz_intal_wxs* por la vía de acceso al directorio raíz de instalación de eXtreme Scale o por la vía de acceso de archivo raíz de *raíz_intal_wxs* de la versión de prueba extraída de eXtreme Scale.
3. Ejecute el siguiente script para iniciar un proceso de servicio de catálogo en el sistema principal local:

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

   El proceso de servicio de catálogo se ejecuta en la ventana actual de terminal.

   También puede iniciar el servicio de catálogo con el mandato **startOgServer**. Ejecute **startOgServer** en el directorio *raíz_intal_wxs*`/ObjectGrid/bin`:

   - UNIX Linux `startOgServer.sh cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`
   - Windows `startOgServer.bat cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`
4. Abra otra ventana de sesión de terminal o de línea de mandatos, y ejecute el siguiente mandato para iniciar una instancia de servidor de contenedor:

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

   El servidor de contenedor se ejecuta en la ventana actual del terminal. Puede repetir este paso con un nombre de servidor distinto si desea iniciar más instancias de servidor de contenedor para dar soporte a la réplica.

   También puede iniciar servidores de contenedor con el mandato **startOgServer**. Ejecute **startOgServer** en el directorio *raíz_intal_wxs*`/ObjectGrid/bin`:

   - 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. Abra otra ventana de sesión de terminal o de línea de mandatos para ejecutar los mandatos de cliente.

   El script `runclient.sh|bat` ejecute el cliente de CRUD sencillo e inicia la operación determinada. El script `runclient.sh|bat` se ejecuta con los parámetros siguientes:

   - UNIX Linux `./runclient.sh` *`mandato valor1 valor2`*
   - Windows `runclient.bat` *`valor valor1 valor2`*

   Para *mandato*, utilice una de las siguientes opciones:

- Especifique `i` para insertar *valor2* en la cuadrícula de datos con la clave *valor1*
- Especifique como `u` para actualizar el objeto con clave de *valor1* a *valor2*
- Especifique como `d` para suprimir el objeto con clave por *valor1*
- Especifique como `g` para recuperar y visualizar el objeto con clave por *valor1*

a. Añada datos a la cuadrícula de datos:
   - UNIX Linux `./runclient.sh i key1 helloWorld`
   - Windows `runclient.bat i key1 helloWorld`

b. Busque y visualice el valor:
   - UNIX Linux `./runclient.sh g key1`
   - Windows `runclient.bat g key1`

c. Actualice el valor:
   - UNIX Linux `./runclient.sh u key1 goodbyeWorld`
   - Windows `runclient.bat u key1 goodbyeWorld`

d. Suprima el valor:
   - UNIX Linux `./runclient.sh d key1`
   - Windows `runclient.bat d key1`

Punto de comprobación de la lección

En esta lección, ha aprendido lo siguiente:

- Cómo iniciar servidores de catálogo y servidores de contenedor
- Cómo ejecutar la aplicación cliente de ejemplo

Lección 4 de la guía de aprendizaje de iniciación: Supervisar el entorno

Puede utilizar el programa de utilidad **xscmd** y las herramientas de la consola web para supervisar el entorno de la cuadrícula de datos.

Supervisión con la consola web

Con la consola web, puede representar gráficos de las estadísticas actuales e históricas. Esta consola proporciona algunos gráficos configurados previamente para visiones generales de alto nivel y tiene una página de informes personalizados que puede utilizar para crear gráficos de las estadísticas disponibles. Puede utilizar las posibilidades de representación gráfica en la consola de supervisión de WebSphere eXtreme Scale para ver el rendimiento general de las cuadrículas de datos del entorno.

Instalar la consola web como una característica opcional cuando se ejecuta el asistente de instalación.

1. Inicie el servidor de consola. El script **startConsoleServer.bat|sh** para iniciar el servidor de la consola se encuentra en el directorio *raíz_intal_wxs*/`ObjectGrid/bin` de la instalación.
2. Inicie la sesión en la consola.
   a. Desde el navegador web, vaya a `https://su.host.consola:7443`, sustituyendo `su.host.consola` por el nombre de host del servidor en el que ha instalado la consola.
   b. Inicie la sesión en la consola.
      - **ID de usuario:** `admin`

- **Contraseña:** `admin`

Se visualiza la página de bienvenida de la consola.

3. Edite la configuración de la consola. Pulse **Valores** > **Configuración** para revisar la configuración de la consola. La configuración de la consola incluye información como:
   - Serie de rastreo del cliente WebSphere eXtreme Scale, como `*=all=disabled`
   - Nombre y contraseña del administrador
   - Dirección de correo electrónico del administrador
4. Establezca y mantenga las conexiones a los servidores de catálogo que desea supervisar. Repita los pasos siguientes para añadir cada servidor de catálogo a la configuración.
   a. Pulse **Valores** > **Servidores de catálogo eXtreme Scale**.
   b. Añada un servidor de catálogo nuevo.

      1) Pulse el icono de añadir ( ) para registrar un servidor de catálogo existente.
      2) Proporcione información, como el nombre de host y el puerto de escucha. Consulte “Planificación de puertos de red” en la página 64 para obtener más información sobre la configuración de puerto y los valores predeterminados.
      3) Pulse **Aceptar**.
      4) Verifique que el servidor de catálogo se ha añadido al árbol de navegación.
5. Consulte el estado de conexión. El campo **Dominio actual** indica el nombre del dominio de servicio de catálogo que se utiliza actualmente para visualizar información en la consola web. El estado de conexión se visualiza junto al nombre del dominio de servicio de catálogo.
6. Visualice estadísticas para las cuadrículas de datos y los servidores, o cree un informe personalizado.

Supervisión con el programa de utilidad xscmd

1. Abra una ventana de línea de mandatos.En la línea de mandatos, establezca las variables de entorno correspondientes.
   a. Establezca la variable de entorno *CLIENT_AUTH_LIB*:
      - Windows `set CLIENT_AUTH_LIB=<`*`vía_acceso_a_JAR_o_clases_seguridad`*`>`
      - UNIX `set CLIENT_AUTH_LIB=<`*`vía_acceso_a_JAR_o_clases_seguridad`*`>`
        `export CLIENT_AUTH_LIB`
2. Vaya al directorio *inicio_wxs*`/bin`.

   `cd` *`inicio_wxs`*`/bin`
3. Ejecute varios mandatos para visualizar información sobre el entorno.
   - Mostrar todos los servidores de contenedor en línea para la cuadrícula de datos Grid y el conjunto de correlaciones mapSet:

     ```
     xscmd -c showPlacement -g Grid -ms mapSet
     ```
   - Visualizar la información de direccionamiento de la cuadrícula de datos.

     ```
     xscmd -c routetable -g Grid
     ```
   - Visualizar el número de entradas de correlación en la cuadrícula de datos.

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

Detención de los servidores

Cuando ha terminado de utilizar la aplicación cliente y de supervisar el entorno de ejemplo de iniciación, puede detener los servidores.

- Si ha utilizado los archivos de script para iniciar los servidores, utilice <ctrl+c> para detener el proceso de servicio de catálogo y los servidores de contenedor en las ventanas respectivas.
- Si ha utilizado el mandato **startOgServer** para iniciar los servidores, utilice el mandato **stopOgServer** para detener los servidores.

  Detenga el servidor de contenedor:

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

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

  Detenga el servidor de catálogo:

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

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

Punto de comprobación de la lección

En esta lección, ha aprendido lo siguiente:

- Cómo iniciar la consola web y conectarla al servidor de catálogo
- Cómo supervisar las estadísticas del servidor y de la cuadrícula de datos
- Cómo detener los servidores

Capítulo 2. Planificación

Antes de instalar WebSphere eXtreme Scale y desplegar las aplicaciones de cuadrícula de datos, debe decidir sobre la topología de almacenamiento en memoria caché, completar la planificación de capacidad, revisar los requisitos de hardware y software, valores de red y ajuste, etc. ambién puede utilizar la lista de comprobación operacional para asegurarse de que el entorno está preparado para tener una aplicación desplegada.

Para obtener una descripción de los métodos recomendados que puede utilizar al diseñar las aplicaciones WebSphere eXtreme Scale, lea el artículo siguiente en developerWorks: Principles and best practices for building high performing and highly resilient WebSphere eXtreme Scale applications (Principios y métodos recomendados para crear aplicaciones de WebSphere eXtreme Scale muy flexibles y de alto rendimiento).

Visión general de la planificación

Antes de utilizar WebSphere eXtreme Scale en un entorno de producción, tenga en cuenta las siguientes cuestiones para optimizar el despliegue.

Consideraciones sobre la instalación

Puede instalar WebSphere eXtreme Scale en un entorno autónomo, o bien puede integrar la instalación con WebSphere Application Server. Para asegurarse de que podrá integrar sin problemas los servidores en el futuro, debe planificar el entorno en consecuencia. Para obtener el mejor rendimiento, los servidores de catálogo se deben ejecutar en distintas máquinas que los servidores de contenedor. Si debe ejecutar los servidores de catálogo y servidores de contenedor en la misma máquina, utilice distintas instalaciones de WebSphere eXtreme Scale para los servidores de catálogo y contenedor. Mediante dos instalaciones, puede actualizar en primer lugar la instalación que ejecuta el servidor de catálogo. Consulte

Consideraciones sobre la topología de memoria caché

La arquitectura puede utilizar almacenamiento en memoria caché de datos cliente-servidor distribuido o almacenamiento en memoria caché de datos en memoria local. Cada tipo de topología de memoria caché tiene ventajas y desventajas. La topología de almacenamiento en memoria caché que implemente dependerá de los requisitos de su entorno y aplicación. Para obtener más información sobre las distintas topologías de almacenamiento en memoria caché, consulte “Planificación de la topología” en la página 10.

Consideraciones sobre la capacidad de datos

A continuación se enumeran los elementos que deben tenerse en cuenta:

- **Número de sistemas y procesadores**: ¿cuántas máquinas físicas y cuántos procesadores se necesitan en el entorno?
- **Número de servidores**: ¿cuántos servidores eXtreme Scale para alojar correlaciones de eXtreme Scale?
- **Número de particiones**: el volumen de datos almacenados en las correlaciones es un factor para determinar el número de particiones necesarias.

- **Número de réplicas**: ¿cuántas réplicas se necesitan para cada fragmento primario del dominio?
- **Réplica síncrona o asíncrona**: ¿son vitales los datos de modo que la réplica síncrona es necesaria? ¿Es el rendimiento, en cambio, una prioridad mayor, por lo que la opción es la réplica asíncrona?
- **Tamaños de almacenamiento dinámico**: ¿cuántos datos se almacenarán en cada servidor?

Para obtener una descripción detallada de cada una de estas consideraciones, consulte "Planificación de la capacidad del entorno" en la página 55..

Planificación de la topología

Con WebSphere eXtreme Scale, la arquitectura puede utilizar el almacenamiento en memoria caché de datos en memoria local o el almacenamiento en memoria caché de datos de cliente-servidor distribuido. La arquitectura puede tener distintas relaciones con las bases de datos. También puede configurar la topología para que abarque varios centros de datos.

Para poder funcionar, WebSphere eXtreme Scale necesita una mínima infraestructura adicional. La infraestructura se compone de scripts que instalan, inician y detienen una aplicación Java Platform, Enterprise Edition en un servidor. Los datos colocados en memoria caché se almacenan en servidores de contenedor, y los clientes se conectan de forma remota al servidor.

Entornos en memoria

Cuando realiza un despliegue en un entorno local en memoria, WebSphere eXtreme Scale se ejecuta en una única Máquina virtual Java y no se replica. Para configurar un entorno local puede utilizar un archivo XML de ObjectGrid o las API de ObjectGrid.

Entornos distribuidos

Cuando realiza un despliegue en un entorno distribuido, WebSphere eXtreme Scale se ejecuta en un conjunto de Máquinas virtuales Java, aumentando el rendimiento, disponibilidad y escalabilidad. Con esta configuración, puede utilizar el particionamiento y la réplica de datos. También puede añadir servidores adicionales sin reiniciar los servidores eXtreme Scale existentes. Igual que en el entorno local, en el entorno distribuido se necesita un archivo XML ObjectGrid, o una configuración equivalente mediante programa. Debe también proporcionar un archivo XML de política de despliegue con detalles de configuración

Puede crear despliegues sencillos o grandes despliegues con terabytes en los que son necesarios miles de servidores.

Almacenamiento local de memoria caché en memoria

En el caso más sencillo, WebSphere eXtreme Scale se puede utilizar como una memoria caché de cuadrícula de datos en memoria local (no distribuida). El caso local beneficia especialmente a las aplicaciones de simultaneidad alta donde varias hebras necesitan acceder y modificar los datos transitorios. Los datos que se mantienen en una cuadrícula de datos local se pueden indexar y recuperar mediante consultas. Las consultas le ayudan a utilizar conjuntos de datos en memoria grandes. El soporte proporcionado con Máquina virtual Java (JVM), aunque está listo para su uso, tiene una estructura de datos limitada.

La topología de la memoria caché en memoria local para WebSphere eXtreme Scale se utiliza para proporcionar un acceso coherente y transaccional a los datos temporales de una única máquina virtual Java.

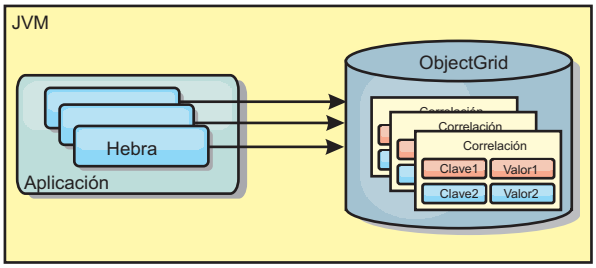


Figura 1. Escenario de memoria caché en memoria local

Ventajas

- Fácil configuración: se puede crear un ObjectGrid a través de un programa o de forma declarativa con el archivo XML descriptor de ObjectGrid o con otras infraestructuras como, por ejemplo, Spring.
- Rápido: cada BackingMap puede adaptarse de forma independiente de modo que la utilización de la memoria y la simultaneidad sean óptimas.
- Es ideal para las topologías de máquina virtual Java única con conjuntos de datos pequeños o para almacenar en memoria caché los datos de acceso frecuente.
- Es transaccional. Las actualizaciones de BackingMap se pueden agrupar en una única unidad de trabajo y se pueden integrar como último participante en transacciones de 2 fases como, por ejemplo, transacciones JTA (Java Transaction Architecture).

Desventajas

- No es tolerante a errores.
- Los datos no se replican. Las memorias caché en memoria son la mejor solución para los datos de referencia de sólo lectura.
- No es escalable. La cantidad de memoria necesaria para la base de datos podría desbordar la máquina virtual Java.
- Se producen problemas al añadir máquinas virtuales Java:
  - Los datos no se pueden particionar fácilmente.
  - Se debe replicar manualmente el estado entre las máquinas virtuales Java o cada instancia podría tener distintas versiones de los mismos datos.
  - La operación de invalidación es muy costosa.
  - Cada memoria caché se debe calentar de forma independientemente. El calentamiento es el periodo de carga de un conjunto de datos, de forma que la memoria caché se rellena con datos válidos.

Cuándo se debe utilizar

La topología de despliegue de la memoria caché en memoria local sólo se debe utilizar cuando la cantidad de datos que se deben almacenar en memoria caché es pequeña (cabe en una única máquina virtual Java) y es relativamente estable. Los datos obsoletos deben tolerarse con este acercamiento. El uso de desalojadores para mantener en la memoria caché los datos usados con más frecuencia o los más

recientes puede ayudar a mantener pequeño el tamaño de la memoria caché y a aumentar la relevancia de los datos.

Memoria caché local replicada de igual

Debe asegurarse de que la memoria caché esté sincronizada si existen varios procesos con instancias de memoria caché independientes. Para asegurarse de que las instancias de memoria caché están sincronizadas, habilite una memoria caché replicada por un igual con JMS (Java Message Service).

WebSphere eXtreme Scale incluye dos plug-ins que propagan automáticamente los cambios de las transacciones entre instancias de ObjectGrid de un igual. El plug-in JMSObjectGridEventListener propaga automáticamente los cambios de eXtreme Scale mediante JMS.

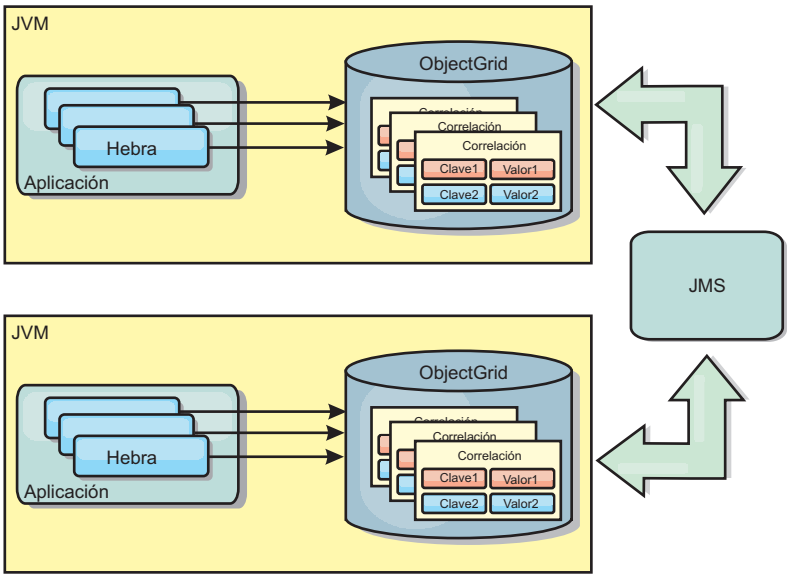


Figura 2. La memoria caché duplicada por un igual con los cambios que se propagan con JMS

Si ejecuta un entorno WebSphere Application Server, el plug-in TranPropListener también está disponible. El plug-in TranPropListener utiliza el gestor de alta disponibilidad (HA) para propagar los cambios a cada instancia de memoria caché de igual.

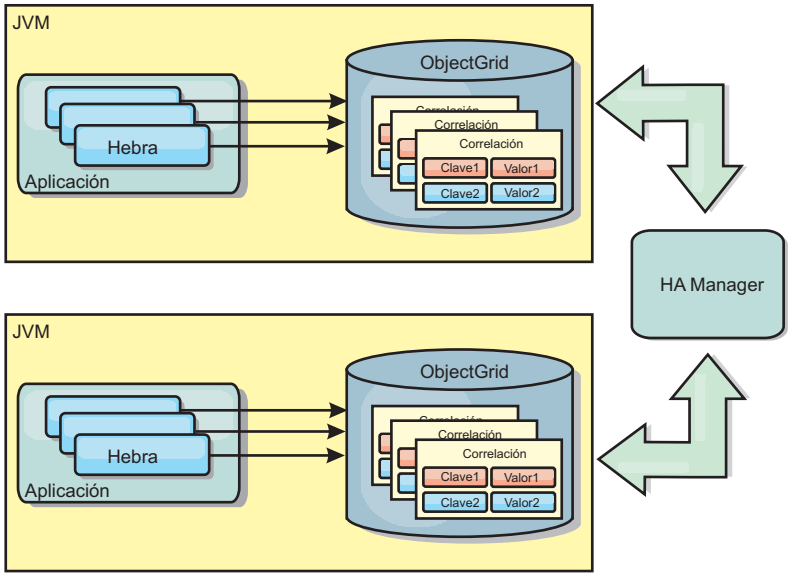


Figura 3. La memoria caché duplicada por un igual con los cambios propagados con el High Availability Manager.

Ventajas

- Los datos son más válidos porque se actualizan con más frecuencia.
- Con el plug-in TranPropListener, igual que el entorno local, eXtreme Scale se puede crear a través de programa o de forma declarativa con el archivo XML de descriptor de despliegue de eXtreme Scale o con otras infraestructuras como, por ejemplo, Spring. La integración con el High Availability Manager se realiza de forma automática.
- Cada BackingMap se puede ajustar independientemente para obtener un uso y una simultaneidad óptimos de la memoria.
- Las actualizaciones de BackingMap se pueden agrupar en una única unidad de trabajo y se pueden integrar como último participante en transacciones de 2 fases como, por ejemplo, transacciones JTA (Java Transaction Architecture).
- Ideal para topologías de pocas JVM con un conjunto de datos razonablemente pequeño o para almacenar en memoria caché datos de acceso frecuente.
- Los cambios en eXtreme Scale se duplican en todas las instancias de eXtreme Scale de igual. Los cambios son coherentes mientras se utilice una suscripción duradera.

Desventajas

- La configuración y el mantenimiento de JMSObjectGridEventListener pueden ser complejos. eXtreme Scale puede crearse mediante programación o de forma declarativa con el archivo XML de descriptor de despliegue de eXtreme Scale o con otras infraestructuras como Spring.
- No es escalable: el volumen de memoria que requiere la base de datos puede desbordar la JVM.
- Funciona de forma incorrecta cuando se añade Máquinas virtuales Java:
  - Los datos no se pueden particionar fácilmente.
  - La operación de invalidación es muy costosa.
  - Cada memoria caché debe calentarse de manera independiente.

Cuándo se debe utilizar

Utilice topología de despliegue solo cuando la cantidad de datos que se deben almacenar en memoria caché sea pequeña, pueda caber en una única JVM y sea relativamente estable.

Memoria caché incorporada

Las cuadrículas de WebSphere eXtreme Scale pueden ejecutarse en procesos existentes como servidores eXtreme Scale incorporados o bien puede gestionarlas como procesos externos.

Las cuadrículas incorporadas son útiles cuando se ejecutan en un servidor de aplicaciones como, por ejemplo, WebSphere Application Server. Puede iniciar los servidores eXtreme Scale que no están incorporados utilizando los scripts de la línea de mandatos y ejecutarlos en un proceso Java.

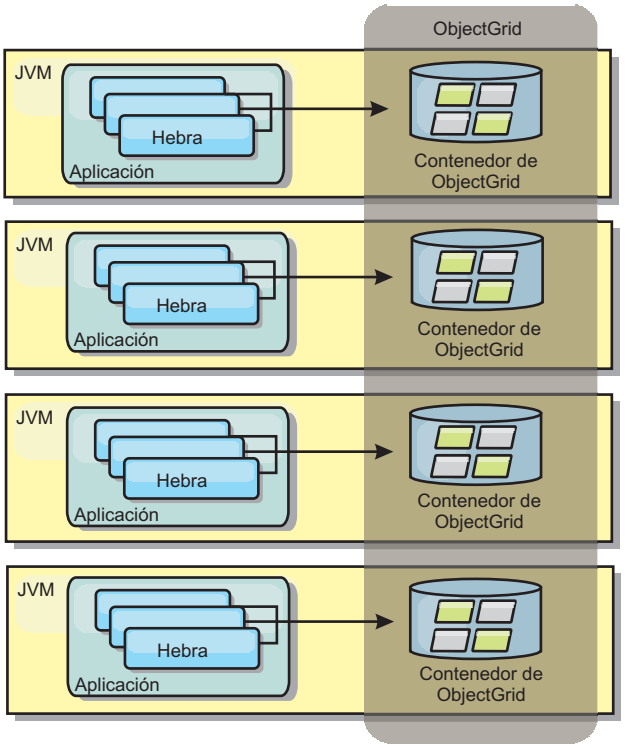


Figura 4. Memoria caché incorporada

Ventajas

- Administración simplificada ya que hay menos procesos que deban gestionarse.
- Despliegue de aplicaciones simplificado ya que la cuadrícula utiliza el cargador de clases de la aplicación cliente.
- Admite particionamiento y alta disponibilidad.

Desventajas

- Aumenta el uso de la memoria en procesos de cliente ya que todos los datos se colocan en el proceso.
- Aumenta el uso de la CPU para dar servicio a las solicitudes de los clientes.

- Es más difícil manejar las actualizaciones de las aplicaciones ya que los clientes utilizan los mismos archivos JAR (Java Archive) de aplicación que los servidores.
- Menos flexible. Escalar clientes y servidores de cuadrícula no puede aumentar a la misma velocidad. Si los servidores se definen externamente, puede tener más flexibilidad al gestionar el número de procesos.

Cuándo se debe utilizar

Utilice cuadrículas incorporadas cuando haya suficiente memoria libre en el proceso de cliente para datos de cuadrícula y posibles datos de sustitución por anomalía.

Para obtener más información, consulte el tema sobre la habilitación del mecanismo de invalidación de cliente en la*Guía de administración*.

Memoria caché distribuida

WebSphere eXtreme Scale se usa con más frecuencia como una memoria caché compartida, para proporcionar acceso transaccional a los datos en varios componentes donde, de lo contrario, se utilizará una base de datos tradicional. La memoria caché compartida elimina la necesidad de configurar una base de datos.

Coherencia de la memoria caché

La memoria caché es coherente porque todos los clientes ven los mismos datos en la memoria caché. Cada dato se almacena exactamente en un servidor de la memoria caché, lo que evita tener copias innecesarias que podrían contener posiblemente distintas versiones de los datos. Una memoria caché coherente también puede contener más datos a medida que se añadan más servidores a la cuadrícula de datos, y se amplía de forma lineal a medida que crece el tamaño de la cuadrícula. Puesto que los clientes acceden a los datos desde esta cuadrícula de datos con llamadas a procedimiento remotas, también se conoce como memoria caché remota, o memoria caché lejana). A través de la partición de datos, cada proceso contiene un subconjunto exclusivo del conjunto de datos total. Las cuadrículas de datos más grandes pueden contener más datos y dar servicio a más solicitudes de esos datos. La coherencia también elimina la necesidad de pasar datos de invalidación por la cuadrícula de datos porque no hay datos obsoletos. La memoria caché coherente sólo contiene la copia más reciente de cada dato.

Si ejecuta un entorno WebSphere Application Server, el plug-in TranPropListener también está disponible. El plug-in TranPropListener utiliza el componente de alta disponibilidad (HA Manager) de WebSphere Application Server para propagar los cambios en cada instancia de memoria caché de ObjectGrid de igual.

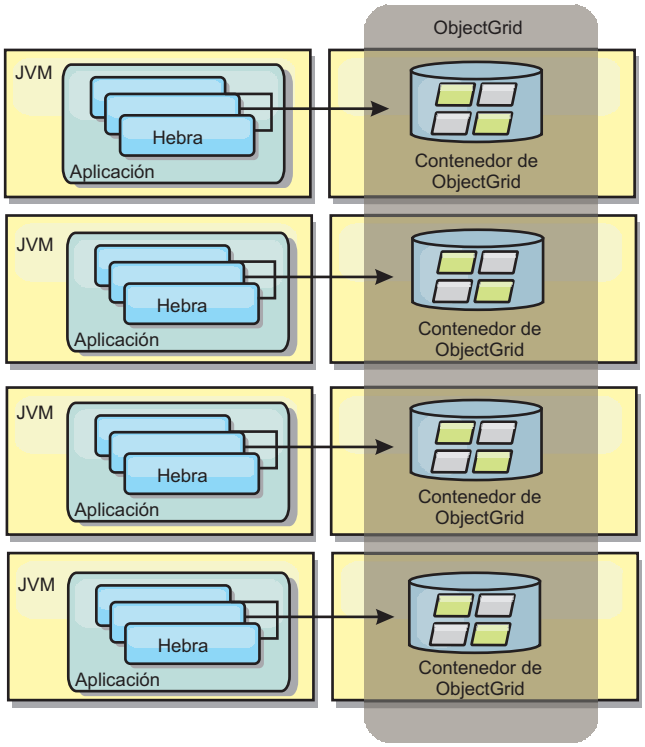


Figura 5. Memoria caché distribuida

Memoria caché cercana

De forma opcional, los clientes pueden tener una memoria caché local en línea cuando se utiliza eXtreme Scale en una topología distribuida. Esta memoria caché opcional se llama memoria caché cercana, es un ObjectGrid independiente en cada cliente, que sirve como memoria caché para la memoria caché remota del lado del servidor. La memoria caché cercana se habilita de manera predeterminada al configurar el bloqueo como optimista o ninguno, y no puede utilizarse si se configura como pesimista.

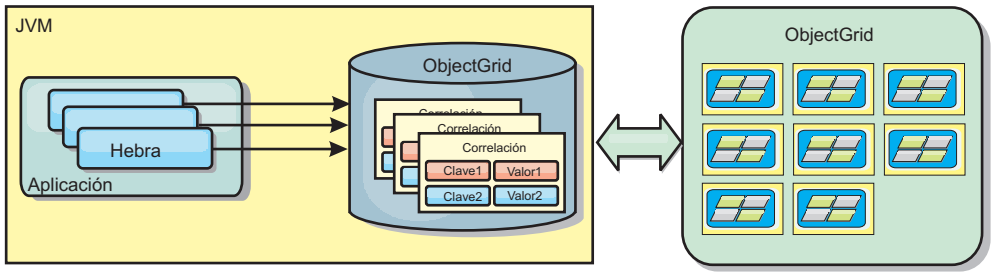


Figura 6. Memoria caché cercana

Una memoria caché cercana es muy rápida porque proporciona un acceso en memoria a un subconjunto de todos los conjuntos de datos almacenados en memoria caché que se almacenan de forma remota en los servidores eXtreme Scale. La memoria caché cercana no está particionada y contiene datos de cualquiera de las particiones eXtreme Scale remotas. WebSphere eXtreme Scale puede tener hasta tres niveles de memoria caché del modo siguiente.

1. La memoria caché de nivel de transacción contiene todos los cambios de una única transacción. La memoria caché de transacción contiene una copia de

trabajo de los datos hasta que la transacción se confirma. Cuando una transacción de cliente solicita datos de un objeto ObjectMap, primero se comprueba la transacción.

2. La memoria caché cercana en el nivel de cliente contiene un subconjunto de datos del nivel de servidor. Cuando el nivel de transacción no tiene los datos, los datos se captan de la capa de cliente, si están disponibles, y se insertan en la memoria caché de transacción
3. La cuadrícula de datos del nivel del servidor contiene la mayoría de los datos y se comparte entre todos los clientes. El nivel de servidor puede particionarse, lo que permite almacenar en memoria caché un gran volumen de datos. Cuando la memoria caché cercana de cliente no tiene los datos, éstos se captan del nivel de servidor y se insertan en la memoria caché de cliente. El nivel de servidor también tiene un plug-in Loader. Si la cuadrícula no tiene los datos solicitados, se invoca el Loader y los datos resultantes se insertan del almacén de datos de proceso de fondo en la cuadrícula.

Para inhabilitar la memoria caché cercana, establezca el atributo numberOfBuckets en 0 en la configuración de descriptor de eXtreme Scale de alteración temporal del cliente. Consulte el tema sobre el bloqueo de entrada de correlación para ver detalles sobre las estrategias de bloqueo de eXtreme Scale. La memoria caché cercana también se puede configurar para tener una política de desalojo separada y distintos plug-ins mediante una configuración de descriptor de eXtreme Scale de alteración temporal del cliente.

Ventaja

- Un tiempo de respuesta rápido porque todos los accesos a los datos son locales. Buscando los datos en la memoria caché cercana primero se guarda un recorrido a la cuadrícula de los servidores, por lo hace que incluso los datos remotos se puedan acceder de forma local.

Desventajas

- Aumenta la duración de los datos obsoletos debido a que la memoria caché cercana en cada nivel puede no estar sincronizada con los datos actuales de la cuadrícula de datos.
- Se basa en un desalojador para invalidar los datos a fin de evitar quedarse sin memoria.

Cuándo se debe utilizar

Debe usarse cuando el tiempo de respuesta sea importante y puedan tolerarse los datos obsoletos.

Integración de base de datos: almacenamiento en memoria caché de grabación diferida, en línea y complementaria

WebSphere eXtreme Scale se utiliza para atender una base de datos tradicional y eliminar la actividad de lectura que normalmente se envía a la base de datos. Puede utilizarse una memoria caché coherente con una aplicación mediante el uso directo o indirecto de un correlacionador de objetos relacionales. La memoria caché coherente puede después descargar de lecturas la base de datos o el programa de fondo. En un escenario ligeramente más complejo, como por ejemplo un acceso transaccional a un conjunto de datos donde sólo algunos de los datos necesitan garantías de persistencia tradicional, puede usarse el filtrado para descargar incluso transacciones de grabación.

Puede configurar WebSphere eXtreme Scale para que funcione como un espacio de proceso de base de datos en memoria muy flexible. No obstante, WebSphere eXtreme Scale no es un correlacionador de objetos relacionales (ORM). No sabe de dónde proceden los datos de la cuadrícula de datos. Una aplicación o un ORM puede colocar datos en un servidor eXtreme Scale. Es responsabilidad del origen de datos garantiza que son coherentes con la base de datos de la que proceden los datos. Esto significa que eXtreme Scale no puede invalidar los datos extraídos de una base de datos automáticamente. La aplicación o el correlacionador debe proporcionar esta función y gestionar los datos almacenados en eXtreme Scale.

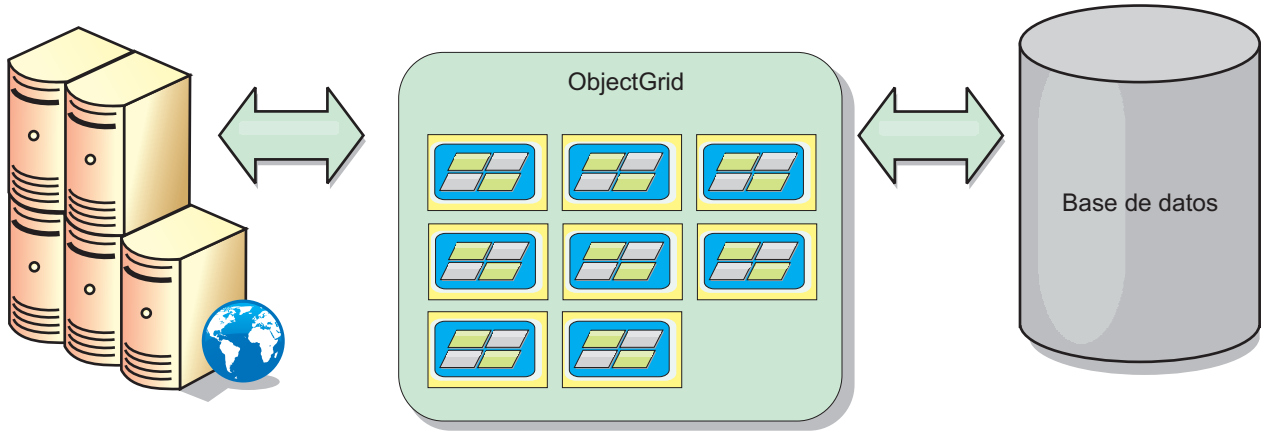


Figura 7. ObjectGrid como un almacenamiento intermedio de base de datos

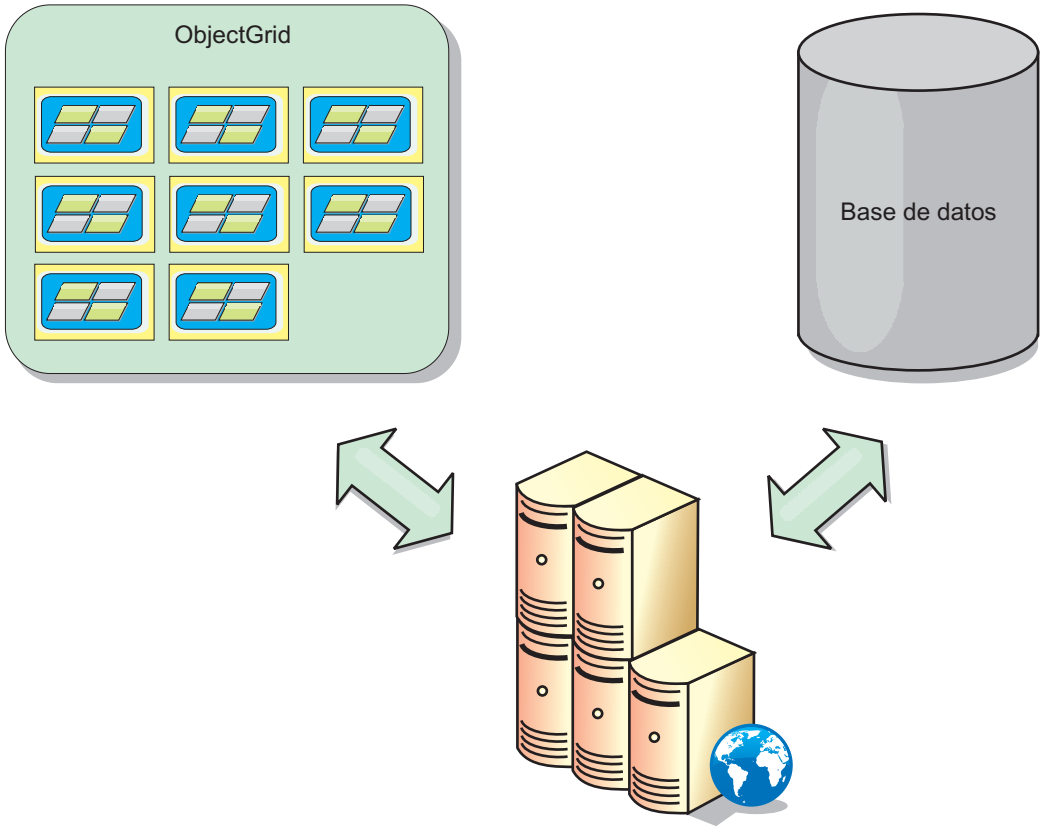


Figura 8. ObjectGrid como una memoria caché secundaria

Memoria caché escasa y completa

WebSphere eXtreme Scale puede utilizarse como una memoria caché escasa o una memoria caché completa. Una memoria caché escasa sólo mantiene un subconjunto de los datos totales, mientras que una memoria caché completa conserva todos los datos y se puede llenar de forma poca activa, conforme se requieran los datos. A las memorias caché escasas normalmente se accede utilizando claves (en lugar de índices o consultas) puesto que los datos sólo están parcialmente disponibles.

memoria caché escasa

Cuando una clave no está presente en una memoria caché escasa, o los datos no están disponibles y se produce una falta de coincidencia de memoria caché, se invoca el siguiente nivel. Los datos se captan, desde una base de datos, por ejemplo, y se insertan en el nivel de la memoria caché de cuadrícula de datos. Si utiliza una consulta o un índice, sólo se accede a los valores cargados actualmente y las solicitudes no se remiten a los demás niveles.

Memoria caché completa

Una memoria caché completa contiene todos los datos necesarios y se puede acceder a la misma utilizando atributos que no son de clave con índices o consultas. Una memoria caché completa se precarga con datos de la base de atos antes de que la aplicación intente acceder a los datos. Una memoria caché completa puede funcionar como una sustitución de base de datos después de que se carguen los datos. Puesto que están disponibles todos los datos, las consultas y los índices se pueden utilizar para encontrar y agregar datos.

Memoria caché complementaria

Cuando se utiliza WebSphere eXtreme Scale como memoria caché complementaria, se utiliza el programa de fondo con la cuadrícula de datos.

Memoria caché complementaria

Puede configurar el producto como una memoria caché complementaria para la capa de acceso a datos de una aplicación. En este escenario, WebSphere eXtreme Scale se utiliza para almacenar temporalmente objetos que normalmente se recuperarían de una base de datos de programa de fondo. Las aplicaciones comprueban si la cuadrícula de datos contiene los datos. Si los datos están en la cuadrícula de datos, los datos se devuelven al emisor. Si los datos no existen, los datos se recuperan de la base de datos de fondo. A continuación, los datos se insertan en la cuadrícula de datos de forma que la siguiente solicitud pueda utilizar la copia almacenada en memoria caché. El diagrama siguiente muestra cómo se puede utilizar WebSphere eXtreme Scale como una memoria caché complementaria con una capa de acceso a datos arbitrarios como por ejemplo OpenJPA o Hibernate.

Plug-ins de memoria caché para Hibernate y OpenJPA

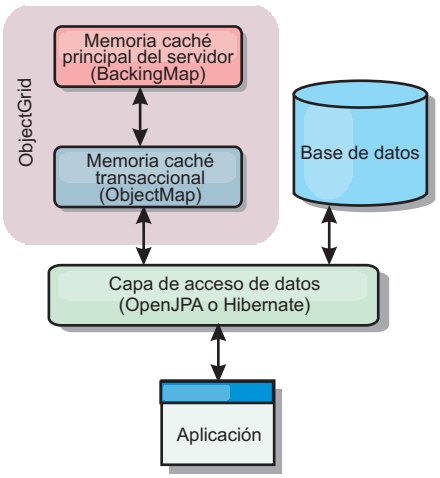


Figura 9. Memoria caché complementaria

Los plug-ins de memoria caché para OpenJPA e Hibernate se incluyen en WebSphere eXtreme Scale, de forma que puede utilizar el producto como una memoria caché complementaria automática. El uso de WebSphere eXtreme Scale como un proveedor de memoria caché aumenta el rendimiento cuando se leen y consultan datos y reduce la carga de la base de datos. WebSphere eXtreme Scale presenta algunas ventajas sobre las implementaciones de memoria caché incorporada ya que la memoria caché se replica automáticamente entre procesos. Cuando un cliente almacena en memoria caché un valor, todos los demás clientes pueden utilizar el valor almacenado en la memoria.

Memoria caché en línea

Puede configurar almacenamiento en memoria caché en línea para un programa de fondo de base de datos o como una memoria complementaria para una base de datos. El almacenamiento en memoria caché en línea utiliza eXtreme Scale como el medio principal para interactuar con los datos. Cuando se utiliza eXtreme Scale como una memoria caché en línea, la aplicación interactúa con el programa de fondo mediante un plug-in Loader.

Memoria caché en línea

Cuando se utiliza como una memoria caché en línea, WebSphere eXtreme Scale interactúa con el programa de fondo utilizando un plug-in Loader. Este escenario puede simplificar el acceso a datos porque las aplicaciones pueden acceder a las API eXtreme Scale directamente. Se da soporte a distintos escenarios de almacenamiento en memoria caché en eXtreme Scale para garantizar que los datos de la memoria caché y los datos del programa de fondo estarán sincronizados. El diagrama siguiente ilustra cómo una memoria caché en línea interactúa con la aplicación y el programa de fondo.

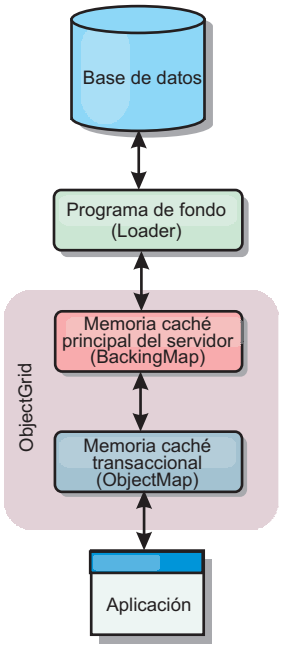


Figura 10. Memoria caché en línea

La opción de memoria caché en línea simplifica el acceso de datos, porque permite a las aplicaciones acceder a las API de eXtreme Scale directamente. WebSphere eXtreme Scale soporta varios escenarios de memoria caché en línea, del modo siguiente.

- Lectura directa
- Grabación directa
- Grabación diferida

Caso de ejemplo de almacenamiento en memoria caché de lectura directa

Una memoria caché de lectura directa es una memoria caché escasa que carga de forma poco activa entradas de datos por clave cuando se solicitan. Esto se lleva a cabo sin que el solicitante sepa cómo se llenan las entradas. Si los datos no se pueden encontrar en la memoria caché de eXtreme Scale, eXtreme Scale recuperará los datos que faltan del plug-in Loader, que carga los datos de la base de datos de programa de fondo y los inserta en la memoria caché. Las solicitudes subsiguientes para la misma clave de datos se encontrarán en la memoria caché hasta que se elimina, anula o desaloja.

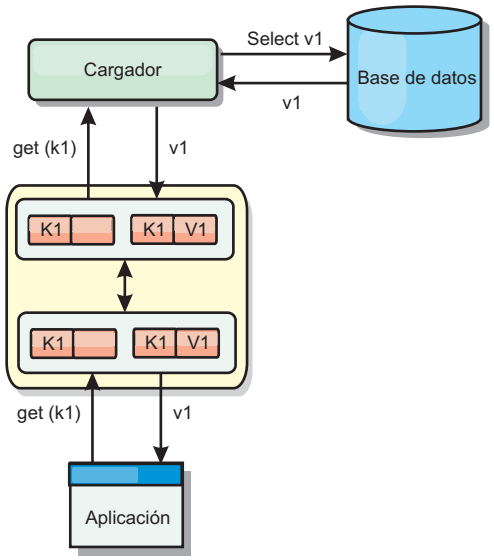


Figura 11. Almacenamiento en memoria caché de lectura directa

Caso de ejemplo de almacenamiento en memoria caché de grabación directa

En una memoria caché de grabación directa, cada grabación en la memoria caché graba de forma síncrona en la base de datos mediante el cargador. Este método proporciona coherencia con el programa de fondo, pero reduce el rendimiento de grabación porque la operación de la base de datos es síncrona. Como que la memoria caché y la base de datos están actualizadas, las lecturas subsiguientes para los mismos datos se encontrarán en la memoria caché, evitando la llamada a la base de datos. Una memoria caché de grabación directa suele utilizarse junto con una memoria caché de lectura directa.

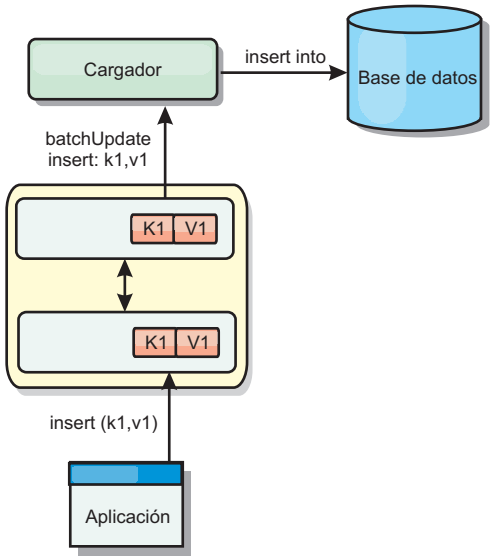


Figura 12. Almacenamiento en memoria caché de grabación directa

Caso de ejemplo de almacenamiento en memoria caché de grabación anticipada

La sincronización de base de datos se puede mejorar grabando los cambios de forma asíncrona. Esto se conoce como memoria caché de grabación diferida o de grabación aplazada. En su lugar, los cambios que normalmente se grabarían de forma síncrona en el cargador se colocarán en el almacenamiento intermedio de eXtreme Scale y se grabarán en la base de datos utilizando una hebra de subordinada. El rendimiento de grabación se mejora de forma significativa porque la operación de la base de datos se elimina de la transacción del cliente y se pueden comprimir las grabaciones de la base de datos.

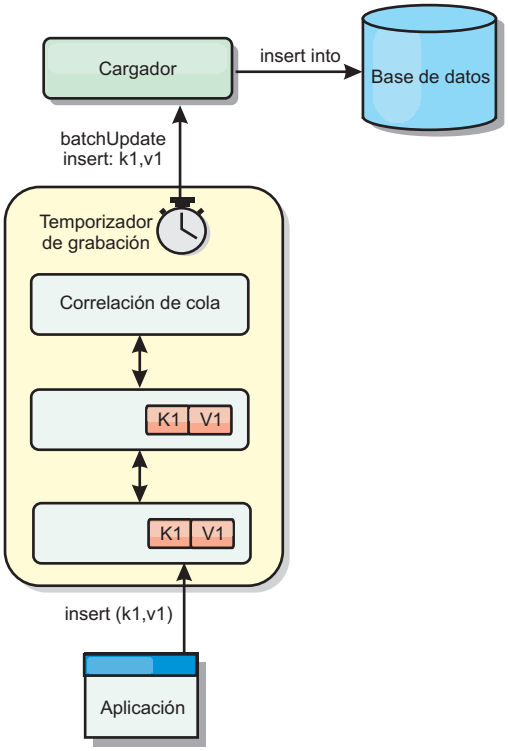


Figura 13. Almacenamiento en memoria caché de grabación diferida

Almacenamiento en memoria caché de grabación diferida

Puede utilizar el almacenamiento en la memoria caché de grabación diferida para reducir la sobrecarga que se produce al actualizar una base de datos utilizada como programa de fondo.

Visión general del almacenamiento en memoria caché con grabación diferida

El almacenamiento en memoria caché de grabación diferida pone en cola de forma asíncrona actualizaciones del plug-in de cargador (Loader). Puede mejorar el rendimiento mediante la desconexión de actualizaciones, inserciones y eliminaciones de una correlación, la sobrecarga de la actualización de la base de datos de programa de fondo. La actualización asíncrona se realiza después de un retardo basado en la hora (por ejemplo, cinco minutos) o un retardo basado en entradas (1000 entradas).

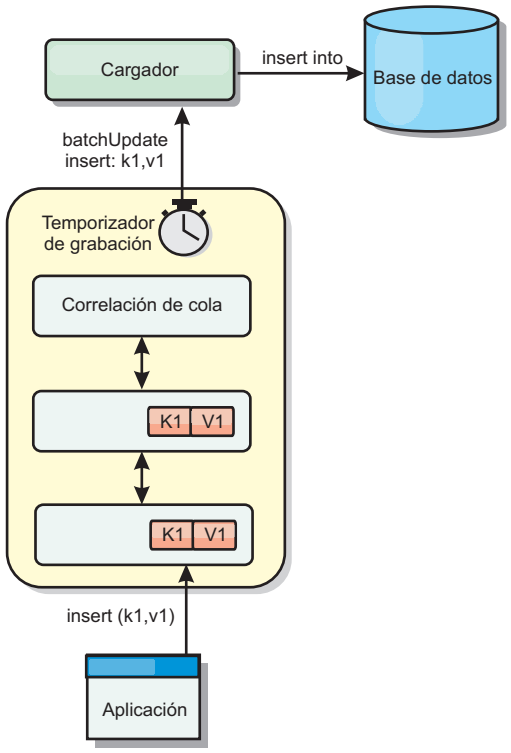


Figura 14. Almacenamiento en memoria caché de grabación diferida

La configuración de la grabación diferida en BackingMap crea una hebra entre el cargador y la correlación. El cargador delega las solicitudes de datos a través de la hebra de acuerdo con los valores de configuración del método BackingMap.setWriteBehind. Cuando una transacción de eXtreme Scale inserta, actualiza o elimina una entrada de una correlación, se crea un objeto LogElement para cada uno de estos registros. Estos elementos se envían al cargador de grabación diferida y se ponen en cola en un objeto ObjectMap especial llamado correlación de cola. Cada correlación de respaldo con el valor de grabación diferida habilitado tiene sus propias correlaciones de cola. Una hebra de grabación diferida elimina periódicamente los datos en cola de las correlaciones de cola y los envía al cargador de programa de fondo real.

El cargador de grabación diferida sólo envía los tipos de inserción, actualización y eliminación de objetos LogElement al cargador real. Todos los demás tipos de objetos LogElement, por ejemplo el tipo EVICT, se pasan por alto.

El soporte de grabación diferida *es* una ampliación del plug-in Loader, que puede utilizar para integrar eXtreme Scale con la base de datos. Por ejemplo, consulte la información del apartado “Configuración de cargadores JPA” en la página 354 sobre cómo configurar un cargador JPA.

Ventajas

La habilitación del soporte de grabación diferida tiene las ventajas siguientes:

- **Aislamiento de anomalía de programa de fondo:** el almacenamiento de grabación diferida proporciona una capa de aislamiento de las anomalías de programa de fondo. Cuando la base de datos de programa de fondo falla, las actualizaciones se ponen en cola en la correlación de cola. Las aplicaciones

pueden continuar con las transacciones a eXtreme Scale. Cuando se recupera el programa de fondo, los datos de la correlación de cola se envían al programa de fondo.

- **Carga reducida de programa de fondo** el cargador de grabación diferida fusiona las actualizaciones según una clave, de forma que sólo existe una actualización fusionada por clave en la correlación de cola. Este procedimiento reduce el número de actualizaciones en la base de datos de programa de fondo.
- **Rendimiento mejorado de transacciones:** los tiempos individuales de las transacciones de eXtreme Scale se reducen porque la transacción no necesita esperar a que los datos se sincronicen con el programa de fondo.

Consideraciones sobre el diseño de aplicaciones

Habilitar el soporte de grabación diferida es sencillo, pero diseñar una aplicación que funcione con el soporte de grabación diferida requiere un cuidado especial. Sin el soporte de grabación diferida, la transacción ObjectGrid encierra la transacción del programa de fondo. La transacción ObjectGrid se inicia antes de que se inicie la transacción de programa de fondo, pero termina después de que termine la transacción de programa de fondo.

Con el soporte de grabación diferida habilitado, la transacción ObjectGrid finaliza antes de que se inicie la transacción de programa de fondo. La transacción ObjectGrid y la transacción del programa de fondo se desacoplan.

Restricciones de la integridad referencial

Cada correlación de respaldo que se configura con soporte de grabación diferida tiene su propia hebra de grabación diferida que empuja los datos al programa de fondo. Por lo tanto, los datos que se actualizan en correlaciones diferentes de una transacción ObjectGrid se actualizan en el programa de fondo en diferentes transacciones de programa de fondo. Por ejemplo, la transacción T1 actualiza la clave key1 en la correlación Map1 y la clave key2 en la correlación Map2. La actualización de key1 en la correlación Map1 se actualiza en el programa de fondo en una transacción de programa de fondo, y la clave key2 actualizada en la correlación Map2 se actualiza en el programa de fondo en otra transacción de programa de fondo mediante distintas hebras de grabación diferida. Si los datos almacenados en Map1 y Map2 tienen relaciones, como restricciones de clave foránea en el programa de fondo, puede que se produzca un error en las actualizaciones.

Al diseñar las restricciones de la integridad referencial en la base de datos de programa de fondo, asegúrese de que se permiten las actualizaciones que no funcionan.

Comportamiento de bloqueo de correlaciones de cola

Otra diferencia principal en el comportamiento de las transacciones es el comportamiento de bloqueo. ObjectGrid admite tres estrategias de bloqueo distintas: pesimista (PESSIMISTIC), optimista (OPTIMISTIC) y ninguno (NONE). Las correlaciones de cola de grabación diferida utilizan la estrategia de bloqueo pesimista independientemente de la estrategia de bloqueo configurada en el mapa de respaldo. Existen dos tipos diferentes de operaciones que adquieren un bloqueo en la correlación de cola:

- Cuando se confirma una transacción ObjectGrid, o se produce un vaciado (vaciado de correlación o vaciado de sesión), la transacción lee la clave de la correlación de cola y coloca un bloqueo S en la clave.
- Cuando se confirma una transacción ObjectGrid, la transacción intenta actualizar el bloqueo S a un bloqueo X en la clave.

Debido a este comportamiento de correlación de colas adicional, puede ver algunas diferencias en el comportamiento del bloqueo.

- Si la correlación de usuarios está configurada como estrategia de bloqueo pesimista (PESSIMISTIC), no hay mucha diferencia de comportamiento en el bloqueo. Cada vez que se llama a una operación de desecho o confirmación, se coloca un bloqueo S en la misma clave de la misma correlación de colas. Durante la confirmación, no sólo se adquiere un bloqueo X para la clave en la correlación de usuarios, sino que además de adquiere para la clave en la correlación de colas.
- Si la correlación de usuarios está configurada como estrategia de bloqueo optimista (OPTIMISTIC) o ninguna (NONE), la transacción de usuario seguirá el patrón de estrategia de bloqueo pesimista (PESSIMISTIC). Cada vez que se llama a una operación de desecho o confirmación, se adquiere un bloqueo S en la misma clave de la misma correlación de colas. Durante la confirmación se adquiere un bloqueo X para la clave en la correlación de colas utilizando la misma transacción.

Reintentos de transacción de cargador

ObjectGrid no admite transacciones XA o en dos fases. La hebra de grabación diferida elimina los registros de la correlación de cola y actualiza los registros del programa de fondo. Si se produce una anomalía en el servidor durante la transacción, puede que se pierdan algunas actualizaciones del programa de fondo.

El cargador de grabación diferida reintentará automáticamente la grabación de las transacciones con anomalías y enviará un objeto LogSequence en duda al programa de fondo para evitar la pérdida de datos. Esta acción requiere que el cargador sea idempotente, que significa que cuando Loader.batchUpdate(TxId, LogSequence) se llama dos veces con el mismo valor, el resultado es como si se aplicara sólo una vez. Las implementaciones de cargador deben implementar la interfaz RetryableLoader para habilitar esta característica. Consulte la documentación de la API para obtener información detallada.

Anomalías del cargador

El plug-in de cargador puede fallar cuando no puede comunicarse con el programa de fondo de la base de datos. Esto puede suceder si el servidor de bases de datos o la conexión de red está inactivo. El cargador de grabación diferida pondrá en cola las actualizaciones e intentará empujar los cambios de los datos al cargador de forma periódica. El cargador debe notificar al tiempo de ejecución de ObjectGrid que hay un problema de conectividad de base de datos; para ello, emitirá una excepción LoaderNotAvailableException.

Por lo tanto, la implementación del cargador debe distinguir entre una anomalía de datos o un anomalía física del cargador. La anomalía de datos debe emitirse o volver a emitirse como excepción LoaderException o OptimisticCollisionException, pero una anomalía física del cargador debe emitirse o volver a emitirse como excepción LoaderNotAvailableException. ObjectGrid maneja estas dos excepciones de manera diferente:

- Si el cargador de grabación diferida obtiene una excepción LoaderException, el cargador de grabación diferida considerará la anomalía como un error de los datos, como por ejemplo un error de clave duplicada. El cargador de grabación diferida anulará el proceso por lotes de la actualización, e intentará actualizar un registro cada vez para aislar la anomalía de los datos. Si se vuelve a obtener una excepción {{LoaderException}} durante la actualización de un registro, se crea un registro de actualización con errores y se anota en la correlación de actualizaciones con errores.
- Si el cargador de grabación diferida obtiene una excepción LoaderNotAvailableException, el cargador de grabación diferida la considerará como un error porque no puede conectase a la base de datos, por ejemplo, el programa de fondo de la base de datos está inactivo, una conexión de base de datos no está disponible, o la red no está activa. El cargador de grabación diferida esperará 15 segundos y después volverá a intentar realizar la actualización por lotes en la base de datos.

El error habitual es emitir una excepción LoaderException cuando debería emitirse una excepción LoaderNotAvailableException. Todos los registros puestos en cola en el cargador de grabación diferida pasan a ser registros de actualizaciones con anomalías, que anula el propósito del aislamiento de anomalías de programa de fondo.

Consideraciones sobre el rendimiento

El soporte de almacenamiento en memoria caché de grabación diferida aumenta el tiempo de respuesta al eliminar la actualización del cargador de la transacción. También aumenta el rendimiento de base de datos ya que las actualizaciones de base de datos se combinan. Es importante comprender la sobrecarga que supone la hebra de grabación diferida, que extrae los datos de la correlación de cola y los envía al cargador.

El número máximo de actualizaciones o el tiempo máximo de actualización debe ajustarse en función del entorno y de los patrones de uso esperados. Si el valor del número máximo de actualizaciones o el tiempo máximo de actualización es demasiado pequeño, la sobrecarga de la hebra de grabación diferida puede sobrepasar las ventajas. Si se especifica un valor elevado para estos dos parámetros, podría aumentarse el uso de memoria al poner en cola los datos y aumentarse el tiempo obsoleto de los registros de la base de datos.

Para obtener un rendimiento óptimo, ajuste los parámetros de grabación diferida de acuerdo con los factores siguientes:

- Índice de transacciones de lectura y grabación.
- Misma frecuencia de actualización de registros.
- Latencia de actualización de la base de datos.

Cargadores

Con un plug-in Loader plug-in, una correlación de cuadrícula de datos puede actuar como una memoria caché de datos para los datos que se mantienen normalmente en un almacén persistente en el mismo sistema o en otro sistema. Generalmente, se utiliza una base de datos o un sistema de archivos como almacenamiento persistente. Una máquina virtual Java (JVM) remota también se puede utilizar como el origen de datos, lo que permite crear memorias caché basadas en hub utilizando eXtreme Scale. Un cargador tiene la lógica para leer y escribir datos en un almacén persistente.

Visión general

Los cargadores son plug-ins de correlaciones de respaldo que se invocan cuando se realizan cambios en la correlación de respaldo o ésta no puede satisfacer una solicitud de datos (una falta de memoria caché). Se invoca el cargador cuando la memoria caché no puede satisfacer la solicitud de una clave, proporcionando la capacidad de lectura a través y el relleno poco activo de la memoria caché. Un cargador también permite actualizar la base de datos cuando los valores de la memoria caché cambian. Todos los cambios de una transacción se agrupan para minimizar el número de interacciones de la base de datos. Se utiliza un plug-in TransactionCallback junto con el cargador para desencadenar la demarcación de la transacción de fondo. Utilizar este plug-in es importante cuando se incluyen varias correlaciones en una única transacción, o cuando se desechan los datos de una transacción en la memoria caché sin confirmar.

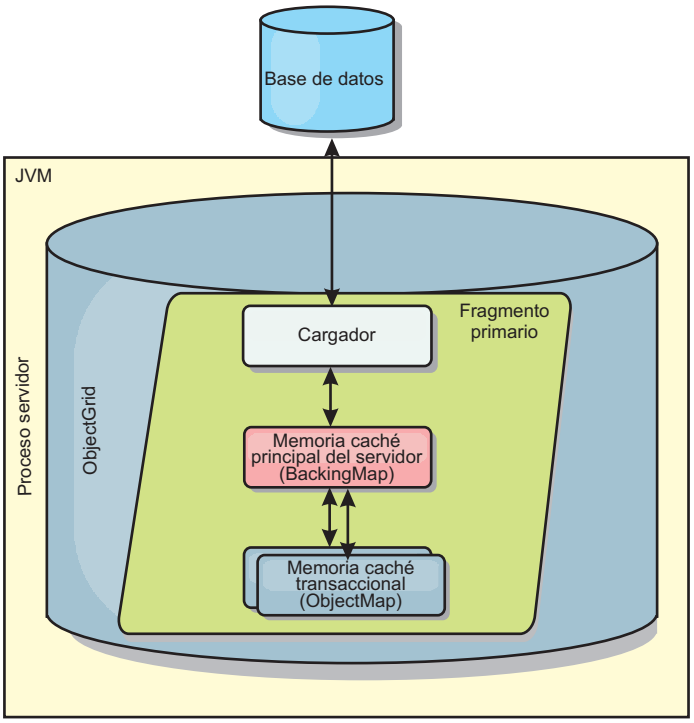


Figura 15. Cargador

El cargador también puede utilizar las actualizaciones sobrecualificadas para evitar mantener los bloqueos de base de datos. Al almacenar un atributo de versión en el valor de memoria caché, el cargador puede ver la imagen antes y después del valor tal como se actualiza en la memoria caché. Este valor se puede utilizar cuando se actualiza la base de datos o cuando se realiza un programa de fondo para verificar que los datos no se han actualizado. Un cargador también se puede configurar para precargar la cuadrícula de datos cuando se inicia. Cuando se realizan particiones, se asocia una instancia de cargador con cada partición. Si la correlación "Company" tiene diez particiones, hay diez instancias de cargador, una por partición primaria. Cuando se activa el fragmento primario de la correlación, se invoca el método preloadMap para el cargador de forma síncrona o asíncrona, que permite cargar automáticamente la partición de la correlación con los datos procedentes del programa de fondo. Cuando se invocan de forma síncrona, todas las transacciones de cliente se bloquean, lo que impide el acceso incoherente a la

cuadrícula de datos. De forma alternativa, se puede utilizar un precargador de cliente para cargar toda la cuadrícula de datos.

Dos cargadores incorporados pueden simplificar en gran medida la integración con los programas de fondo de la base de datos relacional. Los cargadores JPA utilizan las funciones de correlación de objetos relacionales (ORM) de ambas implementaciones, OpenJPA e Hibernate, de la especificación de JPA (Java Persistence API). Si desea más información, consulte Cargadores JPA.

Si utiliza cargadores en una configuración de varios centros de datos, debe considerar cómo se mantiene la coherencia de los datos y la memoria caché entre las cuadrículas de datos. Para obtener más información, consulte "Consideraciones sobre el cargador en una topología multimaestro" en la página 41.

Configuración de cargador

Para añadir un cargador a la configuración de BackingMap, puede utilizar la configuración mediante programa o la configuración del archivo XML. Un cargador tiene la siguiente relación con una correlación de respaldo.

- Una correlación de respaldo sólo puede tener un cargador.
- Una correlación de respaldo de cliente (memoria caché cercana) no puede tener un cargador.
- Una definición de cargador se puede aplicar a varias correlaciones de respaldo, pero cada una de éstas tiene su propia instancia de cargador.

Precarga de datos y calentamiento

En muchos escenarios que incorporan el uso de un cargador, puede preparar la cuadrícula de datos precargándola con datos.

Cuando se utiliza como una memoria caché completa, la cuadrícula de datos debe alojar todos los datos y se debe cargar antes de que los clientes se puedan conectar a ella. Cuando se utiliza una memoria caché escasa, puede preparar la memoria caché con datos de forma que los clientes tengan acceso inmediato a los datos cuando estos se conecten.

Existen dos enfoques para la precarga de datos en la cuadrícula de datos: mediante un plug-in Loader o mediante un cargador de clientes, tal como se describe en las secciones siguientes.

Plug-in Loader

El plug-in Loader está asociado con cada correlación y es responsable de sincronizar un fragmento de partición primaria con la base de datos. El método preloadMap del plug-in Loader se invoca automáticamente cuando se activa un fragmento. Por ejemplo, si tiene 100 particiones, existen 100 instancias de cargador, y cada una carga los datos para su partición. Si se ejecuta de forma síncrona, todos los clientes se bloquean hasta que se complete la precarga.

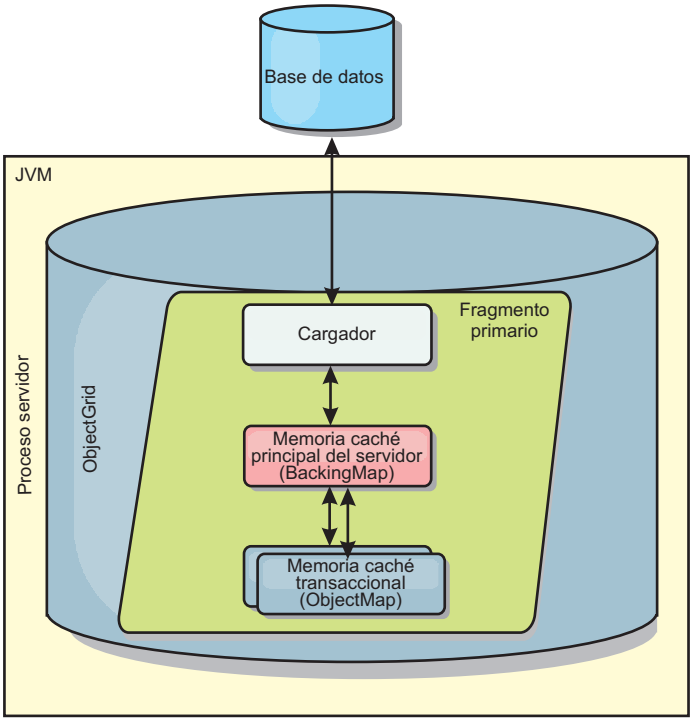


Figura 16. Plug-in Loader

Cargador de clientes

Un cargador de clientes es un patrón para utilizar uno o más clientes para carga la cuadrícula con datos. El uso de varios clientes para cargar los datos de cuadrícula puede ser eficaz cuando el esquema de partición no se almacena en la base de datos. Puede invocar los cargadores de clientes manual o automáticamente cuando se inicia la cuadrícula de datos. De forma opcional, los cargadores de clientes pueden utilizar StateManager para establecer el estado de la cuadrícula de datos en la modalidad de precarga, de forma que los clientes no pueden acceder a la cuadrícula mientras está precargando los datos. WebSphere eXtreme Scale incluye un cargador basado en JPA (Java Persistence API) que puede utilizar para cargar automáticamente la cuadrícula de datos con los proveedores OpenJPA o Hibernate JPA. Para obtener más información sobre los proveedores de memoria caché, consulte “Plug-in de memoria caché de nivel 2 (L2) JPA” en la página 331.

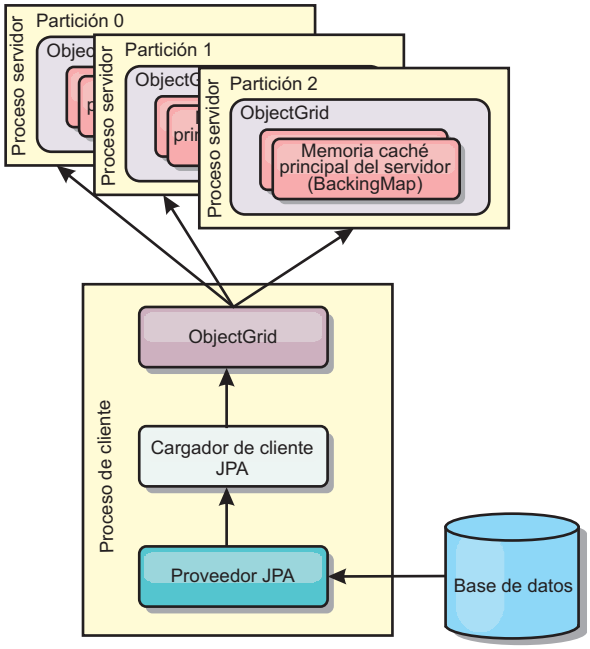


Figura 17. Cargador de clientes

Técnicas de sincronización de base de datos

Cuando se utiliza WebSphere eXtreme Scale como memoria caché, se deben escribir aplicaciones que admitan datos obsoletos si la base de datos puede actualizarse de forma independiente a una transacción de eXtreme Scale. Para servir como un espacio de proceso de base de datos en memoria sincronizado, eXtreme Scale proporciona distintos métodos para mantener la memoria caché actualizada.

Técnicas de sincronización de base de datos

Renovación periódica

La memoria caché se puede invalidar o actualizar de forma automática y periódica utilizando el actualizados de base de datos basado en el tiempo de JPA (Java Persistence API). El actualizador consulta periódicamente la base de datos utilizando un proveedor JPA para cualquier actualización o inserción que se haya producido desde la actualización anterior. Todos los cambios identificados se anulan o actualizan automáticamente cuando se utilizan con una memoria caché escasa. Si se utilizan con una memoria caché completa, las entradas se pueden descubrir e insertar en la memoria caché. Las entradas nunca se eliminan de la memoria caché.

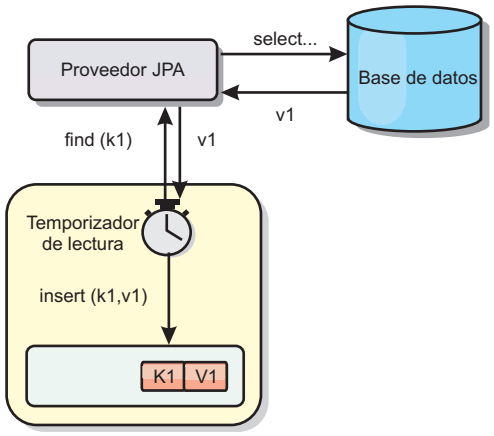


Figura 18. Renovación periódica

Desalojo

Las memorias caché escasas pueden utilizar políticas de desalojo para eliminar automáticamente datos de la memoria caché sin afectar a la base de datos. Existen tres políticas incorporadas incluidas en eXtreme Scale: tiempo de vida, menos usada recientemente y usada con menos frecuencia. Las tres políticas pueden, de forma opcional, desalojar datos de forma más agresiva a medida que la memoria pasa a estar limitada habilitando la opción de desalojo basado en memoria.

Anulación basada en sucesos

Las memorias caché escasas y completas se pueden invalidar o actualizar utilizando un generador de sucesos como, por ejemplo, JMS (Java Message Service). La anulación utilizando JMS puede unirse manualmente a cualquier proceso que actualiza el programa de fondo utilizando un desencadenante de base de datos. Se proporciona un plug-in JMS ObjectGridEventListener en eXtreme Scale que puede notificar a los clientes cuando la memoria caché del servidor tiene algún cambio. Esto puede disminuir la cantidad de tiempo que el cliente puede ver los datos obsoletos.

Anulación programática

Las API eXtreme Scale permiten la interacción manual de la memoria caché cercana y de servidor utilizando los métodos de API Session.beginNoWriteThrough(), ObjectMap.invalidate() y EntityManager.invalidate(). Si un proceso de cliente o servidor ya no necesita una parte de los datos, los métodos de anulación se pueden utilizar para eliminar datos de la memoria caché cercana o del servidor. El método beginNoWriteThrough se aplica cualquier operación ObjectMap o EntityManager a la memoria caché local sin llamar al cargador. Si se invoca desde un cliente, la operación sólo se aplica a la memoria caché cercana (el cargador remoto no se invoca). Si se invoca en el servidor, la operación sólo se aplica a la memoria caché principal del servidor sin invocar el cargador.

Invalidación de datos

Para eliminar los datos de memoria caché de escala, puede utilizar un mecanismo de invalidación basado en suceso o mediante programa.

Invalidación basada en sucesos

Las memorias caché escasas y completas se pueden invalidar o actualizar utilizando un generador de sucesos como, por ejemplo, JMS (Java Message Service). La anulación utilizando JMS puede unirse manualmente a cualquier proceso que actualiza el programa de fondo utilizando un desencadenante de base de datos. Se proporciona un plug-in JMS ObjectGridEventListener en eXtreme Scale que puede notificar a los clientes cuando la memoria caché de servidor cambia. Este tipo de notificación disminuye la cantidad de tiempo que el cliente puede ver los datos obsoletos.

La invalidación basada en sucesos consta normalmente de los tres componentes siguientes.

- **Cola de sucesos:** Una cola de sucesos almacena los sucesos de cambio de datos. Puede ser una cola JMS, una base de datos, una cola FIFO o cualquier clase de siempre que pueda gestionar los sucesos de cambio de datos.
- **Editor de sucesos:** Un editor de sucesos publica los sucesos de cambio de datos en la cola de sucesos. Un editor de sucesos es normalmente una aplicación que usted mismo crea o una implementación de plug-in de eXtreme Scale. El editor de sucesos sabe cuándo se cambian los datos o cambia los datos por sí mismo. Cuando se confirma una transacción, se generan los sucesos para los datos cambiados y el editor de sucesos publica estos sucesos en la cola de sucesos.
- **Consumidor de sucesos:** Un consumidor de sucesos consume sucesos de cambio de datos. El consumidor de sucesos es por lo general una aplicación para garantizar que los datos de la cuadrícula de destino se actualizan con el cambio más reciente de otras cuadrículas. Este consumidor de sucesos interactúa con la cola de sucesos para obtener los cambios de datos más recientes y aplica los cambios de datos en la cuadrícula de destino. Los consumidores de sucesos pueden utilizar las API de eXtreme Scale para invalidar datos obsoletos o actualizar la cuadrícula con los datos más recientes.

Por ejemplo, JMSObjectGridEventListener tiene una opción para un modelo cliente-servidor, en el cual la cola de sucesos es un destino de JMS designado. Todos los procesos del servidor son editores de sucesos. Cuando se confirma una transacción, el servidor obtiene los cambios de datos y los publica en la JMS de destino designada. Todos los procesos de cliente son consumidores de sucesos. Reciben los cambios de datos del destino de JMS designado y aplican los cambios en la memoria caché cercana del cliente.

Consulte el tema sobre la habilitación del mecanismo de invalidación del cliente en la *Guía de administración* si desea más información.

Anulación programática

Las API WebSphere eXtreme Scale permiten la interacción manual de la memoria caché cercana y de servidor utilizando los métodos de API Session.beginNoWriteThrough(), ObjectMap.invalidate() y EntityManager.invalidate(). Si un proceso de cliente o servidor ya no necesita una parte de los datos, los métodos de anulación se pueden utilizar para eliminar datos de la memoria caché cercana o del servidor. El método beginNoWriteThrough se aplica cualquier operación ObjectMap o EntityManager a la memoria caché local sin llamar al cargador. Si se invoca desde un cliente, la operación sólo se aplica a la memoria caché cercana (el cargador remoto no se invoca). Si se invoca en el servidor, la operación sólo se aplica a la memoria caché principal del servidor sin invocar el cargador.

Puede utilizar la anulación mediante programa con otras técnicas para determinar cuándo invalidar los datos. Por ejemplo, este método de invalidación utiliza mecanismos de invalidación basados en sucesos para recibir los sucesos de cambio de datos y luego utiliza interfaces de programación de aplicaciones para invalidar los datos obsoletos.

Índices

Utilice el plug-in MapIndexPlugin para crear un índice o varios índice en una BackingMap para dar soporte al acceso a datos no de clave.

Tipos de índices y configuración

La característica de indexación la representa el plug-in MapIndexPlugin, o Index de forma abreviada. Index es un plug-in BackingMap. Una BackingMap puede tener varios plug-ins Index configurados, siempre que cada uno de ellos siga las normas de configuración de Index.

Puede utilizar la característica de indexación para crear uno o más índices en una BackingMap. Un índice se crea a partir de un atributo o una lista de atributos de un objeto en la BackingMap. De esta manera, las aplicaciones pueden encontrar rápidamente determinados objetos. Con la característica de índices, las aplicaciones pueden encontrar objetos con un valor específico o dentro de un intervalo de valores de atributos indizados.

Existen dos tipos de índice: estático y dinámico. Con el índice estático, debe configurar el plug-in de índices en BackingMap antes de inicializar la instancia de ObjectGrid. Puede realizar esta configuración con una configuración de XML o mediante programa de la BackingMap. Los índices estáticos empiezan a construir un índice durante la inicialización de ObjectGrid. El índice siempre está sincronizado con la BackingMap y listo para ser utilizado. Después de que se inicie el proceso de indexación estática, el mantenimiento del índice forma parte del proceso de gestión de transacciones de eXtreme Scale. Cuando las transacciones confirman cambios, estos cambios también actualizan el índice estático y los cambios de índice se retrotraen si la transacción se retrotrae.

Con el índice dinámico, puede crear un índice en una correlación BackingMap antes o después de la inicialización de la instancia de ObjectGrid que contiene. Las aplicaciones tienen un control del ciclo de vida sobre el proceso de indexación dinámica, de forma que pueda eliminar un índice dinámico, cuando ya no sea necesario. Cuando una aplicación crea un índice dinámico, éste podría no estar listo para su uso inmediato debido al tiempo que tarda en completarse el proceso de creación del índice. Puesto que la cantidad de tiempo depende de la cantidad de datos indexados, se proporciona la interfaz DynamicIndexCallback para aplicaciones que desean recibir notificaciones cuando se produzcan determinados sucesos de indexación. Estos sucesos pueden incluir sucesos de error, destrucción y preparado. Las aplicaciones pueden implementar esta interfaz de devolución de llamada y registrarla con el proceso de índices dinámicos.

Si una BackingMap tiene un plug-in de índice configurado, podrá obtener el proxy de índice de aplicaciones de la ObjectMap correspondiente. Si se llama al método getIndex en la ObjectMap y se proporciona el nombre del plug-in de índice, se devolverá el objeto de proxy de índice. Debe difundir el objeto de proxy de índice en una interfaz apropiada de índice de aplicaciones como, por ejemplo, MapIndex, MapRangeIndex, o una interfaz personalizada de índices. Después de obtener el objeto de proxy de índice, puede utilizar los métodos definidos en la interfaz de índices de aplicación para buscar objetos almacenados en memoria caché.

En la lista siguiente se resumen los pasos que debe seguir para utilizar los índices:

- Añada plug-ins de índices estáticos o dinámicos a BackingMap.
- Obtenga el objeto de proxy de índice de aplicación; para ello, emita el método getIndex de ObjectMap.
- Difunda el objeto de proxy de índice a una interfaz de índices de aplicación apropiada, como MapIndex, MapRangeIndex, o a una interfaz de índices personalizada.
- Utilice los métodos definidos en una interfaz de índices de aplicación para buscar los objetos almacenados en memoria caché.

La clase HashIndex es la implementación de plug-in de índice que puede soportar ambas interfaces de índice de aplicación incorporadas: MapIndex y MapRangeIndex. También puede crear sus propios índices. Puede añadir HashIndex como un índice estático o dinámico en BackingMap, obtener un objeto proxy de índice MapIndex o MapRangeIndex y utilizar el objeto proxy de índice para encontrar los objetos almacenados en memoria caché.

Índice predeterminado

Si desea iterar a través de las claves en una correlación local, puede utilizar el índice predeterminado. Este índice no requiere ninguna configuración, pero se debe utilizar en el fragmento, utilizando una instancia de ObjectGrid o agente recuperada del método ShardEvents.shardActivated(ObjectGrid shard).

Consideraciones sobre la calidad de los datos

Los resultados de los métodos de consulta de índice sólo representan una instantánea de los datos en un momento puntual. No se obtiene ningún bloqueo contra la entrada de datos después de que los resultados vuelvan a la aplicación. La aplicación tiene que ser consciente de que se pueden producir actualizaciones de datos en un conjunto de datos devuelto. Por ejemplo, la aplicación obtiene la clave de un objeto almacenado en memoria caché ejecutando el método findAll de MapIndex. Este objeto de clave devuelto se asocia a una entrada de datos de la memoria caché. La aplicación debe poder ejecutar el método get en ObjectMap para encontrar un objeto proporcionando el objeto de clave. Si otra transacción elimina el objeto de datos de la memoria caché, justo antes de que se llame al método get, el resultado devuelto será nulo.

Consideraciones sobre el rendimiento de los índices

Uno de los principales objetivos de la característica de índices es mejorar el rendimiento global de BackingMap. Si los índices no se utilizan correctamente, podría verse afectado el rendimiento de la aplicación. Tenga en cuenta los siguientes factores antes de utilizar esta característica.

- **El número de transacciones de escritura simultáneas:** el proceso de índices se puede producir cada vez que una transacción escribe datos en una BackingMap. El rendimiento disminuye si hay muchas transacciones grabando datos en una correlación al mismo tiempo que una aplicación realiza operaciones de consulta de índices.
- **El tamaño del conjunto de resultados devuelto por una operación de consulta:** a medida que el tamaño del conjunto de resultados aumenta, el rendimiento de la consulta disminuye. El rendimiento tiene tendencia a disminuir si el tamaño del conjunto de resultados es un 15% o más de la BackingMap.

- **El número de índices creados sobre la misma BackingMap:** cada índice consume recursos del sistema. A medida que el número de índices creados sobre la BackingMap aumenta, disminuye el rendimiento.

La función de indexación puede mejorar el rendimiento de BackingMap de forma drástica. Los casos ideales se producen cuando la BackingMap tiene operaciones básicamente de lectura, el conjunto de resultados de la consulta es un pequeño porcentaje de las entradas de BackingMap, y sólo se crean unos pocos índices sobre la BackingMap.

Planificación de topologías de varios centros de datos

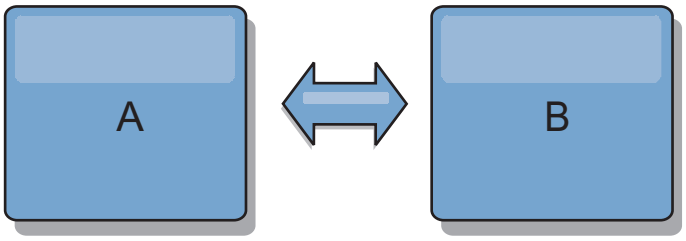
Mediante la utilización de la réplica asíncrona multimaestro, dos o más cuadrículas de datos pueden convertirse en copias exactas entre ellas. Cada cuadrícula de datos está alojada en un dominio de servicio de catálogo independiente, con su propio servicio de catálogo, servidores de contenedor y un nombre exclusivo. Con la réplica asíncrona multimaestro, puede utilizar enlaces para conectar una colección de dominios de servicio de catálogo. A continuación, los dominios de servicio de catálogo se sincronizan utilizando la réplica mediante los enlaces. Puede construir casi cada topología mediante la definición de enlaces entre los dominios de servicio de catálogo.

Topologías para réplica multimaestro

Tiene diversas opciones cuando elige una topología para el despliegue que incorpora réplica multimaestro.

Enlaces que conectan dominios de servicio de catálogo

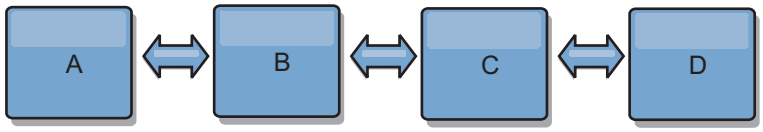
Una infraestructura de cuadrícula de datos de réplica es un gráfico conectado de dominios de servicio de catálogo con enlaces bidireccionales entre ellos. Con un enlace, dos dominios de servicio de catálogo pueden comunicar los cambios en los datos. Por ejemplo, la topología más sencilla es un par de dominios de servicio de catálogo con un único enlace entre ellos. Los dominios de servicio de catálogo se nombran alfabéticamente: A, B, C y así sucesivamente, desde la izquierda. Un enlace puede cruzar una red de área amplia (WAN), abarcando distancias grandes. Incluso si se interrumpe el enlace, aún puede cambiar los datos en cualquiera de los dos dominios de servicio de catálogo. La topología reconcilia los cambios cuando el enlace reconecta los dominios de servicio de catálogo. Los enlaces intentan volverse a conectar automáticamente si se interrumpe la conexión de red.



Después de haber configurado los enlaces, eXtreme Scale en primer lugar intenta hacer que cada dominio de servicio de catálogo sea idéntico. A continuación, eXtreme Scale intenta mantener las condiciones idénticas a los cambios producidos en cualquier dominio de servicio de catálogo. El objetivo es que cada dominio de servicio de catálogo sea un reflejo exacto de cada uno de los otros dominios de servicio de catálogo conectados mediante los enlaces. Los enlaces de réplica entre los dominios de servicio de catálogo ayudan a garantizar que los cambios realizados en un dominio se copian en los otros dominios.

Topologías de línea

Aunque es un despliegue muy simple, una topología de línea muestra algunas cualidades de los enlaces. En primer lugar, no es necesario que un dominio de servicio de catálogo esté conectado directamente a cada uno de los otros dominios de servicio de catálogo para que reciba los cambios. El Dominio B obtiene los cambios del Dominio A. El Dominio C recibe los cambios del Dominio A a través del Dominio B, que se conecta a los Dominios A y C. De forma similar, el Dominio D recibe los cambios de los otros dominios mediante el Dominio C. Esta capacidad esparce la carga de distribuir los cambios lejos del origen de los cambios.



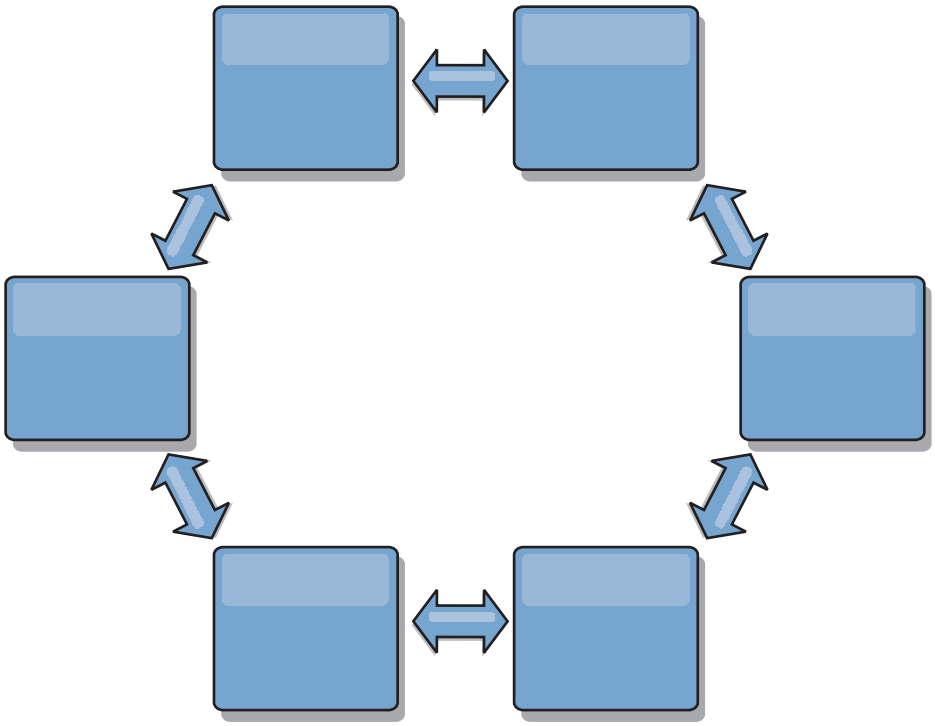
Tenga en cuenta que si el Dominio C falla, se producirán las acciones siguientes:

1. El Dominio D se quedará huérfano hasta que se reinicie el Dominio C
2. El Dominio C se sincronizará a sí mismo con el Dominio B, que es una copia del Dominio A
3. El Dominio D utilizará el Dominio C para sincronizarse a sí mismo con los cambios de los Dominios A y B. Estos cambios inicialmente se han producido mientras el Dominio D estaba huérfano (mientras el Dominio C estaba inactivo).

En última instancia, los Dominios A, B, C y D serán todos ellos idénticos entre ellos de nuevo.

Topologías de anillo

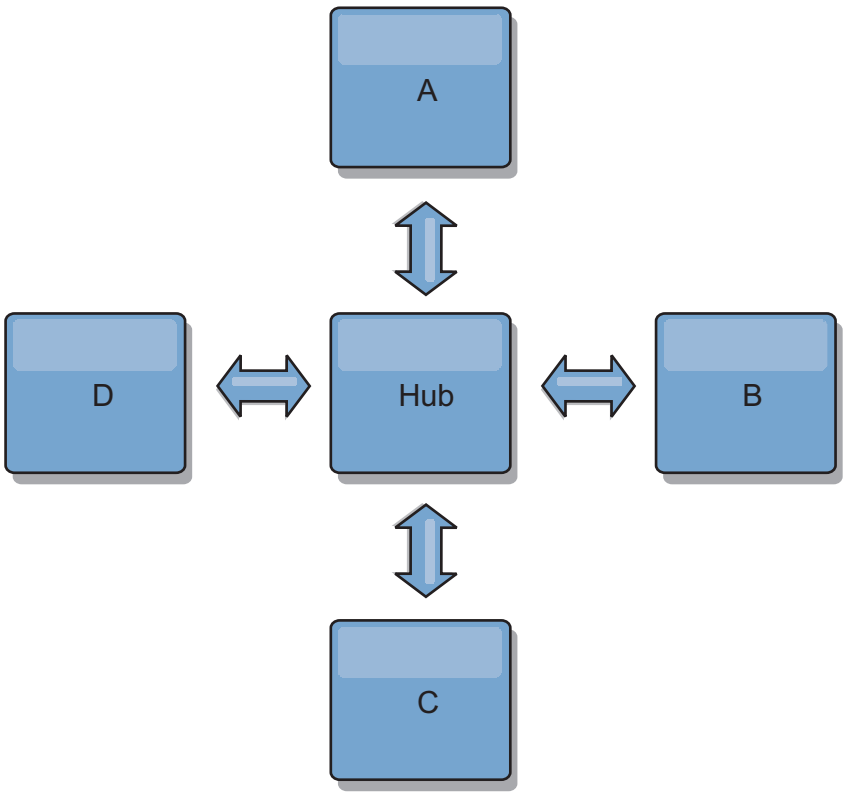
Las topologías de anillo son un ejemplo de una topología más flexible. Cuando un dominio de servicio de catálogo o un único enlace falla, los dominios de servicio de catálogo supervivientes todavía pueden obtener los cambios. Los dominios de servicio de catálogo viajan alrededor del anillo, alejándose de la anomalía. Cada dominio de servicio de catálogo tiene como máximo dos enlaces a otros dominios de servicio de catálogo, independientemente del tamaño de la topología de anillo. La latencia para propagar los cambios puede ser grande. Podría ser necesario que los cambios de un dominio de servicio de catálogo determinado viajaran a través de varios enlaces antes de que todos los dominios de servicio de catálogo tengan los cambios. Una topología de línea tiene la misma característica.



También puede desplegar una topología de anillo más sofisticada, con un dominio de servicio de catálogo raíz en el centro del anillo. El dominio de servicio de catálogo raíz funciona como el punto central de reconciliación. Los otros dominios de servicio de catálogo actúan como puntos remotos de reconciliación para los cambios que se producen en el dominio de servicio de catálogo raíz. El dominio de servicio de catálogo raíz puede arbitrar los cambios entre los dominios de servicio de catálogo. Si una topología de anillo contiene más de un anillo alrededor de un dominio de servicio de catálogo raíz, el dominio sólo puede arbitrar los cambios entre el anillo más interno. Sin embargo, los resultados del arbitraje se distribuyen por los dominios de servicio de catálogo de los otros anillos.

Topologías de hub y radio

Con una topología de hub y radio, los cambios viajan a través de un dominio de servicio de catálogo de hub. Debido a que el hub es el único dominio de servicio de catálogo intermedio especificado, las topologías de hub y radio tienen una latencia menor. El dominio de hub está conectado a cada dominio de radio mediante un enlace. El hub distribuye los cambios entre los dominios de servicio de catálogo. El hub actúa como un punto de reconciliación para las colisiones. En un entorno con una tasa de actualización alta, es posible que el hub necesite ejecutarse en más hardware que los radios para permanecer sincronizado. WebSphere eXtreme Scale está diseñado para escalar de forma lineal, lo que significa que puede ampliar el hub, según sea necesario, sin dificultad. Sin embargo, si el hub falla, los cambios no se distribuyen hasta que se reinicia el hub. Los cambios en los dominios de servicio de catálogo de radio se distribuirán una vez que se vuelva a conectar el hub.



También puede utilizar una estrategia con clientes completamente replicados, una variación de la topología que utiliza un par de servidores eXtreme Scale en ejecución como hub. Cada cliente crea una cuadrícula de datos de un solo contenedor autocontenida con un catálogo en la JVM de cliente. Un cliente utiliza su cuadrícula de datos para conectarse al catálogo de hub. Esta conexión hace que el cliente se sincronice con el hub tan pronto como el cliente obtenga una conexión del hub.

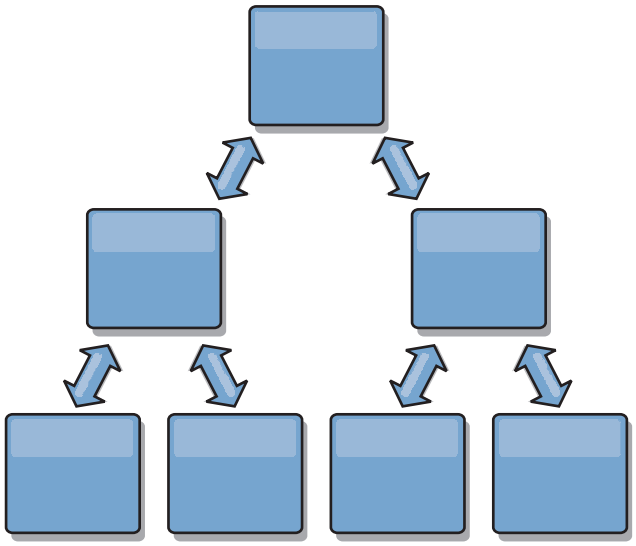
Los cambios realizados por el cliente son locales al cliente y se hace una réplica de ellos asíncrona en el hub. El hub actúa como un dominio de arbitraje, que distribuye los cambios a todos los clientes conectados. La topología de clientes completamente replicados proporciona una memoria caché L2 fiable para una correlacionador relacional de objetos como, por ejemplo, OpenJPA. Los cambios se distribuyen rápidamente entre las JVM de cliente a través del hub. Si el tamaño de memoria caché puede estar contenido en el espacio de almacenamiento intermedio disponible, la topología es una arquitectura fiable para este estilo de L2.

Utilice varias particiones para escalar el dominio del hub en varias JVM, si es necesario. Debido a que todos los datos aún deben caber en una única JVM de cliente, varias plataformas aumentan la capacidad del concentrador para distribuir y arbitrar cambios. Sin embargo, tener varias particiones no cambia la capacidad de un único dominio.

Topologías de árbol

También puede utilizar un árbol dirigido acíclico. Un árbol acíclico no tiene ciclos ni bucles, y una configuración dirigida limita los enlaces a los existentes solo entre padres e hijos. Esta configuración puede ser útil para topologías que tengan muchos dominios de servicio de catálogo y no es práctico tener un hub central que

esté conectado a todos los radios posibles. Este tipo de topología también puede resultar útil cuando debe añadir dominios de servicio de catálogo actualizando el dominio de servicio de catálogo raíz.



Una topología de árbol aún puede tener un punto central de reconciliación en el dominio de servicio de catálogo raíz. El segundo nivel aún puede funcionar como un punto remoto de reconciliación para los cambios que se producen en el dominio de servicio de catálogo por debajo de ellos. El dominio de servicio de catálogo raíz puede arbitrar los cambios entre los dominios de servicio de catálogo en el segundo nivel solamente. También puede utilizar árboles "n-ários", cada uno de los cuales tiene N hijos en cada nivel. Cada dominio de servicio de catálogo se conecta a *n* enlaces.

Clientes totalmente replicados

En esta variación de topología interviene un par de servidores eXtreme Scale que se ejecutan como un hub. Cada cliente crea una cuadrícula de datos de un solo contenedor autocontenida con un catálogo en la JVM de cliente. Un cliente utiliza su cuadrícula de datos para conectarse al catálogo de hub, lo que hace que el cliente se sincronice con el hub tan pronto como el cliente obtiene una conexión del hub.

Los cambios realizados por el cliente son locales al cliente y se hace una réplica de ellos asíncrona en el hub. El hub actúa como un dominio de arbitraje, que distribuye los cambios a todos los clientes conectados. La topología de clientes totalmente replicados proporciona una buena memoria caché L2 para un correlacionador relacional de objetos, como OpenJPA. Los cambios se distribuyen rápidamente entre las JVM de cliente a través del hub. Siempre que el tamaño de la memoria caché se pueda incluir en el espacio de almacenamiento dinámico disponible de los clientes, esta topología es una buena arquitectura para este estilo de L2.

Utilice varias particiones para escalar el dominio del hub en varias JVM, si es necesario. Dado que todos los datos todavía deben caber en una sola JVM cliente, el uso de varias particiones aumenta la capacidad del hub para distribuir y arbitrar los cambios, pero no cambia la capacidad de un dominio único.

Consideraciones sobre la configuración para topologías multimaestro

Considere los puntos siguientes cuando decida si desea utilizar topologías de réplica multimaestro y cómo utilizarlas.

- **Requisitos de conjunto de correlaciones**

  Los conjuntos de correlaciones deben tener las características siguientes para replicar cambios en todos los enlaces del dominio de servicio de catálogo:

  – El nombre de ObjectGrid y el nombre de conjunto de correlaciones de un dominio de servicio de catálogo deben coincidir con el nombre de ObjectGrid y el nombre de conjunto de correlaciones de otros dominios de servicio de catálogo. Por ejemplo, el ObjectGrid "og1" y el conjunto de correlaciones "ms1" deben estar configurados en el dominio de servicio de catálogo A y el dominio de servicio de catálogo B para replicar los datos del conjunto de correlaciones entre los dominios de servicio de catálogo.

  – Es una cuadrícula de datos `FIXED_PARTITION`. Las cuadrículas de datos `PER_CONTAINER` no se pueden replicar.

  – Tiene el mismo número de particiones en cada dominio de servicio de catálogo. El conjunto de correlaciones podría tener o no el mismo número y los mismos tipos de réplicas.

  – Tiene los mismos tipos de datos que se están replicando en cada dominio de servicio de catálogo.

  – Contiene las mismas correlaciones y plantillas de correlación dinámica en cada uno de los dominios de servicio de catálogo.

  – No utiliza el gestor de entidades. Un conjunto de correlaciones que contiene una correlación de entidades no se replica en todos los dominios de servicio de catálogo.

  – No utiliza el soporte de almacenamiento en memoria caché de grabación diferida. Un conjunto de correlaciones que contiene una correlación que está configurada con soporte de grabación diferida no se replica en todos los dominios de servicio de catálogo.

  Los conjuntos de correlaciones con las características anteriores empiezan la réplica una vez que se han iniciado los dominios de servicio de catálogo en la topología.

- **Cargadores de clases con varios dominios de servicio de catálogo**

  Los dominios de servicio de catálogo deben tener acceso a todas las clases utilizadas como claves y valores. Todas las dependencias se deben reflejar en todas las vías de acceso de clases para máquinas virtuales Java (JVM) de contenedor de cuadrícula de datos para todos los dominios. Si un plug-in CollisionArbiter recupera el valor para una entrada de memoria caché, las clases para los valores deben estar presentes para el dominio que inicia el árbitro.

Consideraciones sobre el cargador en una topología multimaestro

Cuando se utilizan cargadores en una topología multimaestro, debe considerar los posibles retos de mantenimiento de la información de revisión y colisión. La cuadrícula de atos mantiene información de revisión sobre los elementos de la cuadrícula de datos de forma que se pueden detectar las colisiones cuando otros fragmentos primarios de la configuración graban entradas en la cuadrícula de datos. Cuando se añaden entradas desde un cargador, esta información de revisión no se incluye y la entrada asume una revisión nueva. Debido a que la revisión de la entrada parece una inserción nueva, se produciría una falta colisión si otro fragmento primario también cambia este estado u obtiene la misma información de un cargador.

Los cambios de la réplica invocan el método get en el cargador con una lista de las claves que no están aún en la cuadrícula de datos pero que se han a cambiar durante la transacción de réplica. Cuando se produce la réplica, estas entradas son entradas de colisión. Cuando se arbitran las colisiones y se aplica la revisión, se llama a una actualización por lotes en el cargador para aplicar los cambios en la base de datos. Todas las correlaciones modificadas en la ventana de revisión se actualizan en la misma transacción.

Interrogante de la precarga

Considere una topología de dos centros de datos con el centro de datos A y el centro de datos B. Ambos centros de datos tienen bases de datos independientes, pero solo el centro de datos A tiene una cuadrícula de datos en ejecución. Al establecer un enlace entre los cnetros de datos para una configuración multimaestro, las cuadrículas de datos del centro de datos A inician el envío de datos a las nuevas cuadrículas de datos del centro de datos B, lo que causa una colisión con cada entrada. Otro problema importante que se produce es con los datos que se encuentra en la base de datos del centro de datos B pero no en la base de atos del centro de datos A. Estas filas no se llenan ni arbitran, lo que genera incoherencias que no se resuelven.

Solución al interrogante de la precarga

Debido a que los datos que se encuentran solo en la base de datos no puede tener revisiones, debe precargar siempre completamente la cuadrícula de datos desde la base de datos local antes de establecer un enlace multimaestro. A continuación, ambas cuadrículas de datos pueden revisar y arbitrar los datos, y finalmente llegar a un estado coherente.

Interrogante de la memoria caché escasa

Con una memoria caché escasa, en primer lugar la aplicación intenta encontrar datos en la cuadrícula de datos. Si los datos no se encuentran en la cuadrícula de datos, se buscan los dato s en la base de atos utilizando el cargador. Se desalojan periodicamente entradas de la cuadrícula de datos para mantener un tamaño pequeño de la memoria caché.

Este tipo de memoria caché puede ser problemático en un escenario de configuración multimaestro porque las entradas de la cuadrícula de datos tienen metadatos de revisión que le ayudarán a detectar qué colisiones se producen y qué lado ha realizado cambios. Cuando los enlaces entre los centros de datos no funcionan, un centro de datos puede actualizar una entrada y a continuación en última instancia actualizar la base de datos e invalidar la entrada en la cuadrícula de datos. Cuando se recupera el enlace, los centros de datos intentan sincronizar las revisiones entre ellos. Sin embargo, debido a que la base de datos se ha actualizado y la entrada de la cuadrícula de datos se ha invalidado, el cambio se pierde desde la perspectiva del centro de datos que se ha interrumpido. Como resultado, los dos lados de la cuadrícula de datos están desincronizados y no son coherentes.

Solución a los interrogantes de memoria caché escasa

Topología y hub y radio:

Puede ejecutar el cargador solo en el hub de una topología de hub y radio, lo que mantiene la coherencia de los datos al mismo tiempo que se escala la cuadrícula de

datos. Sin embargo, si está considerando el despliegue, tenga en cuenta que los cargadores pueden permitir que la cuadrícula de datos se cargue parcialmente, lo que significa que se ha configurado el desalojador. Si los radios de la configuración son memorias caché escasas pero no tienen cargadores, las faltas de coincidencia de memoria caché no tienen ninguna manera de recuperar los datos de la base de datos. Debido a esta restricción, debe utilizar una topología de memoria caché llenada completamente con una configuración de hub y radio.

Invalidaciones y desalojo

La invalidación crea coherencia entre la cuadrícula de datos y la base de datos. Los datos se pueden eliminar de la cuadrícula de datos mediante programación o con desalojo. Al desarrollar la aplicación, debe tener en cuenta que el manejo de revisiones no replica los cambios que se han invalidado, lo que genera incoherencias entre fragmentos primarios.

Los sucesos de invalidación no son cambios de estado de memoria caché y no generan réplica. Los desalojadores configurados se ejecutan independientemente de otros desalojadores de la configuración. Por ejemplo, podría tener un desalojador configurado para un umbral de memoria en un dominio de servicio de catálogo, pero un tipo distinto de desalojador menos agresivo en el servicio de catálogo enlazado. Cuando se eliminan entradas de cuadrícula de datos debido a la política de umbral de memoria, las entradas del otro dominio de servicio de catálogo no resultan afectadas.

Actualizaciones de base de datos e invalidación de cuadrícula de datos

Se producen problemas al actualizar la base de datos directamente en segundo plano al llamar a la invalidación en la cuadrícula de datos para las entradas actualizadas en una configuración multimaestro. Este problema se produce porque la cuadrícula de datos no puede replicar el cambio en otros fragmentos primarios hasta que algún tipo de acceso de memoria caché mueve la entrada a la cuadrícula de datos.

Varios grabadores en una única base de datos lógica

Cuando utiliza una única base de datos con varios fragmentos primarios que se conectan mediante un cargador, se producen conflictos de transacciones. La implementación de cargador debe manejar especialmente estos tipos de escenarios.

Duplicación de datos utilizando réplica multimaestro

Puede configurar bases de datos independientes conectadas a dominios de servicio de catálogo independiente. En esta configuración, el cargador puede enviar cambios de un centro de datos al otro centro de datos.

Consideraciones sobre el diseño para la réplica multimaestro

Al implementar la réplica multimaestro, debe tener en cuenta aspectos del diseño como los siguientes: arbitraje, enlace y rendimiento.

Consideraciones sobre arbitraje en el diseño de topología

Se podrían producir colisiones de cambio si se pueden cambiar en dos lugares a la vez los mismos registros. Configure cada uno de los dominios de servicio de catálogo para que tenga aproximadamente la misma cantidad de recursos de procesador, memoria y red. Podría observar que los dominios de servicio de

catálogo que realicen el manejo de colisiones de cambio (arbitraje) utilicen más recursos que otros dominios de servicio de catálogo. Las colisiones se detectan automáticamente. Se manejan con uno de dos mecanismos:

- **Árbitro de colisión predeterminado**: el protocolo predeterminado utilizará los cambios del dominio de servicio de catálogo con el nombre léxicamente inferior. Por ejemplo, si los dominios de servicio de catálogo A y B generan un conflicto para un registro, el cambio del dominio de servicio de catálogo B se ignorará. El dominio de servicio de catálogo A mantiene su versión y el registro en el dominio de servicio de catálogo B se cambia para que coincida con el registro del dominio de servicio de catálogo A. Este comportamiento se aplica también a las aplicaciones en las que los usuarios o sesiones normalmente se enlazan o tienen una afinidad con una de las cuadrículas siguientes.
- **Árbitro de colisiones personalizado**: las aplicaciones pueden proporcionar un árbitro personalizado. Cuando un dominio de servicio de catálogo detecta una colisión, se inicia un árbitro. Para obtener información sobre cómo desarrollar un árbitro personalizado útil, consulte Desarrollo de árbitros personalizados para la réplica con varios maestros.

Para topologías en las que las colisiones son posibles, considere implementar una topología de hub y radio o una topología de árbol. Estas dos topologías son propicias para evitar colisiones constantes, lo que puede suceder en los escenarios siguientes:

1. Varios dominios de servicio de catálogo sufren una colisión
2. Cada dominio de servicio de catálogo maneja la colisión localmente, lo que genera revisiones
3. Las revisiones colisionan, con lo que se producen revisiones de revisiones

Para evitar colisiones, elija un dominio de servicio de catálogo específico, denominado un *dominio de servicio de catálogo de arbitraje* como el árbitro de colisión para un subconjunto de dominios de servicio de catálogo. Por ejemplo, una topología de hub y radio podría utiliza el hub como el manejador de colisiones. El manejador de colisiones de radio ignora las colisiones detectadas por los dominios de servicio de catálogo de radio. El dominio de servicio de catálogo de hub crea revisiones, lo que evita revisiones de colisiones inesperadas. El dominio de servicio de catálogo que se asigna para manejar colisiones debe enlazar a todos los dominios para los que es responsable para manejar colisiones. En una topología de árbol, los dominios padre internos manejan colisiones para sus hijos inmediatos. Por el contrario, si utiliza una topología en anillo, no puede designar un dominio de servicio de catálogo en el anillo como el árbitro.

En la tabla siguiente se resumen los enfoques de arbitraje que son más compatibles con distintas topologías.

Tabla 1. Enfoques de arbitraje. En esta tabla se indica si el arbitraje de la aplicación es compatible con distintas tecnologías.

Topología	¿Arbitraje de aplicación?	Notas
Una línea de dos dominios de servicio de catálogo	Sí	Elija un dominio de servicio de catálogo como árbitro.

Tabla 1. Enfoques de arbitraje (continuación). En esta tabla se indica si el arbitraje de la aplicación es compatible con distintas tecnologías.

Topología	¿Arbitraje de aplicación?	Notas
Una línea de tres dominios de servicio de catálogo	Sí	El dominio de servicio de catálogo intermedio debe ser el árbitro. Considere el dominio de servicio de catálogo intermedio como hub en una topología de hub y radio simple.
Una línea de más de tres dominios de servicio de catálogo	No	No se admite el arbitraje de aplicaciones.
Un hub con N radios	Sí	El hub con enlaces a todos los radios debe ser el dominio de servicio de catálogo de arbitraje.
Un anillo de N dominios de servicio de catálogo	No	No se admite el arbitraje de aplicaciones.
Un árbol dirigido acíclico (árbol n-ario)	Sí	Todos los nodos raíz deben evaluar solo sus descendientes directos.

Consideraciones sobre enlaces en el diseño de topología

De forma ideal, una topología incluye el número mínimo de enlaces cuando optimiza los compromisos entre las características de latencia de cambios, tolerancia a errores y rendimiento.

- **Latencia de cambios**

  La latencia de cambios la determina el número de dominios de servicio de catálogo intermedio por los que debe pasar un cambio antes de llegar a un dominio de servicio de catálogo específico.

  Una topología tiene la mejor latencia de cambios cuando elimina dominios de servicio de catálogo intermedios enlazando cada dominio de servicio de catálogo a cada uno de los otros dominios de servicio de catálogo. Sin embargo, un dominio de servicio de catálogo debe realizar trabajo de réplica en proporción a su número de enlaces. Para topologías grandes, el gran número de enlaces que se definirán puede causar carga administrativa.

  La velocidad a la que se copia un cambio en otros dominios de servicio de catálogo depende de factores adicionales, como por ejemplo:

  – Procesador y ancho de banda de red en el dominio de servicio de catálogo de origen
  – Número de dominios de servicio de catálogo intermedios y enlaces entre los dominios de servicio de catálogo de origen y de destino
  – Recursos de procesador y de red disponibles a los dominios de servicio de catálogo de origen, de destino e intermedio

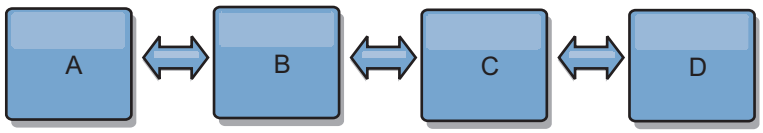
- **Tolerancia al error**

  La tolerancia a errores la determina el número de vías de acceso existentes entre los dos dominios de servicio de catálogo para la réplica de cambios.

  Si solo tiene un enlace entre un par determinado de dominios de servicio de catálogo, una anomalía de enlace no permite la propagación de cambios. De forma similar, los cambios no se propagan entre los dominios de servicio de catálogo si alguno de los dominios intermedios experimenta anomalía de enlace. La topología podría tener un único enlace desde un dominio de servicio de

catálogo a otro de tal forma que el enlace pase por dominios intermedios. Si es así, los cambios no se propagarán si alguno de los dominios de servicio de catálogo intermedios está inactivo.

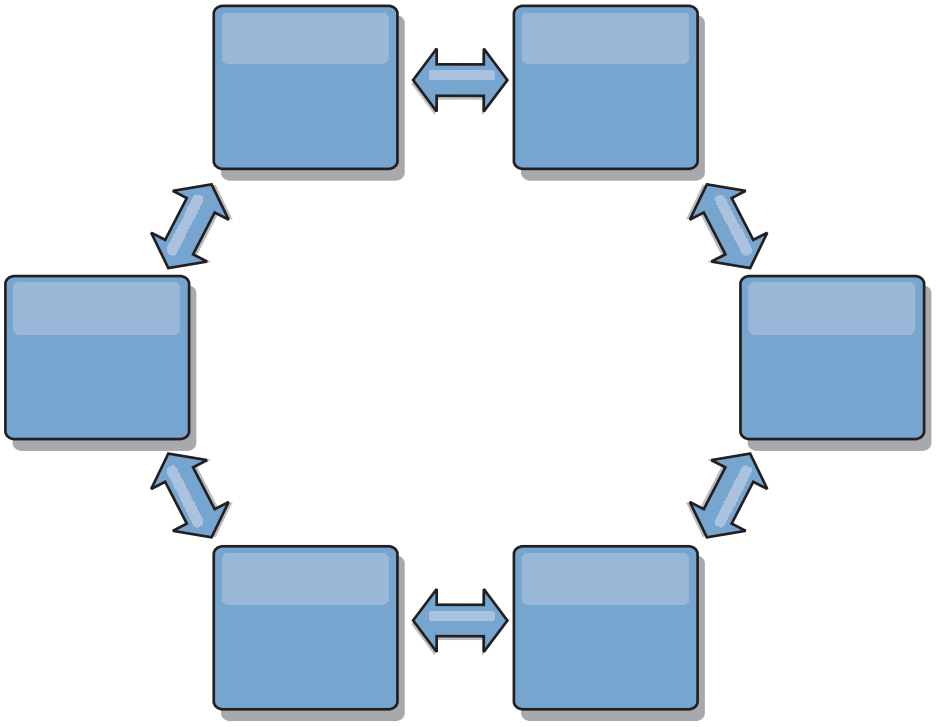
Considere la topología de línea con cuatro dominios de servicio de catálogo A, B, C, y D:



Si se mantiene alguna de estas condiciones, el Dominio D no verá los cambios de A:

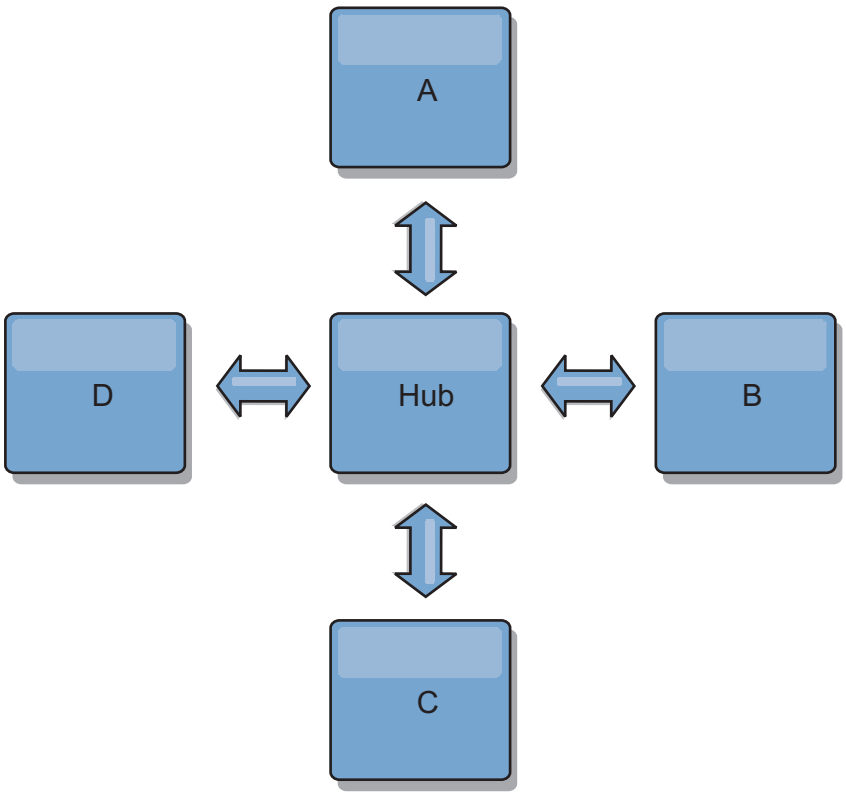
- El dominio A está activo y el dominio B está inactivo
- Los dominios A y B están activos y el dominio C está inactivo
- El enlace entre A y B está inactivo
- El enlace entre B y C está inactivo
- El enlace entre C y D está inactivo

En cambio, con una topología de anillo, cada uno de los dominios de servicio de catálogo puede recibir cambios desde cualquier dirección.



Por ejemplo, si un servicio de catálogo determinado de la topología de anillo está inactivo, los dos dominios adyacentes aún pueden obtener cambios directamente uno del otro.

Todos los cambios se propagan mediante el hub. Por lo tanto, a diferencia de las topologías de línea y de anillo, el diseño de hub y radio puede desglosarse, si el hub falla.



Un único dominio de servicio de topología es resistente a una determinada cantidad de pérdida de servicio. Sin embargo, anomalías mayores como interrupciones de la red amplia o la pérdida de enlaces entre centros de datos físicos puede interrumpir cualquiera de los dominios de servicio de catálogo.

- **Enlace y rendimiento**

  El número de enlaces definidos en un dominio de servicio de catálogo afecta al rendimiento. Más enlaces utilizan más recursos y como resultado el rendimiento de la réplica puede disminuir. La posibilidad de recuperar cambios para un dominio A mediante otros dominios libera de forma efectiva al dominio A de tener que replicar las transacciones en todas partes. La carga de distribución de cambios de un dominio está limitada por el número de enlaces que utiliza, no por cuántos dominios haya en la topología. Esta propiedad de carga proporciona escalabilidad, de forma que los dominios de la topología pueden compartir la carga de la distribución de cambios.

  Un dominio de servicio de catálogo puede recuperar los cambios indirectamente mediante otros dominios de servicio de catálogo. Considere una topología de línea con cinco dominios de servicio de catálogo.

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

  – A extrae los cambios de B, C, D y E a B
  – B extrae los cambios de A y C directamente y los cambios de D y E a C
  – C realiza los cambios de B y D directamente y los cambios de A a B y de E a D
  – D extrae los cambios de C y E directamente y los cambios de A y B a C
  – E extrae los cambios de D directamente, y los cambios de A, B y C a D

  La carga de distribución de los dominios de servicio de catálogo A y E es la menor, ya que cada uno de ellos tiene un enlace a un único dominio de servicio de catálogo. Cada uno de los dominios B, C y D tiene un enlace a dos dominios. Por lo tanto, la carga de distribución de los dominios B, C y D es el doble de la

carga de los dominios A y E. La carga de trabajo depende del número de enlaces de cada dominio, no del número global de dominios de la topología. Por lo tanto, la distribución de cargas descrita permanecería constante, incluso si la línea contuviera 1000 dominios.

Consideraciones sobre el rendimiento de réplica multimaestros

Tenga en cuenta las limitaciones siguientes cuando utilice topologías de réplica multimaestro:

- **Cambiar ajuste de distribución**, se trata en la sección anterior.
- **Rendimiento de enlace de réplica** WebSphere eXtreme Scale crea un único socket TCP/IP entre cualquier par de JVM. Todos el tráfico entre las JVM se produce entre el único socket, incluido tráfico de la réplica multimaestro. Los dominios de servicio de catálogo se alojan en como mínimo *n* JVM de contenedor, lo que proporciona como mínimo *n* enlaces TCP a dominios de servicio de catálogo de igual. Por lo tanto, los dominios de servicio de catálogo con una gran cantidad de contenedores tienen niveles más altos de rendimiento de la réplica. Más contenedores requieren más recursos de procesador y red.
- **Ajuste de la ventana deslizante TCP y RFC 1323** El soporte de RFC 1323 en ambos extremos de un enlace proporciona más datos para un viaje de ida y vuelta. Este soporte produce un mejor rendimiento, ampliando la capacidad de la ventana en un factor de aproximadamente 16.000.

  Recuerde que los sockets TCP utilizan un mecanismo de ventana deslizando para controlar el flujo de datos masivo. Este mecanismo normalmente limita el socket a 64 KB para un intervalo de viaje de ida y vuelta. Si el intervalo de viaje de ida y vuelta es 100 ms, el ancho de banda se limita a 640 KB/segundo sin ajuste adicional. El uso de todo el ancho de banda disponible en un enlace podría requerir un ajuste específico de un sistema operativo. La mayoría de sistemas operativos incluyen parámetros de ajuste, incluidas las opciones de RFC 1323, para ampliar el rendimiento sobre los enlaces de latencia alta.

  Varios factores pueden afectar al rendimiento de la réplica:

  – La velocidad a la que eXtreme Scale recupera cambios.
  – La velocidad a la que eXtreme Scale puede dar servicio a solicitudes de recuperación de réplica.
  – La capacidad de la ventana deslizante.
  – Con el ajuste de almacenamiento intermedio de red en ambos lados de un enlace, eXtreme Scale recupera cambios sobre el socket de forma eficiente.
- **Serialización de objetos** Todos los datos deben ser serializables. Si un dominio de servicio de catálogo no utiliza `COPY_TO_BYTES`, el dominio de servicio de catálogo debe utilizar Java o ObjectTransformers para optimizar el rendimiento de serialización.
- **Compresión** WebSphere eXtreme Scale comprime todos los datos enviados entre dominios de servicio de catálogo de forma predeterminada. La inhabilitación de la compresión no está disponible actualmente.
- **Ajuste de la memoria** El uso de memoria para una topología de réplica multimaestro es considerablemente independiente del número de dominios de servicio de catálogo de la topología.

  La réplica multimaestro añade una cantidad fija de proceso por entrada Map para manejar el mantenimiento de versiones. Cada contenedor también realiza un seguimiento de una cantidad fija de datos para cada dominio de servicio de catálogo de la topología. Una topología con dos dominios de servicio de catálogo utiliza aproximadamente la misma memoria que una topología con 50 dominios de servicio de catálogo. WebSphere eXtreme Scale no utiliza registros

de reproducción o colas similares en su implementación. Por lo tanto, no hay ninguna estructura de recuperación lista en el caso de que un enlace de réplica no esté disponible durante el periodo de tiempo considerable y se reinicie posteriormente.

Interoperatividad con otros productos WebSphere

Puede integrar WebSphere eXtreme Scale con otros productos de servidor como, por ejemplo, WebSphere Application Server y WebSphere Application Server Community Edition.

WebSphere Application Server

Puede integrar WebSphere Application Server con diversos aspectos de la configuración de WebSphere eXtreme Scale. Puede desplegar aplicaciones de cuadrícula de datos y utilizar WebSphere Application Server para alojar los servidores de contenedor y catálogo. También puede utilizar la seguridad de WebSphere Application Server en el entorno de WebSphere eXtreme Scale.

WebSphere Portal

Puede persistir sesiones HTTP de WebSphere Portal en una cuadrícula de datos en WebSphere eXtreme Scale.

WebSphere Application Server Community Edition

WebSphere Application Server Community Edition puede compartir el estado de sesión, pero no de una forma eficaz y escalable. WebSphere eXtreme Scale proporciona un alto rendimiento, una capa de persistencia distribuida que puede utilizarse para replicar el estado, pero que no se integra fácilmente con otro servidor de aplicaciones fuera de WebSphere Application Server. Puede integrar estos dos productos para proporcionar una solución de gestión de sesiones escalable.

WebSphere Real Time

Con el soporte de WebSphere Real Time, la oferta Java de tiempo real líder del sector, WebSphere eXtreme Scale, permite a las aplicaciones Extreme Transaction Processing (XTP) tener tiempos de respuesta coherentes y predecibles.

Planificación de la instalación

Antes de instalar el producto, debe tener en cuenta su entorno.

Requisitos de hardware y software

Examine una visión general de requisitos de hardware y de sistema operativo. Aunque no es necesario que utilice un nivel específico de hardware o sistema operativo para WebSphere eXtreme Scale, están disponibles opciones de hardware y software soportadas formalmente en la página Systems Requirements (Requisitos de sistema) del sitio de soporte del producto. Si existe un conflicto entre el Information Center y la página de requisitos de sistema, tiene prioridad la información del sitio web. La información de requisitos previos en el centro de información sólo se proporciona por comodidad.

Consulte la página Requisitos del sistema para ver el conjunto oficial de requisitos de hardware y software.

No es necesario que instale y despliegue eXtreme Scale en un nivel específico de sistema operativo. Cada instalación de Java Platform, Standard Edition (Java SE) y Java Platform, Enterprise Edition (Java EE) necesita diferentes niveles de sistema operativo o arreglos.

Puede instalar y desplegar el producto en los entornos de Java EE y Java SE. También puede empaquetar el componente de cliente con las aplicaciones Java EE directamente si integrarse con WebSphere Application Server. WebSphere eXtreme Scale soporta Java SE 5 o posteriores y WebSphere Application Server Versión 6.1 y posteriores.

Requisitos de hardware

WebSphere eXtreme Scale no requiere un nivel específico de hardware. Los requisitos de hardware dependen del hardware soportado para la instalación de Java Platform, Standard Edition que utiliza para ejecutar WebSphere eXtreme Scale. Si utiliza eXtreme Scale con WebSphere Application Server u otra implementación de Java Platform, Enterprise Edition, los requisitos de hardware de estas plataformas son suficientes para WebSphere eXtreme Scale.

Requisitos de sistema operativo

- **Sin la consola web**

  eXtreme Scale no requiere un nivel específico de sistema operativo. Cada implementación de Java SE y Java EE requiere niveles o arreglos distintos de sistema operativo para problemas que se han descubierto durante la comprobación de la implementación de Java. Los niveles necesarios para estas implementaciones son suficientes para eXtreme Scale.
- **Con la consola web**

  Los requisitos siguientes se aplican a cada sistema operativo si se utiliza la consola:
  - Linux: JVM de 32 bits o 64 bits
  - Linux PPC: sólo JVM de 32 bits
  - Windows: sólo JVM de 32 bits
  - AIX: sólo JVM de 32 bits

Requisitos del navegador web

La consola web da soporte a los siguientes navegadores web:

- Mozilla Firefox, versión 3.5.x y posteriores
- Mozilla Firefox, versión 3.6.x y posteriores
- Microsoft Internet Explorer, versión 7 o 8

Requisitos de WebSphere Application Server

- WebSphere Application Server Versión 6.1.0.39 o posterior
- WebSphere Application Server Versión 7.0.0.19 o posterior
- WebSphere Application Server Versión 8.0.0.1 o posterior

Consulte los Arreglos recomendados para WebSphere Application Server si desea más información.

Requisitos de otros servidores de aplicaciones

Otras implementaciones de Java EE pueden utilizar el tiempo de ejecución de eXtreme Scale como una instancia local o como un cliente para los servidores eXtreme Scale. Para implementar Java SE, debe utilizar la versión 5 o posterior.

Consideraciones sobre Java SE

WebSphere eXtreme Scale requires Java SE 5 o posterior. En general, las nuevas versiones de Java SE tienen mejor funcionalidad y rendimiento.

Versiones soportadas

Puede utilizar WebSphere eXtreme Scale conJava SE 5 o posterior. La versión que utilice debe estar soportada actualmente por el proveedor JRE (Java Runtime Environment).

Se instala un JRE completamente soportado como parte de las instalaciones autónomas de WebSphere eXtreme Scale y WebSphere eXtreme Scale Client en el directorio *raíz_intal_wxs*/java y está disponible para ser utilizado por clientes y servidores. Si está instalando WebSphere eXtreme Scale en WebSphere Application Server, puede utilizar el JRE incluido en la instalación de WebSphere Application Server.

WebSphere eXtreme Scale aprovecha la función de Java Development Kit (JDK) 5 o posterior cuando pasa a estar disponible. Normalmente, las versiones más nuevas de Java Development Kit (JDK) y Java SE tiene mejor rendimiento y funcionalidad.

Consulte Software soportado para obtener más información.

Características dependientes de Java de WebSphere eXtreme Scale

Tabla 2. Características que requieren Java SE 5 o Java SE 6.

WebSphere eXtreme Scale utiliza una funcionalidad que se ha introducido en Java SE 5 o Java SE 6 para proporcionar las siguientes características del producto.

Característica	**Soportada en Java SE 5 y posterior**	**Soportada en Java SE 6 y posterior**
Anotaciones de la API EntityManager (Opcional: también puede utilizar archivos XML)	X	X
Java Persistence API (JPA): cargador JPA, cargador de clientes JPA y actualizador basado en tiempo JPA	X	X
Desalojo basado en memoria (utiliza MemoryPoolMXBean)	X	X

Tabla 2. Características que requieren Java SE 5 o Java SE 6 (continuación).

WebSphere eXtreme Scale utiliza una funcionalidad que se ha introducido en Java SE 5 o Java SE 6 para proporcionar las siguientes características del producto.

Característica	Soportada en Java SE 5 y posterior	Soportada en Java SE 6 y posterior
Agentes de instrumentación: • `wxssizeagent.jar`: aumenta la precisión de las métricas de correlaciones de bytes utilizadas. • `ogagent.jar`: aumenta el rendimiento de las entidades de acceso a campos.	X	X
Consola web para la supervisión		X

Consideraciones sobre Java EE

Mientras se prepara para integrar WebSphere eXtreme Scale en un entorno Java Platform, Enterprise Edition, debe tener en cuenta ciertos elementos, como versiones, opciones de configuración, requisitos y limitaciones y desarrollo y gestión de aplicaciones.

Ejecución de aplicaciones de eXtreme Scale en un entorno Java EE

Una aplicación Java EE puede conectarse a una aplicación de eXtreme Scale remota. Además, el entorno de WebSphere Application Server permite el inicio de un servidor eXtreme Scale mientras se inicia una aplicación en el servidor de aplicaciones.

Si utiliza un archivo XML para crear una instancia de ObjectGrid y el archivo XML está en el módulo del archivador empresarial (EAR), acceda al archivo mediante el método getClass().getClassLoader().getResource("META-INF/objGrid.xml") para obtener un objeto URL y utilizarlo para crear una instancia de ObjectGrid. Substituya el nombre del archivo XML que utilice en la llamada de método.

Puede utilizar beans de arranque para que una aplicación cree una rutina de carga para una instancia de ObjectGrid cuando una aplicación se inicie y para que destruya la instancia de ObjectGrid al detenerse la aplicación. Un bean de arranque es un bean de sesión sin estado con una ubicación remota com.ibm.websphere.startupservice.AppStartUpHome y una interfaz remota com.ibm.websphere.startupservice.AppStartUp. La interfaz remota tiene dos métodos: el método start y el método stop. Utilice el método start para crear una rutina de carga de la instancia y utilice el método stop para destruir la instancia. La aplicación utiliza el método ObjectGridManager.getObjectGrid para mantener una referencia a la instancia. Consulte la información sobre cómo acceder a un ObjectGrid con ObjectGridManager en la *Guía de programación* para obtener más información.

Uso de cargadores de clases

Cuando los módulos de aplicación que utilizan cargadores de clases diferentes comparten una sola instancia de ObjectGrid en una aplicación Java EE, compruebe que los objetos que se almacenan en eXtreme Scale y los plug-ins para el producto están en un cargador común en la aplicación.

Gestión del ciclo de vida de las instancias de ObjectGrid en un servlet

Para gestionar el ciclo de vida de una instancia de ObjectGrid en un servlet, puede utilizar el método init para crear la instancia y el método destroy para eliminar la instancia. Si la instancia se almacena en memoria caché, se recupera y manipula en el código del servlet. Consulte la información sobre cómo acceder a un ObjectGrid con la interfaz ObjectGridManager en la *Guía de programación* para obtener más información.

Convenios de directorio

Se utilizan los siguientes convenios de directorio en toda la documentación para hacer referencia a directorios como por ejemplo *raíz_instalación_wxs* e *inicio_wxs*. Accede a estos directorios durante distintos escenarios, incluido durante la instalación y la utilización de las herramientas de línea de mandatos.

raíz_intal_wxs
El directorio *raíz_instalación_wxs* es el directorio raíz donde se instalan los archivos del producto WebSphere eXtreme Scale. El directorio *raíz_instalación_wxs* puede ser el directorio en el que se extrae el archivado de prueba o el directorio en el que se instala el producto WebSphere eXtreme Scale.

- Ejemplo al extraer la prueba:

  Ejemplo: `/opt/IBM/WebSphere/eXtremeScale`
- Ejemplo cuando se instala WebSphere eXtreme Scale en un directorio autónomo:

  Ejemplo: `/opt/IBM/eXtremeScale`
- Ejemplo cuando se integra WebSphere eXtreme Scale con WebSphere Application Server:

  Ejemplo: `/opt/IBM/WebSphere/AppServer`

inicio_wxs
El directorio *inicio_wxs* es el directorio raíz de los componentes, ejemplos y bibliotecas del producto WebSphere eXtreme Scale. Este directorio es el mismo que el directorio *raíz_instalación_wxs* cuando se ha extraído la versión de prueba. Para instalaciones autónomas, el directorio *inicio_wxs* es el subdirectorio `ObjectGrid` del directorio *raíz_instalación_wxs*. Para instalaciones integradas con WebSphere Application Server, este directorio es el directorio `optionalLibraries/ObjectGrid` del directorio *raíz_instalación_wxs*.

- Ejemplo al extraer la prueba:

  Ejemplo: `/opt/IBM/WebSphere/eXtremeScale`
- Ejemplo cuando se instala WebSphere eXtreme Scale en un directorio autónomo:

  Ejemplo: `/opt/IBM/eXtremeScale/ObjectGrid`
- Ejemplo cuando se integra WebSphere eXtreme Scale con WebSphere Application Server:

Ejemplo: /opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid

raíz_was
El directorio *raíz_was* es el directorio raíz de una instalación de WebSphere Application Server:

Ejemplo: /opt/IBM/WebSphere/AppServer

inicio_servicioRest
El directorio *inicio_servicioRest* es el directorio en el que se encuentran las bibliotecas y los ejemplos del servicio de datos REST de WebSphere eXtreme Scale. Este directorio se denomina `restservice` y es un subdirectorio del directorio *inicio_wxs*.

- Ejemplo para despliegues autónomos:
  Ejemplo: /opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice
- Ejemplo para despliegues integrados de WebSphere Application Server:
  Ejemplo: /opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice

raíz_tomcat
raíz_tomcat es el directorio raíz de la instalación de Apache Tomcat.

Ejemplo: /opt/tomcat5.5

raíz_wasce
raíz_wasce es el directorio raíz de la instalación de WebSphere Application Server Community Edition.

Ejemplo: /opt/IBM/WebSphere/AppServerCE

inicio_java
inicio_java es el directorio raíz de una instalación de Java Runtime Environment (JRE).

Ejemplo: /opt/IBM/WebSphere/eXtremeScale/java

inicio_samples
inicio_samples es el directorio en el que extrae los archivos de ejemplo que se utilizan para las guías de aprendizaje.

Ejemplo: /wxs-samples/

raíz_dvd
El directorio *raíz_dvd* es el directorio raíz del DVD que contiene el producto.

Ejemplo: raíz_dvd/docs/

raíz_equinox
El directorio *raíz_equinox* es el directorio raíz de la instalación de infraestructura OSGi de Eclipse Equinox.

Ejemplo:/opt/equinox

inicio_usuario
El directorio *inicio_usuario* es la ubicación donde se almacenan los archivos de usuario, por ejemplo los perfiles de seguridad.

Windows c:\Documents and Settings*nombre_usuario*

UNIX /home/*nombre_usuario*

Planificación de la capacidad del entorno

Si tiene un tamaño de conjunto de datos inicial y un tamaño de conjunto de datos proyectado, puede planificar la capacidad que necesita para ejecutar WebSphere eXtreme Scale. Mediante estos ejercicios de planificación, puede desplegar WebSphere eXtreme Scale de forma eficaz para futuros cambios, lo que le permite maximizar la elasticidad de la cuadrícula de datos, que no tendría con un escenario distinto como, por ejemplo, una base de datos en memoria u otro tipo de base de datos.

Dimensionamiento de la memoria y cálculo del número de particiones

Puede calcular la cantidad de memoria y particiones necesarias para la configuración específica.

Atención: Este tema es aplicable cuando no utiliza la modalidad de copia COPY_TO_BYTES. Si utiliza la modalidad de copia COPY_TO_BYTES, el tamaño de memoria es mucho menor y el procedimiento de cálculo es distinto.

WebSphere eXtreme Scale almacena datos dentro del espacio de direcciones de Mäquinas virtuales Java (JVM). Cada JVM proporciona espacio de procesador para atender a llamadas para crear, recuperar, actualizar y suprimir, para los datos que están almacenados en la JVM . Además, cada JVM proporciona espacio de memoria para réplicas y entradas de datos. Los objetos Java varían en tamaño, por lo tanto, debe realizar una medición para calcular la cantidad de memoria necesaria.

Para calcular el tamaño de la memoria necesaria, cargue los datos de aplicación en una sola JVM . Cuando el uso del almacenamiento dinámico alcanza el 60%, anote el número de objetos que se utilizan. Este número máximo de objetos recomendado para cada una de las Mäquinas virtuales Java. Para obtener el tamaño más preciso, utilice datos realistas e incluya todos los índices definidos en el tamaño porque los índices también consumen memoria. El mejor forma de dimensionar el uso de memoria es ejecutar la salida **verbosegc** de la recogida de basura, ya que esta salida le proporciona los números después de la recogida. Puede consultar el uso del almacenamiento dinámico en cualquier momento mediante los MBeans o mediante programación, pero estas consultas le proporcionan solo una instantánea actual del almacenamiento dinámico. Es posible que esta instantánea incluya basura sin recoger, así que la utilización de ese método no es una indicación precisa de la memoria consumida.

Dimensionamiento de la configuración

Número de fragmentos por partición (valor de numShardsPerPartition)

Para calcular el número de fragmentos por partición, o el valor de numShardsPerPartition, añada 1 para el fragmento primario además del número total de fragmentos de réplica que desea.

```
numShardsPerPartition = 1 + número_total_de_réplicas
```

Número de Mäquinas virtuales Java (valor minNumJVMs)

Para dimensionar la configuración, primero decida sobre el número máximo de objetos que es necesario almacenar en total. Para determinar el número de Mäquinas virtuales Java que necesita, utilice la siguiente fórmula:

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

Redondee este valor al valor entero más cercano.

Número de fragmentos (valor de numShards)

En el tamaño de crecimiento final, utilice 10 fragmentos para cada JVM . Como se ha descrito anteriormente, cada JVM tiene un fragmento primario y (N-1) fragmentos para las réplicas o, en este caso, nuevo réplicas. Puesto que ya tiene un número de Mäquinas virtuales Java para almacenar los datos, puede multiplicar el número de Mäquinas virtuales Java por 10 para determinar el número de fragmentos:

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

Número de particiones

Si una partición tiene un fragmento primario y un fragmento de réplica, la partición tiene dos fragmentos (primario y réplica). El número particiones es el total de fragmentos dividido por 2, redondeado por arriba hasta el número primo más cercano. Si la partición tiene un fragmento primario y dos réplicas, el número de particiones es el total de fragmentos dividido por 3, redondeado por arriba hasta el número primo más cercano.

```
numPartitions = numShards / numShardsPerPartition
```

Ejemplo de dimensionamiento

En este ejemplo, el número de entradas empieza en 250 millones. Cada año, el número de entradas aumenta aproximadamente un 14%. Después de siete años, el número total de entradas es de 500 millones, así que debe planificar la capacidad de la forma correspondiente. Para alta disponibilidad, es necesario una sola réplica. Con una réplica, el número de entradas se dobla, o 1.000.000,000 entradas. Como prueba, dos millones de entradas pueden almacenarse en cada JVM . Si se utilizan los cálculos en este escenario, es necesaria la siguiente configuración:

- 500 Mäquinas virtuales Java para almacenar el número final de entradas.
- 5000 fragmentos, que se calculan multiplicando 500 Mäquinas virtuales Java por 10.
- 2500 particiones, o 2503 como el siguiente número primo más cercado, que se calculan tomando 5000 fragmentos divididos por dos para los fragmentos primario y de réplica.

Inicio de la configuración

En función de los cálculos anteriores, empiece con 250 Mäquinas virtuales Java y vaya creciendo hacia 500 Mäquinas virtuales Java a lo largo de cinco años. Con esta configuración, puede gestionar el crecimiento incremental hasta que llegue al número final de entradas.

En esta configuración, se almacenan aproximadamente 200.000 entradas por partición (500 millones de entradas divididas entre 2503 particiones). Establezca el parámetro **numberOfBuckets** en la correlación que aloja las entradas en el número primo más alto más cercano, en este ejemplo 70887, lo que mantiene la proporción en 3 aproximadamente.

Cuando se alcanza el máximo número de Mäquinas virtuales Java

Cuando llegue al número máximo de 500 Mäquinas virtuales Java, aún podrá seguir aumentando el tamaño de la cuadrícula de datos. Como el número de Mäquinas virtuales Java aumentas hasta superar 500, el total de fragmentos empieza a caer por debajo de 10 para cada JVM , que está por debajo del número recomendado. Los fragmentos empiezan a crecer, lo que puede causar problemas. Repita el proceso de dimensionamiento considerando de nuevo el crecimiento futuro, y restablezca el recuento de particiones. Éste método requiere un reinicio completo o una interrupción de la cuadrícula de datos.

Número de servidores

Atención: No utilice la transferencia de páginas en un servidor en ninguna circunstancia.

Una sola JVM utiliza más memoria que el tamaño de almacenamiento dinámico. Por ejemplo, 1 GB de almacenamiento dinámico para una JVM en realidad utiliza 1,4 GB de memoria real. Determine la RAM libre disponible en el servidor. Divida la cantidad de RAM por la memoria por JVM para obtener el número máximo de Mäquinas virtuales Java en el servidor.

Tamaño de CPU por partición en transacciones

Aunque una funcionalidad principal de eXtreme Scale es su capacidad de escaladas elásticas, también es importante considerar el dimensionamiento y ajustar el número ideal de CPU para escalar.

El coste del procesador incluye lo siguiente:

- Coste de los servicios de las operaciones crear, recuperar, actualizar y eliminar en los clientes
- Coste de la réplica de otras Mäquinas virtuales Java
- Coste de la invalidación
- Coste de la política de desalojo
- Coste de la recogida de basura
- Coste de la lógica de la aplicación
- Coste de la serialización

Máquinas virtuales Java por servidor

Utilice dos servidores e inicie el número máximo de JVM por servidor. Utilice el número de particiones calculadas en el apartado anterior. A continuación, precargue las Mäquinas virtuales Java con un volumen de datos que quepa en estos dos sistemas. Utilice un servidor independiente como cliente. Ejecute una simulación de transacciones realista en esta cuadrícula de datos de dos servidores.

Para calcular la línea base, intente saturar el uso del procesador. Si no puede, es probable que la red esté saturada. Si la red está saturada, añada más tarjetas de red y disponga las Mäquinas virtuales Java por turno circular en las diversas tarjetas de red.

Ejecute los sistemas con un uso del procesador del 60%, y mida la velocidad de las transacciones crear, recuperar, actualizar y eliminar. El valor que obtenga proporciona el rendimiento de los dos servidores. Este número se dobla con cuatro servidores, y se vuelve a doblar con ocho servidores, y así sucesivamente. Esta escala presupone que la capacidad de la red y la capacidad del cliente también pueden escalarse.

Como resultado, el tiempo de respuesta de eXtreme Scale debe ser estable a medida que se aumenta el número de servidores. El rendimiento de la transacción se debe ampliar de forma lineal a medida que se añadan sistemas a la cuadrícula de datos.

Dimensionamiento de las CPU para transacciones paralelas

Las transacciones de una sola partición dimensionan el rendimiento de forma lineal a medida que la cuadrícula de datos va creciendo. Las transacciones paralelas son distintas de las transacciones de una sola partición porque afectan a un conjunto de los servidores (puede tratarse de todos los servidores).

Si una transacción afecta a todos los servidores, el rendimiento se limita al rendimiento del cliente que inicia la transacción o el servidor más lento que resulta afectado. Las cuadrículas de datos más grandes esparcen los datos más y proporcionan más espacio de procesador, memoria, red, etc. No obstante, el cliente debe esperar a que el servidor más lento responda, y el cliente debe hacer uso los resultados de la transacción.

Cuando una transacción afecta a un subconjunto de servidores, M de N servidores obtienen una solicitud. A continuación el rendimiento será N dividido por M veces más rápido que el rendimiento del servidor más lento. Por ejemplo, si tiene 20 servidores y una transacción que afecta a 5 servidores, el rendimiento es 4 veces el rendimiento del servidor más lento de la cuadrícula de datos.

Cuando una transacción paralela finaliza, los resultados se envían a la hebra de cliente que ha iniciado la transacción. Este cliente deberá agregar los resultados con una sola hebra. Este tiempo de agregación aumenta a medida que aumenta el número de servidores afectados por la transacción. No obstante, esta vez depende de la aplicación porque es posible que cada servidor devuelva aun resultado más pequeño a medida que va creciendo la cuadrícula de datos.

Normalmente, las transacciones paralelas afectan a todos los servidores en la cuadrícula de datos porque las particiones se distribuyen de forma uniforme por la cuadrícula. En este caso, el rendimiento se limita al primer caso.

Resumen

Con este dimensionamiento, tiene tres medidas, del modo siguiente.

- Número de particiones.
- Número de servidores necesarios para la memoria que es necesaria.
- Número de servidores necesarios para el rendimiento necesario.

Si necesita 10 servidores para los requisitos de memoria, pero sólo obtiene el 50% del rendimiento necesario debido a la saturación en el procesador, necesitará el doble de servidores.

Para obtener la estabilidad más alta, debe ejecutar los servidores al 60% de la carga de procesador y los almacenamientos dinámicos deJVM al 60% de la carga de almacenamiento dinámico. Los picos de utilización pueden conducir al uso del procesador a un 80–90%, aunque de forma habitual no debe ejecutar los servidores a niveles más altos que éstos.

Planificación de la capacidad de memoria caché dinámica

La API de memoria caché dinámica está disponible para las aplicaciones Java EE desplegadas en WebSphere Application Server. Se puede sacar el máximo partido de la memoria caché dinámica para almacenar en la memoria caché los datos empresariales, el HTML generado o para sincronizar los datos de la memoria caché en la célula utilizando el servicio de duplicación de datos (DRS).

Visión general

De forma predeterminada, todas las instancias de memoria caché dinámica creadas con el proveedor de memoria caché dinámica WebSphere eXtreme Scale tienen una alta disponibilidad. El coste del nivel y de la memoria de la alta disponibilidad depende de la topología utilizada.

Cuando se utiliza la topología incorporada, el tamaño de memoria caché está limitado a la cantidad de memoria libre de un único proceso de servidor y cada proceso de servidor almacena una copia completa de la memoria caché. Mientras el proceso de servidor único se sigue ejecutando, la memoria caché sobrevive. Los datos de la memoria caché sólo se perderán si todos los servidores que acceden a la memoria caché se concluyen.

Para el almacenamiento en la memoria caché que utiliza la topología particionada incorporada, el tamaño de memoria caché está limitado a un agregado del espacio libre disponible en todos los procesos del servidor. De forma predeterminada, el proveedor de memoria caché dinámica eXtreme Scale utiliza 1 réplica para cada fragmento primario, de forma que cada conjunto de datos de la memoria caché se almacena dos veces.

Utilice la siguiente fórmula A para determinar la capacidad de una memoria caché incorporada con particiones:

Fórmula A

F * C / (1 + R) = M

Donde:

- F = memoria libre por proceso de contenedor
- C = número de contenedores
- R = número de réplicas
- M = tamaño total de la memoria caché

Para una cuadrícula de datos de WebSphere Application Server Network Deployment que tenga 256 MB de espacio disponible en cada proceso, con 4 procesos de servidor en total, una instancia de memoria caché en todos estos servidores podría almacenar hasta 512 megabytes de datos. En esta modalidad, la memoria caché puede sobrevivir a que se cuelgue un servidor sin perder datos. Además, se podrían concluir hasta dos servidores de forma secuencial sin perder datos. Así puede, para el ejemplo anterior, la fórmula es la siguiente:

256mb * 4 contenedores/ (1 primario + 1 réplica) = 512mb.

Las memorias caché que utilizan la topología remota tienen unas características de tamaño similares a las de las memorias caché que utilizan las particiones

incorporadas, pero están limitadas por la cantidad de espacio disponible en todos los procesos de contenedor de eXtreme Scale.

En las topologías remotas, es posible aumentar el número de réplicas para proporcionar un nivel superior de disponibilidad con el coste de la sobrecarga de memoria adicional. En la mayoría de aplicaciones de memoria caché dinámica, esto debería ser innecesario, pero puede editar el archivo `dynacache-remote-deployment.xml` para aumentar el número de réplicas.

Utilice las fórmulas siguientes, B y C, para determinar el efecto de la adición de réplicas adicionales en la alta disponibilidad de la memoria caché.

Fórmula B

N = Minimum(T -1, R)

Donde:
- N = el número de procesos que se pueden colgar simultáneamente
- T = el número total de contenedores
- R = el número total de réplicas

Fórmula C

Ceiling(T/ (1+N)) = m

Donde:
- T = el número total de contenedores
- N = el número total de réplicas
- m = el número mínimo de contenedores necesarios para soportar los datos de la memoria caché.

Para el ajuste del rendimiento con el proveedor de memoria caché dinámica, consulte "Ajuste del proveedor de la memoria caché dinámica" en la página 505.

Tamaño de la memoria caché

Antes de que se pueda desplegar una aplicación utilizando el proveedor de memoria caché dinámica de WebSphere eXtreme Scale, los principios generales descritos en la sección anterior se deben combinar con los datos de entorno para los sistemas de producción. La primera figura para establecer es el número total de procesos de contenedor y la cantidad de memoria disponible en cada procesa para contener datos de memoria caché. Al utilizar la topología incorporada, los contenedores de memoria caché se volverán a colocar dentro de los procesos de WebSphere Application Server, de forma que haya un contenedor para cada servidor que comparte la memoria caché. Determinar la sobrecarga de memoria de la aplicación sin la memoria caché habilitada y WebSphere Application Server es el mejor método para descubrir la cantidad de espacio disponible en el proceso. Esto se puede realizar analizando los datos de la recogida de basura verbosa. Al utilizar la topología remota, esta información se puede encontrar consultando la salida de la recogida de basura verbosa de un contenedor autónomo iniciado recientemente, que todavía no se haya rellenado con datos de la memoria caché. El último concepto que se debe tener en cuenta al descubrir la cantidad de espacio disponible para el proceso para los datos de memoria caché es reservar algo de espacio de almacenamiento dinámico para la recogida de basura. La sobrecarga del

contenedor, WebSphere Application Server o servidor autónomo, además del tamaño reservado para la memoria caché no debe representar más del 70% del almacenamiento dinámico total.

Una vez que se ha recopilado esta información, los valores se puede conectar a la fórmula A, descrita anteriormente, para determinar el tamaño máximo de la memoria caché particionada. Una vez que se conoce el tamaño máximo, el siguiente paso es determinar el número total de entradas de la memoria caché que se pueden soportar, que requiere que se determine el tamaño medio por entrada de memoria caché. El método sencillo par hacer esto es añadir un 10% al tamaño del objeto del cliente. Consulte la Guía de ajustes para memoria caché dinámica y réplica de datos para obtener información más deetallada sobre el dimensionamiento de las entradas de memoria caché al utilizar memoria caché dinámica.

Cuando está habilitada la compresión, afecta al tamaño del objeto del cliente, no a la sobrecarga del sistemas de colocación en la memoria caché. Utilice la siguiente fórmula para determinar el tamaño de un objeto guardado en la memoria caché cuando utilice la compresión:

S = O * C + O * 0.10

Donde:

- S = tamaño medio del objeto almacenado en memoria caché
- O = tamaño medio de un objeto de cliente no comprimido
- C = proporción de compresión expresada como una fracción.

Así, una proporción de comprensión de 2 a 1 es 1/2 = 0,50. Para este valor es mejor valores pequeños. Si el objeto que se almacena es un POJO normal, básicamente lleno de tipos primitivos, presuponga una proporción de compresión de 0,60 a 0,70. Si el objeto almacenado en memoria caché es un Servlet, JSP, o un objeto WebServices, el método óptimo para determinar la proporción de comprensión es comprimir un ejemplo representativo con un programa de utilidad de comprensión ZIP. Si esto no es posible, una proporción de compresión de 0,2 a 0,35 es común para este tipo de datos.

A continuación, utilice esta información para determinar el número total de entradas de memoria caché que se pueden soportar. Utilice la siguiente fórmula D:

Fórmula D

T = S / A

Donde:

- T= número total de entradas de la memoria caché
- S = tamaño total disponible para los datos de la memoria caché calculados utilizando la fórmula A
- A = tamaño medio de cada entrada de la memoria caché

Finalmente, debe establecer el tamaño de memoria caché en la instancia de la memoria caché dinámica para aplicar este límite. El proveedor de la memoria caché dinámica WebSphere eXtreme Scale difiere del proveedor de la memoria caché dinámica en este aspecto. Utilice la siguiente fórmula para determinar el valor que se debe establecer para el tamaño de memoria caché en la instancia de la memoria caché dinámica. Utilice la siguiente fórmula E:

Fórmula E

Cs = Ts / Np

Donde:

- Ts = tamaño total de la memoria caché
- Cs = valor del tamaño de memoria caché para establecer en la instancia de la memoria caché dinámica
- Np = número de particiones. El valor predeterminado es 47.

Establezca el tamaño de la instancia de memoria caché dinámica en un valor calculado por la fórmula E en cada servidor que comparta la instancia de memoria caché.

Planificación de la configuración

Antes de configurar el hardware o software, comprenda las siguientes consideraciones.

Lista de comprobación operacional

Utilice la lista de comprobación operacional para preparar el entorno para desplegar WebSphere eXtreme Scale.

Tabla 3. Lista de comprobación operacional

Elemento de lista de comprobación	Para más información

Si utiliza AIX, ajuste los siguientes valores del sistema operativo:
TCP_KEEPINTVL
El valor TCP_KEEPINTVL forma parte de un protocolo de actividad de socket que permite la detección de una caída de red. La propiedad especifica el intervalo entre paquetes que se envían para validar la conexión. Si se utiliza WebSphere eXtreme Scale, establezca el valor en `10`. Para comprobar el valor actual, ejecute el mandato siguiente:
`# no –o tcp_keepintvl`
Para cambiar el valor actual, ejecute el siguiente mandato:
`# no –o tcp_keepintvl=10`
El valor TCP_KEEPINTVL está en medios segundos.
TCP_KEEPINIT
El valor TCP_KEEPINIT forma parte de un protocolo de actividad de socket que permite la detección de una caída de red. La propiedad especifica el valor de tiempo de espera inicial para la conexión TCP. Si se utiliza WebSphere eXtreme Scale, establezca el valor en `40`. Para comprobar el valor actual, ejecute los siguiente mandatos:
`# no –o tcp_keepinit`
Para cambiar el valor actual, ejecute el siguiente mandato:
`# no –o tcp_keepinit=40`
El valor TCP_KEEPINIT está en medios segundos.
	• Si desea la información de ajuste de AIX, consulte Ajuste de los sistemas AIX.

Actualice el archivo `orb.properties` para modificar el comportamiento de transporte de la cuadrícula. El archivo `orb.properties` se encuentra en el directorio `java/jre/lib`.	"Propiedades ORB" en la página 492

Tabla 3. Lista de comprobación operacional (continuación)

Elemento de lista de comprobación	Para más información
Utilice los parámetros del script **startOgServer**. En particular, utilice los siguientes parámetros: • Establezca los valores de almacenamiento dinámico con el parámetro **-jvmArgs**. • Establezca las classpath y las propiedades de la aplicación con el parámetro **-jvmArgs**. • Establezca los parámetros **-jvmArgs** para configurar la supervisión del agente. **Valores de puerto** WebSphere eXtreme Scale debe abrir los puertos para las comunicaciones para algunos transportes. Estos puertos se han definido todos dinámicamente. Sin embargo, si se utiliza un cortafuegos entre los contenedores, debe especificar los puertos. Utilice la siguiente información sobre los puertos: **Puerto de escucha** Puede utilizar el argumento **-listenerPort** para especificar el puerto que se utiliza para la comunicación entre procesos. **Puerto de grupo principal** Puede utilizar el argumento **-haManagerPort** para especificar el puerto que se utiliza para la detección de anomalías. Este argumento es igual que peerPort. Tenga en cuenta que los grupos principales no necesitan comunicarse entre zonas, de forma que es posible que no tenga que establecer este puerto, si el cortafuegos está abierto para todos los miembros de una única zona. **Puerto de servicio JMX** Puede utilizar el argumento **-JMXServicePort** para especificar el puerto que debe utilizar el servicio JMX. **Puerto SSL** Pasar `-Dcom.ibm.CSI.SSLPort=1234` como un argumento **-jvmArgs** establece el puerto SSL en `1234`. El puerto SSL es el puerto seguro igual al puerto de escucha. **Puerto de cliente** Sólo se utiliza en el servicio de catálogo. Puede especificar este valor con el argumento **-catalogServiceEndPoints**. El formato del valor de este parámetro está en el formato: `nombreservidor:nombrehost:puertocliente:puertoigual`	"Script **startOgServer**" en la página 401
Verifique que los valores de seguridad se han configurado correctamente: • Transporte (SSL) • Aplicación (Autenticación y Autorización) Para verificar los valores de seguridad, puede intentar utilizar un cliente dañino para conectarse a la configuración. Por ejemplo, cuando está configurado el valor SSL-Required, un cliente que tiene un valor TCP_IP con o un cliente con el almacén de confianza erróneo no debe poder conectarse al servidor. Si la autenticación es necesaria, un cliente sin credenciales como, por ejemplo, un ID de usuario y una contraseña, no debe poder conectarse al servidor. Si la autorización se aplica, a un cliente sin autorización de acceso no se le debe otorgar el acceso a los recursos del servidor.	"Integración de la seguridad con proveedores externos" en la página 520

Tabla 3. Lista de comprobación operacional (continuación)

Elemento de lista de comprobación	Para más información
Elija cómo va a supervisar el entorno. • Herramienta **xscmd**: – Los puertos JMX de los servidores de catálogo necesitan estar visibles en la herramienta **xscmd**. También es necesario que los puertos del servidor de contenedor sean accesibles para algunos mandatos que recopilan información de los contenedores. • Consola de supervisión: Con la consola de supervisión, puede representar gráficos de estadísticas actuales e históricas. • Herramientas de supervisión de proveedor: – Tivoli Enterprise Monitoring Agent – CA Wily Introscope – Hyperic HQ	• "Supervisión con el programa de utilidad **xscmd**" en la página 462 • "Seguridad JMX (Java Management Extensions)" en la página 517 • "Supervisión con la consola web" en la página 443 • "Supervisión con IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale" en la página 477 • "Supervisión de eXtreme Scale con Hyperic HQ" en la página 486 • "Supervisión de aplicaciones de eXtreme Scale con CA Wily Introscope" en la página 483

Planificación de puertos de red

WebSphere eXtreme Scale es una memoria caché distribuida que requiere puertos de apertura para la comunicación con la pila del intermedio de solicitud de objetos (ORB) y Transmission Control Protocol (TCP) entre máquinas virtuales Java. Planee y controle los puertos, especialmente en un entorno que tiene un cortafuegos, y cuando utiliza un servicio de catálogo y contenedores en varios puertos.

Importante: Al especificar números de puerto, evite establecer puertos que estén en el rango efímero para el sistema operativo. Si utiliza un puerto que está en el rango efímero, se podrían producir conflictos de puertos.

Dominio de servicio de catálogo

Un dominio de servicio de catálogo requiere que se definan los puertos siguientes:

peerPort
Especifica el puerto para que el gestor de alta disponibilidad (HA) se comunique entre servidores de catálogo iguales sobre una pila TCP. En WebSphere Application Server, este valor se hereda mediante la configuración del puerto del gestor de alta disponibilidad.

clientPort
Especifica el puerto para que los servidores de catálogo accedan a los datos de servicio de catálogo. En WebSphere Application Server, este puerto se establece mediante la configuración de dominio de servicio de catálogo.

listenerPort
Especifica el número de puerto al que se enlaza el intermediario para solicitudes de objetos (ORB). Este valor configura los contenedores y clientes para que se comuniquen con el servicio de catálogo mediante el ORB. En WebSphere Application Server, el listenerPort se hereda de la configuración de puerto BOOTSTRAP_ADDRESS.Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.

Valor predeterminado:2809

JMXConnectorPort
Define el puerto SSL (Capa de sockets seguros) al que se enlaza el servicio JMX (Java Management Extensions).

Servidores de contenedor

Los servidores de contenedor WebSphere eXtreme Scale también requieren varios puertos para funcionar. De forma predeterminada, el servidor de contenedor eXtreme Scale genera su puerto de gestor HA y puerto de escucha ORB automáticamente con los puertos dinámicos. Para un entorno que tiene un cortafuegos, presenta ventajas para planificar y controlar los puertos. Para que los servidores de contenedor se inicien con puertos específicos, puede utilizar las opciones siguientes en el mandato **startOgServer**.

haManagerPort
Sinónimo con puerto de igual. Especifica el número de puerto que utiliza el High Availability Manager. Si esta propiedad no está establecida, el servicio de catálogos genera un puerto disponible de forma automática. Esta propiedad se aplica tanto al servidor de contenedor, como al servicio catálogos. (Sólo necesario para entornos de WebSphere Application Server.)

listenerPort
Especifica el número de puerto al que se enlaza el intermediario para solicitudes de objetos (ORB). Este valor configura los contenedores y clientes para que se comuniquen con el servicio de catálogo mediante el ORB. En WebSphere Application Server, el listenerPort se hereda de la configuración de puerto BOOTSTRAP_ADDRESS.Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.

Valor predeterminado:2809

JMXConnectorPort
Define el puerto SSL (Capa de sockets seguros) al que se enlaza el servicio JMX (Java Management Extensions).

7.1.1+ **xioChannel.xioContainerTCPSecure.Port**
Especifica el número de puerto SSL de eXtremeIO en el servidor. Esta propiedad sólo se utiliza cuando la propiedad **transportType** se establece en `SSL-Supported` o `SSL-Required`.

7.1.1+ **xioChannel.xioContainerTCPNonSecure.Port**
Especifica el número de puerto de escucha no seguro de eXtremeIO en el servidor. Si no establece el valor, se utiliza un puerto efímero. Esta propiedad sólo se utiliza cuando la propiedad **transportType** se establece en `TCP/IP`.

La planificación adecuada del control de puertos es esencial cuando se inician cientos de máquinas virtuales Java en un servidor. Si existe un conflicto de puertos, los servidores de contenedor no se iniciarán.

Clientes

Los clientes de WebSphere eXtreme Scale pueden recibir devoluciones de llamada de servidores al utilizar la API DataGrid o diversos otros mandatos. Utilice la propiedad **listenerPort** en el archivo de propiedades de cliente para especificar el puerto en el que el cliente está a la escucha de devoluciones de llamada del servidor.

haManagerPort
Sinónimo con puerto de igual. Especifica el número de puerto que utiliza el High Availability Manager. Si esta propiedad no está establecida, el servicio de catálogos genera un puerto disponible de forma automática.

Esta propiedad se aplica tanto al servidor de contenedor, como al servicio catálogos. (Sólo necesario para entornos de WebSphere Application Server.)

jvmArgs (opcional)
Especifica una lista de argumentos de máquina virtual Java (JVM). Cuando la seguridad está habilitada, debe utilizar el siguiente argumento para configurar el puerto SSL (Capa de sockets seguros): `-jvmArgs -Dcom.ibm.CSI.SSLPort=<puerto_SSL>`.

listenerPort
Especifica el número de puerto al que se enlaza el intermediario para solicitudes de objetos (ORB). Este valor configura los contenedores y clientes para que se comuniquen con el servicio de catálogo mediante el ORB. En WebSphere Application Server, el listenerPort se hereda de la configuración de puerto BOOTSTRAP_ADDRESS.Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.

Valor predeterminado:2809

Puertos en WebSphere Application Server

- El valor de **listenerPort** se hereda del valor de **BOOTSTRAP_ADDRESS** para cada servidor de aplicaciones WebSphere Application Server.
- Los valores **haManagerPort** y **peerPort** heredados del valor **DCS_UNICAST_ADDRESS** para cada servidor de aplicaciones WebSphere Application Server.

Puede definir un dominio de servicio de catálogo en la consola administrativa, tal como se describe en "Creación de dominios de servicio de catálogo en WebSphere Application Server" en la página 257.

PUede visualizar los puertos para un servidor determinado pulsando una de las vías de acceso siguientes en la consola administrativa:

- WebSphere Application Server Network Deployment Versión 6.1: **Servers** > **Servidores de aplicaciones** > ***nombre_servidor*** > **Puertos** > ***nombre_puento_final***.
- WebSphere Application Server Network Deployment Versión 7.0: **Servers** > **Tipos de servidor** > **WebSphere Application Servers** > *nombre_servidor* > **Puertos** > ***nombre_puerto***

Visión general de seguridad

WebSphere eXtreme Scale puede proteger el acceso a los datos, incluida la posibilidad de integración con proveedores de datos externos.

Nota: En un almacén de datos no almacenado en memoria caché existente, como una base de datos, probablemente, tendrá características de seguridad incorporadas que podría necesitar para configurar o habilitar de forma activa. No obstante, después de haber almacenado en memoria caché los datos con eXtreme Scale, debe considerar la situación resultante importante de que las características de seguridad del programa de fondo ya no están en vigor. Puede configurar la seguridad de eXtreme Scale en los niveles necesarios de modo que la nueva arquitectura almacenada en memoria caché para los datos también esté protegida.
A continuación, aparece un breve resumen de las características de seguridad de eXtreme Scale. Si desea más información detallada sobre cómo configurar la seguridad, consulte *Guía de administración* y *Guía de programación*.

Conceptos básicos de la seguridad distribuida

La seguridad distribuida de eXtreme Scale se basa en tres conceptos clave:

Autenticación de confianza
La capacidad de determinar la identidad del solicitante. WebSphere eXtreme Scale da soporte a la autenticación de cliente a servidor y servidor a servidor.

Autorización
La capacidad de dar permisos para otorgar derechos de acceso al solicitante. WebSphere eXtreme Scale da soporte a distintas autorizaciones para diversas operaciones.

Transporte seguro
La transmisión segura de datos a través de una red. WebSphere eXtreme Scale soporta los protocolos TLS/SSL (Transport Layer Security/Secure Sockets Layer).

Autenticación

WebSphere eXtreme Scale da soporte a la infraestructura distribuida de cliente-servidor. La infraestructura de seguridad de cliente-servidor existe para proteger el acceso a los servidores de eXtreme Scale. Por ejemplo, cuando el servidor eXtreme Scale requiere una autenticación, el cliente de eXtreme Scale debe proporcionar las credenciales para autenticar el servidor. Estas credenciales pueden ser un par de nombre de usuario y contraseña, un certificado de cliente, un ticket de Kerberos o datos que se presentan en un formato acordado por el cliente y el servidor.

Autorización

Las autorizaciones de WebSphere eXtreme Scale se basan en sujetos y permisos. Puede utilizar JAAS (Java Authentication and Authorization Services) para autoriza el acceso, o puede conectar un método personalizado, como Tivoli Access Manager (TAM), para manejar las autorizaciones. Pueden otorgarse las siguientes autorizaciones a un cliente o grupo:

Autorización de correlaciones
Realizar operaciones de inserción, lectura, actualización o supresión en correlaciones.

Autorización de ObjectGrid
Realizar consultas de objetos o entidades y consultas de secuencias en objetos ObjectGrid.

Autorización de agentes de DataGrid
Permitir que los agentes de DataGrid se desplieguen en un ObjectGrid.

Autorización de correlaciones del lado del servidor
Duplicar una correlación de servidor con el lado del cliente o crear un índice dinámico con la correlación de servidor.

Autorización de administración
Realizar tareas de administración.

Seguridad de transporte

Para proteger la comunicación cliente-servidor, WebSphere eXtreme Scale soporta TLS/SSL. Estos protocolos proporcionan el nivel de seguridad de la capa de transporte con la autenticidad, integridad y confidencialidad para una conexión segura entre un cliente y un servidor de eXtreme Scale.

Seguridad de la cuadrícula

En un entorno seguro, un servidor debe poder comprobar la autenticidad de otro servidor. Para ello WebSphere eXtreme Scale utiliza un mecanismo de serie de clave secreta compartida. Este mecanismo de clave secreta es parecido a una contraseña secreta. Todos los servidores de eXtreme Scale acuerdan una serie secreta compartida. Cuando un servidor se une a la cuadrícula de datos, el servidor se ve obligado a presentar la serie secreta. Si la serie secreta del servidor que se une coincide con una del servidor maestro, este servidor se puede unir a la cuadrícula. De lo contrario, la solicitud se rechaza.

El envío de una serie secreta en texto normal no es seguro. La infraestructura de seguridad de eXtreme Scale proporciona un plug-in SecureTokenManager para permitir al servidor proteger este secreto antes de enviarlo. Puede elegir cómo implementar la operación segura. WebSphere eXtreme Scale proporciona una implementación, en la que se implementa la operación segura para cifrar y firmar la serie secreta.

Seguridad JMX (Java Management Extensions) en una topología de despliegue dinámico

La seguridad de JMX MBean recibe soporte en todas las versiones de eXtreme Scale. Los clientes de MBeans de servidor de catálogo y MBeans de servidor de contenedor pueden autenticarse, y se puede forzar el acceso a operaciones de MBean.

Seguridad de eXtreme Scale local

La seguridad de eXtreme Scale local es distinta del modelo de eXtreme Scale distribuido porque la aplicación crea una instancia y utiliza una instancia de ObjectGrid directamente. La aplicación y las instancias de eXtreme Scale están en la misma JVM (Java Virtual Machine). Puesto que no hay ningún concepto de cliente-servidor en este modelo, no se da soporte a la autenticación. Las aplicaciones deben gestionar su propia autenticación y, a continuación, pasar el objeto Subject autenticado aeXtreme Scale. Sin embargo, el mecanismo de autorización que se utiliza para el modelo de programación de eXtreme Scale local es el mismo que se ha utilizado para el modelo cliente-servidor.

Configuración y programación

Para obtener más información sobre cómo configurar y programar la seguridad, consulte “Integración de la seguridad con proveedores externos” en la página 520 y API de seguridad.

Capítulo 3. Guías de aprendizaje

Puede utilizar guías de aprendizaje como ayuda para comprender los escenarios de uso del producto, incluido el gestor de entidades, las consultas y la seguridad.

Guía de aprendizaje: Configuración de la seguridad de Java SE

Con la siguiente guía de aprendizaje, puede crear un entorno distribuido de eXtreme Scale en un entorno de Java Platform, Standard Edition.

Antes de empezar

Asegúrese de que está familiarizado con los conceptos básicos de una configuración de eXtreme Scale distribuido.

Acerca de esta tarea

En esta guía de aprendizaje, el servidor de catálogo, el servidor de contenedor y el cliente se ejecutan todos en un entorno Java SE. Cada paso de la guía de aprendizaje se basa en el anterior. Siga cada uno de los pasos para proteger un eXtreme Scale distribuido y desarrollar una aplicación Java SE sencilla para acceder al eXtreme Scale seguro.

Inicio de la guía de aprendizaje

Procedimiento

1. "Guía de aprendizaje de seguridad Java SE - Paso 1" en la página 70
   - Iniciar un servidor de catálogo no seguro
   - Iniciar un servidor de contenedor no seguro
   - Iniciar un cliente para acceder a los datos
   - Utilizar el programa de utilidad **xscmd** para que se muestre el tamaño de la correlación
   - Detener el servidor
2. "Guía de aprendizaje de seguridad de Java SE - Paso 2" en la página 73
   - Uso del generador de credenciales
   - Uso del autenticador
   - Iniciar un servidor de catálogo seguro
   - Iniciar un servidor de contenedor seguro
   - Iniciar el cliente para acceder a ObjectGrid seguro
   - Utilizar el programa de utilidad **xscmd** para que se muestre el tamaño de la correlación
   - Detener el servidor seguro
3. "Guía de aprendizaje de seguridad de Java SE - Paso 3" en la página 79
   - Uso de la política de autorización JAAS
4. "Guía de aprendizaje de seguridad de Java SE - Paso 4" en la página 83
   - Crear un almacén de claves y un almacén de confianza
   - Configurar propiedades SSL para el servidor

- Configurar propiedades SSL para el cliente
- Utilizar el programa de utilidad **xscmd** para que se muestre el tamaño de la correlación
- Detener el servidor seguro

Guía de aprendizaje de seguridad Java SE - Paso 1

En este tema se describe un *ejemplo no seguro simple*. En los pasos de la guía de aprendizaje se añaden características de seguridad adicionales para aumentar la cantidad de seguridad integrada que está disponible.

Antes de empezar

Nota: Todos los archivos necesarios para este paso de la guía de aprendizaje se proporcionan en la siguiente sección.

Procedimiento

Ejecución del ejemplo
Inicie el servicio de catálogo utilizando los siguientes scripts. Para obtener más información sobre cómo iniciar el servicio de catálogo, consulte "Inicio de un servicio de catálogo autónomo" en la página 395.

1. Vaya al directorio bin: `cd objectgridRoot/bin`
2. Inicie el servidor de catálogo denominado catalogServer:
   - UNIX Linux `startOgServer.sh catalogServer`
   - Windows `startOgServer.bat catalogServer`
3. Vaya hasta el directorio bin `cd objectgridRoot/bin`
4. A continuación inicie un servidor de contenedor llamado c0 con el siguiente script:
   - 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
     ```

Ejemplo

Para obtener más información sobre cómo iniciar servidores de contenedores, consulte "Inicio de servidores de contenedor" en la página 398.

Después de iniciar el servidor de catálogo y el servidor de contenedor, inicie el cliente tal como se muestra a continuación:

1. Vaya hasta el directorio bin una vez más.
2. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SimpleApp`

El archivo `secsample.jar` contiene la clase SimpleApp.

La salida de este programa es:

```
El nombre de cliente para el ID 0001 es fName lName
```

También puede utilizar el programa de utilidad **xscmd** para que se muestren los tamaños máximos de la cuadrícula "accounting".

- Vaya hasta el directorio `objectgridRoot/bin`.
- Utilice el programa de utilidad **xscmd** para que se muestren los tamaños de correlación:
  - UNIX Linux `xscmd.sh -c showMapSizes -g accounting -ms mapSet1`
  - Windows `xscmd.bat -c showMapSizes -g accounting -ms mapSet1`

Cómo detener los servidores

Servidor de contenedor

Utilice el siguiente mandato para detener el servidor de contenedor c0.

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

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

Verá el siguiente mensaje.

```
CWOBJ2512I: el servidor ObjectGrid c0 se ha detenido.
```

Servidor de catálogo

Puede detener un servidor de catálogo utilizando el siguiente mandato.

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

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

Si concluye el servidor de catálogo, verá el siguiente mensaje.

```
CWOBJ2512I: el servidor ObjectGrid catalogServer se ha detenido.
```

Archivos necesarios

El archivo siguiente es la clase Java para SimpleApp.

SimpleApp.java

```
// Este programa de ejemplo se proporciona TAL CUAL y se puede utilizar, ejecutar, copiar y modificar
// sin que el cliente tenga que pagar derechos
// (a) para su propia formación,
// (b) para desarrollar aplicaciones diseñadas para ejecutarse con un producto IBM WebSphere,
// para uso interno propio del cliente o para su redistribución por parte del cliente, como parte de una
// aplicación de ese tipo, en los productos propios del cliente.
// Material bajo licencia - Propiedad de 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);
    }

    /**
     * leer y grabar la correlación
     * @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);
    }

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

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

        return og;

    }

}
```

El método getObjectGrid de esta clase obtiene un ObjectGrid, y el método run lee un registro de la correlación del cliente y actualiza el valor.

Para ejecutar este ejemplo en un entorno distribuido, se crean un archivo XML descriptor de ObjectGrid `SimpleApp.xml` y un archivo XML de despliegue `SimpleDP.xml`. Los archivos se muestran en el siguiente ejemplo:

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>
```

El siguiente archivo XML configura el entorno de despliegue.

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>
```

Se trata de una configuración de ObjectGrid sencilla con una instancia de ObjectGrid llamada "accounting" y una correlación llamada "customer" (dentro del mapSet "mapSet1"). El archivo `SimpleDP.xml` incorpora un conjunto de correlaciones que se configura con 1 partición y 0 réplicas mínimas necesarias.

Paso siguiente de la guía de aprendizaje

Guía de aprendizaje de seguridad de Java SE - Paso 2

Basándose en el paso anterior, el siguiente tema muestra cómo implementar la autenticación de cliente en un entorno distribuido de eXtreme Scale.

Antes de empezar

Asegúrese de que ha completado “Guía de aprendizaje de seguridad Java SE - Paso 1” en la página 70.

Acerca de esta tarea

Con la autenticación de cliente habilitada, un cliente se autentica antes de conectarse al servidor eXtreme Scale. Esta sección muestra cómo puede realizarse la autenticación de cliente en un entorno de servidor de eXtreme Scale, e incluye código de ejemplo y scripts para demostrarlo.

Al igual que cualquier otro mecanismos de autenticación, la autenticación mínima consta de los siguientes pasos:

1. El administrador efectúa cambios en las configuraciones de modo que la autenticación sea un requisito.
2. El cliente proporciona una credencial al servidor.
3. El servidor autentica la credencial en el registro.

Procedimiento

1. **Credencial del cliente**

   Una credencial de cliente se representa mediante una interfaz com.ibm.websphere.objectgrid.security.plugins.Credential. Una credencial de cliente puede ser un par de nombre de usuario y contraseña, un ticket Kerberos, un certificado de cliente o datos en cualquier formato que hayan acordado el cliente y el servidor. Consulte la la documentación de la API Credential para ver más detalles.

   Esta interfaz define de forma explícita los métodos equals(Object) y hashCode(). Estos dos métodos son importantes porque los objetos Subject autenticados se almacenan en memoria caché utilizando el objeto Credential como la clave en el lado del servidor.

   eXtreme Scale también proporciona un plug-in para generar una credencial. Este plug-in se representa mediante la interfaz com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator, y se utiliza para generar una credencial de cliente. Esto resulta útil cuando la credencial

puede caducar. En este caso, se llama al método getCredential() para renovar una credencial. Consulte la documentación de la API CredentialGenerator para obtener más detalles.

Puede implementar estas dos interfaces para que el tiempo de ejecución del cliente de eXtreme Scale obtenga credenciales de cliente.

Este ejemplo utiliza las dos siguientes implementaciones de plug-in de ejemplo proporcionadas por eXtreme Scale.

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

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

Para obtener más información sobre estos plug-ins, consulte Programación de la autenticación de cliente .

2. **Autenticación de servidor** Después de que el cliente de eXtreme Scale recupere el objeto Credential mediante el objeto CredentialGenerator, el objeto Credential de este cliente se envía junto con la solicitud del cliente al servidor de eXtreme Scale. El servidor de eXtreme Scale autentica el objeto Credential antes de procesar la solicitud. Si el objeto Credential se autentica correctamente, se devuelve un objeto Subject para representar este cliente.

   A continuación, el objeto Subject se almacena en memoria caché y caduca después de que su vida útil alcance el valor de tiempo de espera de la sesión. El valor de tiempo de espera del inicio de sesión puede establecerse mediante la propiedad loginSessionExpirationTime del archivo XML del clúster. Por ejemplo, establecer `loginSessionExpirationTime="300"` hace que el objeto Subject caduque en 300 segundos.Este objeto Subject se utilizará para autorizar la solicitud que se muestra más adelante.

   Un servidor de eXtreme Scale utiliza el plug-in Authenticator para autenticar el objeto Credential. Consulte la documentación de la API de Authenticator para obtener más detalles.

   Este ejemplo utiliza una implementación incorporada de eXtreme Scale: KeyStoreLoginAuthenticator, que es para fines de prueba y ejemplo (un almacén de claves es un registro de usuarios simple y no debe utilizarse en un entorno de producción). Para obtener más información, consulte el tema sobre el plug-in de autenticador en Programación de la autenticación de cliente .

   Este KeyStoreLoginAuthenticator utiliza un KeyStoreLoginModule para autenticar el usuario con el almacén de claves utilizando el módulo de inicio de sesión JAAS "KeyStoreLogin". El almacén de claves se puede configurar como una opción para la clase KeyStoreLoginModule. En el siguiente ejemplo se muestra el alias keyStoreLogin configurado en el archivo de configuración de JAAS og_jaas.config:

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

   Los siguientes mandatos crean un almacén de claves sampleKS.jks en el directorio %OBJECTGRID_HOME%/security con la contraseña sampleKS1. Además, se crean tres certificados de usuario que representan el usuario administrator, el usuario manager y el usuario cashier con sus propias contraseñas.

   a. Vaya hasta el directorio raíz de eXtreme Scale.

      ```
      cd objectgridRoot
      ```

   b. Cree un directorio llamado "security".

      ```
      mkdir security
      ```

   c. Vaya hasta el directorio de seguridad acabado de crear.

      ```
      cd security
      ```

d. Utilice keytool (en el directorio javaHOME/bin) para crear un usuario "administator" con la contraseña "administrator1" en el almacén de claves sampleKS.jks.

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

e. Utilice keytool (en el directorio javaHOME/bin) para crear un usuario "manager" con la contraseña "manager1" en el almacén de claves sampleKS.jks.

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

f. Utilice keytool (en el directorio javaHOME/bin) para crear un usuario "cashier" con la contraseña "cashier1" en el almacén de claves sampleKS.jks.

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

La configuración de seguridad de cliente se configura en el archivo de propiedades del cliente. Utilice el siguiente mandato para crear una copia en el directorio %OBJECTGRID_HOME%/security:

a. Vaya al directorio de seguridad.

```
cd objectgridRoot/security
```

b. Copie el archivo sampleClient.properties en el archivo client.properties.

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

Las siguientes propiedades aparecen resaltadas en el archivo client.properties en el directorio de seguridad.

a. **securityEnabled:** establecer securityEnabled en true (valor predeterminado) habilita la seguridad de cliente, que incluye la autenticación.

b. **credentialAuthentication:** establezca credentialAuthentication en Supported (valor predeterminado), que significa que el cliente da soporte a la autenticación de credenciales.

c. **transportType:** establezca transportType en TCP/IP, que significa que no se utilizará SSL.

d. **singleSignOnEnabled:** establézcalo en false (valor predeterminado). El inicio de sesión único no está disponible.

3. **Configuración de seguridad de servidor**

La configuración de seguridad de servidor se especifica en el archivo XML de descriptor de seguridad y en el archivo de propiedades de seguridad del servidor.El archivo XML de descriptor de seguridad describe las propiedades de seguridad comunes a todos los servidores (incluidos los servidores de catálogo y los servidores de contenedor). Un ejemplo de propiedad es la configuración de autenticador que representa el mecanismo de autenticación y el registro de usuarios.

A continuación se muestra el archivo security.xml que se va a utilizar en este ejemplo:

```
<?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:** establézcalo en true, que habilita la seguridad de servidor que incluye la autenticación.

b. **loginSessionExpirationTime:** establezca el valor en 300 (valor predeterminado).

c. **authenticator:** añada la clase de autenticador KeyStoreLoginAuthenticator al archivo XML del clúster tal como se muestra a continuación:

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

d. **credentialAuthentication:** establezca el atributo credentialAuthentication en Required de forma que el servidor requiera la autenticación

Para obtener una explicación más detallada en el archivo `security.xml`, consulte Archivo XML de descriptor de seguridad .

Copie el archivo de propiedad de servidor en el directorio de seguridad. En este momento no es necesario modificar nada en este archivo.

a. Vaya hasta el directorio de seguridad.

```
cd objectgridRoot/security
```

b. Copie el archivo de ejemplo de objectGrid `sampleServer.properties` del directorio de propiedades en el nuevo archivo `server.properties`.

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

Realice los cambios siguiente en el archivo `server.properties`:

a. **securityEnabled:** establezca el atributo **`securityEnabled`** en true.

b. **transportType:** establezca el atributo **`transportType`** en TCP/IP, que significa que no se utiliza SSL.

c. **secureTokenManagerType:** establezca el atributo **`secureTokenManagerType`** en none para no configurar el gestor de señales seguro.

4. **Cliente seguro** Conecte la aplicación cliente al servidor de forma segura tal como se muestra en el siguiente ejemplo:

```
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);
    }

    /**
     * Obtener ObjectGrid
     * @return an ObjectGrid instance
     * @throws Exception
     */
    protected ObjectGrid getObjectGrid(String[] args) throws Exception {
        ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();
        ogManager.setTraceFileName("logs/client.log");
        ogManager.setTraceSpecification("ObjectGrid*=all=enabled:ORBRas=all=enabled");

        // Crear un objeto ClientSecurityConfiguration utilizando el archivo especificado
        ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
                .getClientSecurityConfiguration(args[0]);

        // Crear un CredentialGenerator utilizando el usuario y la contraseña pasados.
```

```
        CredentialGenerator credGen = new UserPasswordCredentialGenerator(args[1], args[2]);
        clientSC.setCredentialGenerator(credGen);

        // Crear un ObjectGrid conectándose al servidor de catálogo.
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", clientSC, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;

    }

}
```

Hay tres cosas distintas de la aplicación no segura:

a. Se ha creado un objeto ClientSecurityConfiguration pasando el archivo `client.properties` configurado.

b. Se ha creado un UserPasswordCredentialGenerator utilizando el ID de usuario y la contraseña pasados.

c. Se ha conectado al servidor de catálogo para obtener un ObjectGrid del ClientClusterContext pasado un objeto ClientSecurityConfiguration.

5. **Emita la aplicación**

   Para ejecutar la aplicación, inicie el servidor de catálogo. Emita las opciones de la línea de mandatos -clusterFile y -serverProps para pasar las propiedades de seguridad:

   a. Vaya al directorio bin:

      ```
      cd objectgridRoot/bin
      ```

   b. Inicie el servidor de catálogo:

      - 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"
        ```

   A continuación, inicie un servidor de contenedor seguro utilizando el siguiente script:

   a. Vuelva a ir hasta el directorio bin:

      ```
      cd objectgridRoot/bin
      ```

   b. Inicie un servidor de contenedor seguro:

      - 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"
        ```

   El archivo de propiedades de servidor se pasa emitiendo -serverProps.

   Una vez que se ha iniciado el servidor, inicie el cliente utilizando el mandato siguiente:

   a. `cd objectgridRoot/bin`

b.

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

Linux Utilice dos puntos (:) para el separador de classpath en lugar de punto y coma (;) como en el ejemplo anterior.

El archivo `secsample.jar` contiene la clase SimpleApp.

SecureSimpleApp utiliza tres parámetros que se proporcionan en la siguiente lista:

a. El archivo `../security/client.properties` es el archivo de propiedades de seguridad del cliente.

b. `manager` es el ID de usuario.

c. `manager1` es la contraseña.

Después de emitir la clase, se obtiene la siguiente salida:

El nombre de cliente para `ID 0001 es fName lName`.

También puede utilizar el programa de utilidad **xscmd** para que se muestren los tamaños máximos de la cuadrícula "accounting".

- Vaya hasta el directorio `objectgridRoot/bin`.
- Utilice el mandato **xscmd** con el mandato -c showMapSizes de la forma siguiente.
  - 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`

Ahora puede utilizar el mandato **stopOgServer** para detener el proceso de servicio de catálogo o servidor de contenedor. Sin embargo tendrá que proporcionar un archivo de configuración de seguridad. El archivo de propiedades de cliente de ejemplo define las siguientes dos propiedades para generar una credencial ID usuario/contraseña (manager/manager1).

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

Detenga el contenedor c0 con el siguiente mandato.

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

Si no proporciona la opción **-clientSecurityFile**, verá una excepción con el mensaje siguiente.

```
>> SERVER (id=39132c79, host=9.10.86.47) TRACE START:
>> org.omg.CORBA.NO_PERMISSION: el servidor requiere la autenticación de
credenciales, pero no hay contexto de seguridad del cliente.
Normalmente, esto sucede cuando el cliente no pasar ninguna credencial
al servidor.
vmcid: 0x0
código menor: 0
completado: No
```

También puede concluir el servidor de catálogo utilizando el mandato siguiente. Sin embargo, si desea continuar intentando el siguiente paso de la guía de aprendizaje, podrá dejar el servidor de catálogo ejecutándose.

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

Si concluye el servidor de catálogo, verá la siguiente salida.

```
CWOBJ2512I: el servidor ObjectGrid catalogServer se ha detenido
```

Ahora el sistema ya es parcialmente seguro y se ha llevado a cabo habilitando la autenticación. Ha configurado el servidor para conectarse en el registro de usuarios, ha configurado el cliente para proporcionar credenciales de cliente y ha cambiado el archivo de propiedades de cliente y el archivo XML del clúster para habilitar la autenticación.

Si proporciona una contraseña no válida, verá una excepción indicando que el nombre de usuario o la contraseña no son correctos.

Para obtener más información sobre la autenticación de cliente, consulte "Autenticación de cliente de aplicaciones" en la página 507.

Paso siguiente de la guía de aprendizaje

Guía de aprendizaje de seguridad de Java SE - Paso 3

Tras autenticar un cliente, como en el paso anterior, puede proporcionar privilegios de seguridad a través de mecanismos de autorización de eXtreme Scale.

Antes de empezar

Asegúrese de haber completado el apartado "Guía de aprendizaje de seguridad de Java SE - Paso 2" en la página 73 antes de llevar a cabo esta tarea.

Acerca de esta tarea

El paso anterior de esta guía de aprendizaje ha demostrado cómo habilitar la autenticación en una cuadrícula de eXtreme Scale. Como resultado, un cliente no autenticado se puede conectar al servidor y enviar solicitudes al sistema. No obstante, cada cliente autenticado tiene el mismo permiso o privilegios que el servidor, como por ejemplo, la lectura, la grabación o la supresión de datos que se almacenan en las correlaciones de ObjectGrid. Los clientes también pueden emitir cualquier tipo de consulta. Esta sección demuestra cómo utilizar la autorización de eXtreme Scale para otorgar distintos privilegios variables de usuarios autenticados.

De forma parecida a muchos otros sistemas, eXtreme Scale adopta un mecanismo de autorización basado en permisos. WebSphere eXtreme Scale tiene distintas categorías de permisos representadas por diferentes clases de permisos. Este tema muestra MapPermission. Para ver la categoría completa de permisos, consulte Programación de autorización de cliente .

En WebSphere eXtreme Scale, la clase com.ibm.websphere.objectgrid.security.MapPermission representa permisos para los recursos de eXtreme Scale, en particular los métodos de las interfaces ObjectMap o JavaMap. WebSphere eXtreme Scale define las siguientes series de permiso para acceder a los métodos de ObjectMap y JavaMap:

- leer: otorga permiso para leer los datos de la correlación.
- grabar: otorga permiso para actualizar los datos de la correlación.
- insertar: otorga permiso para insertar los datos en la correlación.

- eliminar: otorga permiso para eliminar los datos de la correlación.
- invalidar: otorga permiso para invalidar los datos de la correlación.
- todos: otorga todos los permisos anteriores: leer, grabar, insertar, eliminar e invalidar.

La autorización tiene lugar cuando un cliente llama a un método de ObjectMap o JavaMap. El tiempo de ejecución de eXtreme Scale comprueba los distintos permisos de correlación para los métodos diferentes. Si los permisos requeridos no se conceden al cliente, se produce una excepción AccessControlException.

Esta guía de aprendizaje muestra cómo utilizar la autorización Java Authentication and Authorization Service (JAAS) para otorgar accesos a correlaciones de autorizaciones para usuarios distintos.

Procedimiento

1. **Habilitación de la autorización de eXtreme Scale** Para habilitar la autorización en ObjectGrid, debe establecer `true` como valor del atributo securityEnabled para ese ObjectGrid determinado en el archivo XML. La habilitación de la seguridad en el ObjectGrid significa que se habilita la autorización. Utilice los siguientes mandatos para crear un nuevo archivo XML de ObjectGrid con la seguridad habilitada.
   a. Vaya al directorio `xml`.
      ```
      cd objectgridRoot/xml
      ```
   b. Copie el archivo `SimpleApp.xml` en el archivo `SecureSimpleApp.xml`.
      ```
      cp SimpleApp.xml SecureSimpleApp.xml
      ```
   c. Abra el archivo `SecureSimpleApp.xml` y añada `securityEnabled="true"` en el nivel de ObjectGrid tal como se muestra en el XML siguiente:
      ```
      <?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. **Definición de la política de autorización.** En la sección de autenticación previa al cliente, ha creado tres usuarios en el almacén de claves: cashier, manager y administrator. En este ejemplo, el usuario "cashier" sólo tiene permisos de lectura para todas las correlaciones y que el usuario "manager" tiene todos los permisos. La autorización JAAS se utiliza en este ejemplo. La autorización JAAS utiliza el archivo de política de autorización para otorgar permisos a principales. El siguiente archivo se define en el directorio de seguridad:
   ```
   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";
   };
   ```

   Nota:

   - El código base `codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"` es un URL especialmente reservado para ObjectGrid. Todos los permisos de ObjectGrid otorgados a principales deben utilizar esta base de código especial.

- La primera sentencia grant otorga permiso de correlación de "lectura" al principal `"CN=cashier,O=acme,OU=OGSample"`, de modo que el usuario cashier sólo tiene permiso de lectura de correlación para todas las correlaciones en el ObjectGrid accounting.
- La segunda sentencia grant otorga "todos" los permisos de correlación al principal `"CN=manager,O=acme,OU=OGSample"`, de modo que el usuario manager tiene todos los permisos para las correlaciones en el ObjectGrid accounting.

Ahora puede iniciar un servidor con una política de autorización. El archivo de política de autorización de JAAS se puede establecer utilizando la propiedad-D estándar: -Djava.security.auth.policy=../security/ogAuth.policy

3. **Ejecute la aplicación.**

   Después de crear los archivos anteriores, puede ejecutar la aplicación.

   Utilice los siguientes mandatos para iniciar el servidor de catálogo. Para obtener más información sobre cómo iniciar el servicio de catálogo, consulte “Inicio de un servicio de catálogo autónomo” en la página 395.

   a. Vaya al directorio bin: `cd objectgridRoot/bin`

   b. Inicie el servidor de catálogo.

      - 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"
      ```

      Los archivos `security.xml` y `server.properties` se crearon en el paso anterior de esta guía de aprendizaje.

      T

   c. Entonces puede iniciar un servidor de contenedor seguro utilizando el script siguiente. Ejecute el script siguiente desde el directorio bin:

      - UNIX Linux
      ```
      # startOgServer.sh c0 -objectGridFile
      ../xml/SecureSimpleApp.xml -deploymentPolicyFile
      ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809
      -serverProps ../security/server.properties -jvmArgs
      -Djava.security.auth.login.config="../security/og_jaas.config"
      -Djava.security.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"
      ```

Tenga en cuenta las siguientes diferencias del mandato de inicio de servidor de contenedor anterior:

- Utilice el archivo `SecureSimpleApp.xml` en lugar del archivo `SimpleApp.xml`.
- Añada otro argumento `-Djava.security.auth.policy` para establecer el archivo de política de autorización de JAAS para el proceso de servidor de contenedor.

Utilice el mismo mandato que en el paso anterior de la guía de aprendizaje:

a. Desplácese al directorio bin.

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

   Como el usuario "manager" tiene todos los permisos para las correlaciones del accounting ObjectGrid, la aplicación se ejecuta correctamente.

   Ahora, en lugar de utilizar el usuario "manager", utilice el usuario "cashier" para iniciar la aplicación cliente.

c. Desplácese al directorio bin.

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

Se genera la siguiente excepción:

```
Excepción en la hebra "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 más
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 más
```

Esta excepción se produce porque el usuario "cashier" no tiene permiso de grabación, y por ello no puede actualizar el cliente de correlación.

Ahora el sistema da soporte a la autorización. Puede definir políticas de autorización para otorgar distintos permisos a usuarios diferentes. Para obtener información sobre la autorización, consulte “Autorización de cliente de aplicaciones” en la página 509.

Qué hacer a continuación

Complete el siguiente paso de la guía de aprendizaje. Consulte “Guía de aprendizaje de seguridad de Java SE - Paso 4”.

Guía de aprendizaje de seguridad de Java SE - Paso 4

El siguiente paso le explica cómo habilitar una capa de seguridad para la comunicación entre los puntos finales del entorno.

Antes de empezar

Asegúrese de haber completado el apartado “Guía de aprendizaje de seguridad de Java SE - Paso 3” en la página 79 antes de llevar a cabo esta tarea.

Acerca de esta tarea

La topología de eXtreme Scale da soporte a Transport Layer Security/Secure Sockets Layer (TLS/SSL) para la comunicación segura entre puntos finales de ObjectGrid (cliente, servidores de contenedor y servidores de catálogo). Este paso de la guía de aprendizaje se basa en los pasos anteriores para habilitar la seguridad de transporte.

Procedimiento

1. **Cree almacenes de claves y claves de TLS/SSL**

   Para habilitar la seguridad de transporte, debe crear un almacén de claves y un almacén de confianza. Este ejercicio sólo crea un par de almacén de claves y almacén de confianza. Estos almacenes se utilizan para los servidores de catálogo, servidores de contenedor y clientes ObjectGrid, y se crean con la herramienta de claves de JDK.

   - *Crear una clave privada en el almacén de claves*

     ```
     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
     ```

     Con este mandato, se crea un almacén de claves key.jks con una clave "ogsample" almacenada en él. Este almacén de claves key.jks se utilizará como el almacén de claves SSL.

   - *Exportar el certificado público*

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

     Con este mandato, se extrae el certificado público de la clave "ogsample" y se almacena en el archivo temp.key.

   - *Importar el certificado público del cliente en el almacén de confianza*

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

     Con este mandato, el certificado público se ha añadido al almacén de claves trust.jks. Este trust.jks se utiliza como el almacén de confianza SSL.

2. **Configuración de los archivos de propiedades de ObjectGrid**

   En este paso, debe configurar los archivos de propiedades de ObjectGrid para habilitar la seguridad de transporte.

   Primero, copie los archivos key.jks y trust.jks en el directorio objectgridRoot/security.

   Establezca las propiedades siguientes en el archivo `client.properties` y `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: el valor de transportType se establece en "SSL-Required", que significa que el transporte requiere SSL. Por lo tanto, todos los puntos finales de ObjectGrid (clientes, servidores de catálogo y servidores de contenedor deben tener establecida la configuración SSL y toda la comunicación de transporte estará cifrada.

   Las otras propiedades se utilizan para establecer las configuraciones SSL. Consulte “Transport Layer Security (TLC) y Secure Sockets Layer (SSL)” en la página 515 para ver una explicación detallada. Asegúrese de seguir las instrucciones de este tema para actualizar el archivo `orb.properties`.

   Asegúrese de que sigue esta página para actualizar el archivo `orb.properties`.

   En el archivo `server.properties` , debe añadir una propiedad adicional clientAuthentication y establecerla en false (falso). En el lado del servidor, no es necesario que confíe en el cliente.

   ```
   clientAuthentication=false
   ```

3. **Ejecute la aplicación**

   Los mandatos son los mismos que en el tema “Guía de aprendizaje de seguridad de Java SE - Paso 3” en la página 79.

   Utilice los siguientes mandatos para iniciar un servidor de catálogo.

   a. Vaya al directorio bin: `cd objectgridRoot/bin`

   b. Inicie el servidor de catálogo:

      - 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"
        ```

      Los archivos `security.xml` y `server.properties` se crearon en la página “Guía de aprendizaje de seguridad de Java SE - Paso 2” en la página 73.

      Utilice la opción **`-JMXServicePort`** para especificar explícitamente el puerto JMX para el servidor. Esta opción es necesaria para utilizar el programa de utilidad **`xscmd`**.

      Ejecute un servidor de contenedor de ObjectGrid seguro:

   c. Vuelva al directorio bin: `cd objectgridRoot/bin`

   d.

- Linux UNIX

```
startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints
localhost:2809 -serverProps ../security/server.properties
-JMXServicePort 11002 -jvmArgs
-Djava.security.auth.login.config="../security/og_jaas.config"
-Djava.security.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"
```

Tenga en cuenta las siguientes diferencias del mandato de inicio de servidor de contenedor anterior:

- Utilice `SecureSimpleApp.xml` en lugar de archivos `SimpleApp.xml`.
- Añada otro -Djava.security.auth.policy para establecer el archivo de política de autorización de JAAS para el proceso de servidor de contenedor.

Ejecute el siguiente mandato para la autenticación de cliente:

a. `cd objectgridRoot/bin`

b.

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

Como el usuario "manager" tiene permiso para todas las correlaciones del accounting ObjectGrid, la aplicación se ejecuta satisfactoriamente.

Puede utilizar el programa de utilidad **xscmd** para que se muestren los tamaños de correlación de la cuadrícula "accounting".

- Vaya hasta el directorio `objectgridRoot/bin`.
- Utilice el mandato **xscmd** para que se muestren los tamaños de correlación:
  - 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
    ```

    Tenga en cuenta que se especifica el puerto JMX del servicio de catálogo utilizando aquí -p 11001.

    Obtendrá la siguiente salida.

```
Este programa de utilidad administrativo se proporciona sólo como un ejemplo y no se debe
considerar como un componente completamente soportado del producto WebSphere eXtreme Scale.
Conexión al servicio de catálogo en localhost:1099
********** Mostrando resultados para Grid - accounting, MapSet - mapSet1 **********
*** Listado de correlaciones para c0 ***
Nombre de correlación: customer Núm. de partición: 0 Tamaño de correlación: 1 Tipo de fragmento: primario
Total de servidores: 1
Recuento total de dominios: 1
```

Ejecución de la aplicación con un almacén de claves incorrecto

Si el almacén de confianza no contiene el certificado público de la clave privada en el almacén de claves, obtendrá una excepción que indica que no se puede confiar en la clave.

Para mostrarlo, cree otro almacén de claves 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
```

Después, modifique server.properties de forma que keyStore señale a este nuevo almacén de claves key2.jks:

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

Ejecute el siguiente mandato para iniciar el servidor de catálogo:

a. Desplácese al directorio bin: `cd objectgridRoot/bin`

b. Inicie el servidor de catálogo:

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"
```

Verá la siguiente excepción:

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

Finalmente, vuelva a cambiar el archivo `server.properties` para utilizar el archivo `key.jks`.

Guía de aprendizaje: Integrar la seguridad de WebSphere eXtreme Scale con WebSphere Application Server

Esta guía de aprendizaje muestra cómo proteger un despliegue de servidor de WebSphere eXtreme Scale en un entorno de WebSphere Application Server.

Objetivos del aprendizaje

Los objetivos de aprendizaje de esta guía de aprendizaje son los siguientes:

- Configurar WebSphere eXtreme Scale para utilizar plug-ins de autenticación de WebSphere Application Server
- Configurar la seguridad de transporte de WebSphere eXtreme Scale para utilizar la configuración CSIv2 de WebSphere Application Server
- Utilizar autenticación JAAS (Java Authentication and Authorization Service) en WebSphere Application Server
- Utilizar un módulo de inicio de sesión personalizado para autorización JAAS basada en grupo
- Utilizar el programa de utilidad **xscmd** de WebSphere eXtreme Scale en el entorno de WebSphere Application Server

Tiempo necesario

Esta guía de aprendizaje requiere aproximadamente 4 horas desde el principio hasta el final.

Introducción: Integrar la seguridad de WebSphere eXtreme Scale con WebSphere Application Server utilizando los plug-ins de autenticación de WebSphere Application Server

En esta guía de aprendizaje, integra la seguridad de WebSphere eXtreme Scale con WebSphere Application Server. En primer lugar, configura la autenticación con una aplicación web simple que utiliza credenciales de usuario autenticadas desde la hebra actual para conectar al ObjectGrid. A continuación, investiga el cifrado de datos transferidos entre el cliente y el servidor con seguridad de capa de transporte. Para otorgar a los usuarios diversos niveles de permisos, puede configurar JAAS (Java Authentication and Authorization Service). Después de completar la configuración, puede utilizar el programa de utilidad **xscmd** para supervisar las cuadrículas de datos y correlaciones.

Esta guía de aprendizaje asume que todos los clientes, servidores de contenedor y servidores de catálogo de WebSphere eXtreme Scale se despliegan en el entorno de WebSphere Application Server.

Objetivos del aprendizaje

Los objetivos de aprendizaje de esta guía de aprendizaje son los siguientes:

- Configurar WebSphere eXtreme Scale para utilizar plug-ins de autenticación de WebSphere Application Server
- Configurar la seguridad de transporte de WebSphere eXtreme Scale para utilizar la configuración CSIv2 de WebSphere Application Server
- Utilizar autenticación JAAS (Java Authentication and Authorization Service) en WebSphere Application Server
- Utilizar un módulo de inicio de sesión personalizado para autorización JAAS basada en grupo
- Utilizar el programa de utilidad **xscmd** de WebSphere eXtreme Scale en el entorno de WebSphere Application Server

Tiempo necesario

Esta guía de aprendizaje requiere aproximadamente 4 horas desde el principio hasta el final.

Nivel de conocimientos

Intermedio.

A quién va dirigida

Los desarrolladores y administradores interesados en la integración de seguridad entre WebSphere eXtreme Scale y WebSphere Application Server.

Requisitos y topología del sistema

- WebSphere Application Server Versión 6.1 o Versión 7.0.0.11 o posterior
- Actualice el tiempo de ejecución de Java para aplicar el arreglo siguiente: IZ79819: IBMJDK NO PUEDE LEER SENTENCIA PRINCIPAL CON ESPACIO EN BLANCO DE ARCHIVO DE SEGURIDAD

Esta guía de aprendizaje utiliza cuatro servidores de aplicaciones WebSphere Application Server y un gestor de despliegue para mostrar el ejemplo.

Requisitos previos

Es útil disponer de conocimientos básicos de los elementos siguientes antes de iniciar esta guía de aprendizaje:

- El modelo de programación de WebSphere eXtreme Scale
- Los conceptos básicos de seguridad de WebSphere eXtreme Scale
- Los conceptos básicos de seguridad de WebSphere Application Server

Para obtener información previa sobre la integración de la seguridad de WebSphere eXtreme Scale y WebSphere Application Server, consulte "Integración de la seguridad con WebSphere Application Server" en la página 525.

Módulo 1: Preparar WebSphere Application Server

Antes de comenzar la guía de aprendizaje para la integración con WebSphere eXtreme Scale, debe crear una configuración de seguridad básica en WebSphere Application Server.

Objetivos del aprendizaje

Con las lecciones de este módulo, aprenderá a:

- Configurar la seguridad de WebSphere Application Server para utilizar un repositorio federado basado en un archivo interno como un registro de cuentas de usuario.
- Crear grupos de usuarios y usuarios.
- Crear clústeres para la aplicación y servidores WebSphere eXtreme Scale.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Lección 1.1: Comprender la topología y obtener los archivos de la guía de aprendizaje

Para preparar el entorno para la guía de aprendizaje, debe configurar la seguridad de WebSphere Application Server. Puede configurar la administración y la seguridad de la aplicación mediante repositorios federados basados en archivo interno como registro de cuentas de usuario.

Esta lección le guía por la topología de ejemplo y las aplicaciones que se utilizan en esta guía de aprendizaje. Para empezar a ejecutar la guía de aprendizaje, debe descargar las aplicaciones y colocar los archivos de configuración en las ubicaciones correctas para su entorno. Puede descargar la aplicación de ejemplo desde la wiki de WebSphere eXtreme Scale.

Topología de ejemplo de WebSphere Application Server: Esta guía de aprendizaje le guía por la creación de cuatro servidores de aplicaciones WebSphere Application Server para mostrar las aplicaciones de ejemplo con la seguridad habilitada. Estos servidores de aplicaciones se agrupan en dos clústeres, cada uno de ellos con dos servidores:

- **Clúster appCluster**: aloja la aplicación empresarial de ejemplo EmployeeManagement. Este clúster tiene dos servidores de aplicaciones: s1 y s2.
- **Clúster xsCluster**: aloja los servidores de contenedor eXtreme Scale. Este clúster tiene dos servidores de aplicaciones: xs1 y xs2.

En esta topología de despliegue, los servidores de aplicaciones s1 y s2 son los servidores de cliente que acceden a los datos que se almacenan en la cuadrícula de datos. Los servidores xs1 y xs2 son los servidores de contenedor que alojan la cuadrícula de datos.

El servidor de catálogo se despliega en el proceso de gestor de despliegue de forma predeterminada. Esta guía de aprendizaje utiliza el comportamiento predeterminado. En un entorno de producción no se recomienda alojar el servidor de catálogo en el gestor de despliegue. En un entorno de producción, debe crear un dominio de servicio de catálogo para definir dónde se inician los servidores de catálogo. Si desea más información, consulte "Creación de dominios de servicio de catálogo en WebSphere Application Server" en la página 257.

Configuración alternativa: puede alojar todos los servidores de aplicaciones en un solo clúster como, por ejemplo, en el clúster appCluster. Con esta configuración, todos los servidores del clúster son tanto clientes como servidores de contenedor. Esta guía de aprendizaje utiliza dos clústeres para distinguir entre los servidores de aplicaciones que alojan los clientes y servidores de contenedor.

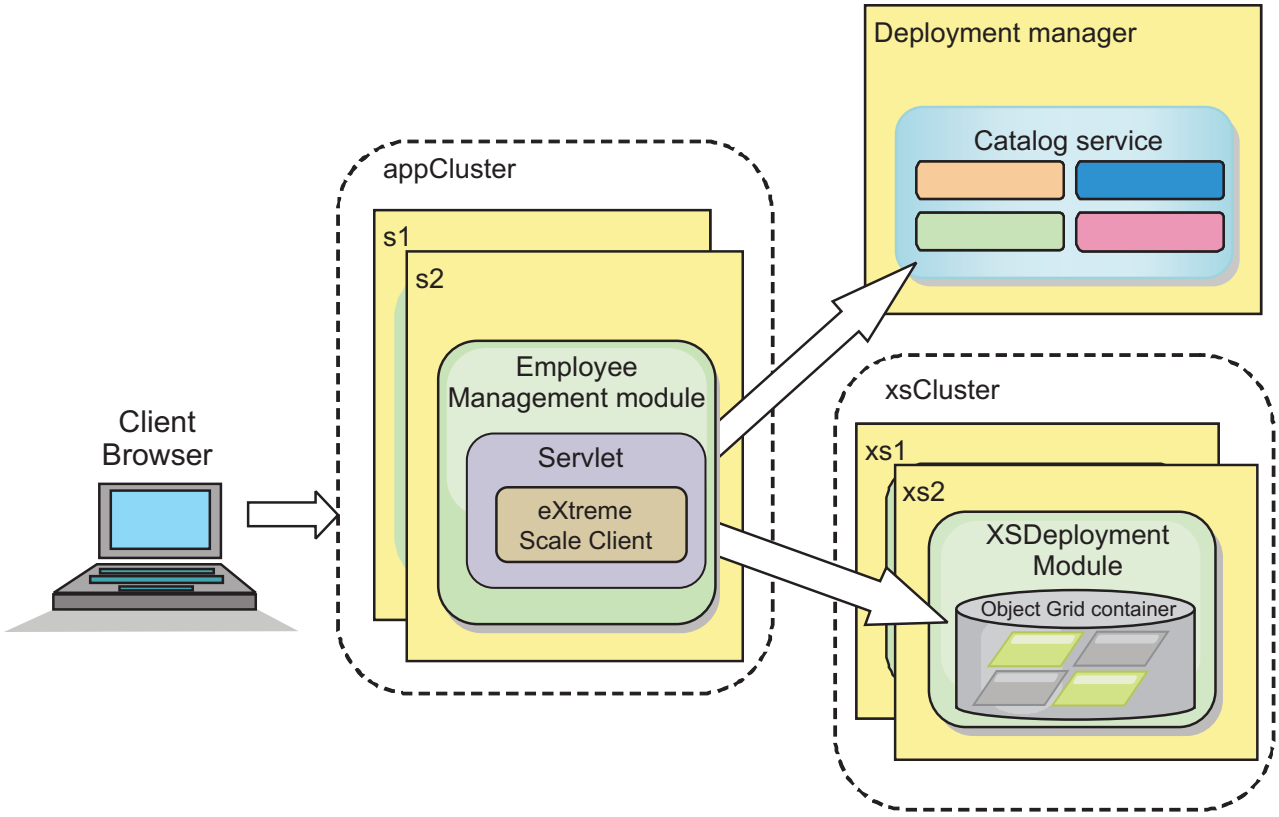


Figura 19. Topología de la guía de aprendizaje

Aplicaciones: En esta guía de aprendizaje, utiliza dos aplicaciones y un archivo de biblioteca compartida:

- **`EmployeeManagement.ear`**: la aplicación `EmployeeManagement.ear` es una aplicación empresarial Java 2 Platform, Enterprise Edition (J2EE) simplificada. Contiene un módulo web para gestionar los perfiles de empleado. El módulo web contiene el archivo `management.jsp` para visualizar, insertar, actualizar y suprimir perfiles de empleado almacenados en los servidores de contenedor.

- **XSDeployment.ear**: esta aplicación contiene un módulo de aplicación empresarial sin artefactos de la aplicación. Los objetos de memoria caché se empaquetan en el archivo EmployeeData.jar. El archivo EmployeeData.jar se despliega como una biblioteca compartida para el archivo XSDeployment.ear, de forma que el archivo XSDeployment.ear pueda acceder a las clases. La finalidad de esta aplicación es empaquetar los archivos de configuración de eXtreme Scale. Cuando se inicia esta aplicación empresarial, la ejecución de eXtreme Scale detecta autómáticamente los archivos de configuración de eXtreme Scale, de forma que se crean los servidores de contenedor. Estos archivos de configuración incluyen los archivos objectGrid.xml y objectGridDeployment.xml.
- **EmployeeData.jar**: este archivo jar contiene una sola clase: la clase com.ibm.websphere.sample.xs.data.EmployeeData. Esta clase representa los datos de los empleados almacenados en la cuadrícula. Este archivo de archivado Java (JAR) se despliega con los archivos EmployeeManagement.ear y XSDeployment.ear como una biblioteca compartida.

Obtener los archivos de la guía de aprendizaje:

1. Descargue los archivos WASSecurity.zip y security.zip. Puede descargar la aplicación de ejemplo desde la wiki de WebSphere eXtreme Scale.
2. Extraiga el archivo WASSecurity.zip en un directorio para visualizar los artefactos binarios y de origen, por ejemplo, el directorio /wxs_samples/. Se hace referencia a este directorio como *inicio_samples* para el resto de la guía de aprendizaje. Para ver una descripción del contenido del archivo WASSecurity.zip y cómo cargar el origen en el espacio de trabajo de Eclipse, consulte el archivo README.txt en el paquete.
3. Extraiga el archivo security.zip en el directorio *inicio_samples*. El archivo security.zip contiene los siguientes archivos de configuración de seguridad utilizados en esta guía de aprendizaje:
   - catServer2.props
   - server2.props
   - client2.props
   - securityWAS2.xml
   - xsAuth2.props

Acerca de los archivos de configuración:

Los archivos objectGrid.xml y objectGridDeployment.xml crean las cuadrículas de datos y correlaciones que almacenan los datos de aplicación.

Estos archivos de configuración se deben denominar objectGrid.xml y objectGridDeployment.xml. Cuando se inicia el servidor de aplicaciones, eXtreme Scale detecta estos archivos en el directorio META-INF de los módulos EJB y web. Si se encuentran estos archivos, se asume que la máquina virtual Java (JVM) actúa como un servidor de contenedor para las cuadrículas de datos definidas en los archivos de configuración.

Archivo objectGrid.xml

El archivo objectGrid.xml ha definido un ObjectGrid denominado Grid. La cuadrícula de datos Grid tiene una cuadrícula, la correlación Map1, que almacena el perfil de empleado para la aplicación.

```
<?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>
```

Archivo `objectGridDeployment.xml`

El archivo `objectGridDeployment.xml` especifica cómo desplegar la cuadrícula de datos `Grid`. Cuando se despliega la cuadrícula, tiene cinco particiones y una réplica síncrona.

```
<?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>
```

Punto de comprobación de la lección:

En esta lección, ha aprendido sobre la topología de la guía de aprendizaje y ha añadido archivos de configuración y aplicaciones de ejemplo al entorno.

Si desea obtener más información sobre cómo iniciar automáticamente los servidores de contenedor, consulte "Configuración de aplicaciones WebSphere Application Server para el inicio automático de servidores de contenedor" en la página 274.

Lección 1.2: Configurar el entorno de WebSphere Application Server

Para preparar el entorno para la guía de aprendizaje, debe configurar la seguridad de WebSphere Application Server. Habilite la administración y la seguridad de la aplicación mediante repositorios federados basados en archivo interno como un registro de cuentas de usuario. A continuación, puede crear clústeres de servidores para alojar la aplicación de cliente y los servidores de contenedor.

Los pasos siguientes se han escrito utilizando WebSphere Application Server Versión 7.0. Sin embargo, también puede aplicar los conceptos en versiones anteriores de WebSphere Application Server.

Configurar la seguridad de WebSphere Application Server:

1. Configure la seguridad de WebSphere Application Server.
   a. En la consola administrativa de WebSphere Application Server, pulse **Seguridad** > **Seguridad global**.
   b. Seleccione **Repositorios federados** como el **Repositorio de cuentas de usuario**. Pulse **Establecer como actual**.
   c. Pulse **Configurar..** para ir al panel **Repositorios federados**.
   d. Especifique el **Nombre de usuario administrativo primario**, por ejemplo, `admin`. Pulse **Aplicar**.
   e. Cuando se le solicite, especifique el usuario administrativo y la contraseña y pulse **Aceptar**. Guarde los cambios.

f. En la página **Seguridad global**, compruebe que el valor **Repositorios federados** esté establecido en el registro de cuentas de usuario actual.
g. Seleccione los elementos siguientes: **Habilitar seguridad administrativa**, **Habilitar seguridad de la aplicación** y **Utilizar seguridad Java 2 para restringir el acceso a la aplicación a recursos locales**. Pulse **Aplicar** y guarde los cambios.
h. Reinicie el gestor de despliegue y los servidores de aplicaciones en ejecución.

La seguridad administrativa de WebSphere Application Server se habilita mediante los repositorios federados basados en archivo internos como registro de cuentas de usuario.

2. Cree dos grupos de usuarios: adminGroup y operatorGroup.
   a. Pulse **Usuarios y grupos** > **Gestionar grupos** > **Crear...**
   b. Especifique `adminGroup` como nombre de grupo. Especifique `Grupo de administración` como la descripción. Pulse **Crear**.
   c. Pulse **Crear similar**. Especifique `operatorGroup` como nombre de grupo. Especifique `Grupo de operadores` como descripción. Pulse **Crear**.
   d. Pulse **Cerrar**.
3. Cree los usuarios `admin1` y `operator1`.
   a. Pulse **Usuarios y grupos** > **Gestionar usuarios** > **Crear...**
   b. Cree un usuario denominado `admin1` con el nombre `Joe` y el apellido `Doe` con la contraseña `admin1`. Pulse **Crear**.
   c. Cree un segundo usuario. Pulse **Crear similar** para crear un usuario denominado `operator1` con el nombre `Jane` y el apellido `Doe` con la contraseña `operator1`. Pulse **Crear**. Pulse **Cerrar**.
4. Añada usuarios a los grupos de usuarios. Añada el usuario `admin1` al grupo `adminGroup` y el usuario `operator1` al grupo `operatorGroup`.
   a. Pulse **Usuarios y grupos** > **Gestionar usuarios**.
   b. Busque usuarios para añadirlos a los grupos. Pulse **Buscar..** y establezca el valor de búsqueda en un asterisco (*) para visualizar todos los usuarios.
   c. En el resultado de la búsqueda, seleccione el usuario `admin1` y pulse el separador **Grupos**. Pulse **Añadir** para añadir el grupo.
   d. Busque en los grupos para encontrar los grupos disponibles. Pulse el grupo `adminGroup` y **Añadir**.
   e. Repita estos pasos para añadir el usuario `operator1` al grupo de usuarios `operatorGroup`.
5. Guarde los cambios, cierre la sesión en la consola administrativa y reinicie el gestor de despliegue y el agente de nodo para habilitar los valores de seguridad.

Ha habilitado la seguridad y los usuarios y grupos de usuarios creados tienen acceso administrativo y de operador a la configuración de WebSphere Application Server.

Crear clústeres de servidores:

Cree dos clústeres de servidores en la configuración de WebSphere Application Server: el clúster `appCluster` para alojar la aplicación de ejemplo para la guía de aprendizaje y el clúster `xsCluster` para alojar la cuadrícula de datos.

1. En la consola administrativa de WebSphere Application Server, abra el panel de clústeres. Pulse **Servidores** > **Clústeres** > **Clústeres de servidores de aplicaciones WebSphere** > **Nuevo**.
2. Especifique `appCluster` como nombre de clúster, deje seleccionada la opción **Preferir local** y pulse **Siguiente**.
3. Cree servidores en el clúster. Cree un servidor denominado `s1`, manteniendo las opciones predeterminadas. Añada un miembro de clúster adicional denominado `s2`.
4. Complete los demás pasos del asistente para crear el clúster. Guarde los cambios.
5. Repita estos pasos para crear el clúster `xsCluster`. Este clúster tiene dos servidores, denominados `xs1` y `xs2`.

Punto de comprobación de la lección:

Ha habilitado la seguridad global para la célula de WebSphere Application Server, ha creado usuarios y grupos de usuarios y ha creado clústeres para alojar la aplicación y la cuadrícula de datos.

Módulo 2: Configurar WebSphere eXtreme Scale para utilizar plug-ins de autenticación de WebSphere Application Server

Después de haber creado la configuración de WebSphere Application Server, puede integrar la autenticación de WebSphere eXtreme Scale con WebSphere Application Server.

Cuando un cliente de WebSphere eXtreme Scale se conecta a un servidor de contenedor que requiere autenticación, el cliente debe proporcionar un generador de credenciales representado por la interfaz com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. Un generador de credenciales es una fábrica para crear una credencial de cliente. Una credencial de cliente puede ser: un par de nombre de usuario y contraseña, un ticket Kerberos, un certificado de cliente o datos de identificación de cliente en cualquier formato que hayan acordado el cliente y el servidor. Consulte la Documentación de la API de credenciales para obtener más información. En este ejemplo, el cliente de WebSphere eXtreme Scale es la aplicación web EmployeeManagment que se despliega en el clúster appCluster. La credencial de cliente es una señal de seguridad de WebSphere que representa la identidad del usuario web.

Objetivos del aprendizaje

Con las lecciones de este módulo, aprenderá a:

- Configurar la seguridad de cliente-servidor.
- Configurar la seguridad del servidor de catálogo.
- Configurar la seguridad del servidor de contenedor.
- Instalar y ejecutar la aplicación de ejemplo.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Lección 2.1: Configurar la seguridad de cliente-servidor

El archivo de propiedades de cliente indica la clase de implementación CredentialGenerator que se utilizará.

Configure el archivo de propiedades de cliente con la propiedad de la JVM **-Dobjectgrid.client.props**. El nombre de archivo especificado para esta propiedad es una vía de acceso de archivo absoluta, por ejemplo, *inicio_samples*/security/client2.props. Consulte Archivo de propiedades de cliente si desea más información sobre el archivo de propiedades de cliente.

Contenido del archivo de propiedades de cliente:

Este ejemplo utiliza señales de seguridad de WebSphere Application Server como la credencial de cliente. El archivo client2.props se encuentra en el directorio *inicio_samples*/security. El archivo client2.props incluye los valores siguientes:

securityEnabled
Cuando se establece en true, indica que el cliente debe enviar la información de seguridad disponible al servidor.

credentialAuthentication
Cuando se establece en Supported, indica que el cliente da soporte a la autenticación de credenciales.

credentialGeneratorClass
Indica la clase com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator para que el cliente recuperar las señales de seguridad de la hebra. Consulte "Integración de la seguridad con WebSphere Application Server" en la página 525 para obtener información sobre cómo se recuperan las señales de seguridad.

Definición del archivo de propiedades de cliente mediante las propiedades de la JVM (Java virtual machine):

En la consola administrativa, complete los pasos siguiente para los servidores s1 y s2 del clúster appCluster. Si está utilizando una topología distinta, complete los pasos siguientes para todos los servidores de aplicaciones en los que se despliega la aplicación EmployeeManagement.

1. **Servidores** > **Servidores de aplicaciones WebSphere** > *nombre_servidor* > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**.
2. Cree la siguiente propiedad de JVM genérica para establecer la ubicación del archivo de propiedades de cliente:
   ```
   -Dobjectgrid.client.props=inicio_samples/security/client2.props
   ```
3. Pulse **Aceptar** y guarde los cambios.

Punto de comprobación de la lección:

Ha editado el archivo de propiedades de cliente y ha configurado los servidores en el clúster appCluster para utilizar el archivo de propiedades de cliente. Este archivo de propiedades indica la clase de implementación CredentialGenerator que se utilizará.

Lección 2.2: Configurar seguridad del servidor de catálogo

Un servidor de catálogo contiene dos niveles distintos de información de seguridad: las propiedades de seguridad comunes a todos los servidores WebSphere eXtreme Scale, incluidos el servicio de catálogo y los servidores de contenedor, y las propiedades de seguridad específicas del servidor de catálogo.

Las propiedades de seguridad comunes a los servidores de catálogo y los servidores de contenedor se configuran en el archivo de descriptor XML de seguridad. Un archivo de propiedades comunes es la configuración de

autenticador, que representa el registro de usuarios y el mecanismo de autenticación. Consulte Archivo XML de descriptor de seguridad para obtener más información sobre las propiedades de seguridad.

Para configurar el archivo de descriptor XML de seguridad, cree una propiedad -Dobjectgrid.cluster.security.xml.url en el argumento de máquina virtual Java (JVM). El nombre de archivo especificado para esta propiedad está en formato de URL, por ejemplo, `file:///`*`inicio_samples`*`/security/securityWAS2.xml`.

Archivo `securityWAS2.xml`:

En esta guía de aprendizaje, el archivo `securityWAS2.xml` está en el directorio *inicio_samples*`/security`. A continuación se muestra el contenido del archivo `securityWAS2.xml` con los comentarios eliminados:

```
<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>
```

Se definen las propiedades siguientes en el archivo `securityWAS2.xml`:

securityEnabled
: La propiedad securityEnabled se establece en `true`, lo que indica al servidor de catálogo que la seguridad global de WebSphere eXtreme Scale está habilitada.

authenticator
: El autenticador se configura como la clase com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenAuthenticator. Con esta implementación incorporada del plug-in Authenticator, el servidor WebSphere eXtreme Scale puede convertir las señales de seguridad en un objeto Subject. Consulte “Integración de la seguridad con WebSphere Application Server” en la página 525 para obtener más información sobre cómo se convierten las señales de seguridad.

Archivo `catServer2.props`:

El archivo de propiedades del servidor almacena las propiedades específicas del servidor, que incluyen las propiedades de seguridad específicas del servidor. Consulte Archivo de propiedades de servidor para obtener más información. Puede configurar el archivo de propiedades del servidor con el programa de utilidad `-Dobjectgrid.server.props` en el argumento de JVM. Especifique el valor de nombre de archivo para esta propiedad en una vía de acceso absoluta, como por ejemplo *inicio_samples*`/security/catServer2.props`. Para esta guía de aprendizaje, se incluye un archivo `catServer2.props` en el directorio *inicio_samples*`/security`. A continuación se muestra el contenido del archivo `catServer2.props` con los comentarios eliminados:

securityEnabled
: La propiedad securityEnabled se establece en `true` para indicar que este servidor de catálogo es un servidor seguro.

credentialAuthentication
La propiedad credentialAuthentication se establece en `Required`, de forma que cualquier cliente que se conecte al servidor necesitará proporcionar una credencial.

secureTokenManagerType
La propiedad secureTokenManagerType se establece en `none` para indicar que el secreto de autenticación no está cifrado al unirse a los servidores existentes.

authenticationSecret
La propiedad authenticationSecret se establece en `ObjectGridDefaultSecret`. Esta serie secreta se utiliza para la unión al clúster de servidores eXtreme Scale. Cuando un servidor se une a la cuadrícula de datos, se solicita que proporcione la serie secreta. Si la serie secreta del servidor que se une coincide con la serie en el servidor de catálogo, se aceptará el servidor que se desea unir. Si la serie no coincide, se rechaza la solicitud de unión.

transportType
La propiedad transportType se establece inicialmente en `TCP/IP`. Más adelante en la guía de aprendizaje, la seguridad de transporte se habilitará.

Definición del archivo de propiedades de servidor con propiedades de JVM:

Establezca el archivo de propiedades del servidor en el servidor del gestor de despliegue. Si utiliza una topología distinta a la topología de esta guía de aprendizaje, establezca el archivo de propiedades del servidor en todos los servidores de aplicaciones que utiliza para alojar los servidores de contenedor.

1. Abra la configuración de la máquina virtual Java para el servidor. En la consola administrativa, pulse **Administración del sistema** > **Gestor de despliegue** > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**.
2. Añada los siguientes argumentos de JVM genéricos:
```
-Dobjectgrid.cluster.security.xml.url=file:///inicio_samples/security/securityWAS2.xml
-Dobjectgrid.server.props=inicio_samples/security/catServer2.props
```
3. Pulse **Aceptar** y guarde los cambios.

Punto de comprobación de la lección:

Ha configurado la seguridad de servidor de catálogo asociando los archivos `securityWAS2.xml` y `catServer2.props` al gestor de despliegue, que aloja el proceso de servidor de catálogo en la configuración de WebSphere Application Server.

Lección 2.3: Configurar la seguridad del servidor de contenedor

Cuando un servidor de contenedor se conecta al servicio de catálogo, el servidor de contenedor obtiene todas las configuraciones de seguridad configuradas en el archivo XML de seguridad de cuadrícula de objetos como, por ejemplo, configuración de autenticación, el valor de tiempo de espera de inicio de sesión y otra información de configuración. Un servidor de contenedor también tiene sus propias propiedades de seguridad específicas del servidor en el archivo de propiedades del servidor.

Configure el archivo de propiedades del servidor con la propiedad de máquina virtual Java (JVM) `-Dobjectgrid.server.props`. El nombre de archivo de esta propiedad es una vía de acceso de archivo absoluta, por ejemplo, *inicio_samples*`/security/server2.props`.

En esta guía de aprendizaje, los servidores de contenedor se alojan en los servidores xs1 y xs2 del clúster xsCluster.

Archivo server2.props:

El archivo server2.props se encuentra en el directorio *inicio_samples*/security del directorio WASSecurity. Las propiedades definidas en el archivo server2.props son las siguientes:

securityEnabled
La propiedad securityEnabled se establece en true para indicar que el servidor de contenedor es un servidor seguro.

credentialAuthentication
La propiedad credentialAuthentication se establece en Required, de forma que cualquier cliente que se conecte al servidor necesitará proporcionar una credencial.

secureTokenManagerType
La propiedad secureTokenManagerType se establece en none para indicar que el secreto de autenticación no está cifrado al unirse a los servidores existentes.

authenticationSecret
La propiedad authenticationSecret se establece en ObjectGridDefaultSecret. Esta serie secreta se utiliza para la unión al clúster de servidores eXtreme Scale. Cuando un servidor se une a la cuadrícula de datos, se solicita que proporcione la serie secreta. Si la serie secreta del servidor que se une coincide con la serie en el servidor de catálogo, se aceptará el servidor que se desea unir. Si la serie no coincide, se rechaza la solicitud de unión.

Definición del archivo de propiedades de servidor con propiedades de JVM:

Establezca el archivo de propiedades del servidor en los servidores xs1 y xs2. Si no está utilizando la topología para esta guía de aprendizaje, establezca el archivo de propiedades del servidor en todos los servidores de aplicaciones que está utilizando para alojar servidores de contenedor.

1. Abra la página de la máquina virtual Java para el servidor. **Servidores** > **Servidores de aplicaciones** > *nombre_servidor* > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**
2. Añada los argumentos de JVM genéricos:
   `-Dobjectgrid.server.props=`*inicio_samples*`/security/server2.props`
3. Pulse **Aceptar** y guarde los cambios.

Punto de comprobación de la lección:

Ahora la autenticación del servidor WebSphere eXtreme Scale está protegida. Si se configura esta seguridad, será necesario que todas las aplicaciones que intenten conectarse a los servidores WebSphere eXtreme Scale proporcionen una credencial. En esta guía de aprendizaje, WSTokenAuthenticator es el autenticador. Como resultado, es necesario que el cliente proporcione una señal de seguridad de WebSphere Application Server.

Lección 2.4: Instalar y ejecutar el ejemplo

Una vez que se ha configurado la autenticación, puede instalar y ejecutar la aplicación de ejemplo.

Creación de una biblioteca compartida para el archivo `EmployeeData.jar`:

1. En la consola administrativa de WebSphere Application Server, abra la página **Bibliotecas compartidas**. Pulse **Entorno** > **Bibliotecas compartidas**.
2. Elija el ámbito **célula**.
3. Cree la biblioteca compartida. Pulse **Nueva**. Especifique `EmployeeManagementLIB` como **Nombre**. Especifique la vía de acceso del archivo `EmployeeData.jar` en la classpath, por ejemplo, *inicio_samples*`/WASSecurity/EmployeeData.jar`.
4. Pulse **Aplicar**.

Instalación del ejemplo:

1. Instale el archivo `EmployeeManagement.ear`.
   a. Para empezar la instalación, pulse **Aplicaciones** > **Nueva aplicación** > **Nueva aplicación empresarial**. Elija la vía de acceso detallada para la instalación de la aplicación.
   b. En el paso **Correlacionar módulos con servidores**, especifique el clúster `appCluster` para instalar el módulo EmployeeManagementWeb.
   c. En el paso **Bibliotecas compartidas de correlación**, seleccione el módulo `EmployeeManagementWeb`.
   d. Pulse **Bibliotecas compartidas de referencia**. Seleccione la biblioteca `EmployeeManagementLIB`.
   e. Correlacione el rol `webUser` con **Todos los autenticados en el reino de la aplicación**.
   f. Pulse **Aceptar**.

   Los clientes se ejecutan en los servidores `s1` y `s2` de este clúster.
2. Instale el archivo `XSDeployment.ear` de ejemplo.
   a. Para empezar la instalación, pulse **Aplicaciones** > **Nueva aplicación** > **Nueva aplicación empresarial**. Elija la vía de acceso detallada para la instalación de la aplicación.
   b. En el paso **Correlacionar módulos con servidores**, especifique el clúster `xsCluster` para instalar el módulo web `XSDeploymentWeb`.
   c. En el paso **Bibliotecas compartidas de correlación**, seleccione el módulo `XSDeploymentWeb`.
   d. Pulse **Bibliotecas compartidas de referencia**. Seleccione la biblioteca `EmployeeManagementLIB`.
   e. Pulse **Aceptar**.

   Los servidores `xs1` y `xs2` de este clúster alojan los servidores de contenedor.
3. Reinicie el gestor de despliegue. Cuando se inicia el gestor de despliegue, se inicia también el servidor de catálogo. Si mira el archivo `SystemOut.log` del gestor de despliegue, puede ver el siguiente mensaje que indica que se ha cargado el archivo de propiedades del servidor eXtreme Scale.

   ```
   CWOBJ0913I: Los archivos de propiedades del servidor se han cargado:
   /wxs_samples/security/catServer2.props.
   ```
4. Reinicie el clúster xsCluster. Cuando se inicia el clúster xsCluster, se inicia la aplicación `XSDeployment` y el servidor de contenedor se inicia en los servidores xs1 y xs2, respectivamente. Si mira el archivo `SystemOut.log` de los servidores xs1 y xs2, se visualiza el mensaje siguiente que indica que el archivo de propiedades del servidor se ha cargado:

   ```
   CWOBJ0913I: Los archivos de propiedades del servidor se han cargado:
   /wxs_samples/security/server2.props.
   ```

5. Reinicie el clúster appClusters. Cuando se inicia el clúster appCluster, se inicia también la aplicación EmployeeManagement. Si mira el archivo `SystemOut.log` de los servidores s1 y s2, puede ver el mensaje siguiente que indica que el archivo de propiedades de cliente se ha cargado.

   ```
   CWOBJ0924I: El archivo de propiedades de cliente {0} se ha cargado.
   ```

   Puede ignorar los mensajes de aviso referentes a las propiedades authenticationRetryCount, transportType y clientCertificateAuthentication. Se utilizarán los valores predeterminados porque no se han especificado los valores en el archivo de propiedades.Si utiliza WebSphere eXtreme Scale Versión 7.0, se visualiza el mensaje CWOBJ9000I, solo en inglés, para indicar que el archivo de propiedades de cliente se ha cargado. Si no ve el mensaje esperado, compruebe que ha configurado la propiedad -Dobjectgrid.server.props o -Dobjectgrid.client.props en el argumento de JVM. Si sí tiene las propiedades configuradas, asegúrese de que el guión (-) sea un carácter UTF.

Ejecución de la aplicación de ejemplo:

1. Ejecute el archivo `management.jsp`. En un navegador web, acceda a `http://<su_nombre_servidor>:<puerto>/EmployeeManagementWeb/management.jsp`. Por ejemplo, podría utilizar el URL siguiente: `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Proporcione autenticación en la aplicación. Especifique las credenciales del usuario que ha correlacionado con el rol webUser. De forma predeterminada, este rol de usuario se correlaciona con todos los usuarios autenticados. Especifique `admin1` como ID de usuario y `admin1` como contraseña. Se visualiza una página para visualizar, añadir, actualizar y suprimir empleados.
3. Visualice los empleados. Pulse **Visualizar un empleado**. Especifique `emp1@acme.com` como la dirección de correo electrónico y pulse **Someter**. Se visualiza un mensaje que indica que no se puede encontrar el empleado.
4. Añada un empleado. Pulse **Añadir un empleado**. Especifique `emp1@acme.com` como dirección de correo electrónico, `Joe` como nombre y `Doe` como apellido. Pulse **Someter**. Se visualiza un mensaje que indica que se ha añadido un empleado con la dirección `emp1@acme.com`.
5. Visualice el nuevo empleado. Pulse **Visualizar un empleado**. Especifique `emp1@acme.com` como dirección de correo electrónico con campos vacíos para el nombre y apellido, y pulse **Someter**. Aparece un mensaje que indica que se ha encontrado el empleado, y se visualizan los nombres correctos en los campos de nombre y apellido.
6. Suprima el empleado. Pulse **Suprimir un empleado**. Especifique `emp1@acme.com` y pulse **Someter**. Aparece un mensaje que indica que se ha suprimido el empleado.

Punto de comprobación de la lección:

Ha instalado y ejecutado la aplicación de ejemplo. Debido a que esta guía de aprendizaje utiliza la integración de WebSphere Application Server, no puede ver el escenario cuando un cliente no se puede autenticar en el servidor eXtreme Scale. Si el usuario se autentica satisfactoriamente en WebSphere Application Server, eXtreme Scale también se autenticará satisfactoriamente.

Módulo 3: Configurar seguridad del transporte

Configure la seguridad del transporte para proteger la transferencia de datos entre los clientes y servidores de la configuración.

En el módulo anterior de la guía de aprendizaje, ha habilitado la autenticación de WebSphere eXtreme Scale. Con la autenticación, es necesario que cualquier aplicación que intente conectarse al servidor WebSphere eXtreme Scale proporcione una credencial. Por lo tanto, ningún cliente no autenticado se puede conectar al servidor WebSphere eXtreme Scale. Los clientes deben ser una aplicación autenticada en ejecución en una célula de WebSphere Application Server.

Con la configuración hasta este módulo, la transferencia de datos entre los clientes en el clúster appCluster y los servidores en el clúster xsCluster no está cifrada. Esta configuración podría ser aceptable si los clústeres de WebSphere Application Server están instalados detrás de un cortafuegos. Sin embargo, en algunos escenarios, no se acepta el tráfico no cifrado por varias razones, incluso aunque la topología esté protegida por cortafuegos. Por ejemplo, una política de gobierno podría obligar a tráfico cifrado. WebSphere eXtreme Scale da soporte a TLS/SSL (Transport Layer Security/Secure Sockets Layer) para la comunicación segura entre puntos finales de ObjectGrid, que incluyen servidores de cliente, servidores de contenedor y servidores de catálogo.

En este despliegue de ejemplo, todos los clientes y los servidores de contenedor de eXtreme Scale se ejecutan en el entorno de WebSphere Application Server. Las propiedades de cliente o servidor no son necesarias para configurar los valores de SSL porque la seguridad de transporte de eXtreme Scale la gestionan los valores de transporte CSIV2 (Common Secure Interoperability Protocol Versión 2) de Application Server. Los servidores WebSphere eXtreme Scale utilizan la misma instancia de intermediario de solicitud de objetos (ORB) que los servidores de aplicaciones en los que se ejecutan. Especifique todos los valores de SSL para los servidores de contenedor y cliente en la configuración de WebSphere Application Server mediante estos valores de transporte CSIv2. El servidor de catálogo tiene sus propias vías de acceso de transporte de propiedad que no utilizan IIOP (Internet Inter-ORB Protocol - Protocolo Inter-ORB de Internet) o RMI (Remote Method Invocation - Invocación a método remoto). Debido a estas vías de acceso de transporte de propietario, el servidor de catálogo no puede ser gestionado por los valores de transporte CSIV2 de WebSphere Application Server. Por lo tanto, debe configurar las propiedades SSL en el archivo de propiedades del servidor para el servidor de catálogo.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, ha aprendido a:

- Configurar transporte de entrada y salida CSIv2.
- Añadir propiedades SSL al archivo de propiedades del servidor de catálogo.
- Comprobar el archivo de propiedades del ORB.
- Ejecutar el ejemplo.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Requisitos previos

Este paso de la guía de aprendizaje se basa en los módulos anteriores. Complete los módulos anteriores de esta guía de aprendizaje antes de configurar la seguridad de transporte.

Lección 3.1: Configurar transporte de entrada y salida CSIv2

Para configurar TLS/SSL (Transport Layer Security/Secure Sockets Layer) para el transporte del servidor, establezca el transporte de entrada CSIv2 (Common Secure Interoperability Protocol Version 2) y el transporte de salida CSIv2 en `SSL-Required` para todos los servidores WebSphere Application Server que alojan clientes, servidores de catálogo y servidores de contenedor.

En la topología de ejemplo de la guía de aprendizaje, debe establecer estas propiedades para los servidores de aplicaciones s1, s2, xs1 y xs2. Los pasos siguientes configuran los transportes de entrada y salida para todos los servidores de la configuración.

Establezca los transportes de entrada y salida en la consola administrativa. Asegúrese de que la seguridad administrativa esté habilitada.

- **WebSphere Application Server Versión 6.1**: pulse **Securidad** > **Administración segura** > **Aplicación..** > **Seguridad RMI/IIOP** y cambie el tipo de transporte a **SSL-Required**.
- **WebSphere Application Server Versión 7.0**: pulse **Seguridad** > **Seguridad global** > **Seguridad RMI/IIOP** > **Comunicaciones de entrada CSIv2**. Cambie el tipo de transporte en la capa de transporte CSIv2 a **SSL-Required**. Repita este paso para configurar las comunicaciones de salida CSIv2s.

Puede utilizar valores de seguridad de punto final gestionados de forma centralizada, o bien puede configurar repositorios SSL. Consulte Valores de entrada de transporte de Common Secure Interoperability Versión 2 para obtener más información.

Lección 3.2: Añadir propiedades SSL al archivo de propiedades de servidor de catálogo

El servidor de catálogo tiene sus propias vías de acceso de transporte de propietario que no pueden gestionar los valores de transporte CSIV2 (WebSphere Application Server Common Secure Interoperability Protocol Versión 2). Por lo tanto, debe configurar las propiedades SSL (Secure Sockets Layer) en el archivo de propiedades del servidor para el servidor de catálogo.

Para configurar la seguridad del servidor de catálogo, se requieren pasos adicionales porque el servidor de catálogo tiene sus propias vías de acceso de transporte de propietario. Estas vías de acceso de transporte no las pueden gestionar los valores de transporte CSIV2 de Application Server.

1. Edite las propiedades SSL en el archivo `catServer2.props`. Para configurar la seguridad del servidor de catálogo, elimine el comentario de las propiedades SSL siguientes en el archivo de propiedades del servidor de catálogo. Para esta guía de aprendizaje, las propiedades del servidor de catálogo se encuentran en el archivo `catServer2.props`. Actualice las propiedades keyStore y trustStore para hacer referencia a la ubicación correcta en su entorno.

```
#alias=default
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<NOMBRE_DMGR>/config/
cells/<NOMBRE_CÉLULA>/nodes/<NOMBRE_NODO>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/<INICIO_WAS>/IBM/WebSphere/AppServer/profiles/<NOMBRE_DMGR>/config/
cells/<NOMBRE_CÉLULA>/nodes/<NOMBRE_NODO>/trust.p12
#trustStorePassword=WebAS
#clientAuthentication=false
```

El archivo `catServer2.props` utiliza el almacén de claves y el almacén de confianza predeterminados de nivel de nodo de WebSphere Application Server. Si está desplegando un entorno de despliegue más complejo, debe elegir el almacén de confianza y el almacén de claves correctos. En algunos casos, debe crear un almacén de claves y un almacén de confianza e importar las claves desde almacenes de claves de los otros servidores. Tenga en cuenta que la serie `WebAS` es la contraseña predeterminada del almacén de confianza y del almacén de claves de WebSphere Application Server. Consulte Configuración predeterminada de los certificados autofirmados para obtener más información.

2. En el archivo `catServer2.props`, actualice el valor de la propiedad transportType. Para los pasos anteriores de la guía de aprendizaje, el valor se ha establecido en `TCP/IP`. Cambie el valor a `SSL-Required`.
3. Reinicie el gestor de despliegue para activar los cambios en los valores de seguridad del servidor de catálogo.

Punto de comprobación de la lección:

Ha configurado las propiedades SSL del servidor de catálogo.

Lección 3.3: Ejecutar el ejemplo

Reinicie todos los servidores y ejecute de nuevo la aplicación de ejemplo. Debería poder ejecutar todos los pasos sin problemas.

Consulte “Lección 2.4: Instalar y ejecutar el ejemplo” en la página 97 para obtener más información sobre la ejecución e instalación de la aplicación de ejemplo.

Punto de comprobación de la lección:

Ha ejecutado la aplicación de ejemplo con la seguridad de transporte habilitada.

Módulo 4: Utilizar autorización JAAS (Java Authentication and Authorization Service) en WebSphere Application Server

Ahora que ha configurado la autenticación para clientes, puede configurar la autenticación para otorgar a distintos usuarios diversos permisos. Por ejemplo, es posible que un usuario operator solo pueda visualizar datos, mientras que un usuario administrador puede realizar todas las operaciones.

Tras autenticar un cliente, como en el módulo anterior de esta guía de aprendizaje, puede otorgar privilegios de seguridad mediante los mecanismos de autorización de eXtreme Scale. El módulo anterior de esta guía de aprendizaje ha demostrado cómo habilitar la autenticación para una cuadrícula de datos mediante la integración con WebSphere Application Server. Como resultado, ningún cliente no autenticado se puede conectar a los servidores eXtreme Scale o enviar solicitudes al sistema. No obstante, cada cliente autenticado tiene el mismo permiso o privilegios que el servidor, como por ejemplo, la lectura, la grabación o la supresión de datos que se almacenan en las correlaciones de ObjectGrid. Los clientes también pueden emitir cualquier tipo de consulta.

Esta parte de la guía de aprendizaje muestra cómo utilizar la autenticación de eXtreme Scale para proporcionar a los usuarios diversos privilegios. WebSphere eXtreme Scale utiliza un mecanismo de autorización basado en permisos. Puede asignar distintas categorías de permiso representadas por distintas clases de permiso. Este módulo presenta la clase MapPermission. Para ver una lista de todos los permisos posibles, consulte Programación de autorización de cliente .

En WebSphere eXtreme Scale, la clase `com.ibm.websphere.objectgrid.security.MapPermission` representa permisos a los recursos eXtreme Scale, específicamente los métodos de las interfaces ObjectMap o JavaMap. WebSphere eXtreme Scale define las siguientes series de permiso para acceder a los métodos de ObjectMap y JavaMap:

- **leer**: otorga permiso para leer los datos de la correlación.
- **grabar**: otorga permiso para actualizar los datos de la correlación.
- **insertar**: otorga permiso para insertar los datos de la correlación.
- **eliminar**: otorga permiso para eliminar los datos de la correlación.
- **invalidar**: otorga permiso para invalidar los datos de la correlación.
- **todo**: otorga todos los permisos para leer, grabar, insertar, eliminar e invalidar.

La autorización se produce cuando un cliente de eXtreme Scale utiliza una API de acceso a datos como, por ejemplo, las API ObjectMap, JavaMap o EntityManager. El tiempo de ejecución de eXtreme Scale comprueba los permisos de correlación correspondientes cuando se llama al método. Si no se otorgan los permisos necesarios al cliente, se genera una excepción AccessControlException. Esta guía de aprendizaje muestra cómo utilizar autorización JAAS (Java Authentication and Authorization Service) para otorgar acceso a la correlación a distintos usuarios.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, ha aprendido a:

- Habilitar la autorización para WebSphere eXtreme Scale.
- Habilitar la autorización basada en usuario.
- Configurar la autorización basada en grupos.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Requisitos previos

Debe completar los módulos anteriores de esta guía de aprendizaje antes de configurar la autenticación.

Lección 4.1: Habilitar la autorización de WebSphere eXtreme Scale

Para habilitar la autorización en WebSphere eXtreme Scale, debe habilitar la seguridad en un ObjectGrid específico.

Para habilitar la autorización en el ObjectGrid, debe establecer el atributo **`securityEnabled`** en true para ese ObjectGrid determinado en el archivo XML. Para esta guía de aprendizaje, puede utilizar el archivo `XSDeployment_sec.ear` en el directorio *inicio_samples*`/WASSecurity`, que ya tiene la seguridad establecida en el archivo `objectGrid.xml`, o puede editar el archivo `objectGrid.xml` existente para habilitar la seguridad. Esta lección muestra cómo editar el archivo para habilitar la seguridad.

1. Extraiga los archivos contenidos en el archivo `XSDeployment.ear` y, a continuación, desempaquete el archivo `XSDeploymentWeb.war`.
2. Abra el archivo `objectGrid.xml` y establezca el atributo securityEnabled en `true` en el nivel de ObjectGrid. Consulte un ejemplo de este atributo en el siguiente ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>

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

    <objectGrids>
        <objectGrid name="Grid" securityEnabled="true">
            <backingMap name="Map1" />
        </objectGrid>
    </objectGrids>

</objectGridConfig>
```

Si tiene varios ObjectGrids definidos, debe establecer este atributo en cada cuadrícula de datos.

3. Vuelva a empaquetar los archivos `XSDeploymentWeb.war` y `XSDeployment.ear` para incluir los cambios. Nombre el archivo `XSDeployment_sec.ear` de modo que no sobrescribirá el paquete original.
4. Desinstale la aplicación `XSDeployment` existente e instale el archivo `XSDeployment_sec.ear`. Consulte "Lección 2.4: Instalar y ejecutar el ejemplo" en la página 97 para obtener más información sobre cómo desplegar aplicaciones.

Punto de comprobación de la lección:

Ha habilitado la seguridad en el ObjectGrid, lo que también habilita la autorización en la cuadrícula de datos.

Lección 4.2: Habilitar autorización basada en usuario

En el módulo de autenticación de esta guía de aprendizaje, ha creado dos usuarios: `operator1` y `admin1`. Puede asignar diversos permisos a estos usuarios con autorización JAAS (Java Authentication and Authorization Service).

Definición de la política de autorización JAAS (Java Authentication and Authorization Service) mediante principales de usuario:

Puede asignar permisos a los usuarios que ha creado anteriormente. Asigne al usuario `operator1` permisos de lectura solo en todas las correlaciones. Asigne al usuario `admin1` todos los permisos. Utilice el archivo de política de autorización JAAS para otorgar permisos a los principales.

Edite el archivo de autorización JAAS. El archivo `xsAuth2.policy` se encuentra en el directorio *inicio_samples*`/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";
};
```

En este archivo, la base de código `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` es un URL reservado especialmente para ObjectGrid. Todos los permisos de ObjectGrid otorgados a los principales deben utilizar esta base de código especial. Se asignan los permisos siguientes a este archivo:

- La primera sentencia de otorgamiento otorga permiso de correlación `read` al principal `operator1`. El usuario `operator1` solo tiene permiso de lectura de correlación en la correlación Map1 de la instancia de la cuadrícula ObjectGrid.

- La segunda sentencia de otorgamiento otorga permiso a todas las correlaciones al principal admin1. El usuario `admin1` tiene todos los permisos en la correlación Map1 de la instancia de la cuadrícula ObjectGrid.
- El nombre de principal es `defaultWIMFileBasedRealm/operator1`, pero no `Operator1`. WebSphere Application Server añade automáticamente el nombre de reino al nombre de principal cuando se utilizan repositorios federados como registro de cuentas de usuario. Ajuste este valor, si es necesario.

Definición del archivo de política de autorización JAAS mediante las propiedades de JVM:

Utilice los pasos siguientes para establecer propiedades de JVM para los servidores xs1 y xs2, que están en el clúster xsCluster. Si utiliza una topología distinta de la topología de ejemplo que se utiliza en esta guía de aprendizaje, establezca el archivo en todos sus servidores de contenedor.

1. En la consola administrativa, pulse **Servidores** > **Servidores de aplicaciones** > *nombre_servidor* > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java.**
2. Añada los siguientes argumentos de JVM genéricos:

   `-Djava.security.auth.policy=`*`inicio_samples`*`/security/xsAuth2.policy`
3. Pulse **Aceptar** y guarde los cambios.

Ejecución de la aplicación de ejemplo para probar la autorización:

Puede utilizar la aplicación de ejemplo para probar los valores de autorización. El usuario administrador continúa teniendo todos los permisos en la correlación Map1, incluida la visualización y adición de empleados. El usuario operator solo debe poder visualizar los empleados, ya que a dicho usuario solo se ha asignado permiso de lectura.

1. Reinicie todos los servidores de aplicaciones que ejecutan servidores de contenedor.
2. Abra la aplicación `EmployeeManagementWeb`. En un navegador web, abra `http://<host>:<puerto>/EmployeeManagermentWeb/management.jsp`.
3. Inicie la sesión en la aplicación como administrador. Utilice el nombre de usuario `admin1` y la contraseña `admin1`.
4. Intente visualizar un empleado. Pulse **Visualizar un empleado** y busque la dirección de correo electrónico `authemp1@acme.com`. Se visualiza un mensaje que indica que no se puede encontrar el usuario.
5. Añada un empleado. Pulse **Añadir un empleado**. Añada el correo electrónico `authemp1@acme.com`, el nombre `Joe` y el apellido `Doe`. Pulse **Someter**. Se visualiza un mensaje que indica que se ha añadido el empleado.
6. Inicie la sesión como el usuario operator. Abra una segunda ventana de navegador web y abra `http://<host>:<puerto>/EmployeeManagermentWeb/management.jsp`. Utilice el nombre de usuario `operator1` y la contraseña `operator1`.
7. Intente visualizar un empleado. Pulse **Visualizar un empleado** y busque la dirección de correo electrónico `authemp1@acme.com`. Se visualizará el empleado.
8. Añada un empleado. Pulse **Añadir un empleado**. Añada el correo electrónico `authemp2@acme.com`, el nombre `Joe` y el apellido `Doe`. Pulse **Someter**. Se visualiza el mensaje siguiente:

   ```
   Se produce una excepción al Añadir el empleado. Consulte a continuación para ver mensajes de
   excepción detallados.
   ```

La siguiente excepción se encuentra en la cadena de la excepción:

```
java.security.AccessControlException: Acceso denegado
(com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
```

Este mensaje se visualiza porque el usuario `operator1` no tiene permiso para insertar datos en la correlación Map1.

Si está ejecutando una versión de WebSphere Application Server anterior a la versión 7.0.0.11, es posible que vea un error java.lang.StackOverflowError en el servidor de contenedor. Este error se debe a un problema de IBM Developer Kit. El problema se ha solucionado en el IBM Developer Kit proporcionado con WebSphere Application Server Versión 7.0.0.11 y posterior.

Punto de comprobación de la lección:

En esta lección, ha configurado la autorización asignando permisos a usuarios específicos.

Lección 4.3: Configurar autorización basada en grupo

En la lección anterior, ha asignado autorización basada en usuario individual con principales de usuario en la política de autorización JAAS (Java Authentication and Authorization Service). Sin embargo, cuando tenga cientos o miles de usuarios, utilice la autorización basada en grupo, que autoriza el acceso en función de los grupos, en lugar de hacerlo en función de usuarios individuales.

Desafortunadamente, el objeto Subject que se autentica desde WebSphere Application Server solo contiene un principal de usuario. Este objeto no contiene un principal de grupo. Puede añadir un módulo de inicio de sesión personalizado para llenar el principal de grupo en el objeto Subject.

Para esta guía de aprendizaje, el módulo de inicio de sesión personalizado se denomina com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule. El módulo se encuentra en el archivo `groupLM.jar`. Coloque este archivo JAR en el directorio `WAS-INSTALL/lib/ext`.

WASAddGroupLoginModule recupera la credencial de grupo pública del tema de WebSphere Application Server y crea un principal de grupo, com.ibm.websphere.samples.objectgrid.security.WSGroupPrincipal, para representar el grupo. A continuación, se puede utilizar este principal de grupo para la autorización de grupo. Los grupos se definen en el archivo `xsAuthGroup2.policy`:

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
  "defaultWIMFileBasedRealm/cn=operatorGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
 "defaultWIMFileBasedRealm/cn=adminGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

El nombre de principal es `WSGroupPrincipal`, que representa el grupo.

Adición del módulo de inicio de sesión personalizado:

El módulo de inicio de sesión personalizado se debe añadir a cada una de las siguientes entradas de módulo de inicio de sesión del sistema: si utiliza LTPA

(Lightweight Third Party Authentication), añada la entrada a los módulos de inicio de sesión de RMI_INBOUND. LTPA es el mecanismo de autenticación predeterminado de WebSphere Application Server Versión 7.0. Para una configuración de WebSphere Application Server Network Deployment, solo debe configurar las entradas de configuración del mecanismo de autenticación LTPA.

Utilice los pasos siguientes para configurar el módulo de inicio de sesión com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule proporcionado:

1. En la consola administrativa, pulse **Seguridad** > **Seguridad global** > **Java Authentication and Authorization Service** > **Inicios de sesión del sistema** > ***nombre_módulos_inicio_sesión*** > **Módulos de inicio de sesión JAAS** > **Nuevo**.
2. Especifique el nombre de clase como `com.ibm.websphere.sample.xs.security.lm.WASAddGroupLoginModule`.
3. Opcional: Añada una propiedad `debug` y establezca el valor en `true`.
4. Pulse **Aplicar** para añadir el nuevo módulo a la lista de módulos de inicio de sesión.

Definición del archivo de política de autorización JAAS mediante las propiedades de JVM:

En la consola administrativa, realice los pasos siguientes en los servidores xs1 y xs2 en el xsCluster. Si se utiliza una topología de despliegue distinta, realice los pasos siguientes en los servidores de aplicaciones que alojan los servidores de contenedor.

1. En la consola administrativa, pulse **Servidores** > **Servidores de aplicaciones** > ***nombre_servidor*** > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**
2. Especifique los siguientes argumentos de JVM genéricos o sustituya la entrada `-Djava.security.auth.policy` por el texto siguiente:

   `-Djava.security.auth.policy=`*`inicio_samples`*`/security/xsAuthGroup2.policy`
3. Pulse **Aceptar** y guarde los cambios.

Prueba de la autorización de grupo con la aplicación de ejemplo:

Puede comprobar si el módulo de inicio de sesión ha configurado la autorización de grupo mediante la aplicación de ejemplo.

1. Reinicie los servidores de contenedor. Para esta guía de aprendizaje, los servidores de contenedor son los servidores xs1 y xs2.
2. Inicie la sesión en la aplicación de ejemplo. En un navegador web, abra `http://<host>:<puerto>/EmployeeManagementWeb/management.jsp` e inicie la sesión con el nombre de usuario `admin1` y la contraseña `admin1`.
3. Visualice un empleado. Pulse **Visualizar un empleado** y busque la dirección de correo electrónico `authemp2@acme.com`. Se visualiza un mensaje que indica que no se puede encontrar el usuario.
4. Añada un empleado. Pulse **Añadir un empleado**. Añada el correo electrónico `authemp2@acme.com`, el nombre `Joe` y el apellido `Doe`. Pulse **Someter**. Se visualiza un mensaje que indica que se ha añadido el empleado.
5. Inicie la sesión como el usuario operator. Abra una segunda ventana de navegador web y abra el siguiente URL: `http://<host>:<puerto>/EmployeeManagermentWeb/management.jsp`. Utilice el nombre de usuario `operator1` y la contraseña `operator1`.

6. Intente visualizar un empleado. Pulse **Visualizar un empleado** y busque la dirección de correo electrónico `authemp2@acme.com`. Se visualizará el empleado.
7. Añada un empleado. Pulse **Añadir un empleado**. Añada el correo electrónico `authemp3@acme.com`, el nombre `Joe` y el apellido `Doe`. Pulse **Someter**. Se visualiza el mensaje siguiente:

   ```
   Se produce una excepción al Añadir el empleado. Consulte a continuación para ver mensajes de excepción detallados.
   ```

   La siguiente excepción se encuentra en la cadena de la excepción:

   ```
   java.security.AccessControlException: Acceso denegado
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Este mensaje se visualiza porque el usuario operator no tiene permiso para insertar datos en la correlación Map1.

Punto de comprobación de la lección:

Ha configurado grupos para simplificar la asignación de permisos a los usuarios de la aplicación.

Módulo 5: Utilizar la herramienta `xscmd` para supervisar cuadrículas de datos y correlaciones

Puede utilizar la herramienta **xscmd** para mostrar las cuadrículas de datos primarias y los tamaños de correlación de la cuadrícula de datos `Grid`. La herramienta **xscmd** utiliza el MBean para consultar todos los artefactos de cuadrícula de datos como, por ejemplo, fragmentos primarios, fragmentos de réplica, servidores de contenedor, tamaños de correlación, etc.

En esta guía de aprendizaje, los servidores de contenedor y catálogo se ejecutan en servidores de aplicaciones WebSphere Application Server. El tiempo de ejecución de WebSphere eXtreme Scale registra los beans gestionados (MBean) con el servidor MBean creado por el tiempo de ejecución de WebSphere Application Server. La seguridad que utiliza la herramienta **xscmd** la proporciona la seguridad de MBean de WebSphere Application Server. Por lo tanto, la configuración de seguridad específica de WebSphere eXtreme Scale no es necesaria.

1. Mediante la herramienta de línea de mandatos, abra el directorio *PERFIL_DMGR*`/bin`.
2. Ejecute la herramienta **xscmd**.

   Utilice el mandato **`-c listObjectGridPlacement -sf P`** para listar la colocación de los fragmentos primarios. Linux UNIX

   ```
   xscmd.sh -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Windows

   ```
   xscmd.bat -g Grid -ms mapSet -c showPlacement -sf P
   ```

   Antes de poder visualizar la salida, se le solicitará que inicie la sesión con el ID y la contraseña de WebSphere Application Server.

Punto de comprobación de la lección

Ha utilizado la herramienta **xscmd** en WebSphere Application Server.

Guía de aprendizaje: Integrar la seguridad de WebSphere eXtreme Scale en un entorno mixto con un autenticador externo

Esta guía de aprendizaje muestra cómo proteger los servidores WebSphere eXtreme Scale desplegados parcialmente en un entorno de WebSphere Application Server.

En el despliegue de esta guía de aprendizaje, los servidores de contenedor se despliegan en WebSphere Application Server. El servidor de catálogo se despliega como servidor autónomo y se inicia en un entorno Java Standard Edition (Java SE).

Debido a que el servidor de catálogo no se despliega en WebSphere Application Server, no puede utilizar los plug-ins de autenticación de WebSphere Application Server. Para obtener más información sobre el proceso de configuración de los plug-ins de autenticación de WebSphere Application Server, consulte "Guía de aprendizaje: Integrar la seguridad de WebSphere eXtreme Scale con WebSphere Application Server" en la página 86. En esta guía de aprendizaje, se requiere un autenticador distinto para la autenticación de servidor de catálogo. Configura un autenticador de almacén de claves para autenticar los clientes.

Objetivos del aprendizaje

Los objetivos de aprendizaje de esta guía de aprendizaje son los siguientes:

- Configurar WebSphere eXtreme Scale para utilizar el plug-in KeyStoreLoginAuthenticator
- Configurar la seguridad de transporte de WebSphere eXtreme Scale para utilizar la configuración CSIv2 de WebSphere Application Server y el archivo de propiedades de WebSphere eXtreme Scale
- Utilizar autenticación JAAS (Java Authentication and Authorization Service) en WebSphere Application Server
- Utilizar el programa de utilidad **xscmd** para supervisar las cuadrículas de datos y las correlaciones que ha creado en la guía de aprendizaje.

Tiempo necesario

Esta guía de aprendizaje requiere aproximadamente 4 horas desde el principio hasta el final.

Introducción: Seguridad en un entorno mixto

En esta guía de aprendizaje, integra la seguridad de WebSphere eXtreme Scale en un entorno mixto. Los servidores de contenedor se ejecutan en WebSphere Application Server y el servicio de catálogo se ejecuta en modalidad autónoma. Debido a que el servidor de catálogo se encuentra en modalidad autónoma, debe configurar un autenticador externo.

Importante: Si tanto los servidores de contenedor como los servidores de catálogo se ejecutan en WebSphere Application Server, puede utilizar los plug-ins de autenticación de WebSphere Application Server o un autenticador externo. Para obtener más información sobre cómo utilizar los plug-ins de autenticación de WebSphere Application Server, consulte "Guía de aprendizaje: Integrar la seguridad de WebSphere eXtreme Scale con WebSphere Application Server" en la página 86.

Objetivos del aprendizaje

Los objetivos de aprendizaje de esta guía de aprendizaje son los siguientes:

- Configurar WebSphere eXtreme Scale para utilizar el plug-in KeyStoreLoginAuthenticator
- Configurar la seguridad de transporte de WebSphere eXtreme Scale para utilizar la configuración CSIv2 de WebSphere Application Server y el archivo de propiedades de WebSphere eXtreme Scale
- Utilizar autenticación JAAS (Java Authentication and Authorization Service) en WebSphere Application Server
- Utilizar el programa de utilidad **xscmd** para supervisar las cuadrículas de datos y las correlaciones que ha creado en la guía de aprendizaje.

Tiempo necesario

Esta guía de aprendizaje requiere aproximadamente 4 horas desde el principio hasta el final.

Nivel de conocimientos

Intermedio.

A quién va dirigida

Los desarrolladores y administradores interesados en la integración de seguridad entre WebSphere eXtreme Scale y WebSphere Application Server y en configurar autenticadores externos.

Requisitos del sistema

- WebSphere Application Server Versión 6.1 o Versión 7.0.0.11 o posterior con los siguientes arreglos aplicados:arreglo temporal PM20613 y arreglo temporal PM15818.
- El servidor de catálogo debe estar en ejecución en una instalación autónoma, no en una instalación integrada con WebSphere Application Server.
- Actualice el tiempo de ejecución de Java para aplicar el arreglo siguiente: IZ79819: IBMJDK NO PUEDE LEER SENTENCIA PRINCIPAL CON ESPACIO EN BLANCO DE ARCHIVO DE SEGURIDAD
- El nodo autónomo que ejecuta el servicio de catálogo debe utilizar IBM Software Development Kit Versión 1.6 J9. Este Software Development Kit se incluye en la instalación de WebSphere Application Server. El nodo del servidor de catálogo debe ser una instalación autónoma ya que no se puede ejecutar el mandato **startOgServer** en una instalación de WebSphere eXtreme Scale en WebSphere Application Server.

Esta guía de aprendizaje utiliza cuatro servidores de aplicaciones WebSphere Application Server y un gestor de despliegue para mostrar el ejemplo.

Requisitos previos

Es útil disponer de conocimientos básicos de los elementos siguientes antes de iniciar esta guía de aprendizaje:

- El modelo de programación de WebSphere eXtreme Scale
- Los conceptos básicos de seguridad de WebSphere eXtreme Scale

- Los conceptos básicos de seguridad de WebSphere Application Server

Para obtener información previa sobre la integración de la seguridad de WebSphere eXtreme Scale y WebSphere Application Server, consulte “Integración de la seguridad con WebSphere Application Server” en la página 525.

Módulo 1: Preparar el entorno autónomo y de WebSphere Application Server mixto

Antes de comenzar la guía de aprendizaje, debe crear una topología básica que incluya servidores de contenedor que se ejecuten en WebSphere Application Server. En esta guía de aprendizaje, los servidores de catálogo se ejecutan en modalidad autónoma.

Objetivos del aprendizaje

Con las lecciones de este módulo, aprenderá a:

- Comprender la topología mixta y los archivos que son necesarios para la guía de aprendizaje
- Configurar WebSphere Application Server para ejecutar los servidores de contenedor

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Lección 1.1: Comprender la topología y obtener los archivos de la guía de aprendizaje

Para preparar el entorno para la guía de aprendizaje, debe configurar los servidores de catálogo y contenedor para la topología.

Esta lección le guía por la topología de ejemplo y las aplicaciones que se utilizan en esta guía de aprendizaje. Para empezar a ejecutar la guía de aprendizaje, debe descargar las aplicaciones y colocar los archivos de configuración en las ubicaciones correctas para su entorno. Puede descargar la aplicación de ejemplo desde la wiki de WebSphere eXtreme Scale.

Topología: En esta guía de aprendizaje, crea los clústeres siguientes en la célula de WebSphere Application Server:

- **Clúster appCluster**: aloja la aplicación empresarial de ejemplo EmployeeManagement. Este clúster tiene dos servidores de aplicaciones: s1 y s2.
- **Clúster xsCluster**: aloja los servidores de contenedor eXtreme Scale. Este clúster tiene dos servidores de aplicaciones: xs1 y xs2.

En esta topología de despliegue, los servidores de aplicaciones s1 y s2 son los servidores de cliente que acceden a los datos que se almacenan en la cuadrícula de datos. Los servidores xs1 y xs2 son los servidores de contenedor que alojan la cuadrícula de datos.

Configuración alternativa: puede alojar todos los servidores de aplicaciones en un solo clúster como, por ejemplo, en el clúster appCluster. Con esta configuración, todos los servidores del clúster son tanto clientes como servidores de contenedor. Esta guía de aprendizaje utiliza dos clústeres para distinguir entre los servidores de aplicaciones que alojan los clientes y servidores de contenedor.

En esta guía de aprendizaje, configura un dominio de servicio de catálogo que consta de un servidor remoto que no está en la célula de WebSphere Application Server. Esta configuración no es la predeterminada, lo que hace que los servidores de catálogo se ejecuten en el gestor de despliegue y otros procesos de la célula de WebSphere Application Server. Consulte "Creación de dominios de servicio de catálogo en WebSphere Application Server" en la página 257 para obtener más información sobre la creación de un dominio de servicio de catálogo que conste de servidores remotos.

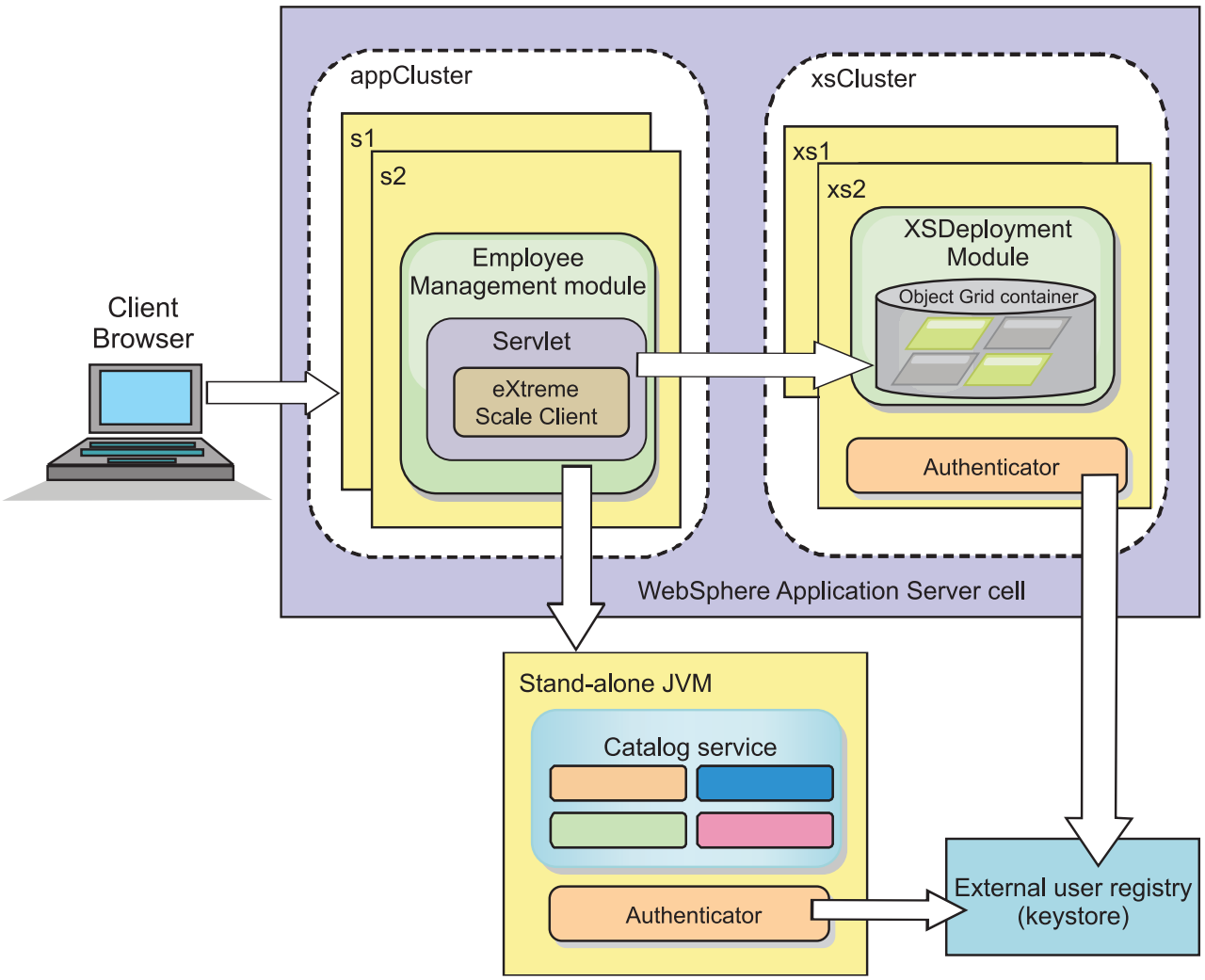


Figura 20. Topología de la guía de aprendizaje

Aplicaciones: En esta guía de aprendizaje, utiliza dos aplicaciones y un archivo de biblioteca compartida:

- **`EmployeeManagement.ear`**: la aplicación `EmployeeManagement.ear` es una aplicación empresarial Java 2 Platform, Enterprise Edition (J2EE) simplificada. Contiene un módulo web para gestionar los perfiles de empleado. El módulo web contiene el archivo `management.jsp` para visualizar, insertar, actualizar y suprimir perfiles de empleado almacenados en los servidores de contenedor.
- **`XSDeployment.ear`**: esta aplicación contiene un módulo de aplicación empresarial sin artefactos de la aplicación. Los objetos de memoria caché se empaquetan en el archivo `EmployeeData.jar`. El archivo `EmployeeData.jar` se despliega como una biblioteca compartida para el archivo `XSDeployment.ear`, de forma que el

archivo XSDeployment.ear pueda acceder a las clases. La finalidad de esta aplicación es empaquetar el archivo de configuración y el archivo de propiedades de eXtreme Scale. Cuando se inicia esta aplicación empresarial, la ejecución de eXtreme Scale detecta autómáticamente los archivos de configuración de eXtreme Scale, de forma que se crean los servidores de contenedor. Estos archivos de configuración incluyen los archivos objectGrid.xml y objectGridDeployment.xml.

- **EmployeeData.jar**: este archivo jar contiene una sola clase: la clase com.ibm.websphere.sample.xs.data.EmployeeData. Esta clase representa los datos de los empleados almacenados en la cuadrícula. Este archivo de archivado Java (JAR) se despliega con los archivos EmployeeManagement.ear y XSDeployment.ear como una biblioteca compartida.

Obtener los archivos de la guía de aprendizaje:

1. Descargue los archivos WASSecurity.zip y security_extauth.zip de la wiki de WebSphere eXtreme Scale.
2. Extraiga el archivo WASSecurity.zip en un directorio para visualizar los artefactos binarios y de origen, por ejemplo, un directorio wxs_samples/. Se hace referencia a este directorio como *inicio_samples* para el resto de la guía de aprendizaje. Consulte el archivo README.txt del paquete para ver una descripción del contenido y de cómo cargar el origen en el espacio de trabajo de Eclipse. Los siguientes archivos de configuración de ObjectGrid están en el directorio META-INF:
   - objectGrid.xml
   - objectGridDeployment.xml
3. Cree un directorio para almacenar los archivos de propiedad utilizados para proteger este entorno. Por ejemplo, podría crear el directorio /opt/wxs/security.
4. Extraiga el archivo security_extauth.zip en *inicio_samples*. El archivo security_extauth.zip contiene los siguientes archivos de configuración de seguridad que se utilizan en esta guía de aprendizaje. A continuación se muestran estos archivos de configuración:
   - catServer3.props
   - server3.props
   - client3.props
   - security3.xml
   - xsAuth3.props
   - xsjaas3.config
   - sampleKS3.jks

Acerca de los archivos de configuración:

Los archivos objectGrid.xml y objectGridDeployment.xml crean las cuadrículas de datos y correlaciones que almacenan los datos de aplicación.

Estos archivos de configuración se deben denominar objectGrid.xml y objectGridDeployment.xml. Cuando se inicia el servidor de aplicaciones, eXtreme Scale detecta estos archivos en el directorio META-INF de los módulos EJB y web. Si se encuentran estos archivos, se asume que la máquina virtual Java (JVM) actúa como un servidor de contenedor para las cuadrículas de datos definidas en los archivos de configuración.

Archivo objectGrid.xml

El archivo objectGrid.xml ha definido un ObjectGrid denominado Grid. La cuadrícula de datos Grid tiene una cuadrícula, la correlación Map1, que almacena el perfil de empleado para la aplicación.

```
<?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>
```

Archivo objectGridDeployment.xml

El archivo objectGridDeployment.xml especifica cómo desplegar la cuadrícula de datos Grid. Cuando se despliega la cuadrícula, tiene cinco particiones y una réplica síncrona.

```
<?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>
```

Punto de comprobación de la lección:

En esta lección, ha aprendido sobre la topología de la guía de aprendizaje y ha añadido archivos de configuración y aplicaciones de ejemplo al entorno.

Lección 1.2: Configurar el entorno de WebSphere Application Server

Para preparar el entorno para la guía de aprendizaje, debe configurar la seguridad de WebSphere Application Server. Habilite la administración y la seguridad de la aplicación mediante repositorios federados basados en archivo interno como un registro de cuentas de usuario. A continuación, puede crear clústeres de servidores para alojar la aplicación de cliente y los servidores de contenedor. Debe crear e iniciar también los servidores de catálogo.

Los pasos siguientes se han escrito utilizando WebSphere Application Server Versión 7.0. Sin embargo, también puede aplicar los conceptos en versiones anteriores de WebSphere Application Server.

Configurar la seguridad de WebSphere Application Server:

Cree y aumente perfiles para el gestor de despliegue y nodos con WebSphere eXtreme Scale. Si desea más información, consulte “Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server” en la página 163.

Configure la seguridad de WebSphere Application Server.

1. En la consola administrativa de WebSphere Application Server, pulse **Seguridad** > **Seguridad global**.
2. Seleccione **Repositorios federados** como el **Repositorio de cuentas de usuario**. Pulse **Establecer como actual**.
3. Pulse **Configurar..** para ir al panel **Repositorios federados**.
4. Especifique el **Nombre de usuario administrativo primario**, por ejemplo, `admin`. Pulse **Aplicar**.
5. Cuando se le solicite, especifique el usuario administrativo y la contraseña y pulse **Aceptar**. Guarde los cambios.
6. En la página **Seguridad global**, compruebe que el valor **Repositorios federados** esté establecido en el registro de cuentas de usuario actual.
7. Seleccione los elementos siguientes: **Habilitar seguridad administrativa**, **Habilitar seguridad de la aplicación** y **Utilizar seguridad Java 2 para restringir el acceso a la aplicación a recursos locales**. Pulse **Aplicar** y guarde los cambios.
8. Reinicie el gestor de despliegue y los servidores de aplicaciones en ejecución.

La seguridad administrativa de WebSphere Application Server se habilita mediante los repositorios federados basados en archivo internos como registro de cuentas de usuario.

Crear clústeres de servidores:

Cree dos clústeres de servidores en la configuración de WebSphere Application Server: el clúster `appCluster` para alojar la aplicación de ejemplo para la guía de aprendizaje y el clúster `xsCluster` para alojar la cuadrícula de datos.

1. En la consola administrativa de WebSphere Application Server, abra el panel de clústeres. Pulse **Servidores** > **Clústeres** > **Clústeres de servidores de aplicaciones WebSphere** > **Nuevo**.
2. Especifique `appCluster` como nombre de clúster, deje seleccionada la opción **Preferir local** y pulse **Siguiente**.
3. Cree servidores en el clúster. Cree un servidor denominado `s1`, manteniendo las opciones predeterminadas. Añada un miembro de clúster adicional denominado `s2`.
4. Complete los demás pasos del asistente para crear el clúster. Guarde los cambios.
5. Repita estos pasos para crear el clúster `xsCluster`. Este clúster tiene dos servidores, denominados `xs1` y `xs2`.

Cree un dominio de servicio de catálogo:

Después de configurar el clúster de servidores y la seguridad, debe definir dónde se inician los servidores de catálogo.

Defina un dominio de servicio de catálogo en WebSphere eXtreme Scale

1. En la consola administrativa de WebSphere Application Server, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo**.
2. Cree el dominio de servicio de catálogo. Pulse **Nueva**. Cree el dominio de servicio de catálogo con el nombre `catalogService1`, y habilite el dominio de servicio de catálogo como valor predeterminado.

3. Añada servidores remotos al dominio de servicio de catálogo. Seleccione **Servidor remoto**. Proporcione el nombre de host donde se está ejecutando el servidor de catálogo. Utilice el valor de puerto de escucha de `16809` para este ejemplo.
4. Pulse **Aceptar** y guarde los cambios.

Punto de comprobación de la lección:

Ha habilitado la seguridad en WebSphere Application Server y ha creado la topología de servidor para WebSphere eXtreme Scale.

Módulo 2: Configurar authentication de WebSphere eXtreme Scale en un entorno mixto

Configurando la autenticación, puede determinar de forma fiable la identidad del solicitante. WebSphere eXtreme Scale da soporte a la autenticación de cliente a servidor y servidor a servidor.

Flujo de autenticación

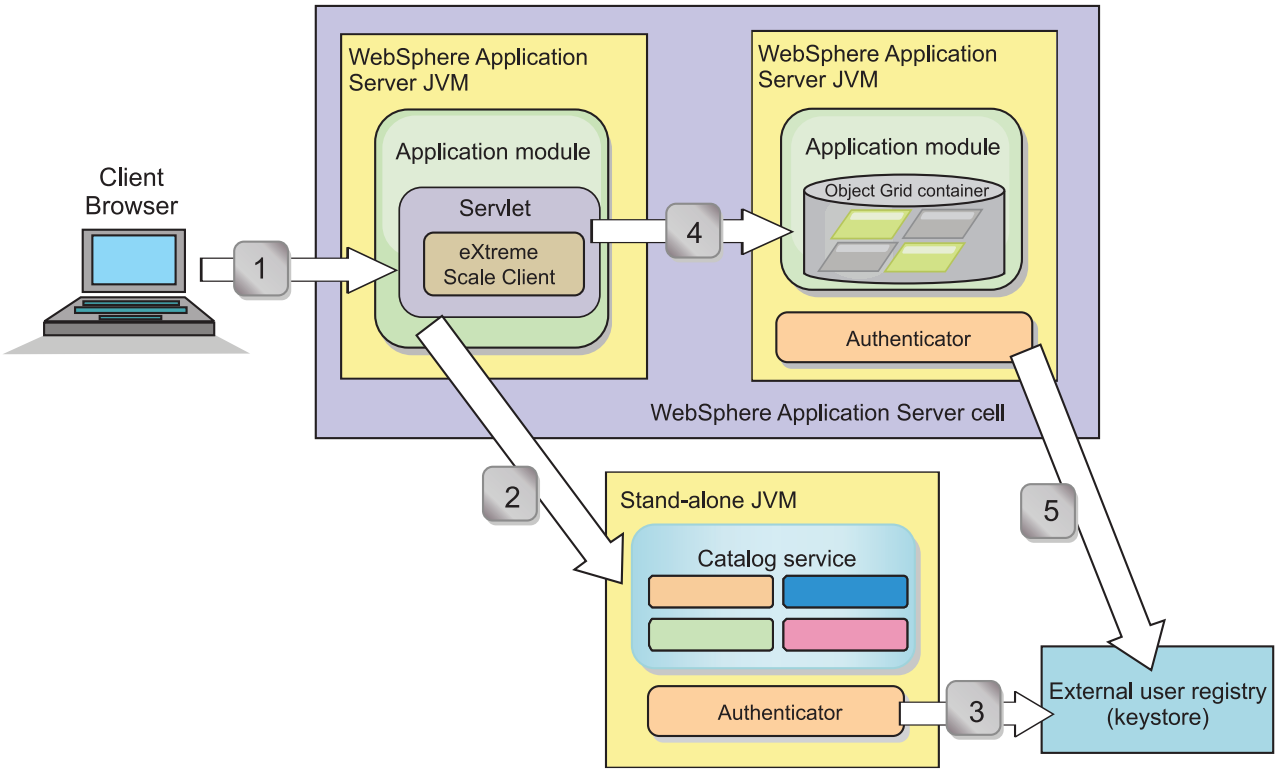


Figura 21. Flujo de autenticación

El diagrama anterior muestra dos servidores de aplicaciones. El primer servidor de aplicaciones aloja la aplicación web, que también es un cliente de WebSphere eXtreme Scale. El segundo servidor de aplicaciones aloja un servidor de contenedor. El servidor de catálogo se ejecuta en una máquina virtual Java (JVM) en lugar de hacerlo en WebSphere Application Server.

Las flechas marcadas con números en el diagrama indican el flujo de autenticación:

1. Un usuario de aplicación empresarial accede al navegador web e inicia la sesión en el primer servidor de aplicaciones con un nombre de usuario y una

contraseña. El primer servidor de aplicaciones envía el nombre de usuario y la contraseña del cliente a la infraestructura de seguridad para su autenticación en el registro de usuarios. Este registro de usuarios es un almacén de claves. Como resultado, la información de seguridad se almacena en la hebra de WebSphere Application Server.

2. El archivo JSP (JavaServer Pages) actúa como un cliente de WebSphere eXtreme Scale para recuperar la información de seguridad del archivo de propiedades de cliente. La aplicación JSP que actúa como el cliente de WebSphere eXtreme Scale envía la credencial de seguridad de cliente de WebSphere eXtreme Scale junto con la solicitud al servidor de catálogo. El envío de la credencial de seguridad con la solicitud se considera un modelo *runAs*. En un modelo runAs, el cliente de navegador web se ejecuta como un cliente de WebSphere eXtreme Scale para acceder a los datos almacenados en el servidor de contenedor. El cliente utiliza una credencial de cliente de toda la máquina virtual Java (JVM) para conectarse a los servidores WebSphere eXtreme Scale. La utilización del modelo runAs es como conectarse a una base de datos con un ID de usuario y una contraseña de nivel de origen de datos.
3. El servidor de catálogo recibe la credencial de cliente de WebSphere eXtreme Scale, que incluye las señales de seguridad de WebSphere Application Server. A continuación, el servidor de catálogo llama al plug-in de autenticador para autenticar la credencial del cliente. El autenticador se conecta al registro de usuarios externo y envía la credencial del cliente al registro de usuarios para su autenticación.
4. El cliente envía el ID y la contraseña de usuario al servidor de contenedor alojado en el servidor de aplicaciones.
5. El servicio de contenedor, alojado en el servidor de aplicaciones, recibe la credencial del cliente de WebSphere eXtreme Scale, que es el par de ID de usuario y contraseña. A continuación, el servidor de contenedor llama al plug-in de autenticador para autenticar la credencial del cliente. El autenticador se conecta al registro de usuario del almacén de claves y envía la credencial del cliente al registro de usuarios para su autenticación.

Objetivos del aprendizaje

Con las lecciones de este módulo, aprenderá a:

- Configurar la seguridad de cliente de WebSphere eXtreme Scale.
- Configurar la seguridad del servidor de catálogo de WebSphere eXtreme Scale.
- Configurar la seguridad del servidor de contenedor de WebSphere eXtreme Scale.
- Instalar y ejecutar la aplicación de ejemplo.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Lección 2.1: Configurar la seguridad de cliente de WebSphere eXtreme Scale

Configura las propiedades de cliente con un archivo de propiedades. El archivo de propiedades de cliente indica la clase de implementación CredentialGenerator que se utilizará.

Contenido del archivo de propiedades de cliente:

La guía de aprendizaje utiliza señales de seguridad de WebSphere Application Server para la credencial de cliente. El directorio *inicio_samples*/security_extauth contiene el archivo client3.props.

El archivo client3.props incluye los valores siguientes:

securityEnabled
Habilita la seguridad de cliente de WebSphere eXtreme Scale. El valor se establece en true para indicar que el cliente debe enviar la información de seguridad disponible al servidor.

credentialAuthentication
Especifica el soporte de autenticación de la credencial del cliente. El valor se establece en Supported para indicar que el cliente da soporte a la autenticación de credenciales.

credentialGeneratorClass
Especifica el nombre de la clase que implementa la interfaz com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. El valor se establece en la clase com.ibm.websphere.objectgrid.security.plugins.builtins. UserPasswordCredentialGenerator de forma que el cliente recupera la información de seguridad de la clase UserPasswordCredentialGenerator.

credentialGeneratorProps
Especifica el nombre de usuario y la contraseña: manager manager1. El nombre de usuario es manager y la contraseña es manager1. También puede utilizar el mandato **FilePasswordEncoder.bat|sh** para codificar esta propiedad utilizando un algoritmo exclusivo o (xor).

Definición del archivo de propiedades de cliente mediante las propiedades de la JVM (Java virtual machine):

En la consola administrativa, complete los pasos siguiente para los servidores s1 y s2 del clúster appCluster. Si está utilizando una topología distinta, complete los pasos siguientes para todos los servidores de aplicaciones en los que se despliega la aplicación EmployeeManagement.

1. **Servidores** > **Servidores de aplicaciones WebSphere** > *nombre_servidor* > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**.
2. Cree la siguiente propiedad de JVM genérica para establecer la ubicación del archivo de propiedades de cliente:
   ```
   -Dobjectgrid.client.props=inicio_samples/security_extauth/client3.props
   ```
3. Pulse **Aceptar** y guarde los cambios.

Punto de comprobación de la lección:

Ha editado el archivo de propiedades de cliente y ha configurado los servidores en el clúster appCluster para utilizar el archivo de propiedades de cliente. Este archivo de propiedades indica la clase de implementación CredentialGenerator que se utilizará.

Lección 2.2: Configurar seguridad del servidor de catálogo

Un servidor de catálogo contiene dos niveles distintos de información de seguridad. El primer nivel contiene las propiedades de seguridad que son comunes a todos los servidores WebSphere eXtreme Scale, incluido el servicio de catálogo y los servidores de contenedor. El segundo nivel contiene las propiedades de seguridad que son específicas del servidor de catálogo.

Las propiedades de seguridad comunes a los servidores de catálogo y los servidores de contenedor se configuran en el archivo de descriptor XML de seguridad. Un archivo de propiedades comunes es la configuración de autenticador, que representa el registro de usuarios y el mecanismo de autenticación. Consulte Archivo XML de descriptor de seguridad para obtener más información sobre las propiedades de seguridad.

Para configurar un archivo de descriptor XML de seguridad en un entorno Java SE, utilice la opción **-clusterSecurityFile** al ejecutar el mandato **startOgServer**. Especifique un valor con formato de archivo como, por ejemplo, *inicio_samples*/security_extauth/security3.xml.

Archivo security3.xml:

En esta guía de aprendizaje, el archivo security3.xml está en el directorio *inicio_samples*/security_extauth. A continuación se muestra el contenido del archivo security3.xml con los comentarios eliminados:

```
<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>
```

Se definen las propiedades siguientes en el archivo security3.xml:

securityEnabled
: La propiedad securityEnabled se establece en true, lo que indica al servidor de catálogo que la seguridad global de WebSphere eXtreme Scale está habilitada.

authenticator
: El autenticador se configura como la clase com.ibm.websphere.objectgrid.security.plugins.builtins. KeyStoreLoginAuthenticator. Con esta implementación incorporada del plug-in Authenticator, se proporciona el ID de usuario y la contraseña para verificar que está configurada en el archivo del almacén de claves. La clase KeyStoreLoginAuthenticator utiliza el alias de módulo de inicio de sesión KeyStoreLogin, así que se requiere una configuración de inicio de sesión JAAS (Java Authentication and Authorization Service).

Archivo catServer3.props:

El archivo de propiedades del servidor almacena las propiedades específicas del servidor, que incluyen las propiedades de seguridad específicas del servidor. Consulte Archivo de propiedades de servidor para obtener más información. Puede utilizar la opción **-serverProps** para especificar la propiedad del servidor de catálogo al ejecutar el mandato **startOgServer**. Para esta guía de aprendizaje, se incluye un archivo catServer3.props en el directorio c. A continuación se muestra el contenido del archivo catServer3.props con los comentarios eliminados:

```
securityEnabled=true
credentialAuthentication=Required
transportType=TCP/IP
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
La propiedad securityEnabled se establece en `true` para indicar que este servidor de catálogo es un servidor seguro.

credentialAuthentication
La propiedad credentialAuthentication se establece en `Required`, de forma que cualquier cliente que se conecte al servidor necesitará proporcionar una credencial. En el archivo de propiedades del cliente, el valor credentialAuthentication se establece en `Supported`, de forma que el servidor recibe las credenciales que envía el cliente.

secureTokenManagerType
La propiedad secureTokenManagerType se establece en `none` para indicar que el secreto de autenticación no está cifrado al unirse a los servidores existentes.

authenticationSecret
La propiedad authenticationSecret se establece en `ObjectGridDefaultSecret`. Esta serie secreta se utiliza para la unión al clúster de servidores eXtreme Scale. Cuando un servidor se une a la cuadrícula de datos, se solicita que proporcione la serie secreta. Si la serie secreta del servidor que se une coincide con la serie en el servidor de catálogo, se aceptará el servidor que se desea unir. Si la serie no coincide, se rechaza la solicitud de unión.

transportType
La propiedad transportType se establece inicialmente en `TCP/IP`. Más adelante en la guía de aprendizaje, la seguridad de transporte se habilitará.

Archivo `xsjaas3.config`:

Puesto que la implementación KeyStoreLoginAuthenticator utiliza un módulo de inicio de sesión, debe configurar el modelo de inicio de sesión con un archivo de configuración de inicio de sesión de autenticación JAAS. A continuación se muestra el contenido del archivo `xsjaas3.config`:

```
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
     keyStoreFile="inicio_samples/security_extauth/sampleKS3.jks" debug = true;
};
```

Si ha utilizado una ubicación para *inicio_samples* distinta a `/wxs_samples/`, debe actualizar la ubicación de keyStoreFile. Esta configuración de inicio de sesión indica que el módulo com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule se utiliza como el módulo de inicio de sesión. El archivo del almacén de claves se establece en el archivo `sampleKS3.jks`.

El archivo de almacén de claves de ejemplo `sampleKS3.jks` almacena dos ID de usuario y las contraseñas: manager/manager1 y cashier/cashier1.

Puede utilizar los siguientes mandatos **`keytool`** para crear este almacén de claves:

- `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`

Iniciar el servidor de catálogo con la seguridad habilitada:

Para iniciar el servidor de catálogo, emita el mandato **startOgServer** con los parámetros **-clusterFile** y **-serverProps** para proporcionar las propiedades de seguridad.

Utilice una instalación autónoma de WebSphere eXtreme Scale para ejecutar el servidor de catálogo. Cuando utilice la imagen de instalación autónoma, debe utilizar el SDK de IBM. Puede utilizar el SDK que se incluye con WebSphere Application Server estableciendo la variable *JAVA_HOME* para que apunte al SDK de IBM. Por ejemplo, `set JAVA_HOME=raíz_was/IBM/WebSphere/AppServer/java/`

1. Vaya al directorio `bin`.

   ```
   cd inicio_wxs/bin
   ```

2. Ejecute el mandato **startOgServer**.

   Linux UNIX

   ```
   ./startOgServer.sh cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
   cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile inicio_samples/security_extauth/security3.xml
   -serverProps inicio_samples/security_extauth/catServer3.props -jvmArgs
   -Djava.security.auth.login.config="inicio_samples/security_extauth/xsjaas3.config"
   ```

   Windows

   ```
   startOgServer.bat cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndPoints
   cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile inicio_samples/security_extauth/security3.xml
   -serverProps inicio_samples/security_extauth/catServer3.props -jvmArgs
   -Djava.security.auth.login.config="inicio_samples/security_extauth/xsjaas3.config"
   ```

Después de ejecutar el mandato **startOgServer**, se inicia un servidor seguro con el puerto de escucha 16809, el puerto de cliente 16601, el puerto de igual 16602 y el puerto JMX 16099. Si existe un conflicto de puertos, cambie el número de puerto a un número de puerto no utilizado.

Detener un servidor de catálogo con la seguridad habilitada:

Puede utilizar el mandato **stopOgServer** para detener el servidor de catálogo.

1. Vaya al directorio `bin`.

   ```
   cd inicio_wxs/bin
   ```

2. Ejecute el mandato **stopOgServer**. Linux UNIX

   ```
   stopOgServer.sh cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   inicio_samples/security_extauth/client3.props
   ```

   Windows

   ```
   stopOgServer.bat cs1 -catalogServiceEndPoints localhost:16809 -clientSecurityFile
   inicio_samples/security_extauth/client3.props
   ```

Punto de comprobación de la lección:

Ha configurado la seguridad del servidor de catálogo asociando los archivos `security3.xml`, `catServer3.props` y `xsjaas3.config` al servicio de catálogo.

Lección 2.3: Configurar la seguridad del servidor de contenedor

Cuando un servidor de contenedor se conecta a un servicio de catálogo, el servidor de contenedor obtiene todas las configuraciones de seguridad configuradas en el archivo XML de seguridad de ObjectGrid. El archivo XML de seguridad de ObjectGrid define la configuración de autenticador, el valor de tiempo de espera de inicio de sesión y otra información de configuración. Un servidor de contenedor también tiene sus propias propiedades de seguridad específicas del servidor en el archivo de propiedades del servidor.

Configure el archivo de propiedades del servidor con la propiedad de máquina virtual Java (JVM) -Dobjectgrid.server.props. El nombre de archivo especificado para esta propiedad es una vía de acceso de archivo absoluta, por ejemplo, *inicio_samples*/security_extauth/server3.props.

En esta guía de aprendizaje, los servidores de contenedor se alojan en los servidores xs1 y xs2 del clúster xsCluster.

Archivo server3.props:

El archivo server3.props se encuentra en el directorio *inicio_samples*/security_extauth/. A continuación se muestra el contenido del archivo server3.props:

```
securityEnabled=true
credentialAuthentication=Required
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

securityEnabled
La propiedad securityEnabled se establece en true para indicar que el servidor de contenedor es un servidor seguro.

credentialAuthentication
La propiedad credentialAuthentication se establece en Required, de forma que cualquier cliente que se conecte al servidor necesitará proporcionar una credencial. En el archivo de propiedades de cliente, la propiedad credentialAuthentication se establece en Supported, de manera que el servidor recibe la credencial enviada por el cliente.

secureTokenManagerType
La propiedad secureTokenManagerType se establece en none para indicar que el secreto de autenticación no está cifrado al unirse a los servidores existentes.

authenticationSecret
La propiedad authenticationSecret se establece en ObjectGridDefaultSecret. Esta serie secreta se utiliza para la unión al clúster de servidores eXtreme Scale. Cuando un servidor se une a la cuadrícula de datos, se solicita que proporcione la serie secreta. Si la serie secreta del servidor que se une coincide con la serie en el servidor de catálogo, se aceptará el servidor que se desea unir. Si la serie no coincide, se rechaza la solicitud de unión.

Definición del archivo de propiedades de servidor con propiedades de JVM:

Establezca el archivo de propiedades del servidor en los servidores xs1 y xs2. Si no está utilizando la topología para esta guía de aprendizaje, establezca el archivo de propiedades del servidor en todos los servidores de aplicaciones que está utilizando para alojar servidores de contenedor.

1. Abra la página de la máquina virtual Java para el servidor. **Servidores** > **Servidores de aplicaciones WebSphere** > *nombre_servidor* > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**.
2. Añada el argumento de JVM genérico:
   ```
   -Dobjectgrid.server.props=inicio_samples/security_extauth/server3.props
   ```
3. Pulse **Aceptar** y guarde los cambios.

Adición del módulo de inicio de sesión personalizado:

El servidor de contenedor utiliza la misma implementación de KeyStoreAuthenticator que el servidor de catálogo. La implementación de KeyStoreAuthenticator utiliza un álias de módulo de inicio de sesión **KeyStoreLogin**, de manera que puede añadir un módulo de inicio de sesión personalizado a las entradas del modelo de inicio de sesión de la aplicación.

1. En la consola administrativa de WebSphere Application Server, pulse **Seguridad** > **Seguridad global** > **Java Authentication and Authorization Service**.
2. Pulse **Inicios de sesión de la aplicación**.
3. Pulse **Nuevo** y añada un alias `KeyStoreLogin`. Pulse **Aplicar**.
4. En **Módulos de inicio de sesión de JAAS**, pulse **Nuevo**.
5. Entre `com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule` como nombre de clase de módulo y elija **SUFFICIENT** como estrategia de autenticación. Pulse **Aplicar**.
6. Añada la propiedad personalizada `keyStoreFile` con el valor *inicio_samples*`/security_extauth/sampleKS.jks`.
7. Opcional: Añada la propiedad personalizada `debug` con el valor `true`.
8. Guarde la configuración.

Punto de comprobación de la lección:

Ahora la autenticación del servidor WebSphere eXtreme Scale está protegida. Si se configura esta seguridad, será necesario que todas las aplicaciones que intenten conectarse a los servidores WebSphere eXtreme Scale proporcionen una credencial. En esta guía de aprendizaje, KeyStoreLoginAuthenticator es el autenticador. Como resultado, se requiere que el cliente proporcione un nombre de usuario y una contraseña.

Lección 2.4: Instalar y ejecutar el ejemplo

Una vez que se ha configurado la autenticación, puede instalar y ejecutar la aplicación de ejemplo.

Creación de una biblioteca compartida para el archivo `EmployeeData.jar`:

1. En la consola administrativa de WebSphere Application Server, abra la página **Bibliotecas compartidas**. Pulse **Entorno** > **Bibliotecas compartidas**.
2. Elija el ámbito **célula**.
3. Cree la biblioteca compartida. Pulse **Nueva**. Especifique `EmployeeManagementLIB` como **Nombre**. Especifique la vía de acceso del archivo `EmployeeData.jar` en la classpath, por ejemplo, *inicio_samples*`/WASSecurity/EmployeeData.jar`.
4. Pulse **Aplicar**.

Instalación del ejemplo:

1. Instale el archivo `EmployeeManagement_extauth.ear` en el directorio *inicio_samples*`/security_extauth`.

   Importante: El archivo `EmployeeManagement_extauth.ear` es distinto del archivo *inicio_samples*`/WASSecurity/EmployeeManagement.ear`. La manera en la que se recupera la sesión de ObjectGrid se ha actualizado para utilizar la credencial almacenada en la memoria caché en el archivo de propiedades de cliente en la aplicación `EmployeeManagement_extauth.ear`. Consulte los comentarios en la clase com.ibm.websphere.sample.xs.DataAccessor del proyecto *inicio_samples*`/WASSecurity/EmployeeManagementWeb` para ver el código que se ha actualizado para este cambio.

a. Para empezar la instalación, pulse **Aplicaciones** > **Nueva aplicación** > **Nueva aplicación empresarial**. Elija la vía de acceso detallada para la instalación de la aplicación.
b. En el paso **Correlacionar módulos con servidores**, especifique el clúster `appCluster` para instalar el módulo EmployeeManagementWeb.
c. En el paso **Bibliotecas compartidas de correlación**, seleccione el módulo `EmployeeManagementWeb`.
d. Pulse **Bibliotecas compartidas de referencia**. Seleccione la biblioteca `EmployeeManagementLIB`.
e. Correlacione el rol `webUser` con **Todos los autenticados en el reino de la aplicación**.
f. Pulse **Aceptar**.

Los clientes se ejecutan en los servidores `s1` y `s2` de este clúster.

2. Instale el archivo `XSDeployment.ear` de ejemplo que se encuentra en el directorio *inicio_samples*`/WASSecurity`.
   a. Para empezar la instalación, pulse **Aplicaciones** > **Nueva aplicación** > **Nueva aplicación empresarial**. Elija la vía de acceso detallada para la instalación de la aplicación.
   b. En el paso **Correlacionar módulos con servidores**, especifique el clúster `xsCluster` para instalar el módulo web `XSDeploymentWeb`.
   c. En el paso **Bibliotecas compartidas de correlación**, seleccione el módulo `XSDeploymentWeb`.
   d. Pulse **Bibliotecas compartidas de referencia**. Seleccione la biblioteca `EmployeeManagementLIB`.
   e. Pulse **Aceptar**.

   Los servidores `xs1` y `xs2` de este clúster alojan los servidores de contenedor.
3. Verifique que el servidor de catálogo se haya iniciado. Para obtener más información sobre cómo iniciar un servidor de catálogo para esta guía de aprendizaje, consulte "Iniciar el servidor de catálogo con la seguridad habilitada" en la página 120.
4. Reinicie el clúster xsCluster. Cuando se inicia el clúster xsCluster, se inicia la aplicación `XSDeployment` y el servidor de contenedor se inicia en los servidores xs1 y xs2, respectivamente. Si mira el archivo `SystemOut.log` de los servidores xs1 y xs2, se visualiza el mensaje siguiente que indica que el archivo de propiedades del servidor se ha cargado:

   ```
   CWOBJ0913I: Los archivos de propiedades del servidor se han cargado:
   inicio_samples/security_extauth/server3.props.
   ```
5. Reinicie el clúster appClusters. Cuando se inicia el clúster appCluster, se inicia también la aplicación EmployeeManagement. Si mira el archivo `SystemOut.log` de los servidores s1 y s2, puede ver el mensaje siguiente que indica que el archivo de propiedades de cliente se ha cargado.

   ```
   CWOBJ0924I: El archivo de propiedades de cliente {0} se ha cargado.
   ```

   Si utiliza WebSphere eXtreme Scale Versión 7.0, se visualiza el mensaje CWOBJ9000I, solo en inglés, para indicar que el archivo de propiedades de cliente se ha cargado. Si no ve el mensaje esperado, compruebe que ha configurado la propiedad -Dobjectgrid.server.props o -Dobjectgrid.client.props en el argumento de JVM. Si sí tiene las propiedades configuradas, asegúrese de que el guión (-) sea un carácter UTF.

Ejecución de la aplicación de ejemplo:

1. Ejecute el archivo `management.jsp`. En un navegador web, acceda a `http://<su_nombre_servidor>:<puerto>/EmployeeManagementWeb/management.jsp`. Por ejemplo, podría utilizar el URL siguiente: `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Proporcione autenticación en la aplicación. Especifique las credenciales del usuario que ha correlacionado con el rol webUser. De forma predeterminada, este rol de usuario se correlaciona con todos los usuarios autenticados. Especifique cualquier nombre de usuario y contraseña válidos, por ejemplo, el nombre de usuario administrativo y la contraseña. Se visualiza una página para visualizar, añadir, actualizar y suprimir empleados.
3. Visualice los empleados. Pulse **Visualizar un empleado**. Especifique `emp1@acme.com` como la dirección de correo electrónico y pulse **Someter**. Se visualiza un mensaje que indica que no se puede encontrar el empleado.
4. Añada un empleado. Pulse **Añadir un empleado**. Especifique `emp1@acme.com` como dirección de correo electrónico, `Joe` como nombre y `Doe` como apellido. Pulse **Someter**. Se visualiza un mensaje que indica que se ha añadido un empleado con la dirección `emp1@acme.com`.
5. Visualice el nuevo empleado. Pulse **Visualizar un empleado**. Especifique `emp1@acme.com` como dirección de correo electrónico con campos vacíos para el nombre y apellido, y pulse **Someter**. Aparece un mensaje que indica que se ha encontrado el empleado, y se visualizan los nombres correctos en los campos de nombre y apellido.
6. Suprima el empleado. Pulse **Suprimir un empleado**. Especifique `emp1@acme.com` y pulse **Someter**. Aparece un mensaje que indica que se ha suprimido el empleado.

Debido a que el tipo de transporte de servidor de catálogo se establece en `TCP/IP`, compruebe que el valor de transporte de salida de los servidores s1 y s2 no esté establecido en `SSL-Required`. De lo contrario, se produce una excepción. Si mira el archivo de salida del sistema del servidor de catálogo, el archivo `logs/cs1/SystemOut.log`, la siguiente salida de depuración indica la autenticación de almacén de claves:

```
SystemOut     O [KeyStoreLoginModule] initialize: Se ha cargado satisfactoriamente
el almacén de claves
SystemOut     O [KeyStoreLoginModule] login: entrada
SystemOut     O [KeyStoreLoginModule] login: nombre de usuario especificado
del usuario: manager
SystemOut     O   Imprimir los certificados:
...
```

Punto de comprobación de la lección:

Ha instalado y ejecutado la aplicación de ejemplo.

Módulo 3: Configurar seguridad del transporte

Configure la seguridad del transporte para proteger la transferencia de datos entre los clientes y servidores de la configuración.

En el módulo anterior de la guía de aprendizaje, ha habilitado la autenticación de WebSphere eXtreme Scale. Con la autenticación, es necesario que cualquier aplicación que intente conectarse al servidor WebSphere eXtreme Scale proporcione una credencial. Por lo tanto, ningún cliente no autenticado se puede conectar al servidor WebSphere eXtreme Scale. Los clientes deben ser una aplicación autenticada en ejecución en una célula de WebSphere Application Server.

Con la configuración hasta este módulo, la transferencia de datos entre los clientes en el clúster appCluster y los servidores en el clúster xsCluster no está cifrada. Esta configuración podría ser aceptable si los clústeres de WebSphere Application Server están instalados detrás de un cortafuegos. Sin embargo, en algunos escenarios, no se acepta el tráfico no cifrado por varias razones, incluso aunque la topología esté protegida por cortafuegos. Por ejemplo, una política de gobierno podría obligar a tráfico cifrado. WebSphere eXtreme Scale da soporte a TLS/SSL (Transport Layer Security/Secure Sockets Layer) para la comunicación segura entre puntos finales de ObjectGrid, que incluyen servidores de cliente, servidores de contenedor y servidores de catálogo.

En este despliegue de ejemplo, todos los clientes y los servidores de contenedor de eXtreme Scale se ejecutan en el entorno de WebSphere Application Server. Las propiedades de cliente o servidor no son necesarias para configurar los valores de SSL porque la seguridad de transporte de eXtreme Scale la gestionan los valores de transporte CSIV2 (Common Secure Interoperability Protocol Versión 2) de Application Server. Los servidores WebSphere eXtreme Scale utilizan la misma instancia de intermediario de solicitud de objetos (ORB) que los servidores de aplicaciones en los que se ejecutan. Especifique todos los valores de SSL para los servidores de contenedor y cliente en la configuración de WebSphere Application Server mediante estos valores de transporte CSIv2. Debe configurar las propiedades SSL en el archivo de propiedades del servidor para el servidor de catálogo.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, ha aprendido a:

- Configurar transporte de entrada y salida CSIv2.
- Añadir propiedades SSL al archivo de propiedades del servidor de catálogo.
- Comprobar el archivo de propiedades del ORB.
- Ejecutar el ejemplo.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Requisitos previos

Este paso de la guía de aprendizaje se basa en los módulos anteriores. Complete los módulos anteriores de esta guía de aprendizaje antes de configurar la seguridad de transporte.

Lección 3.1: Configurar transporte de entrada y salida CSIv2

Para configurar TLS/SSL (Transport Layer Security/Secure Sockets Layer) para el transporte del servidor, establezca el transporte de entrada CSIv2 (Common Secure Interoperability Protocol Version 2) y el transporte de salida CSIv2 en `SSL-Required` para todos los servidores WebSphere Application Server que alojan clientes, servidores de catálogo y servidores de contenedor.

En la topología de ejemplo de la guía de aprendizaje, debe establecer estas propiedades para los servidores de aplicaciones s1, s2, xs1 y xs2. Los pasos siguientes configuran los transportes de entrada y salida para todos los servidores de la configuración.

Establezca los transportes de entrada y salida en la consola administrativa. Asegúrese de que la seguridad administrativa esté habilitada.

- **WebSphere Application Server Versión 6.1**: pulse **Securidad** > **Administración segura** > **Aplicación..** > **Seguridad RMI/IIOP** y cambie el tipo de transporte a **SSL-Required**.
- **WebSphere Application Server Versión 7.0**: pulse **Seguridad** > **Seguridad global** > **Seguridad RMI/IIOP** > **Comunicaciones de entrada CSIv2**. Cambie el tipo de transporte en la capa de transporte CSIv2 a **SSL-Required**. Repita este paso para configurar las comunicaciones de salida CSIv2s.

Puede utilizar valores de seguridad de punto final gestionados de forma centralizada, o bien puede configurar repositorios SSL. Consulte Valores de entrada de transporte de Common Secure Interoperability Versión 2 para obtener más información.

Lección 3.2: Añadir propiedades SSL al archivo de propiedades de servidor de catálogo

El servidor de catálogo se ejecuta fuera de WebSphere Application Server, por lo que debe configurar las propiedades SSL en el archivo de propiedades del servidor.

La otra razón para configurar las propiedades SSL en el archivo de propiedades del servidor es que el servidor de catálogo tiene sus propias vías de acceso de transporte de propietario que no pueden gestionar los valores de transporte CSIV2 (WebSphere Application Server Common Secure Interoperability Protocol Versión 2). Por lo tanto, debe configurar las propiedades SSL (Secure Sockets Layer) en el archivo de propiedades del servidor para el servidor de catálogo.

Propiedades SSL del archivo `catServer3.props`:

```
alias=default
contextProvider=IBMJSSE2
protocol=SSL
keyStoreType=PKCS12
keyStore=/raíz_was/IBM/WebSphere/AppServer/profiles/
<nombre_gestor_despliegue>/config/cells/<nombre_célula>/nodes/
<nombre_nodo>/key.p12
keyStorePassword=WebAS
trustStoreType=PKCS12
trustStore=/raíz_was/IBM/WebSphere/AppServer/profiles/
<nombre_gestor_despliegue>/config/cells/<nombre_célula>/nodes/
<nombre_nodo>/trust.p12
trustStorePassword=WebAS
clientAuthentication=false
```

El archivo `catServer3.props` utiliza el almacén de confianza y el almacén de claves predeterminados de nivel de nodo de WebSphere Application Server. Si está desplegando un entorno de despliegue más complejo, debe elegir el almacén de confianza y el almacén de claves correctos. En algunos casos, debe crear un almacén de claves y un almacén de confianza e importar las claves desde almacenes de claves de los otros servidores. Tenga en cuenta que la serie `WebAS` es la contraseña predeterminada del almacén de confianza y del almacén de claves de WebSphere Application Server. Consulte Configuración predeterminada de los certificados autofirmados para obtener más información.

Estas entradas ya se incluyen en el archivo *inicio_samples*`/security_extauth/catServer3.props` como comentarios. Puede descomentar las entradas y realizar las actualizaciones correspondientes a la instalación de las variables *raíz_was*, <*nombre_gestor_despliegue*>, <*nombre_célula*> y <*nombre_nodo*>.

Después de configurar las propiedades SSL, cambie el valor de la propiedad transportType de `TCP/IP` a `SSL-Required`.

Propiedades SSL del archivo `client3.props`:

También debe configurar las propiedades SSL del archivo `client3.props` porque se utiliza este archivo al detener el servidor de catálogo que se ejecuta fuera de WebSphere Application Server.

Estas propiedades no afectan a los servidores de cliente que se ejecutan en WebSphere Application Server porque utilizan los valores de transporte CSIV2 (WebSphere Application Server Common Security Interoperability Protocol Versión 2). Sin embargo, al detener el servidor de catálogo, debe proporcionar un archivo de propiedades de cliente en el mandato **`stopOgServer`**. Establezca las propiedades siguientes en el archivo <SAMPLES_HOME>/security_extauth/client3.props para que coincida con los valores especificados anteriormente en el archivo catServer3.props:

```
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/raíz_was/IBM/WebSphere/AppServer/profiles/
<nombre_gestor_despliegue>/config/cells/<nombre_célula>/nodes/
<nombre_nodo>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/raíz_was/IBM/WebSphere/AppServer/profiles/
<nombre_gestor_despliegue>/config/cells/<nombre_célula>/nodes/
<nombre_nodo>/trust.p12
#trustStorePassword=WebAS
```

Al igual que con el archivo `catServer3.props`, puede utilizar los comentarios que ya se proporcionan en el archivo *inicio_samples*`/security_extauth/client3.props` con las actualizaciones correspondientes de las variables *raíz_was*, <*nombre_gestor_despliegue*>, <*nombre_célula*> y <*nombre_nodo*> para que coincidan con su entorno.

Punto de comprobación de la lección:

Ha configurado las propiedades SSL del servidor de catálogo.

Lección 3.3: Ejecutar el ejemplo

Reinicie todos los servidores y ejecute de nuevo la aplicación de ejemplo. Debería poder ejecutar todos los pasos sin problemas.

Consulte “Lección 2.4: Instalar y ejecutar el ejemplo” en la página 123 para obtener más información sobre la ejecución e instalación de la aplicación de ejemplo.

Módulo 4: Utilizar autorización JAAS (Java Authentication and Authorization Service) en WebSphere Application Server

Ahora que ha configurado la autenticación de clientes, puede configurar adicionalmente la autorización para proporcionar a distintos usuarios diversos permisos. Por ejemplo, es posible que un usuario "operator" solo pueda visualizar datos, mientras que un usuario "manager" pueda realizar todas las operaciones.

Tras autenticar un cliente, como en el módulo anterior de esta guía de aprendizaje, puede otorgar privilegios de seguridad mediante los mecanismos de autorización de eXtreme Scale. El módulo anterior de esta guía de aprendizaje ha demostrado cómo habilitar la autenticación para una cuadrícula de datos mediante la integración con WebSphere Application Server. Como resultado, ningún cliente no

autenticado se puede conectar a los servidores eXtreme Scale o enviar solicitudes al sistema. No obstante, cada cliente autenticado tiene el mismo permiso o privilegios que el servidor, como por ejemplo, la lectura, la grabación o la supresión de datos que se almacenan en las correlaciones de ObjectGrid. Los clientes también pueden emitir cualquier tipo de consulta.

Esta parte de la guía de aprendizaje muestra cómo utilizar la autenticación de eXtreme Scale para proporcionar a los usuarios diversos privilegios. WebSphere eXtreme Scale utiliza un mecanismo de autorización basado en permisos. Puede asignar distintas categorías de permiso representadas por distintas clases de permiso. Este módulo presenta la clase MapPermission. Para ver una lista de todos los permisos posibles, consulte Programación de autorización de cliente .

En WebSphere eXtreme Scale, la clase `com.ibm.websphere.objectgrid.security.MapPermission` representa permisos a los recursos eXtreme Scale, específicamente los métodos de las interfaces ObjectMap o JavaMap. WebSphere eXtreme Scale define las siguientes series de permiso para acceder a los métodos de ObjectMap y JavaMap:

- **leer**: otorga permiso para leer los datos de la correlación.
- **grabar**: otorga permiso para actualizar los datos de la correlación.
- **insertar**: otorga permiso para insertar los datos de la correlación.
- **eliminar**: otorga permiso para eliminar los datos de la correlación.
- **invalidar**: otorga permiso para invalidar los datos de la correlación.
- **todo**: otorga todos los permisos para leer, grabar, insertar, eliminar e invalidar.

La autorización se produce cuando un cliente de eXtreme Scale utiliza una API de acceso a datos como, por ejemplo, las API ObjectMap, JavaMap o EntityManager. El tiempo de ejecución de eXtreme Scale comprueba los permisos de correlación correspondientes cuando se llama al método. Si no se otorgan los permisos necesarios al cliente, se genera una excepción AccessControlException. Esta guía de aprendizaje muestra cómo utilizar autorización JAAS (Java Authentication and Authorization Service) para otorgar acceso a la correlación a distintos usuarios.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, ha aprendido a:

- Habilitar la autorización para WebSphere eXtreme Scale.
- Habilitar la autorización basada en usuario.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos.

Lección 4.1: Habilitar la autorización de WebSphere eXtreme Scale

Para habilitar la autorización en WebSphere eXtreme Scale, debe habilitar la seguridad en un ObjectGrid específico.

Para habilitar la autorización en el ObjectGrid, debe establecer el atributo **`securityEnabled`** en true para ese ObjectGrid determinado en el archivo XML. Para esta guía de aprendizaje, puede utilizar el archivo `XSDeployment_sec.ear` del directorio *inicio_samples*`/WASSecurity`, que ya tiene la seguridad establecida en el archivo `objectGrid.xml`, o puede editar el archivo `objectGrid.xml` existente para habilitar la seguridad. Esta lección muestra cómo editar el archivo para habilitar la seguridad.

1. Opcional: Extraiga los archivos contenidos en el archivo `XSDeployment.ear` y, a continuación, desempaquete el archivo `XSDeploymentWeb.war`.
2. Opcional: Abra el archivo `objectGrid.xml` y establezca el atributo **securityEnabled** en `true` en el nivel de ObjectGrid. Consulte un ejemplo de este atributo en el siguiente ejemplo:

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

    <objectGrids>
        <objectGrid name="Grid" txTimeout="15" securityEnabled="true">
<backingMap name="Map1" />
</objectGrid>
    </objectGrids>

</objectGridConfig>
```

   Si tiene varias ObjectGrids definidas, debe establecer este atributo en cada cuadrícula.
3. Opcional: Vuelva a empaquetar los archivos `XSDeploymentWeb.war` y `XSDeployment.ear` para incluir los cambios.
4. Necesario: Desinstale el archivo `XSDeployment.ear` y a continuación instale el archivo `XSDeployment.ear` actualizado. Puede utilizar el archivo que ha modificado en los pasos anteriores, o bien puede instalar el archivo `XSDeployment_sec.ear` que se proporciona en el directorio *inicio_samples*/`WASSecurity`. Consulte "Lección 2.4: Instalar y ejecutar el ejemplo" en la página 123 si desea más información sobre la instalación de la aplicación.
5. Reinicie todos los servidores de aplicaciones para habilitar la autorización de WebSphere eXtreme Scale.

Punto de comprobación de la lección:

Ha habilitado la seguridad en el ObjectGrid, lo que también habilita la autorización en la cuadrícula de datos.

Lección 4.2: Habilitar autorización basada en usuario

En el módulo de autenticación de esta guía de aprendizaje, ha creado dos usuarios: `operator` y `manager`. Puede asignar diversos permisos a estos usuarios con autorización JAAS (Java Authentication and Authorization Service).

Definición de la política de autorización JAAS (Java Authentication and Authorization Service) mediante principales de usuario:

Puede asignar permisos a los usuarios que ha creado anteriormente. Asigne al usuario `operator` permiso de solo lectura en todas las correlaciones. Asigne al usuario `manager` todos los permisos. Utilice el archivo de política de autorización JAAS para otorgar permisos a los principales.

Edite el archivo de autorización JAAS. El archivo `xsAuth3.policy` se encuentra en el directorio *inicio_samples*/`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";
};
```

En este archivo, la base de código `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` es un URL reservado especialmente para ObjectGrid. Todos los permisos de ObjectGrid otorgados a los principales deben utilizar esta base de código especial. Se asignan los permisos siguientes a este archivo:

- La primera sentencia de otorgamiento otorga el permiso de correlación `read` al principal `"CN=operator,O=acme,OU=OGSample"`. El usuario `"CN=operator,O=acme,OU=OGSample"` solo tiene permiso de lectura de correlación a la correlación Map1 de la instancia de la cuadrícula ObjectGrid.
- La segunda sentencia de otorgamiento otorga todos los permisos de correlación al principal `"CN=manager,O=acme,OU=OGSample"`. El usuario `"CN=manager,O=acme,OU=OGSample"` tiene todos los permisos a la correlación Map1 en la instancia de la cuadrícula ObjectGrid.

Definición del archivo de política de autorización JAAS mediante las propiedades de JVM:

Utilice los pasos siguientes para establecer propiedades de JVM para los servidores xs1 y xs2, que están en el clúster xsCluster. Si utiliza una topología distinta de la topología de ejemplo que se utiliza en esta guía de aprendizaje, establezca el archivo en todos sus servidores de contenedor.

1. En la consola administrativa, pulse **Servidores** > **Servidores de aplicaciones** > ***nombre_servidor*** > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java.**
2. Añada los siguientes argumentos de JVM genéricos:

   `-Djava.security.auth.policy=`*`inicio_samples`*`/security_extauth/xsAuth3.policy`
3. Pulse **Aceptar** y guarde los cambios.

Ejecución de la aplicación de ejemplo para probar la autorización:

Puede utilizar la aplicación de ejemplo para probar los valores de autorización. El usuario manager continúa teniendo todos los permisos en la correlación Map1, incluida la visualización y adición de empleados. El usuario operator solo debe poder visualizar los empleados, ya que a dicho usuario solo se ha asignado permiso de lectura.

1. Reinicie todos los servidores de aplicaciones que ejecutan servidores de contenedor. Para esta guía de aprendizaje, reinicie los servidores `xs1` y `xs2`.
2. Abra la aplicación `EmployeeManagementWeb`. En un navegador web, abra `http://<host>:<puerto>/EmployeeManagermentWeb/management.jsp`.
3. Inicie sesión en la aplicación utilizando cualquier nombre de usuario y contraseña válidos.
4. Intente visualizar un empleado. Pulse **Visualizar un empleado** y busque la dirección de correo electrónico `authemp1@acme.com`. Se visualiza un mensaje que indica que no se puede encontrar el usuario.
5. Añada un empleado. Pulse **Añadir un empleado**. Añada el correo electrónico `authemp1@acme.com`, el nombre `Joe` y el apellido `Doe`. Pulse **Someter**. Se visualiza un mensaje que indica que se ha añadido el empleado.
6. Edite el archivo *inicio_samples*`/security_extauth/client3.props`. Cambie el valor de la propiedad credentialGeneratorProps de `manager manager1` a

operator operator1. Después de editar el archivo, el servlet utiliza el nombre de usuario "operator" y la contraseña "operator1" para la autenticación en los servidores WebSphere eXtreme Scale.

7. Reinicie el clúster appCluster para que se apliquen los cambios en el archivo *inicio_samples*/security_extauth/client3.props.
8. Intente visualizar un empleado. Pulse **Visualizar un empleado** y busque la dirección de correo electrónico authemp1@acme.com. Se visualizará el empleado.
9. Añada un empleado. Pulse **Añadir un empleado**. Añada el correo electrónico authemp2@acme.com, el nombre Joe y el apellido Doe. Pulse **Someter**. Se visualiza el mensaje siguiente:

   ```
   Se produce una excepción al Añadir el empleado. Consulte a continuación para ver mensajes de
   excepción detallados.
   ```

   A continuación se muestra el texto detallado de la excepción:

   ```
   java.security.AccessControlException: Acceso denegado
   (com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
   ```

   Este mensaje se visualiza porque el usuario operator no tiene permiso para insertar datos en la correlación Map1.

Si está ejecutando una versión de WebSphere Application Server anterior a la versión 7.0.0.11, es posible que vea un error java.lang.StackOverflowError en el servidor de contenedor. Este error se debe a un problema de IBM Developer Kit. El problema se ha solucionado en el IBM Developer Kit proporcionado con WebSphere Application Server Versión 7.0.0.11 y posterior.

Punto de comprobación de la lección:

En esta lección, ha configurado la autorización asignando permisos a usuarios específicos.

Módulo 5: Utilizar el programa de utilidad xscmd para supervisar cuadrículas de datos y correlaciones

Puede utilizar el programa de utilidad **xscmd** para mostrar las cuadrículas de datos primarias y los tamaños de correlación de la cuadrícula de datos Grid. La herramienta **xscmd** utiliza el MBean para consultar todos los artefactos de cuadrícula de datos como, por ejemplo, fragmentos primarios, fragmentos de réplica, servidores de contenedor, tamaños de correlación y otros datos.

En esta guía de aprendizaje, el servidor de catálogo se ejecuta como un servidor Java SE autónomo. Los servidores de contenedor se ejecutan en servidores de aplicaciones WebSphere Application Server.

Para el servidor de catálogo, se crea un servidor MBean en la máquina virtual Java (JVM) autónoma. Cuando se utiliza la herramienta **xscmd** en el servidor de catálogo, se utiliza la seguridad de WebSphere eXtreme Scale.

Para los servidores de contenedor, el tiempo de ejecución de WebSphere eXtreme Scale registra los beans gestionados (MBean) en el servidor MBean creado por el tiempo de ejecución de WebSphere Application Server. La seguridad que utiliza la herramienta **xscmd** la proporciona la seguridad de MBean de WebSphere Application Server.

1. Mediante la herramienta de línea de mandatos, abra el directorio *PERFIL_DMGR*/bin.

2. Ejecute la herramienta **xscmd**. Utilice los parámetros **-c showPlacement -st P** como en los ejemplos siguientes:

   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
   ```

   El nombre de usuario y la contraseña se proporcionan al servidor de catálogo para la autenticación.

3. Visualice los resultados del mandato.

   ```
   *** Mostrando todos los primarios para la cuadrícula - Grid & mapset - mapSet
   Partición Contenedor Host Servidor
   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. Ejecute la herramienta **xscmd**. Utilice el parámetro **-c showMapSizes** como en los ejemplos siguientes:

   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
   ```

   El nombre de usuario y la contraseña se proporcionan al servidor de catálogo para la autenticación. Después de ejecutar el mandato, se le solicita el ID de usuario y la contraseña de WebSphere Application Server para la autenticación en WebSphere Application Server. Debe proporcionar esta información de inicio de sesión porque la opción **-c showMapSizes** obtiene el tamaño de correlación de cada servidor de contenedor, lo que requiere la seguridad de WebSphere Application Server.

5. Opcional: Puede modificar el archivo `PROFILE/properties/sas.client.props` para ejecutar el mandato sin que se requiera el ID de usuario y la contraseña. Modifique la propiedad com.ibm.CORBA.loginSource de `prompt` a `properties` y a continuación proporcione el ID de usuario y la contraseña. A continuación se muestra un ejemplo de las propiedades del archivo `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. Opcional: Si está utilizando el mandato **xscmd** en una instalación autónoma de WebSphere eXtreme Scale, debe añadir las opciones siguientes:
   - Si está utilizando la seguridad de WebSphere eXtreme Scale:

     ```
     -user
     -pwd
     ```

   - Si está utilizando la seguridad de WebSphere eXtreme Scale con la generación de credenciales personalizada:

```
-user
-pwd
-cgc
-cgp
```

- Si SSL está habilitado:

```
-tt
-cxpv
-prot
-ks
-ksp
-kst
-ts
-tsp
-tst
```

Si la seguridad de WebSphere eXtreme Scale y SSL están habilitadas, se necesitan ambos conjuntos de parámetros.

Punto de comprobación de la lección

Ha utilizado la herramienta **xscmd** para supervisar cuadrículas de datos y correlaciones en la configuración.

Guía de aprendizaje: Ejecución de paquetes de eXtreme Scale en la infraestructura OSGi

El ejemplo de OSGi se basa en los ejemplos de serializador de Google Protocol Buffers. Cuando haya completado este conjunto de lecciones, habrá ejecutado los plug-ins de ejemplo de serializador en la infraestructura OSGi.

Objetivos del aprendizaje

Este ejemplo muestra los paquetes OSGi. El plug-in de serializador es secundario y no es necesario. El ejemplo de OSGi está disponible en la galería de ejemplos de WebSphere eXtreme Scale. Debe descargar el ejemplo y extraerlo en el directorio *inicio_wxs*/samples. El directorio raíz para el ejemplo de OSGi es wxs_home/samples/OSGiProto.

El ejemplo de serializador de Google Protocol Buffers se encuentra en el directorio *inicio_wxs*/samples/SerializerProto.

El ejemplo de serializador de Binary JSON (BSON) se encuentra en el directorio *inicio_wxs*/samples/SerializerBSON.

Los ejemplos de mandato de esta guía de aprendizaje supone que se ejecuta en el sistema operativo UNIX. Debe ajustar el ejemplo de mandato para que se ejecute en un sistema operativo Windows.

Después de completar las lecciones de este módulo, comprenderá los conceptos del ejemplo de OSGi y sabrá cómo completar los objetivos siguientes:

- Instalar el paquete de servidor WebSphere eXtreme Scale en el contenedor OSGi para iniciar el servidor eXtreme Scale.
- Configurar el entorno de desarrollo de eXtreme Scale para ejecutar el cliente de ejemplo.
- Utilizar el mandato xscmd para consultar la clasificación de servicio del paquete de ejemplo, actualizarlo a una nueva clasificación de servicio y verificar la nueva clasificación de servicio.

Tiempo necesario

Este módulo requiere aproximadamente 60 minutos para completarse.

Requisitos previos

Además de descargar y extraer los ejemplos de serializador, esta guía de aprendizaje también tiene los requisitos previos siguientes:

- Instale y extraiga el producto eXtreme Scale
- Configure el entorno Eclipse Equinox

Introducción: Inicio y configuración del servidor y contenedor de eXtreme Scale para ejecutar plug-ins en la infraestructura OSGi

En esta guía de aprendizaje inicia un servidor eXtreme Scale en la infraestructura OSGi, inicia un contenedor de eXtreme Scale y conecta los plug-ins de ejemplo al entorno de ejecución de eXtreme Scale.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, comprenderá los conceptos del ejemplo de OSGi y sabrá cómo completar los objetivos siguientes:

- Instalar el paquete de servidor WebSphere eXtreme Scale en el contenedor OSGi para iniciar el servidor eXtreme Scale.
- Configurar el entorno de desarrollo de eXtreme Scale para ejecutar el cliente de ejemplo.
- Utilizar el mandato xscmd para consultar la clasificación de servicio del paquete de ejemplo, actualizarlo a una nueva clasificación de servicio y verificar la nueva clasificación de servicio.

Tiempo necesario

Esta guía de aprendizaje requiere aproximadamente 60 minutos para completarse. Si explora otros conceptos relacionados con esta guía de aprendizaje, podría requerir más tiempo para completarse.

Nivel de conocimientos

Intermedio.

A quién va dirigida

Desarrolladores y administradores que deseen crear, instalar y ejecutar paquetes de eXtreme Scale en la infraestructura OSGi.

Requisitos del sistema

- El cliente de línea de mandatos Luminis OSGi Configuration Admin, versión 0.2.5
- Apache Felix File Install, versión 3.0.2
- Cuando se utiliza Eclipse Gemini como proveedor de contenedor Blueprint, se requiere lo siguiente:
  - Eclipse Gemini Blueprint, versión 1.0.0

- Spring Framework, versión 3.0.5
- SpringSource AOP Alliance API, versión 1.0.0
- SpringSource Apache Commons Logging, versión 1.1.1

- Cuando se utiliza Apache Aries como proveedor de contenedor Blueprint, debe tener los requisitos siguientes:
  - Apache Aries, instantánea más reciente
  - Biblioteca ASM
  - Registro PAX

Requisitos previos

Para completar esta guía de aprendizaje, debe descargar el ejemplo y extraerlo en el directorio `wxs_home/samples`. El directorio raíz para el ejemplo de OSGi es `wxs_home/samples/OSGiProto`.

Resultados esperados

Cuando haya completado esta guía de aprendizaje, habrá instalado los paquetes de ejemplo y ejecutado un cliente de eXtreme Scale para insertar datos en la cuadrícula. También puede esperar consultar y actualizar estos paquetes de ejemplo utilizando las prestaciones dinámicas que proporciona el contenedor OSGi.

Módulo 1: Preparación para instalar y configurar los paquetes del servidor de eXtreme Scale

Complete este módulo para explorar los paquetes de ejemplo de OSGi y examinar los archivos de configuración que utiliza para configurar el servidor de eXtreme Scale.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, comprenderá los conceptos y sabrá cómo completar los objetivos siguientes:

- Localizar y explorar los paquetes incluidos en el ejemplo de OSGi.
- Examinar los archivos de configuración que se utilizan para configurar el servidor y la cuadrícula de eXtreme Scale.

Lección 1.1: Comprender los paquetes de ejemplo de OSGi

Complete esta lección para localizar y explorar los paquetes que se proporcionan en el ejemplo de OSGi.

Paquetes de ejemplo de OSGi:

Además de los paquetes configurados en el archivo `config.ini`, que se muestra en el tema sobre configuración del entorno de Eclipse Equinox, se utilizan los siguientes paquetes adicionales en el ejemplo de OSGi:

`objectgrid.jar`
Paquete de tiempo de ejecución del servidor WebSphere eXtreme Scale. Este paquete se encuentra en el directorio *`inicio_wxs`*`/lib`.

`com.google.protobuf_2.4.0a.jar`
Paquete de Google Protocol Buffers, versión 2.4.0a. Este paquete se encuentra en el directorio *`raíz_wxs_sample_osgi`*`/lib`.

ProtoBufSamplePlugins-1.0.0.jar
Versión 1.0.0 del paquete de plug-in de usuario con las implementaciones de los plug-ins ObjectGridEventListener y MapSerializerPlugin de ejemplo. Este paquete se encuentra en el directorio *raíz_wxs_sample_osgi*/lib. Los servicios se configuran con clasificación de servicio 1.

Esta versión utiliza XML Blueprint estándar para configurar los servicios de plug-in de eXtreme Scale. La clase de servicio es una clase implementada por el usuario para la interfaz de WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. La clase implementada por el usuario crea un bean para cada solicitud y funciona de forma similar a un bean con ámbito de prototipo.

ProtoBufSamplePlugins-2.0.0.jar
Versión 2.0.0 del paquete de plug-in de usuario con implementaciones de los plug-ins ObjectGridEventListener y MapSerializerPlugin de ejemplo. Este paquete se encuentra en el directorio *raíz_wxs_sample_osgi*/lib. Los servicios se configuran con clasificación de servicio 2.

Esta versión utiliza XML Blueprint estándar para configurar los servicios de plug-in de eXtreme Scale. La clase de servicio utiliza una clase incorporada WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, que utiliza el servicio BlueprintContainer. Utilizando la configuración XML Blueprint estándar, los beans se pueden configurar como ámbito de prototipo o ámbito de singleton. El bean no se configura como ámbito de fragmento.

ProtoBufSamplePlugins-Gemini-3.0.0.jar
Versión 3.0.0 del paquete de plug-in de usuario con las implementaciones de los plug-ins ObjectGridEventListener y MapSerializerPlugin de ejemplo. Este paquete se encuentra en el directorio *raíz_wxs_sample_osgi*/lib. Los servicios se configuran con clasificación de servicio 3.

Esta versión utiliza el XML Blueprint específico de Eclipse Gemini para configurar los servicios de plug-in de eXtreme Scale. La clase de servicio utiliza una clase incorporada WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, que utiliza el servicio BlueprintContainer. La forma de configurar un bean con ámbito de fragmento utiliza un enfoque específico de Gemini. Esta versión configura el bean myShardListener como un bean con ámbito de fragmento proporcionando {http://www.ibm.com/schema/objectgrid}shard como valor de ámbito, y configurando un atributo ficticio para que Gemini reconozca el ámbito personalizado. Esto se debe al siguiente problema de Eclipse: https://bugs.eclipse.org/bugs/show_bug.cgi?id=348776

ProtoBufSamplePlugins-Aries-4.0.0.jar
Versión 4.0.0 del paquete de plug-in de usuario con las implementaciones de los plug-ins ObjectGridEventListener y MapSerializerPlugin de ejemplo. Este paquete se encuentra en el directorio *raíz_wxs_sample_osgi*/lib. Los servicios se configuran con clasificación de servicio 4.

Esta versión utiliza XML Blueprint estándar para configurar los servicios de plug-in de eXtreme Scale. La clase de servicio utiliza una clase incorporada WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl, que utiliza el servicio BlueprintContainer. Utilizando la configuración de XML Blueprint estándar, los beans se pueden configurar mediante un ámbito personalizado. Esta versión configura myShardListenerbean como un bean

con ámbito de fragmento proporcionando {http://www.ibm.com/schema/objectgrid}shard como valor de ámbito.

ProtoBufSamplePlugins-Activator-5.0.0.jar
Versión 5.0.0 del paquete de plug-in de usuario con las implementaciones de los plug-ins ObjectGridEventListener y MapSerializerPlugin de ejemplo. Este paquete se encuentra en el directorio *raíz_wxs_sample_osgi*/lib. Los servicios se configuran con clasificación de servicio 5.

Esta versión no utiliza contenedor Blueprint en absoluto. En esta versión, los servicios se registran utilizando el registro de servicios OSGi. La clase de servicio es una clase implementada por el usuario para la interfaz de WebSphere eXtreme Scale, com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. La clase implementada por el usuario crea un bean para cada solicitud. Funciona de forma similar a un bean con ámbito de prototipo.

Punto de comprobación de la lección:

Explorando los paquetes proporcionados con el ejemplo de OSGi, podrá comprender mejor cómo desarrollar sus propias implementaciones que se ejecutarán en el contenedor OSGi.

Ha aprendido sobre lo siguiente:

- Los paquetes incluidos con el ejemplo de OSGi
- La ubicación de estos paquetes
- La clasificación de servicio con la que se ha configurado cada uno de los paquetes

Lección 1.2: Comprender los archivos de configuración de OSGi

El ejemplo de OSGi incluye tres archivos de configuración. Utiliza estos archivos para iniciar y configurar la cuadrícula y el servidor WebSphere eXtreme Scale.

Archivos de configuración de OSGi:

En esta lección, explorará los siguientes archivos de configuración:

- collocated.server.properties
- protoBufObjectGrid.xml
- protoBufDeployment.xml

collocated.server.properties

Se requiere una configuración de servidor para iniciar un servidor. Cuando se inicia el paquete de servidor de eXtreme Scale, no inicia un servidor. Espera que se cree el PID de configuración, com.ibm.websphere.xs.server, con un archivo de propiedades del servidor. Este archivo de propiedades del servidor especifica el nombre del servidor, el número de puerto y otras propiedades del servidor.

En la mayoría de casos, creará una configuración para establecer el archivo de propiedades del servidor. En casos excepcionales, es posible que solo desee iniciar un servidor con todas las propiedades establecidas en un valor predeterminado. En ese caso, puede crear una configuración denominada com.ibm.websphere.xs.server con el valor establecido en default.

Para obtener más detalles sobre el archivo de propiedades de servidor, consulte el tema Archivo de propiedades de servidor .

El ejemplo de OSGi incluye el archivo de propiedades del servidor de ejemplo, *raíz_wxs_sample_osgi*/server/properties/collocated.server.properties. Este archivo de propiedades de ejemplo inicia un único servicio de catálogo y un servidor de contenedor en el proceso de la infraestructura OSGi. Los clientes de eXtreme Scale se conectan al puerto 2809 y los clientes JMX se conectan al puerto 1099. El contenido del archivo de propiedades del servidor de ejemplo es el siguiente:

```
serverName=collocatedServer
isCatalog=true
catalogClusterEndPoints=collocatedServer:
localhost:6601:6602traceSpec=
ObjectGridOSGi=all=enabled
traceFile=logs/trace.log
listenerPort=2809
JMXServicePort=1099
```

protoBufObjectGrid.xml

El archivo XML de descriptor de ObjectGrid protoBufObjectGrid.xml de ejemplo contiene lo siguiente, con los comentarios eliminados.

```
<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>
```

Hay dos plug-ins configurados en este archivo XML de descriptor de ObjectGrid:

ObjectGridEventListener
Plug-in de nivel de fragmento. Para cada instancia de ObjectGrid, hay una instancia de ObjectGridEventListener. Está configurada para utilizar el servicio OSCi myShardListener. Esto significa que cuando se crea la cuadrícula, el plug-in ObjectGridEventListener utiliza el servicio OSGi myShardListener con la clasificación de servicio más alta disponible.

MapSerializerPlugin
Plug-in de nivel de correlación. Para la correlación de respaldo denominada Map, hay un plug-in MapSerializerPlugin configurado. Está configurado para utilizar el servicio OSGi myProtoBufSerializer. Esto significa que cuando se crea la correlación, el plug-in MapSerializerPlugin utiliza el servicio, myProtoBufSerializer, con la clasificación de servicio con el rango más alto disponible.

protoBufDeployment.xml

El archivo XML de descriptor de despliegue describe la política de despliegue de la cuadrícula denominada `Grid`, que utiliza cinco particiones. Consulte el siguiente ejemplo de código de este archivo XML:

```
<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

Como alternativa a la utilización del archivo `collocated.server.properties` junto con el PID de configuración, `com.ibm.websphere.xs.server`, puede incluir los archivos XML del ObjectGrid y XML de despliegue en un paquete OSGi, junto con un archivo XML Blueprint, tal como se muestra en el ejemplo siguiente:

```
<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>
```

Punto de comprobación de la lección:

En esta lección, ha aprendido acerca de los archivos de configuración que se utilizan en el ejemplo de OSGi. Ahora, cuando inicie y configure el servidor y la cuadrícula de eXtreme Scale, comprenderá qué archivos se utilizan en estos procesos y cómo interactúan estos archivos con los plug-ins de la infraestructura OSGi.

Módulo 2: Instalación e inicio de paquetes de eXtreme Scale en la infraestructura OSGi

Utilice los módulos de estas lecciones para instalar el paquete de servidor eXtreme Scale en el contenedor OSGi e iniciar el servidor WebSphere eXtreme Scale.

El inicio del servidor en la infraestructura OSGi no significa que los paquetes OSGi estén listos para su ejecución. Debe configurar las propiedades del servidor y los contenedores para que los paquetes OSGi que instale se reconozcan y se ejecuten correctamente.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, comprenderá los conceptos y sabrá cómo completar las tareas siguientes:

- Instalar los paquetes de eXtreme Scale utilizando la consola de Equinox OSGi.
- Configurar el servidor eXtreme Scale.
- Configurar el contenedor de eXtreme Scale.
- Iniciar paquetes de ejemplo de eXtreme Scale.

Requisitos previos

Para completar este módulo, se requieren las tareas siguientes antes de empezar:

- Instale y extraiga el producto eXtreme Scale
- Configure el entorno Eclipse Equinox

También debe prepararse para acceder a los archivos siguientes para completar las lecciones de este módulo:

- El paquete `objectgrid.jar`. Instala este paquete de eXtreme Scale.
- El archivo `collocated.server.properties`. Añade las propiedades del servidor a este archivo de configuración.
- Puede esperar instalar e iniciar los paquetes siguientes:
- El paquete `protobuf-java-2.4.0a-bundle.jar`
- El paquete `ProtoBufSamplePlugins-1.0.0.jar`
- El paquete `ProtoBufSamplePlugins-2.0.0.jar`

Lección 2.1: Iniciar la consola e instalar el paquete de servidor de eXtreme Scale

En esta lección, utiliza la consola de Equinox OSGi para iniciar e instalar un WebSphere eXtreme Scale

1. Utilice el mandato siguiente para iniciar la consola de Equinox OSGi:

```
cd raíz_equinox

java -jar
plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar
-console
```

2. Una vez que se haya iniciado la consola OSGI, emita el mandato `ss` en la consola y se iniciarán los paquetes siguientes:

Salida de Eclipse Gemini:

```
osgi> ss
Framework is launched.
id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE com.springsource.org.apache.commons.logging_1.1.1
5 ACTIVE com.springsource.org.aopalliance_1.0.0
6 ACTIVE org.springframework.aop_3.0.5.RELEASE
7 ACTIVE org.springframework.asm_3.0.5.RELEASE
8 ACTIVE org.springframework.beans_3.0.5.RELEASE
9 ACTIVE org.springframework.context_3.0.5.RELEASE
10 ACTIVE org.springframework.core_3.0.5.RELEASE
11 ACTIVE org.springframework.expression_3.0.5.RELEASE
12 ACTIVE org.apache.felix.fileinstall_3.0.2
13 ACTIVE net.luminis.cmc_0.2.5
```

```
14 ACTIVE org.eclipse.gemini.blueprint.core_1.0.0.RELEASE
15 ACTIVE org.eclipse.gemini.blueprint.extender_1.0.0.RELEASE
16 ACTIVE org.eclipse.gemini.blueprint.io_1.0.0.RELEASE
```

Salida de Apache Aries:

```
osgi> ss
Framework is launched.
id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE org.ops4j.pax.logging.pax-logging-api_1.6.3
5 ACTIVE org.ops4j.pax.logging.pax-logging-service_1.6.3
6 ACTIVE org.objectweb.asm.all_3.3.0
7 ACTIVE org.apache.aries.blueprint_0.3.2.SNAPSHOT
8 ACTIVE org.apache.aries.util_0.4.0.SNAPSHOT
9 ACTIVE org.apache.aries.proxy_0.4.0.SNAPSHOT
10 ACTIVE org.apache.felix.fileinstall_3.0.2
11 ACTIVE net.luminis.cmc_0.2.5
```

3. Instale el paquete `objectgrid.jar`. Para iniciar un servidor en la máquina virtual Java (JVM), necesita instalar un paquete de servidor de eXtreme Scale. Este paquete de servidor eXtreme Scale puede iniciar un servidor y crear contenedores. Utilice el mandato siguiente para instalar el archivo `objectgrid.jar`:

   ```
   osgi> install file:///inicio_wxs/lib/objectgrid.jar
   ```

   Consulte el siguiente ejemplo:

   ```
   osgi> install
   file:///opt/wxs/ObjectGrid/lib/objectgrid.jar
   ```

   Equinox visualiza su ID de paquete; por ejemplo:

   ```
   El ID de paquete es 19
   ```

   Recuerde: El ID de paquete puede ser distinto. La vía de acceso del archivo debe ser un URL absoluto a la vía de acceso del paquete. No se da soporte a vías de acceso relativas.

Punto de comprobación de la lección:

En esta lección, ha utilizado la consola de Equinox OSGi para instalar el paquete `objectgrid.jar`, que utilizará para iniciar un servidor y crear un contenedor posteriormente en esta guía de aprendizaje.

Lección 2.2: Personalizar y configurar el servidor eXtreme Scale

Utilice esta lección para personalizar y añadir las propiedades de servidor al servidor WebSphere eXtreme Scale.

1. Edite el archivo `raíz_wxs_sample_osgi/server/properties/collocated.server.properties`.
   a. Cambie la propiedad workingDirectory a `raíz_equinox`.
   b. Cambie la propiedad traceFile a `raíz_equinox/logs/trace.log`.
2. Guarde el archivo.
3. Especifique las siguientes líneas de código en la consola OSGI para crear la configuración de servidor desde el archivo:

```
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. Para visualizar la configuración, ejecute el mandato siguiente:

```
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
```

Punto de comprobación de la lección:

En esta lección, ha editado el archivo `raíz_wxs_sample_osgi/server/properties/collocated.server.properties` para especificar valores de servidor como, por ejemplo, el directorio de trabajo y la ubicación de los archivos de registro de rastreo.

Lección 2.3: Configurar el contenedor de eXtreme Scale

Complete esta lección para configurar un contenedor, que incluye el archivo XML de descriptor de ObjectGrid de WebSphere eXtreme Scale y el archivo XML de despliegue de ObjectGrid. Estos archivos incluyen la configuración de la cuadrícula y su topología.

Para crear un contenedor, primero cree un servicio de configuración utilizando el número de identificador de proceso (PID) de la fábrica de servicios gestionados, com.ibm.websphere.xs.container. La configuración de servicio es una fábrica de servicios gestionados, así que puede crear varios PID de servicio a partir de un PID de fábrica. A continuación, para iniciar el servicio de contenedor, establezca los PID de objectgridFile y deploymentPolicyFile para cada PID de servicio.

Complete los pasos siguientes para personalizar y añadir las propiedades de servicio a la infraestructura OSGi:

1. En la consola OSGI, especifique el mandato siguiente para crear el contenedor a partir del archivo:

```
osgi> cm createf com.ibm.websphere.xs.container
PID: com.ibm.websphere.xs.container-1291179621421-0
```

2. Especifique el mandato siguiente para enlazar el PID que se acaba de crear con los archivos XML de ObjectGrid.

   Recuerde: El número de PID será distinto a lo que se incluye en este ejemplo.

```
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. Utilice el mandato siguiente para visualizar la configuración:

```
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
```

Punto de comprobación de la lección:

En esta lección, ha creado un servicio de configuración, que ha utilizado para crear un contenedor de eXtreme Scale. Puesto que los archivos XML de ObjectGrid contienen la configuración para la cuadrícula y su topología, debía enlazar el contenedor que había creado a estos archivos XML de ObjectGrid. Con esta configuración, el contenedor de eXtreme Scale puede reconocer los paquetes OSGi que ejecutará posteriormente en esta guía de aprendizaje.

Lección 2.4: Instalar los paquetes Google Protocol Buffers y de plug-in de ejemplo

Complete esta guía de aprendizaje para instalar el paquete`protobuf-java-2.4.0a-bundle.jar` y el paquete del plug-in `ProtoBufSamplePlugins-1.0.0.jar` mediante la consola de Equinox OSGi.

Complete los pasos siguientes para instalar el paquete Google Protocol Buffers.

En la consola OSGI, especifique el mandato siguiente para instalar el paquete:

```
osgi> install file:///wxs_sample_osgi_root/common/lib/com.google.protobuf_2.4.0a.jar
```

Se visualiza la salida siguiente:

```
El ID de paquete es 21
```

Visión general de los paquetes de plug-in de ejemplo:

Este ejemplo de OSGi incluye cinco paquetes de ejemplo que incluyen plug-ins eXtreme Scale, incluido un plug-in ObjectGridEventListener y un plug-in MapSerializerPlugin personalizados. El plug-in MapSerializerPlugin utiliza el ejemplo Google Protocol Buffers y los mensajes proporcionados por el ejemplo MapSerializerPlugin.

Los paquetes siguientes se encuentran en el directorio *raíz_osgi_ejemplo_wxs*`/lib`: `ProtoBufSamplePlugins-1.0.0.jar` y `ProtoBufSamplePlugins-2.0.0.jar`.

El archivo `blueprint.xml` tiene el siguiente contenido con los comentarios eliminados:

```
<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>
```

El archivo XML Blueprint exporta dos servicios, myShardListener y myProtoBufSerializer. Se hace referencia a estos dos servicios en el archivo `protoBufObjectgrid.xml`.

Instalar el paquete de plug-in de ejemplo:

Complete los pasos siguientes para instalar el paquete `ProtoBufSamplePlugins-1.0.0.jar`.

Ejecute el mandato siguiente en la consola de Equinox OSGi para instalar el paquete del plug-in `ProtoBufSamplePlugins-1.0.0.jar`:

```
osgi> install file:///wxs_sample_osgi_root/common/lib/ProtoBufSamplePlugins-1.0.0.jar
```

Se visualiza la salida siguiente:

```
El ID de paquete es 22
```

Punto de comprobación de la lección:

En esta sesión, ha instalado el paquete `protobuf-java-2.4.0a-bundle.jar` y el paquete de plug-in `ProtoBufSamplePlugins-1.0.0.jar`.

Lección 2.5: Iniciar los paquetes OSGi

El servidor WebSphere eXtreme Scale se empaqueta como un paquete de servidor OSGi. Complete esta lección para instalar el paquete de servidor eXtreme Scale así como otros paquetes OSGi que ha instalado.

1. Inicie el paquete del plug-in de ejemplo. Ejecute el mandato siguiente en la consola de Equinox OSGi para iniciar el paquete. En este ejemplo, el ID del paquete del plug-in de ejemplo es 22.

   ```
   osgi> start 22
   ```

2. Inicie el paquete de Google Protocol Buffers. Ejecute el mandato siguiente en la consola de Equinox OSGi para iniciar el paquete. En este ejemplo, el ID del paquete del plug-in de Google Protocol Buffers es 21.

   ```
   osgi> start 21
   ```

3. Inicie el paquete del servidor. Ejecute el mandato siguiente en la consola de OSGi para iniciar el servidor. En este ejemplo, el ID de paquete del paquete de servidor eXtreme Scale es 19.

   ```
   osgi> start 19
   ```

Después de iniciar el servidor, el escucha de sucesos de MyShardListener se ha iniciado y está preparado para insertar o actualizar registros. Puede ver la salida siguiente en la consola de OSGI para confirmar que el paquete del plug-in se ha iniciado satisfactoriamente:

```
SystemOut O MyShardListener@1253853884(version=1.0.0) order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Order$Builder
@1aba1aba(22) inserted
```

Punto de comprobación de la lección:

En esta lección, ha iniciado dos paquetes de plug-in y el paquete del servidor en el contenedor de eXtreme Scale que ha configurado en la infraestructura OSGi.

Módulo 3: Ejecución del cliente de ejemplo de eXtreme Scale

El servidor de WebSphere eXtreme Scale ahora se ejecuta en un entorno OSGi. Complete los pasos de este módulo para ejecutar un cliente de WebSphere eXtreme Scale que inserte datos en la cuadrícula.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, sabrá cómo completar las tareas siguientes:

- Ejecutar una aplicación cliente que se conecta a la cuadrícula e inserta y recupera datos de ella.
- Iniciar un pedido utilizando una aplicación cliente no OSGi.

Requisitos previos

Complete el Módulo 2: Instalación e inicio de paquetes de eXtreme Scale en la infraestructura OSGi.

Lección 3.1: Configurar Eclipse para ejecutar el cliente y construir los ejemplos

Complete esta lección para importar el proyecto Eclipse que utilizará para ejecutar el cliente y construir los plug-ins de ejemplo.

El ejemplo incluye un programa cliente Java SE que se conecta a la cuadrícula e inserta y recupera datos de la misma. También incluye proyectos que puede utilizar para construir y volver a desplegar los paquetes OSGi.

El proyecto proporcionado se ha probado con Eclipse 3.x y posterior y sólo necesita la perspectiva de proyecto de desarrollo Java estándar. Complete los pasos siguientes para configurar el entorno de desarrollo de WebSphere eXtreme Scale.

1. Abra Eclipse en un espacio de trabajo nuevo o existente.
2. En el menú Archivo, seleccione **Importar**.
3. Expanda la carpeta General. Seleccione **Proyectos existentes en espacio de trabajo** y pulse **Siguiente**.
4. En el campo **Seleccionar directorio raíz**, escriba o vaya al directorio *raíz_wxs_sample_osgi*. Pulse **Finalizar**. Se visualizan varios proyectos en el espacio de trabajo. Debe corregir varios errores de construcción definiendo la biblioteca de usuario de eXtreme Scale. Complete los pasos siguientes para definir la biblioteca de usuario.
5. En el menú Ventana, seleccione **Preferencias**.
6. Expanda la rama **Java** > **Vía de acceso de compilación** y seleccione **Bibliotecas de usuario**.
7. Pulse **Nueva**.
8. Especifique `eXtremeScale` en el campo **Nombre de biblioteca de usuario** y pulse **Aceptar**.
9. Seleccione la nueva biblioteca de usuario y pulse **Añadir JAR**.
   a. Vaya al archivo `objectgrid.jar` del directorio *raíz_instalación_wxs*`/lib` y selecciónelo. Pulse **Aceptar**.
   b. Para incluir la documentación de la API para las API de ObjectGrid, seleccione la ubicación de la documentación de la API para el archivo `objectgrid.jar` que ha añadido en el paso anterior. Pulse **Editar**.
   c. En el recuadro de vía de acceso de ubicación de la documentación de la API, seleccione el archivo `Javadoc.zip` incluido en el directorio siguiente: *raíz_instalación_wxs*`/docs/javadoc.zip`.

Punto de comprobación de la lección:

En esta lección, ha importado el proyecto Eclipse de ejemplo, ha definido la biblioteca de usuario de eXtreme Scale y ha incluido documentación de la API de soporte para el proyecto de ejemplo. Ahora está preparado para iniciar la aplicación cliente de ejemplo.

Lección 3.2: Iniciar un cliente e insertar datos en la cuadrícula

Complete esta lección para iniciar un cliente no OSGi y ejecutar una aplicación cliente.

La aplicación de cliente Java es
`com.ibm.websphere.samples.xs.proto.client.Client`.

Este cliente utiliza una sustitución del cliente, el archivo XML de descriptor de ObjectGrid para sustituir la configuraación de OSGi, de forma que el cliente pueda ejecutarse en un entorno no OSGi. Consulte el contenido siguiente del archivo donde se han eliminado los comentarios y las cabeceras.Algunas líneas de código se muestran en varias líneas por razones de formato.

```
<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>
```

Complete los pasos siguientes para iniciar la aplicación cliente.

1. Utilice el siguiente ejemplo de código para modificar los atributos de la clase cliente para que reflejen su entorno.

   ```
   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. Ejecute la aplicación cliente.

Al ejecutar la aplicación, se visualiza el mensaje siguiente. El mensaje indica que se ha insertado un pedido:

```
order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Or
der$Builder@5d165d16(5000000) inserted
```

Punto de comprobación de la lección:

En esta lección, ha iniciado la aplicación
`com.ibm.websphere.samples.xs.proto.client.Client`, que ha generado un pedido.

Módulo 4: Consulta y actualización del paquete de ejemplo

Complete las lecciones de este módulo para utilizar el mandato **xscmd** para consultar la clasificación de servicio del paquete de ejemplo, actualizarla a una nueva clasificación de servicio y verificar la nueva clasificación de servicio.

Se proporciona un proyecto eclipse como una manera cómoda de ejecutar las aplicaciones de ejemplo.

Objetivos del aprendizaje

Después de completar las lecciones de este módulo, sabrá cómo completar las tareas siguientes:

- Consultar la clasificación de servicio actual del servicio.
- Consultar la clasificación actual de todos los servicios.
- Consultar todas las clasificaciones disponibles para un servicio.
- Consultar todas las clasificaciones de servicio disponibles.
- Utilizar la herramienta xscmd para verificar si hay disponibles clasificaciones de servicio específicas.
- Actualizar las clasificaciones de servicio para servicios OSGi de ejemplo.

Requisitos previos

Complete el Módulo 3: Ejecución del cliente de ejemplo de eXtreme Scale.

Lección 4.1: Consultar clasificaciones de servicio

Complete esta lección para consultar las clasificaciones de servicio actuales así como las clasificaciones de servicio disponibles para su actualización.

- Consulte la clasificación de servicio actual del servicio. Especifique el mandato siguiente para consultar la clasificación de servicio actual que se utiliza para el servicio, myShardListener, que utiliza el ObjectGrid denominado `Grid` y el conjunto de correlaciones denominado `MapSet`.
  1. Cambie al directorio siguiente:
     ```
     cd inicio_wxs/bin
     ```
  2. Especifique el mandato siguiente para consultar la clasificación de servicio actual correspondiente al servicio, myShardListener.
     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet -sn myShardListener
     ```

     Se visualiza la salida siguiente:
     ```
     OSGi Service Name: myShardListener
     ObjectGrid Name MapSet Name Server Name        Current Ranking
     --------------- ----------- -----------        ---------------
     Grid            MapSet      collocatedServer   1

     CWXSI0040I: The command osgiCurrent has completed successfully.
     ```
- Consulte la clasificación actual de todos los servicios. Especifique el mandato siguiente para consultar la clasificación de servicio actual de todos los servicios utilizados por el ObjectGrid denominado `Grid` y el conjunto de correlaciones denominado `MapSet`.
  1. Cambie al directorio siguiente:
     ```
     cd inicio_wxs/bin
     ```
  2. Especifique el mandato siguiente para consultar la clasificación de servicio actual de todos los servicios.
     ```
     ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
     ```

     Se visualiza la salida siguiente:

```
OSGi Service Name    Current Ranking ObjectGrid Name MapSet Name Server Name
-----------------    --------------- --------------- ----------- -----------
myProtoBufSerializer 1               Grid            MapSet      collocatedServer
```

```
myShardListener      1              Grid           MapSet      collocatedServer

CWXSI0040I: The command osgiCurrent has completed successfully.
```

- Consulte todas las clasificaciones disponibles para un servicio. Especifique el mandato siguiente para consultar todas las clasificaciones de servicio disponibles para el servicio denominado myShardListener.
  1. Cambie al directorio siguiente:
     ```
     cd inicio_wxs/bin
     ```
  2. Especifique el mandato siguiente para consultar todas las clasificaciones disponibles para un servicio.
     ```
     ./xscmd.sh -c osgiAll -sn myShardListener
     ```

     Se visualiza la salida siguiente:
     ```
     Server: collocatedServer
        OSGi Service Name Available Rankings
        ---------------- ------------------
        myShardListener 1   Summary - All servers have the same service rankings.

     CWXSI0040I: The command osgiAll has completed successfully.
     ```

     La salida la agrupa el servidor. En este ejemplo, sólo existe el siguiente servidor: collocatedServer.
- Consulte todas las clasificaciones de servicio disponibles. Entre el mandato siguiente para consultar todas las clasificaciones de servicio disponible para todos los servicios.
  1. Cambie al directorio siguiente:
     ```
     cd inicio_wxs/bin
     ```
  2. Especifique el mandato siguiente para consultar todas las clasificaciones de servicio disponibles.
     ```
     ./xscmd.sh -c osgiAll
     ```

     Se visualiza la salida siguiente:
     ```
     Server: collocatedServer
        OSGi Service Name    Available Rankings
        -----------------    ------------------
        myProtoBufSerializer 1
        myShardListener      1

     Summary - All servers have the same service rankings.
     ```
- Instale e inicie la versión 2 del paquete de plug-in. En la consola OSGi del servidor, instale un paquete nuevo que contenga una nueva versión de la clase Order y el plug-in MapSerializerPlugin. Consulte Lección 2.4: Instalar los paquetes Google Protocol Buffers y de plug-in de ejemplo para obtener más información sobre cómo instalar el paquete `ProtoBufSamplePlugins-2.0.0.jar`.
  1. Después de la instalación, inicie el nuevo paquete. Los servicios para el nuevo paquete están disponibles, pero el servidor eXtreme Scale no los utiliza aún. Debe ejecutar una solicitud de actualización de servicio para utilizar un servicio con una versión específica.
- Ahora al consultar de nuevo todas las clasificaciones de servicio disponibles, la clasificación de servicio 2 se añadirá a la salida.
  1. Cambie al directorio siguiente:
     ```
     cd inicio_wxs/bin
     ```
  2. Especifique el mandato siguiente para consultar todas las clasificaciones de servicio disponibles.

```
./xscmd.sh -c osgiAll
```

Se visualiza la salida siguiente:

```
Server: collocatedServer
   OSGi Service Name    Available Rankings
   -----------------    ------------------
   myProtoBufSerializer 1, 2
   myShardListener      1, 2

Summary - All servers have the same service rankings.
```

Punto de comprobación de la lección:

En esta guía de aprendizaje, ha consultado las clasificaciones de servicio especificadas actualmente y todas las disponibles. También ha visualizado la clasificación de servicio para un nuevo paquete que ha instalado e iniciado.

Lección 4.2: Determinar si hay clasificaciones de servicio específicas disponibles

Complete esta lección para determinar si hay clasificaciones de servicio específicas disponibles para los nombres de servicio que especifique.

1. Especifique el mandato siguiente para determinar si el servicio denominado myShardListener, con la clasificación de servicio 2 y el servicio denominado myProtoBufSerializer, con la clasificación de servicio 2, están disponibles. La lista de clasificaciones de servicio se proporciona utilizando la opción `-sr`.
   a. Cambie al directorio siguiente:

      ```
      cd inicio_wxs/bin
      ```

   b. Especifique el mandato siguiente para determinar si los servicios están disponibles:

      ```
      ./xscmd.sh -c osgiCheck -g Grid -ms MapSet -sr
      "myShardListener;2,myProtoBufSerializer;2"
      ```

      Se visualiza la salida siguiente:

      ```
      CWXSI0040I: The command osgiCheck has completed successfully.
      ```

2. Especifique el mandato siguiente para determinar si el servicio denominado myShardListener, con la clasificación de servicio 2 y el servicio denominado myProtoBufSerializer, con la clasificación de servicio 3, están disponibles.
   a. Cambie al directorio siguiente:

      ```
      cd inicio_wxs/bin
      ```

   b. Especifique el mandato siguiente para determinar si los servicios están disponibles:

      ```
      ./xsadmin.sh -c osgiCheck -g Grid -ms MapSet -sr
      "myShardListener;2,myProtoBufSerializer;3"
      ```

      Se visualiza la salida siguiente:

      ```
      Server OSGi Service Unavailable Rankings
      ------ ------------ --------------------
      collocatedServer myProtoBufSerializer 3
      ```

Punto de comprobación de la lección:

En esta lección, ha especificado los servicios myShardListener y myProtoBufSerializer, junto con clasificaciones de servicio específicas para determinar si estas clasificaciones estaban disponibles.

Lección 4.3: Actualizar las clasificaciones de servicio

Complete esta lección para actualizar clasificaciones de servicio actuales que haya consultado.

1. Si se especifica el mandato siguiente, se actualizarán las clasificaciones de servicio de los servicios denominados myShardListener y myProtoBufSerializer a la clasificación de servicio 2. La lista de clasificaciones de servicio se proporciona mediante la opción `-sr`.
   a. Cambie al directorio siguiente:
      ```
      cd inicio_wxs/bin
      ```
   b. Especifique el mandato siguiente para actualizar las clasificaciones de servicio:
      ```
      ./xscmd.sh -c osgiUpdate -g Grid -ms MapSet
      -sr "myShardListener;2,myProtoBufSerializer;2"
      ```

      Se visualiza la salida siguiente:
      ```
      La actualización ha sido satisfactoria para las siguientes clasificaciones
      de servicio:
      Service Ranking
      ------- -------
      myProtoBufSerializer 2
      myShardListener 2

      CWXSI0040I: The command osgiUpdate has completed successfully.
      ```

      Se visualiza la salida siguiente en la consola OSGi:
      ```
      SystemOut O MyShardListener@326505334(version=2.0.0) order
      com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order$Builder@
      22342234(34) updated
      ```

      Tenga en cuenta que el servicio MyShardListener es ahora la versión 2.0.0, que tiene clasificación de servicio 2.
2. Si ejecuta el mandato **xscmd** para consultar la clasificación de servicio actual que se utiliza para todos los servicios que utiliza el ObjectGrid denominado `Grid` y el conjunto de correlaciones denominado `MapSet`.
   a. Cambie al directorio siguiente:
      ```
      cd inicio_wxs/bin
      ```
   b. Especifique el mandato siguiente para consultar las clasificaciones de servicio correspondientes a todos los servicios que utilizan `Grid` y `MapSet`:
      ```
      ./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
      ```

      Se visualiza la salida siguiente:
      ```
      OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
      ----------------- --------------- --------------- ----------- -----------
      myProtoBufSerializer 2 Grid MapSet collocatedServer
      myShardListener 2 Grid MapSet collocatedServer

      CWXSI0040I: The command osgiCurrent has completed successfully.
      ```

Punto de comprobación de la lección:

En esta lección, ha actualizado las clasificaciones de servicio para los servicios myShardListener y myProtoBufSerializer.

.

Capítulo 4. Instalación

WebSphere eXtreme Scale es una cuadrícula de datos en memoria que puede utilizar para crear particiones, replicar y gestionar de forma dinámica datos de aplicación y la lógica empresarial entre varios servidores. Después de determinar los objetivos y los requisitos de su despliegue, instale eXtreme Scale en el sistema.

Antes de empezar

- Antes de empezar la instalación, debe tener conocimientos sobre las arquitecturas de memoria caché de WebSphere eXtreme Scale, la integración de memoria caché y base de datos, la serialización, la escalabilidad y la disponibilidad. Consulte Visión general del producto para obtener más información.
- Planee el despliegue de WebSphere eXtreme Scale. Para obtener más información sobre las distintas topologías de almacenamiento en memoria caché, información de tamaños y más, consulte Capítulo 2, “Planificación”, en la página 9.
- Verifique que el entorno cumple los requisitos previos para instalar eXtreme Scale. Si desea más información, consulte “Requisitos de hardware y software” en la página 49.
- Para obtener más información sobre los entornos y otros requisitos, consulte “Planificación de la instalación” en la página 49.
- Si está instalando una actualización de una versión anterior de WebSphere eXtreme Scale, siga los pasos descritos en “Actualización de servidores eXtreme Scale” en la página 211.

Visión general de la instalación

Puede utilizar la instalación completa o de cliente para instalar WebSphere eXtreme Scale en un entorno de WebSphere Application Server o autónomo.

Tipos de instalación

El instalador completo y el instalador de cliente aparte que puede descargar del sitio de soporte le proporcionan diversas opciones de instalación. Cuando se utiliza el instalador completo, puede ejecutar tanto servidores de catálogo como servidores de contenedor. En los servidores que ejecutan aplicaciones cliente que acceden a la cuadrícula de datos, puede utilizar una instalación de solo cliente. Utilice la instalación de servidor o la instalación de cliente y servidor en nodos que ejecuten servidores de catálogo o servidores de contenedor.

- **Instalación completa:**
  - Al instalar en WebSphere Application Server, puede elegir instalar solo el cliente o tanto el servidor como el cliente.
  - Al instalar en un entorno autónomo, puede instalar tanto el cliente como el servidor. Si desea instalar solo el cliente, utilice la instalación de WebSphere eXtreme Scale Client.
- **Instalación de cliente:**

  Puede utilizar la instalación solo de cliente en nodos que ejecutan las aplicaciones cliente. Para instalar solo el cliente, puede descargar el instalador solo de cliente para la plataforma correspondiente en la sección de descargar del sitio de soporte.

Opciones del entorno

Puede instalar WebSphere eXtreme Scale en un entorno autónomo o de WebSphere Application Server.

- **Entorno de WebSphere Application Server:**

  Instalando WebSphere eXtreme Scale en los nodos del entorno de WebSphere Application Server, puede iniciar automáticamente servidores de catálogo y servidores de contenedor de la misma célula que el gestor de despliegue y otros servidores de aplicaciones.
- **Entorno autónomo:**

  En una instalación autónoma, instala WebSphere eXtreme Scale en un entorno que no tiene WebSphere Application Server. Con un entorno autónomo, configura e inicia manualmente los procesos de servidor de catálogo y servidor de contenedor.

Planificación de la instalación

Antes de instalar el producto, debe tener en cuenta su entorno.

Topologías de instalación

Con WebSphere eXtreme Scale, puede crear muchas topologías de instalación que incluyan servidores autónomos y/o WebSphere Application Server. Los ejemplos siguientes son algunas de las posibles topología que puede crear.

Nodo de desarrollo

El escenario de instalación más simple es la creación de un nodo de desarrollo. En este escenario, instala el cliente y el servidor de WebSphere eXtreme Scale una vez en el nodo donde desea desarrollar la aplicación.

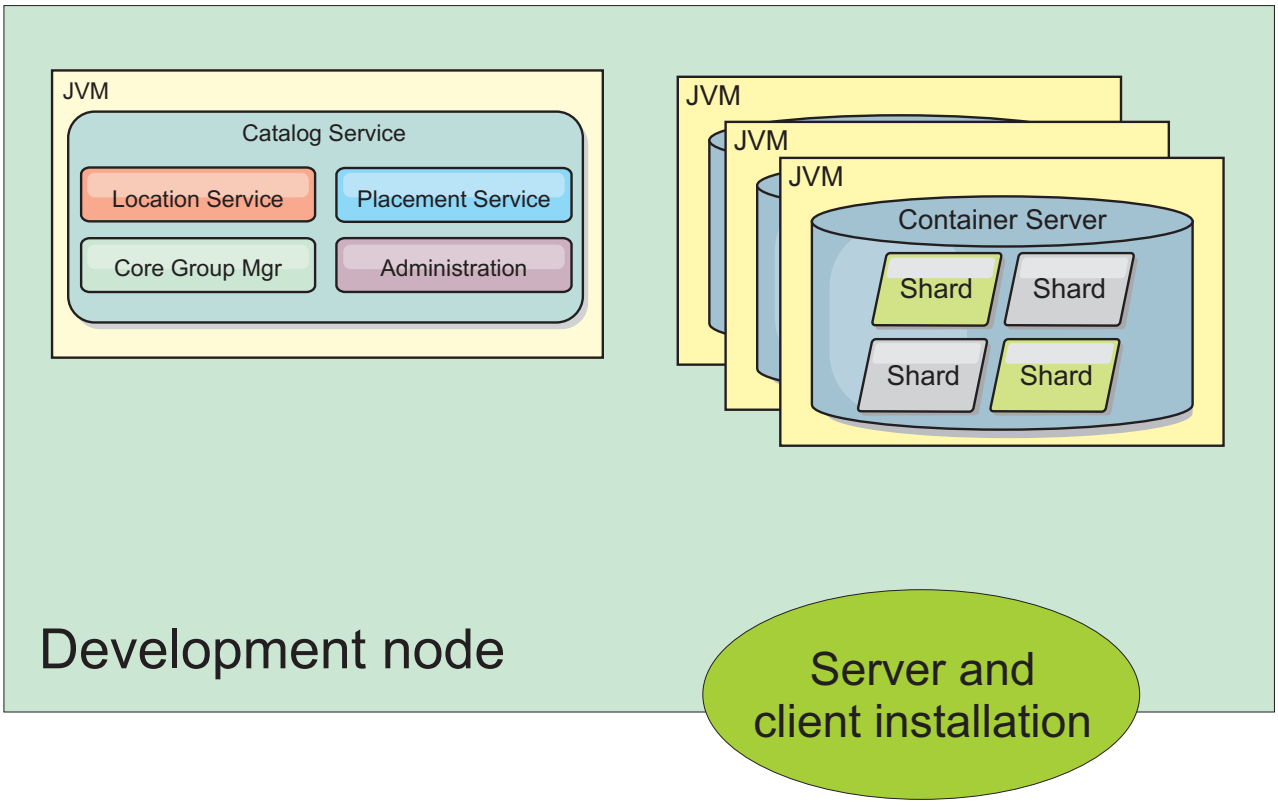


Figura 22. Nodo de desarrollo

Después de completar la instalación en la modalidad de desarrollo, puede configurar el entorno de desarrollo y empezar a escribir las aplicaciones.

Topología autónoma

Una topología autónoma consta de servidores que no se ejecutan en WebSphere Application Server. Puede crear muchas topologías autónomas diferentes, pero se incluye la siguiente topología como ejemplo. En esta topología, existen dos centros de datos. En cada centro de datos, se realizan instalaciones completas (cliente y servidor) e instalaciones de sólo cliente de WebSphere eXtreme Scale en los servidores físicos. Las instalaciones de sólo cliente están en los nodos que ejecutan las aplicaciones web que utilizan la cuadrícula de datos. Estos nodos no ejecutan servidores de catálogo o contenedor, de modo que la instalación de servidor no es necesaria. Un enlace multimaestro conecta los dos dominios de servicio de catálogo en la configuración. El enlace multimaestro permite la réplica entre los fragmentos de los servidores de contenedor de los diferentes centros de datos.

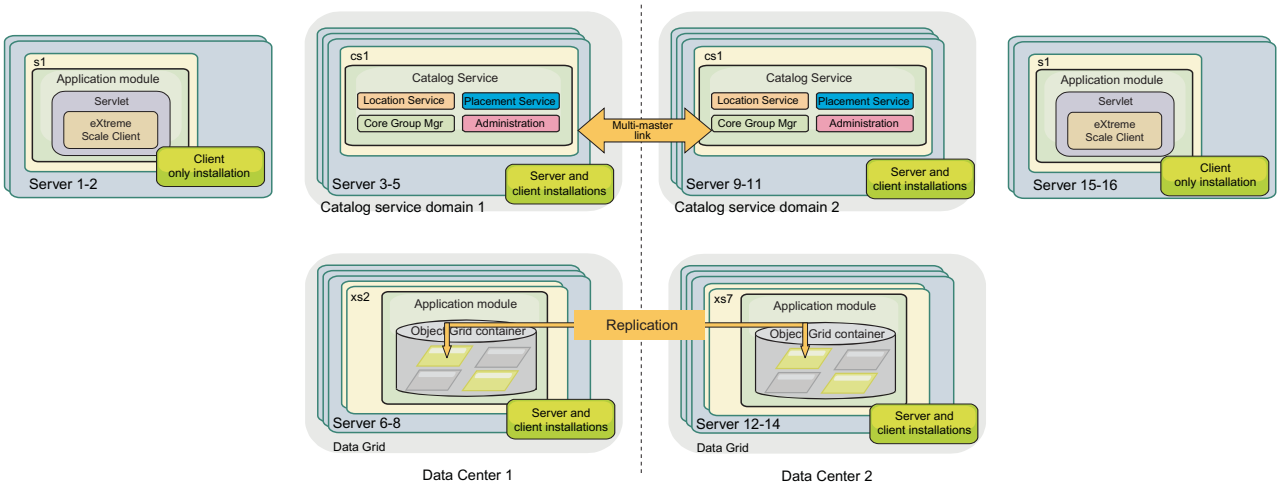


Figura 23. Topología autónoma con dos centros de datos

Ventajas de la utilización de una topología autónoma:

- Opciones de integración flexibles que se pueden incluir con bibliotecas e infraestructuras de proveedor.
- Menor ocupación que una topología de WebSphere Application Server.
- Menos requisitos de licencia que una topología de WebSphere Application Server.
- Opciones de Java Runtime Environment (JRE) expandidas.

Topología de WebSphere Application Server

También puede crear una instalación que se ejecute por completo en una célula de WebSphere Application Server. Los clientes, los servidores de catálogo y los servidores de contenedor tienen cada uno un clúster asociado. Los nodos que ejecutan la aplicación tienen la instalación de sólo cliente. Los demás nodos tienen la instalación de cliente y servidor.

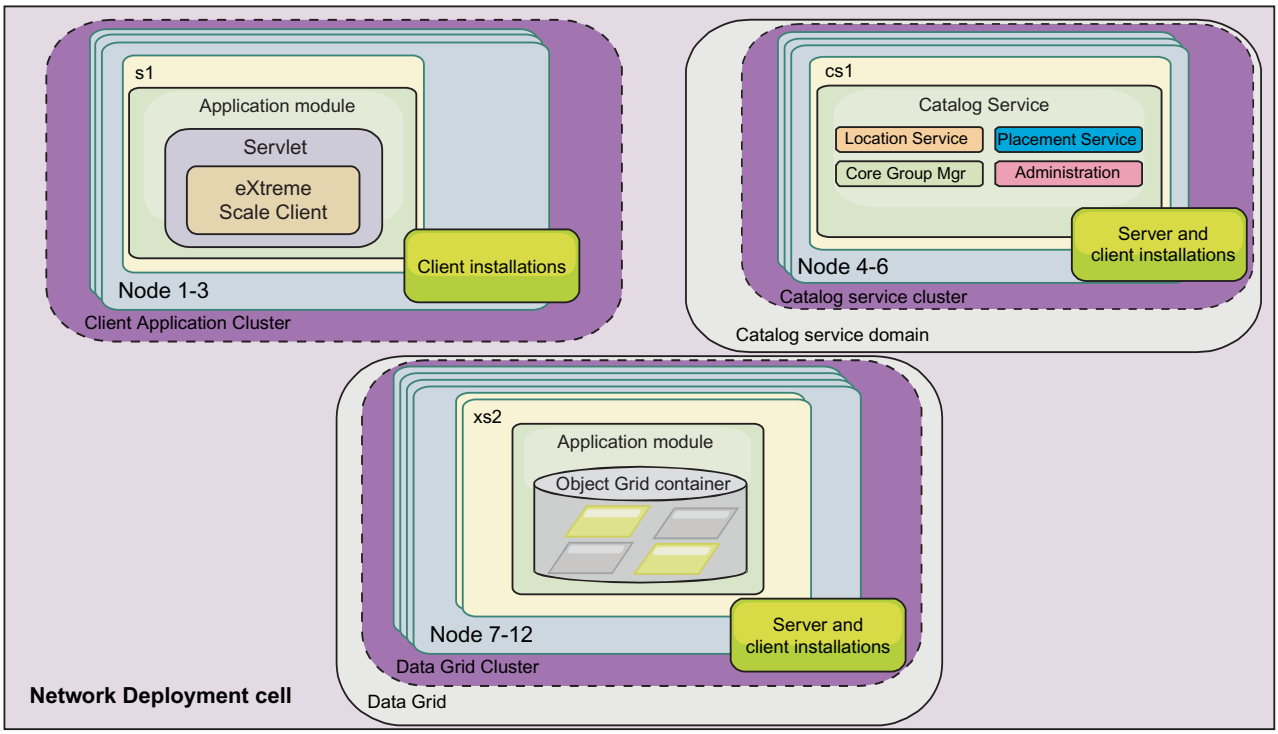


Figura 24. Ejemplo de topología de WebSphere Application Server

Ventajas de utilizar una topología de WebSphere Application Server.

- Administración y configuración centralizadas y coherentes.
- Integración de seguridad.
- Integración de aplicación Java EE.
- integración de PMI (Performance monitoring infrastructure).
- Integración con los siguientes componentes de WebSphere Application Server: memoria caché OpenJPA L2, memoria caché dinámica y persistencia de sesión HTTP.

Topología mixta

Puede crear una topología mixta que contenga WebSphere Application Server y servidores autónomos. En el siguiente ejemplo, las aplicaciones de cliente se ejecutan en la célula de WebSphere Application Server, mientras que los servidores de catálogo y los servidores de contenedor se ejecutan en modalidad autónoma.

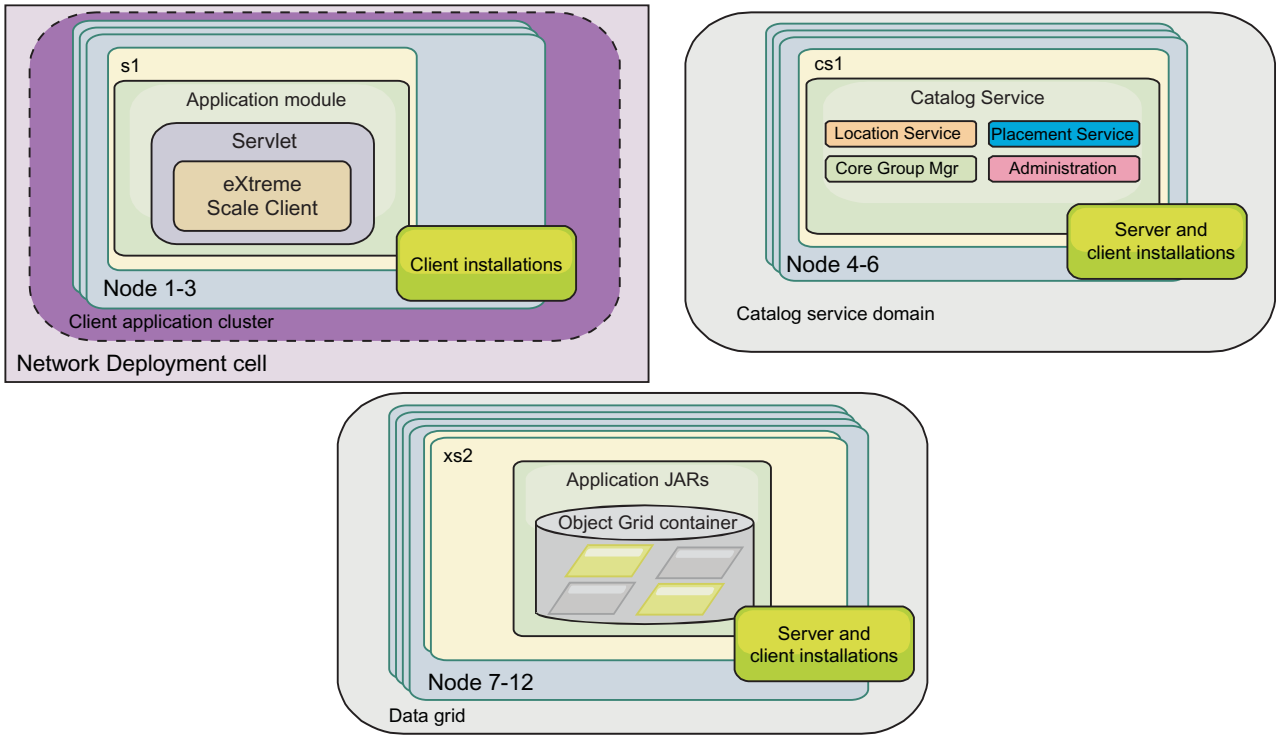


Figura 25. Ejemplo de topología mixta

Requisitos de hardware y software

Examine una visión general de requisitos de hardware y de sistema operativo. Aunque no es necesario que utilice un nivel específico de hardware o sistema operativo para WebSphere eXtreme Scale, están disponibles opciones de hardware y software soportadas formalmente en la página Systems Requirements (Requisitos de sistema) del sitio de soporte del producto. Si existe un conflicto entre el Information Center y la página de requisitos de sistema, tiene prioridad la información del sitio web. La información de requisitos previos en el centro de información sólo se proporciona por comodidad.

Consulte la página Requisitos del sistema para ver el conjunto oficial de requisitos de hardware y software.

No es necesario que instale y despliegue eXtreme Scale en un nivel específico de sistema operativo. Cada instalación de Java Platform, Standard Edition (Java SE) y Java Platform, Enterprise Edition (Java EE) necesita diferentes niveles de sistema operativo o arreglos.

Puede instalar y desplegar el producto en los entornos de Java EE y Java SE. También puede empaquetar el componente de cliente con las aplicaciones Java EE directamente si integrarse con WebSphere Application Server. WebSphere eXtreme Scale soporta Java SE 5 o posteriores y WebSphere Application Server Versión 6.1 y posteriores.

Requisitos de hardware

WebSphere eXtreme Scale no requiere un nivel específico de hardware. Los requisitos de hardware dependen del hardware soportado para la instalación de Java Platform, Standard Edition que utiliza para ejecutar WebSphere eXtreme Scale. Si utiliza eXtreme Scale con WebSphere Application Server u otra implementación de Java Platform, Enterprise Edition, los requisitos de hardware de estas plataformas son suficientes para WebSphere eXtreme Scale.

Requisitos de sistema operativo

- **Sin la consola web**

  eXtreme Scale no requiere un nivel específico de sistema operativo. Cada implementación de Java SE y Java EE requiere niveles o arreglos distintos de sistema operativo para problemas que se han descubierto durante la comprobación de la implementación de Java. Los niveles necesarios para estas implementaciones son suficientes para eXtreme Scale.
- **Con la consola web**

  Los requisitos siguientes se aplican a cada sistema operativo si se utiliza la consola:
  - Linux: JVM de 32 bits o 64 bits
  - Linux PPC: sólo JVM de 32 bits
  - Windows: sólo JVM de 32 bits
  - AIX: sólo JVM de 32 bits

Requisitos del navegador web

La consola web da soporte a los siguientes navegadores web:

- Mozilla Firefox, versión 3.5.x y posteriores
- Mozilla Firefox, versión 3.6.x y posteriores
- Microsoft Internet Explorer, versión 7 o 8

Requisitos de WebSphere Application Server

- WebSphere Application Server Versión 6.1.0.39 o posterior
- WebSphere Application Server Versión 7.0.0.19 o posterior
- WebSphere Application Server Versión 8.0.0.1 o posterior

Consulte los Arreglos recomendados para WebSphere Application Server si desea más información.

Requisitos de otros servidores de aplicaciones

Otras implementaciones de Java EE pueden utilizar el tiempo de ejecución de eXtreme Scale como una instancia local o como un cliente para los servidores eXtreme Scale. Para implementar Java SE, debe utilizar la versión 5 o posterior.

Consideraciones sobre Java SE

WebSphere eXtreme Scale requires Java SE 5 o posterior. En general, las nuevas versiones de Java SE tienen mejor funcionalidad y rendimiento.

Versiones soportadas

Puede utilizar WebSphere eXtreme Scale conJava SE 5 o posterior. La versión que utilice debe estar soportada actualmente por el proveedor JRE (Java Runtime Environment).

Se instala un JRE completamente soportado como parte de las instalaciones autónomas de WebSphere eXtreme Scale y WebSphere eXtreme Scale Client en el directorio *raíz_intal_wxs*/java y está disponible para ser utilizado por clientes y servidores. Si está instalando WebSphere eXtreme Scale en WebSphere Application Server, puede utilizar el JRE incluido en la instalación de WebSphere Application Server.

WebSphere eXtreme Scale aprovecha la función de Java Development Kit (JDK) 5 o posterior cuando pasa a estar disponible. Normalmente, las versiones más nuevas de Java Development Kit (JDK) y Java SE tiene mejor rendimiento y funcionalidad.

Consulte Software soportado para obtener más información.

Características dependientes de Java de WebSphere eXtreme Scale

Tabla 4. Características que requieren Java SE 5 o Java SE 6.

WebSphere eXtreme Scale utiliza una funcionalidad que se ha introducido en Java SE 5 o Java SE 6 para proporcionar las siguientes características del producto.

Característica	Soportada en Java SE 5 y posterior	Soportada en Java SE 6 y posterior
Anotaciones de la API EntityManager (Opcional: también puede utilizar archivos XML)	X	X
Java Persistence API (JPA): cargador JPA, cargador de clientes JPA y actualizador basado en tiempo JPA	X	X
Desalojo basado en memoria (utiliza MemoryPoolMXBean)	X	X
Agentes de instrumentación: • `wxssizeagent.jar`: aumenta la precisión de las métricas de correlaciones de bytes utilizadas. • `ogagent.jar`: aumenta el rendimiento de las entidades de acceso a campos.	X	X
Consola web para la supervisión		X

Consideraciones sobre Java EE

Mientras se prepara para integrar WebSphere eXtreme Scale en un entorno Java Platform, Enterprise Edition, debe tener en cuenta ciertos elementos, como versiones, opciones de configuración, requisitos y limitaciones y desarrollo y gestión de aplicaciones.

Ejecución de aplicaciones de eXtreme Scale en un entorno Java EE

Una aplicación Java EE puede conectarse a una aplicación de eXtreme Scale remota. Además, el entorno de WebSphere Application Server permite el inicio de un servidor eXtreme Scale mientras se inicia una aplicación en el servidor de aplicaciones.

Si utiliza un archivo XML para crear una instancia de ObjectGrid y el archivo XML está en el módulo del archivador empresarial (EAR), acceda al archivo mediante el método getClass().getClassLoader().getResource("META-INF/objGrid.xml") para obtener un objeto URL y utilizarlo para crear una instancia de ObjectGrid. Substituya el nombre del archivo XML que utilice en la llamada de método.

Puede utilizar beans de arranque para que una aplicación cree una rutina de carga para una instancia de ObjectGrid cuando una aplicación se inicie y para que destruya la instancia de ObjectGrid al detenerse la aplicación. Un bean de arranque es un bean de sesión sin estado con una ubicación remota com.ibm.websphere.startupservice.AppStartUpHome y una interfaz remota com.ibm.websphere.startupservice.AppStartUp. La interfaz remota tiene dos métodos: el método start y el método stop. Utilice el método start para crear una rutina de carga de la instancia y utilice el método stop para destruir la instancia. La aplicación utiliza el método ObjectGridManager.getObjectGrid para mantener una referencia a la instancia. Consulte la información sobre cómo acceder a un ObjectGrid con ObjectGridManager en la *Guía de programación* para obtener más información.

Uso de cargadores de clases

Cuando los módulos de aplicación que utilizan cargadores de clases diferentes comparten una sola instancia de ObjectGrid en una aplicación Java EE, compruebe que los objetos que se almacenan en eXtreme Scale y los plug-ins para el producto están en un cargador común en la aplicación.

Gestión del ciclo de vida de las instancias de ObjectGrid en un servlet

Para gestionar el ciclo de vida de una instancia de ObjectGrid en un servlet, puede utilizar el método init para crear la instancia y el método destroy para eliminar la instancia. Si la instancia se almacena en memoria caché, se recupera y manipula en el código del servlet. Consulte la información sobre cómo acceder a un ObjectGrid con la interfaz ObjectGridManager en la *Guía de programación* para obtener más información.

Convenios de directorio

Se utilizan los siguientes convenios de directorio en toda la documentación para hacer referencia a directorios como por ejemplo *raíz_instalación_wxs* e *inicio_wxs*. Accede a estos directorios durante distintos escenarios, incluido durante la instalación y la utilización de las herramientas de línea de mandatos.

raíz_intal_wxs
El directorio *raíz_instalación_wxs* es el directorio raíz donde se instalan los archivos del producto WebSphere eXtreme Scale. El directorio *raíz_instalación_wxs* puede ser el directorio en el que se extrae el archivado de prueba o el directorio en el que se instala el producto WebSphere eXtreme Scale.

- Ejemplo al extraer la prueba:

  Ejemplo: `/opt/IBM/WebSphere/eXtremeScale`
- Ejemplo cuando se instala WebSphere eXtreme Scale en un directorio autónomo:

  Ejemplo: `/opt/IBM/eXtremeScale`
- Ejemplo cuando se integra WebSphere eXtreme Scale con WebSphere Application Server:

  Ejemplo: `/opt/IBM/WebSphere/AppServer`

inicio_wxs
El directorio *inicio_wxs* es el directorio raíz de los componentes, ejemplos y bibliotecas del producto WebSphere eXtreme Scale. Este directorio es el mismo que el directorio *raíz_instalación_wxs* cuando se ha extraído la versión de prueba. Para instalaciones autónomas, el directorio *inicio_wxs* es el subdirectorio `ObjectGrid` del directorio *raíz_instalación_wxs*. Para instalaciones integradas con WebSphere Application Server, este directorio es el directorio `optionalLibraries/ObjectGrid` del directorio *raíz_instalación_wxs*.

- Ejemplo al extraer la prueba:

  Ejemplo: `/opt/IBM/WebSphere/eXtremeScale`
- Ejemplo cuando se instala WebSphere eXtreme Scale en un directorio autónomo:

  Ejemplo: `/opt/IBM/eXtremeScale/ObjectGrid`
- Ejemplo cuando se integra WebSphere eXtreme Scale con WebSphere Application Server:

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

raíz_was
El directorio *raíz_was* es el directorio raíz de una instalación de WebSphere Application Server:

Ejemplo: `/opt/IBM/WebSphere/AppServer`

inicio_servicioRest
El directorio *inicio_servicioRest* es el directorio en el que se encuentran las bibliotecas y los ejemplos del servicio de datos REST de WebSphere eXtreme Scale. Este directorio se denomina `restservice` y es un subdirectorio del directorio *inicio_wxs*.

- Ejemplo para despliegues autónomos:

  Ejemplo: `/opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice`
- Ejemplo para despliegues integrados de WebSphere Application Server:

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

raíz_tomcat
raíz_tomcat es el directorio raíz de la instalación de Apache Tomcat.

Ejemplo: `/opt/tomcat5.5`

raíz_wasce
raíz_wasce es el directorio raíz de la instalación de WebSphere Application Server Community Edition.

Ejemplo: `/opt/IBM/WebSphere/AppServerCE`

inicio_java
inicio_java es el directorio raíz de una instalación de Java Runtime Environment (JRE).

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

inicio_samples
inicio_samples es el directorio en el que extrae los archivos de ejemplo que se utilizan para las guías de aprendizaje.

Ejemplo: `/wxs-samples/`

raíz_dvd
El directorio *raíz_dvd* es el directorio raíz del DVD que contiene el producto.

Ejemplo: `raíz_dvd/docs/`

raíz_equinox
El directorio *raíz_equinox* es el directorio raíz de la instalación de infraestructura OSGi de Eclipse Equinox.

Ejemplo:`/opt/equinox`

inicio_usuario
El directorio *inicio_usuario* es la ubicación donde se almacenan los archivos de usuario, por ejemplo los perfiles de seguridad.

Windows `c:\Documents and Settings\nombre_usuario`

UNIX `/home/nombre_usuario`

Instalación de WebSphere eXtreme Scale con el asistente de instalación

Puede utilizar el asistente de instalación para instalar WebSphere eXtreme Scale para configuraciones autónomas o de WebSphere Application Server.

Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server

Puede instalar WebSphere eXtreme Scale o WebSphere eXtreme Scale Client en un entorno en el que esté instalado WebSphere Application Server o WebSphere Application Server Network Deployment. Puede utilizar las características existentes de WebSphere Application Server o WebSphere Application Server Network Deployment para mejorar sus aplicaciones de eXtreme Scale.

Antes de empezar

- Instale WebSphere Application Server o WebSphere Application Server Network Deployment. Consulte Instalación del entorno de servicio de aplicaciones si desea más información.
- En función de la versión que instale, la versión 6.1 o la versión 7.0, aplique el fixpack más reciente para WebSphere Application Server o WebSphere Application Server Network Deployment para actualizar el nivel de producto. Consulte los Fixpacks más recientes de WebSphere Application Server para obtener más información.

- Verifique que el directorio de instalación de destino no contenga una instalación existente de WebSphere eXtreme Scale ni WebSphere eXtreme Scale Client.
- Detenga todos los procesos que se están ejecutando en el entorno WebSphere Application Server o WebSphere Application Server Network Deployment. Consulte Programas de utilidad de línea de mandatos para obtener más información sobre los mandatos **stopManager**, **stopNode** y **stopServer**.

  PRECAUCIÓN:
  asegúrese de que todos los procesos en ejecución se hayan detenido. Si no se detienen los procesos en ejecución, la instalación sigue, creando resultados imprevisibles y dejando la instalación en un estado indeterminado en algunas plataformas.
- Si está instalando sólo el cliente, puede utilizar el DVD para instalar el cliente o descargar WebSphere eXtreme Scale Client para la plataforma específica de la sección descargas en el Sitio de soporte.

Importante: Cuando instale WebSphere eXtreme Scale o WebSphere eXtreme Scale Client, deberá estar en el mismo directorio en el que haya instalado WebSphere Application Server. Por ejemplo, si instaló WebSphere Application Server en `C:\`*raíz_was*, también debería elegir `C:`*raíz_was* como el directorio destino para la instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client.

Acerca de esta tarea

Integre eXtreme Scale con WebSphere Application Server o WebSphere Application Server Network Deployment para aplicar las características de eXtreme Scale a las aplicaciones Java Platform, Enterprise Edition. Las aplicaciones Java EE alojan cuadrículas de datos y acceden a las cuadrículas de datos utilizando una conexión de cliente.

Procedimiento

1. Utilice el asistente para completar la instalación.
   - Ejecute el siguiente script para iniciar el asistente de la instalación completa de WebSphere eXtreme Scale. Puede optar por instalar solo el cliente o el cliente y el servidor:
     - Linux UNIX *raíz_dvd*`/install`
     - Windows *raíz_dvd*`\install.bat`
   - Ejecute el siguiente script para iniciar el asistente de la instalación de WebSphere eXtreme Scale Client. Los archivos de instalación están en el archivo zip que ha descargado de la sección de descargas del Sitio de soporte:
     - Linux UNIX `root/WXS_Client/install`
     - Windows `root\WXS_Client\install.bat`

     Atención: Si utiliza el convenio de denominación universal (UNC) para identificar vías de acceso en el mandato de instalación, es posible que los elementos cuya instalación anticipa no se instalen una vez ejecutado el mandato. Para evitar problemas, correlacione la vía de acceso del archivo con una unidad de red. Ejecute el mandato **install** contra la unidad correlacionada. La utilización de una unidad de red correlacionada permite asegurarse de que se instalan todos los elementos.
2. Siga las indicaciones del asistente.

El panel de características opcionales lista las características que puede optar por instalar. Sin embargo, las características no se pueden añadir de forma incremental en el entorno del producto después de que se instale el producto. Si elige no instalar una característica con la instalación inicial del producto, deberá desinstalar y volver a instalar el producto para añadir la característica.

El panel de aumento de perfil lista los perfiles existentes que puede seleccionar para aumentar con las características de eXtreme Scale. Sin embargo, si selecciona perfiles que ya están siendo utilizados, se visualiza un panel de aviso. Para continuar con la instalación, detenga los servidores que están configurados en los perfiles, o bien pulse **Atrás** para eliminar los perfiles de la selección.

Resultados

Windows Si va a instalar WebSphere eXtreme Scale Client en Windows, es posible que aparezca el texto siguiente en los resultados de la instalación:

```
Finalizado con éxito: Se ha instalado de forma satisfactoria el siguiente
producto:
cliente de WebSphere eXtreme Scale.  Algunos pasos de la configuración
han dado errores.
Consulte el siguiente archivo de anotaciones cronológicas para obtener más
información:
<raíz de instalación de WebSphere Application Server>\logs\wxs_client\install\log.txt"
Revise el archivo de registro de instalación (log.txt) y revise el
registro de aumento del gestor de despliegue.
```

Si aparece una anomalía en el archivo `iscdeploy.sh`, puede pasar por alto el error. Este error no causa problemas.

Qué hacer a continuación

- Si ejecuta WebSphere Application Server versión 6.1 o versión 7.0, puede utilizar el plug-in de herramienta de gestión de perfiles o el mandato **`manageprofiles`**. Para obtener más información, consulte “Creación y aumento de perfiles para WebSphere eXtreme Scale” en la página 184.
- Verifique la instalación. Para obtener más información, consulte “Verificación de la instalación” en la página 206.
- Empiece configurando la instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client. Para obtener más información, consulte “Primeros pasos después de la instalación” en la página 207.

Archivos de ejecución de WebSphere eXtreme Scale integrado con WebSphere Application Server

Los archivos de archivado Java (JAR) se incluyen en la instalación. Puede utilizar los archivos JAR incluidos y la ubicación en la que se instalan.

Tabla 5. Archivos de tiempo de ejecución de WebSphere eXtreme Scale. La siguiente tabla lista los archivos JAR (Java Archive) que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
`wxsdynacache.jar`	Cliente y servidor	`lib`	El archivo `wxsdynacache.jar` contiene las clases necesarias para utilizar con el proveedor de la memoria caché dinámica.
`wsobjectgrid.jar`	Local y cliente	`lib`	El archivo `wsobjectgrid.jar` contiene los tiempos de ejecución de eXtreme Scale local, cliente y servidor.

Tabla 5. Archivos de tiempo de ejecución de WebSphere eXtreme Scale (continuación). La siguiente tabla lista los archivos JAR (Java Archive) que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
ogagent.jar	Local, cliente y servidor	lib	El archivo ogagent.jar contiene las clases de tiempo de ejecución necesarias para ejecutar el agente de instrumentación Java que se utiliza con la API EntityManager.
ogsip.jar	Servidor	lib	El archivo ogsip.jar contiene el tiempo de ejecución de la gestión de sesiones SIP (Scale Session Initiation Protocol) de eXtreme que es compatible con WebSphere Application Server Versión 6.1.x.
sessionobjectgrid.jar	Cliente y servidor	lib	El archivo sessionobjectgrid.jar contiene el tiempo de ejecución de gestión de sesiones HTTP de eXtreme Scale.
sessionobjectgridsip.jar	Servidor	lib	El archivo sessionobjectgridsip.jar contiene el tiempo de ejecución de gestión de sesiones SIP de eXtreme Scale que es compatible con WebSphere Application Server Versión 7.x.
wsogclient.jar	Local y cliente	lib	El archivo wsogclient.jar instalado cuando se utiliza un entorno que contiene WebSphere Application Server versión 6.0.2 y posterior. Este archivo sólo contiene los entornos de ejecución local y de cliente.
wxssizeagent.jar	Local, cliente y servidor	lib	El archivo wxssizeagent.jar se utiliza para proporcionar información sobre dimensionamiento de entradas de memoria caché más precisa al utilizar el entorno de tiempo de ejecución Java (JRE) Versión 1.5 o posterior.
oghibernate-cache.jar	Cliente y servidor	optionalLibraries/ObjectGrid	El archivo oghibernate-cache.jar contiene el plug-in de memoria caché de eXtreme Scale de nivel 2 para JBoss Hibernate.
ogspring.jar	Local, cliente y servidor	optionalLibraries/ObjectGrid	El archivo ogspring.jar contiene clases de soporte para la integración de la infraestructura de Spring SpringSource.
xsadmin.jar	programa de utilidad	optionalLibraries/ObjectGrid	El archivo xsadmin.jar contiene el programa de utilidad de ejemplo de administración de eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Cliente y servidor	optionalLibraries/ObjectGrid/endorsed	Este conjunto de archivos incluye un módulo de tiempo de ejecución de intermediario de solicitud de objetos (ORB) que se utiliza para ejecutar las aplicaciones en los procesos Java SE.
wxshyperic.jar	programa de utilidad	optionalLibraries/ObjectGrid/hyperic/lib	El plug-in de detección de servidor de WebSphere eXtreme Scale para el agente de supervisión SpringSource Hyperic.
restservice.ear	Cliente	optionalLibraries/ObjectGrid/restservice/lib	El archivo restservice.ear contiene el archivador empresarial de la aplicación de servicio de datos REST de eXtreme Scale para los entornos de WebSphere Application Server.
restservice.war	Cliente	optionalLibraries/ObjectGrid/restservice/lib	El archivo restservice.war contiene el archivo web del servicio de datos REST de eXtreme Scale para los servidores de aplicaciones adquiridos de otro proveedor.
splicerlistener.jar	programa de utilidad	optionalLibraries/ObjectGrid/session/lib	El archivo splicerlistener.jar contiene el programa de utilidad splicer para el filtro del gestor de sesiones HTTP de eXtreme Scale.
splicer.jar	programa de utilidad	optionalLibraries/ObjectGrid/legacy/session/lib	El archivo splicer.jar contiene el programa de utilidad splicer Versión 7.0 para el filtro del gestor de sesiones HTTP de eXtreme Scale.

Tabla 6. Archivos de tiempo de ejecución de WebSphere eXtreme Scale Client. La siguiente tabla lista los archivos JAR (Java Archive) que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
wxsdynacache.jar	Cliente y servidor	lib	El archivo wxsdynacache.jar contiene las clases necesarias para utilizar con el proveedor de la memoria caché dinámica.
ogagent.jar	Local, cliente y servidor	lib	El archivo ogagent.jar contiene las clases de tiempo de ejecución necesarias para ejecutar el agente de instrumentación Java que se utiliza con la API EntityManager.
ogsip.jar	Servidor	lib	El archivo ogsip.jar contiene el tiempo de ejecución de la gestión de sesiones SIP (Scale Session Initiation Protocol) de eXtreme que es compatible con WebSphere Application Server Versión 6.1.x.
sessionobjectgrid.jar	Cliente y servidor	lib	El archivo sessionobjectgrid.jar contiene el tiempo de ejecución de gestión de sesiones HTTP de eXtreme Scale.
sessionobjectgridsip.jar	Servidor	lib	El archivo sessionobjectgridsip.jar contiene el tiempo de ejecución de gestión de sesiones SIP de eXtreme Scale que es compatible con WebSphere Application Server Versión 7.x.
wsogclient.jar	Local y cliente	lib	El archivo wsogclient.jar instalado cuando se utiliza un entorno que contiene WebSphere Application Server versión 6.0.2 y posterior. Este archivo sólo contiene los entornos de ejecución local y de cliente.
wxssizeagent.jar	Local, cliente y servidor	lib	El archivo wxssizeagent.jar se utiliza para proporcionar información sobre dimensionamiento de entradas de memoria caché más precisa al utilizar el entorno de tiempo de ejecución Java (JRE) Versión 1.5 o posterior.
oghibernate-cache.jar	Cliente y servidor	optionalLibraries/ObjectGrid	El archivo oghibernate-cache.jar contiene el plug-in de memoria caché de eXtreme Scale de nivel 2 para JBoss Hibernate.
ogspring.jar	Local, cliente y servidor	optionalLibraries/ObjectGrid	El archivo ogspring.jar contiene clases de soporte para la integarción de la infraestructura de Spring SpringSource.
xsadmin.jar	programa de utilidad	optionalLibraries/ObjectGrid	El archivo xsadmin.jar contiene el programa de utilidad de ejemplo de administración de eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Cliente y servidor	optionalLibraries/ObjectGrid/endorsed	Este conjunto de archivos incluye un módulo de tiempo de ejecución de intermediario de solicitud de objetos (ORB) que se utiliza para ejecutar las aplicaciones en los procesos Java SE.
wxshyperic.jar	programa de utilidad	optionalLibraries/ObjectGrid/hyperic/lib	El plug-in de detección de servidor de WebSphere eXtreme Scale para el agente de supervisión SpringSource Hyperic.
restservice.ear	Cliente	optionalLibraries/ObjectGrid/restservice/lib	El archivo restservice.ear contiene el archivador empresarial de la aplicación de servicio de datos REST de eXtreme Scale para los entornos de WebSphere Application Server.
restservice.war	Cliente	optionalLibraries/ObjectGrid/restservice/lib	El archivo restservice.war contiene el archivo web del servicio de datos REST de eXtreme Scale para los servidores de aplicaciones adquiridos de otro proveedor.
splicerlistener.jar	programa de utilidad	optionalLibraries/ObjectGrid/session/lib	El archivo splicerlistener.jar contiene el programa de utilidad splicer para el filtro del gestor de sesiones HTTP de eXtreme Scale.
splicer.jar	programa de utilidad	optionalLibraries/ObjectGrid/legacy/session/lib	El archivo splicer.jar contiene el programa de utilidad splicer Versión 7.0 para el filtro del gestor de sesiones HTTP de eXtreme Scale.

Utilización del plug-in Installation Factory para crear e instalar paquetes personalizados

Utilice el plug-in IBM® Installation Factory para WebSphere eXtreme Scale para crear un paquete de instalación personalizado (CIP) o un paquete de instalación integrado (IIP). Un CIP contiene un único paquete de instalación y varios activos opcionales. Un IIP combina una o más paquetes de instalación en un único flujo de trabajo de instalación que diseñe.

Antes de empezar

Antes de crear e instalar paquetes personalizados para eXtreme Scale, en primer lugar, debe descargar los siguientes productos:

- IBM Installation Factory para WebSphere Application Server
- Plug-in IBM Installation Factory para WebSphere eXtreme Scale

Acerca de esta tarea

Mediante Installation Factory, puede crear un CIP combinando un único componente de producto con paquetes de mantenimiento, scripts de personalización y otros archivos. Cuando cree un IIP, agregue componentes individuales o paquetes de instalación en un único paquete de instalación.

Archivo de definición de build:

Un archivo de definición de build es un documento XML que especifica cómo crear e instalar un paquete de instalación personalizado (CIP) o un paquete de instalación integrado (IIP). IBM Installation Factory for WebSphere eXtreme Scale lee los detalles del paquete del archivo de definición de build para generar un CIP o un IIP.

Antes de poder crear un CIP o un IIP, debe crear un archivo de definición de build para cada paquete personalizado. El archivo de definición de build describe qué componentes de producto o paquetes de instalación para instalar, la ubicación del CIP o el IIP, los paquetes de mantenimiento para incluir, los scripts de instalación y otros archivos que elija incluir. También puede especificar en el archivo de definición de build para el IIP el orden en el que Installation Factory instalar cada paquete de instalación.

El asistente Definición de build le guía a través del proceso de crear un archivo de definición de build. También puede utilizar el asistente para modificar un archivo de definición de build existente. Cada panel del asistente Definición de build le solicita información sobre un paquete personalizado como, por ejemplo, la identificación del paquete, la ubicación de instalación para la definición del build y la ubicación de instalación para el paquete personalizado. Toda esta información se guarda en el nuevo archivo de definición de build, o se modifican o guardan en un archivo de definición de build existente. Si desea más información, consulte los Paneles del asistente Definición del build del CIP y los Paneles del asistente de definición del build del IIP.

Para crear sólo el archivo de definición de build, puede utilizar la herramienta de la interfaz de línea de mandatos para generar el paquete personalizado fuera de la GUI. Si desea más información, consulte “Instalación silenciosa de un CIP o un IIP” en la página 175.

Creación y generación de archivo y generación de un CIP:

El plug-in IBM Installation Factory para WebSphere eXtreme Scale genera un paquete de instalación personalizado (CIP) de acuerdo con los detalles que especifique en el archivo de definición de build. La definición de build especifica el paquete de producto para instalar, la ubicación del CIP, los paquetes de mantenimiento para incluir en la instalación, los archivos de script de instalación y cualquier archivo adicional para incluir en el CIP.

Acerca de esta tarea

Puede utilizar el asistente Definición de build para crear un archivo de definición de build y generar un CIP.

Procedimiento

1. Ejecute el siguiente script desde el directorio *IF_HOME*/bin para iniciar Installation Factory:
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`

   Pulse el icono **Nueva definición de build**.
2. Seleccione el producto para incluir en el archivo de definición de build y pulse **Finalizar** para iniciar el asistente Definición de build.
3. Siga las indicaciones del asistente.

   En el panel Instalar y desinstalar scripts, pulse **Añadir scripts...** para llenar la tabla con ningún script de instalación personalizado. Escriba la ubicación de los archivos de script y desactive el recuadro de selección para continuar si se visualiza un mensaje de error. La operación se detiene de forma predeterminada. Pulse **Aceptar** para volver al panel.

Resultados

Ha creado y personalizado el archivo de definición de build, y ha generado el CIP si ha elegido trabajar en la modalidad conectada.

Si el asistente Definición de build no le proporciona la opción para generar el CIP a partir del archivo de definición de build, podrá seguir generándolo ejecutando el script `ifcli.sh|bat` desde el directorio *IF_HOME*/bin.

Qué hacer a continuación

Instale el CIP. Si desea más información, consulte “Instalación de un CIP”.

Instalación de un CIP:

Simplifique el proceso de instalación del producto instalando un paquete de instalación personalizado (CIP). Un CIP es una imagen de instalación de producto única que puede incluir uno o más paquetes de mantenimiento, scripts de configuración y otros archivos.

Antes de empezar

Antes de poder instalar un CIP, debe crear un archivo de definición de build para especificar qué opciones incluir en el CIP. Si desea más información, consulte “Creación y generación de archivo y generación de un CIP” en la página 168.

Acerca de esta tarea

Un CIP combina e instala un único componente de producto con paquetes de mantenimiento, scripts de personalización y otros archivos.

Procedimiento

1. Detenga todos los procesos que se ejecutan en la estación de trabajo que está preparando para la instalación. Para detener el gestor de despliegue, ejecute el siguiente script:
   - Linux UNIX *raíz_perfil*/bin/stopManager.sh
   - Windows *raíz_perfil*\bin\stopManager.bat

   Para detener los nodos, ejecute el siguiente script:
   - Linux UNIX *raíz_perfil*/bin/stopNode.sh
   - Windows *raíz_perfil*\bin\stopNode.bat
2. Ejecute el siguiente script para iniciar la instalación:
   - Linux UNIX *inicio_CIP*/bin/install
   - Windows *inicio_CIP*\bin\install.bat
3. Siga las indicaciones del asistente para completar la instalación.

   El panel de características opcionales lista las características que puede optar por instalar. Sin embargo, las características no se pueden añadir de forma incremental en el entorno del producto después de que se instale el producto. Si elige no instalar una característica con la instalación inicial del producto, deberá desinstalar y volver a instalar el producto para añadir la característica.

   El panel de aumento de perfil lista los perfiles existentes que puede seleccionar para aumentar con las características de eXtreme Scale. Sin embargo, si selecciona perfiles que ya están siendo utilizados, se visualiza un panel de aviso. Para continuar con la instalación, detenga los servidores que están configurados en los perfiles, o bien pulse **Atrás** para eliminar los perfiles de la selección.

Resultados

Ha instalado correctamente el CIP.

Qué hacer a continuación

Si ejecute WebSphere Application Server versión 6.1 o versión 7.0, puede utilizar el plug-in de Herramienta de gestión de perfiles o el mandato **manageprofiles** para crear y aumentar perfiles. Si desea más información, consulte “Creación y aumento de perfiles para WebSphere eXtreme Scale” en la página 184.

Si ha aumentado los perfiles para eXtreme Scale durante el proceso de instalación, puede desplegar aplicaciones, iniciar un servicio de catálogo e iniciar los contenedores en el entorno de WebSphere Application Server. Si desea más información, consulte “Configuración de WebSphere eXtreme Scale con WebSphere Application Server” en la página 256.

Instalación de un CIP para aplicar el mantenimiento a una instalación del producto existente:

Puede aplicar paquetes de mantenimiento a una instalación de producto existente instalando un paquete de instalación personalizado (CIP). Normalmente se hace referencia al proceso de aplicar el mantenimiento a una instalación existente con un CIP como *instalación de SLIP*.

Antes de empezar

Cree un archivo de definición de build para especificar qué opciones incluir en el CIP. Si desea más información, consulte "Creación y generación de archivo y generación de un CIP" en la página 168.

Acerca de esta tarea

Cuando se aplica el mantenimiento con un CIP que contiene un paquete de renovación, un fixpack, o ambos, el asistente desinstala todos los informes autorizados de análisis de programa (APAR) instalados previamente. Si el CIP está en el mismo nivel que el producto, los APAR instalados previamente se conservan sólo si se han empaquetado en el CIP. Para aplicar correctamente el mantenimiento a una instalación existente, debe incluir las características instaladas en el CIP.

Procedimiento

1. Detenga todos los procesos que se ejecutan en la estación de trabajo que está preparando para la instalación. Para detener el gestor de despliegue, ejecute el siguiente script:
   - Linux UNIX *raíz_perfil*/bin/stopManager.sh
   - Windows *raíz_perfil*\bin\stopManager.bat

   Para detener los nodos, ejecute el siguiente script:
   - Linux UNIX *raíz_perfil*\bin\stopNode.sh
   - Windows *raíz_perfil*\bin\stopNode.bat
2. Ejecute el siguiente script para iniciar la instalación:
   - Linux UNIX *inicio_CIP*/bin/install
   - Windows *inicio_CIP*\bin\install.bat
3. Siga las indicaciones del asistente para completar la instalación.

   El resumen de la vista previa de instalación lista la versión de producto resultante y las características y los arreglos temporales aplicables. A continuación, el asistente aplica correctamente el mantenimiento y actualiza la características del producto.

Resultados

Los archivos binarios del producto se copian en el directorio *raíz_was*/properties/version/nif/backup. Puede utilizar el IBM Update Installer para desinstalar la actualización y restaurar la estación de trabajo. Si desea más información, consulte "Desinstalación de actualizaciones del CIP de una instalación de producto existente".

Desinstalación de actualizaciones del CIP de una instalación de producto existente:

Puede eliminar las actualizaciones del CIP de una instalación de producto existente sin eliminar todo el producto. Utilice IBM Update Installer versión 7.0.0.4 para desinstalar cualquier actualización. También se hace referencia a esta tarea como *desinstalación de SLIP*.

Antes de empezar

Debe tener, como mínimo, una copia existente del producto instalado en el sistema.

Procedimiento

1. Descargue la versión 7.0.0.4 del instalador de actualización desde el siguiente sitio FTP:

   ftp://ftp.software.ibm.com/software/websphere/cw/process_server/FEP/UPDI/7004
2. Instale el instalador de actualización. Consulte Instalación del instalador de actualización para el software WebSphere en el centro de información de WebSphere Application Server si desea más información.
3. Desinstale los fixpacks, paquetes de renovación o arreglos temporales que ha añadido en el entorno después de haber instalado el CIP.
4. Desinstale los arreglos temporales que ha incluido en la instalación de SLIP. Este proceso es el mismo que la desinstalación de un único fixpack o paquete de renovación. Sin embargo, el mantenimiento que se incluyó en el CIP ahora se incluye en una sola operación.
5. Desinstale el CIP utilizando el instalador de actualización. Los niveles de mantenimiento lo devuelven al estado previo de la actualización y el CIP se denota a través del identificador del CIP que se añade como prefijo a su nombre de archivo. El siguiente ejemplo muestra cómo se visualiza un CIP de forma diferente que los otros paquetes de mantenimiento regulares en el panel de selección del paquete de mantenimiento:

   CIP

   `com.ibm.ws.cip.7000.wxs.primary.ext.pak`

Resultados

Ha eliminado correctamente las actualizaciones del CIP de una instalación de producto existente.

Creación de un archivo de definición de build y generación de un IIP:

El plug-in IBM Installation Factory para WebSphere eXtreme Scale genera un IIP basado en las propiedades que proporciona el archivo de definición de build. El archivo de definición de build contiene información como, por ejemplo, qué paquetes de instalación incluir en el IIP, el orden en el que Installation Factory instala cada paquete y la ubicación del IIP.

Acerca de esta tarea

Puede utilizar el asistente Definición de build para crear un archivo de definición de build y generar un IIP.

Procedimiento

1. Ejecute el siguiente script desde el directorio *IF_HOME*`/bin` para iniciar Installation Factory:
   - UNIX Linux `ifgui.sh`
   - Windows `ifgui.bat`
2. Pulse el icono **Crear nuevo paquete de instalación integrado** para iniciar el asistente Definición de build.

3. Siga las indicaciones del asistente.
   a. En el panel Construir el IIP, seleccione un paquete de instalación soportado en la lista y pulse **Añadir instalador** para añadir el paquete de instalación al IIP. Se visualiza un panel que muestra el nombre de paquete, el identificador del paquete y las propiedades del paquete. Para ver información específica sobre el paquete seleccionado, pulse **Ver información de paquete de instalación**. Pulse **Modificar** para especificar la vía de acceso del directorio al paquete de instalación para cada sistema operativo. Si está añadiendo actualmente un paquete de instalación para WebSphere Extended Deployment, active el recuadro de selección, que le proporciona la opción de utilizar el mismo paquete para todos los sistemas operativos soportados. Pulse **Aceptar** y vuelva al panel Construir el IIP. Se crea una invocación de forma predeterminada.
      - Para modificar la vía de acceso del directorio de un paquete de instalación, seleccione el paquete en los paquetes de instalación utilizados en la lista de IIP y pulse **Modificar**.
      - Para modificar una invocación, selecciónela y pulse **Modificar**. Especifique la ubicación de instalación predeterminada para la invocación en cada sistema operativo. Especifique la ubicación del archivo de respuestas si selecciona una instalación silenciosa como la modalidad de instalación predeterminada.
      - Pulse **Añadir invocación** para añadir una contribución de invocación al paquete de instalación. Se visualiza un panel desde el que puede especificar las propiedades para la invocación.
      - Pulse **Eliminar** para eliminar los paquetes de instalación o las invocaciones.
4. Revise el resumen de las selecciones, seleccione la opción **Guardar archivo de definición de build y generar el paquete de instalación integrado** y pulse **Finalizar**.

   De forma alternativa, puede guardar el archivo de definición de build sin generar el IIP. Con esta opción, genera realmente el IIP fuera del asistente ejecutando el script `ifcli.bat | ifcli.sh` desde el directorio *inicio_IF*/bin/.

Resultados

Ha creado y personalizado el archivo de definición de build para un IIP.

Qué hacer a continuación

Instale el IIP.

Instalación de un IIP:

Utilice el plug-in IBM Installation Factory para WebSphere eXtreme Scale para instalar un paquete de instalación integrado (IIP). Un IIP combina uno o más paquetes de instalación en una flujo de trabajo único que diseñe.

Antes de empezar

Antes de poder instalar un CIP, debe crear un archivo de definición de build para especificar qué opciones incluir en el CIP. Si desea más información, consulte "Creación de un archivo de definición de build y generación de un IIP" en la página 172.

Acerca de esta tarea

Un IIP puede incluir uno o más paquetes de instalación disponibles de forma general, uno o más CIP y otros archivos y directorios opcionales. Mediante la instalación de un IIP, se agregan varios paquetes de instalación, o *contribuciones*, en un único paquete y, a continuación, se instalan las contribuciones en un orden específico para completar una instalación completa.

Procedimiento

1. Ejecute el siguiente script para iniciar el asistente:
   - Linux UNIX *inicio_IIP*/bin/install
   - Windows *inicio_IIP*\bin\install.bat
2. Pulse **Acerca de** en el panel de bienvenida para ver los detalles del IIP como, por ejemplo, el identificador del paquete, los sistemas operativos soportados y los paquetes de instalación incluidos.

   Opcional: Para modificar las opciones de instalación para cada paquete, pulse **Modificar**.

   Opcional: Se visualizan dos botones **Ver registro** en el panel del asistente. Para ver el registro de cada paquete, pulse el botón **Ver registro** que se visualiza junto a la lista que lista los paquetes de instalación. Para ver los detalles generales del registro del IIP, pulse el botón **Ver registro** que se visualiza junto a la información del estado.
3. Seleccione los paquetes de instalación para ejecutar y pulse **Instalar**. Se visualiza una lista de todas las contribuciones en el orden de invocación que contiene el IIP. Para designar qué invocaciones de contribución no se deben ejecutar durante la instalación, desactive el recuadro de selección situado junto al campo **Nombre de instalación**.

Resultados

Ha instalado correctamente un IIP.

Modificación de un archivo de definición de build existente para un IIP:

Puede editar o añadir las propiedades de un IIP para personalizar de forma adicional la instalación.

Acerca de esta tarea

Para cambiar las propiedades de un IIP, modifique el archivo de definición de build existente.

Procedimiento

1. Ejecute el siguiente script desde el directorio *IF_HOME*/bin para iniciar Installation Factory:
   - UNIX Linux ifgui.sh
   - Windows ifgui.bat
2. Pulse el icono **Abrir definición de build** y seleccione el archivo de definición de build que desee modificar.
3. Seleccione las propiedades específicas del IIP que desee modificar. La siguiente lista contiene las posibles modificaciones que puede realizar:

- Cambiar la selección de modalidad actual. En la modalidad conectada, crear la definición del build para utilizar y, de forma opcional, generar el IIP, desde la estación de trabajo actual. En la modalidad desconectada, crear el archivo de definición de build para utilizar en otra estación de trabajo.
- Añadir o eliminar los sistemas operativos existentes que soporta el IIP.
- Editar el identificador y la versión existentes para el IIP.
- Editar la ubicación de destino para el archivo de definición de build.
- Editar la ubicación de destino para el IIP.
- Cambiar si se visualiza un asistente de instalación para el IIP. El asistente proporciona información sobre el IIP y las opciones de instalación cuando se ejecuta el IIP.
- Añadir, eliminar y editar los paquetes de instalación que se incluyen en el IIP.

  Importante: Si ha añadido un sistema operativo soportado y no ha actualizado las propiedades del paquete de instalación en el IIP, recibirá un mensaje de aviso que indica que las contribuciones seleccionadas no contienen paquetes de instalación que se hayan identificado para todos los sistemas operativos que soporta el IIP. Pulse **Sí** para continuar, o pulse **No** para editar el paquete de instalación.

4. Revise el resumen de las selecciones, seleccione **Guardar archivo de definición de build y generar paquete de instalación integrado** y pulse **Finalizar**.

Instalación silenciosa de un CIP o un IIP:

Puede instalar de forma silenciosa un paquete de instalación personalizado (CIP) o un paquete de instalación integrado (IIP) para el producto utilizando un archivo de respuestas plenamente cualificado, que configura de forma específica según sus necesidades, o parámetros que pasa a la línea de mandatos.

Antes de empezar

Cree el archivo de definición de build para el CIP o el IIP. Si desea más información, consulte “Creación y generación de archivo y generación de un CIP” en la página 168.

Acerca de esta tarea

Una instalación silenciosa utiliza el mismo programa de instalación que utiliza la versión de la interfaz gráfica de usuario (GUI). Sin embargo, en lugar de visualizar una interfaz de asistente, la instalación silenciosa lee todas las respuestas de un archivo que personaliza, o de los parámetros que pase a la línea de mandatos. Si instala de forma silenciosa un IIP, puede invocar una contribución con una combinación de opciones que especifica directamente en la línea de mandatos, así como las opciones que especifique en un archivo de respuestas. Sin embargo, las opciones de contribución que pase a la línea de mandatos provoca que el instalador del IIP ignore todas las opciones que se especifican en un archivo de respuestas de contribución específica. Consulte Instalación de un IIP de forma silenciosa para obtener más información detallada.

Nota: Debe especificar el nombre de archivo de respuestas completo. La especificación de la vía de acceso relativa provoca que la instalación falles sin ninguna indicación de que se haya producido un error.

Procedimiento

1. Opcional: Si opta por instalar el CIP o IIP utilizando un archivo de respuestas, en primer lugar, personalice el archivo.
   a. Copie el archivo de respuestas, `wxssetup.response.txt`, del DVD del producto en la unidad de disco.
   b. Abra y edite el archivo de respuestas en el editor de texto que elija. El archivo incluye comentarios para ayudar al proceso de configuración y debe incluir estos parámetros:
      - El acuerdo de licencia
      - La ubicación de la instalación del producto

      Consejo: El instalador utiliza la ubicación que seleccione para la instalación para determinar dónde está instalada la instancia de WebSphere Application Server. Si realiza la instalación en un nodo con varias instancias de WebSphere Application Server, defina de forma clara la ubicación.
   c. Ejecute el siguiente script para iniciar el archivo de respuestas personalizado.
      - Linux UNIX `install -options /vía_acceso_absoluto/archivo_respuesta.txt -silent`
      - Windows `install.bat -options C:\vía_acceso_unidad\archivo_respuesta.txt -silent`
2. Opcional: Si opta por instalar el CIP o IIP pasando determinados parámetros a la línea de mandatos, ejecute el siguiente script para iniciar la instalación:
   - Linux UNIX `install -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=ubicación_instalación`
   - Windows `install.bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=ubicación_instalación`

   donde *ubicación_instalación* es la ubicación de la instalación existente de WebSphere Application Server.
3. Revise los registros resultantes para ver los errores o una anomalía de instalación.

Resultados

Ha instalado de forma silenciosa el CIP o IIP.

Qué hacer a continuación

Si ejecute WebSphere Application Server versión 6.1 o versión 7.0, puede utilizar el plug-in de Herramienta de gestión de perfiles o el mandato **manageprofiles** para crear y aumentar perfiles.

Si ha aumentado los perfiles para eXtreme Scale durante el proceso de instalación, puede desplegar aplicaciones, iniciar un servicio de catálogo e iniciar los contenedores en el entorno de WebSphere Application Server. Si desea más información, consulte “Configuración de WebSphere eXtreme Scale con WebSphere Application Server” en la página 256.

Archivo `wxssetup.response.txt`:

Puede utilizar un archivo de respuestas completo para instalarWebSphere eXtreme Scale o WebSphere eXtreme Scale Client de forma silenciosa.

PRECAUCIÓN:
No añada barras finales, como / o \, al final de las vías de acceso de ubicación de la instalación. Estas vías de acceso se especifican con el atributo installLocation. Si se añade una barra al final de la ubicación de la instalación puede provocar un error en la instalación. Por ejemplo, la vía de acceso siguiente provocaría un error en la instalación:

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale/"
```

La vía de acceso se debe especificar como:

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
```

Archivo de respuestas para la instalación completa de WebSphere eXtreme Scale

```
############################################################################
#
# Archivo de opciones de InstallShield de IBM WebSphere eXtreme Scale V7.1.1
#
# Nombre del asistente: Instalación
# Origen del asistente: setup.jar
#
# Este archivo se puede utilizar la Instalación con las opciones especificadas a
# continuación cuando se ejecuta el asistente con la opción de la línea de mandatos
# "-options". Lea la documentación sobre cada valor para obtener información sobre
# cómo cambiar su valor. Incluya todos los valores entre un solo par de comillas
# dobles.
#
# Un uso común de un archivo de opciones es ejecutar el asistente en modalidad
# silenciosa. Esto permite al autor del archivo de opciones especificar valores
# del asistente sin tener que ejecutar el asistente en modalidad gráfica o de
# consola. Para utilizar este archivo de opciones para la ejecución en modalidad
# silenciosa, utilice los siguientes argumentos de la línea de mandatos al
# ejecutar el asistente:
#
#    -options "D:\installImage\WXS\wxssetup.response" -silent
#
# Observe que se debe utilizar el nombre de archivo de respuestas totalmente
# calificado.
#
############################################################################

############################################################################
#
# Aceptación de la licencia
#
# Valores válidos:
#  true - Acepta la licencia. Instalará el producto.
#  false - Rechaza la licencia. La instalación no se llevará a cabo.
#
# Si no se realiza la instalación, este hecho se registrará en un archivo de
# registro temporal en el directorio temporal del usuario.
#
# Al cambiar el valor de la propiedad silentInstallLicenseAcceptance de
# este archivo de respuestas por "true", afirma que ha revisado y acepta
# las condiciones del Acuerdo Internacional de Programas Bajo Licencia
# de IBM que acompaña a este programa, que se encuentra en el archivo
# CD_ROOT\XD\wxs.primary.pak\repository\legal.xs\license.xs. Si no
# acepta estos términos, no cambie el valor ni descargue, instale, use
# o acceda al programa y devuelva rápidamente el programa y la prueba
# de titularidad al vendedor para obtener un reintegro por el importe
# que haya pagado.
#
```

```
-OPT silentInstallLicenseAcceptance="false"


################################################################################
# Comprobación de requisitos previo que no causen bloqueos
#
# Si desea inhabilitar la comprobación de requisitos previos que no causen
# bloqueos, elimine la marca de comentario de la línea siguiente. Esto
# notificará al instalador que debe continuar con la instalación y registrar
# los avisos aunque la comprobación de requisitos previos haya fallado.
#
#-OPT disableNonBlockingPrereqChecking="true"


################################################################################
#
# Ubicación de instalación
#
# La ubicación de instalación del producto. Especifique un directorio válido en el
# que se instalará el producto. Si el directorio contiene espacios, inclúyalos entre
# comillas dobles tal como se muestra en el ejemplo de Windows siguiente. Tenga en
# cuenta que los espacios en la ubicación de instalación sólo se permiten en los
# sistemas operativos Windows. La longitud máxima de la vía de acceso es de 60
# caracteres para Windows.
#
# A continuación está la lista de ubicaciones de instalación predeterminadas
# para cada sistema operativo soportado cuando se instala como usuario root.  De
# forma predeterminada, en este archivo de respuestas, se utiliza la  ubicación
# de instalación de Windows. Si desea utiliza la ubicación de instalación
# predeterminada para otro sistema operativo, quite las marcas de comentario
# de la entrada de la ubicación de instalación predeterminada apropiada
# (eliminando '#') y luego comente (añadiendo '#') la entrada del sistema
# operativo Windows a continuación.
#
# La ubicación de instalación se utiliza para determinar si WebSphere eXtreme
# Scale se debe instalar como despliegue autónomo o si se debe integrar con
# una instalación existente de WebSphere Application Server.
#
# Si la ubicación especificada es una instalación existente de WebSphere
# Application Server o WebSphere Network Deployment, eXtreme Scale se
# integrará con el WebSphere Application Server existente.  Si la ubicación
# especificada es un directorio nuevo o vacío, WebSphere eXtreme Scale se
# instalará como despliegue autónomo.
#
# Nota: Si la ubicación de instalación especificada contiene una
# instalación anterior de WebSphere eXtreme Scale, WebSphere eXtended
# Deployment DataGrid u ObjectGrid, la instalación fallará.
#
# Ubicación de la instalación predeterminada de AIX
#
#  -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Ubicación de la instalación predeterminada de HP-UX, Solaris o Linux:
#
#  -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Ubicación de la instalación predeterminada de Windows:
#
-OPT installLocation="C:\Archivos de programa\IBM\WebSphere\eXtremeScale"

#
# Si está realizando la instalación como usuario no root en Unix o sin ser
# administrador en Windows, se recomiendan las siguientes ubicaciones de
# instalación predeterminadas. Asegúrese de tener permiso de grabación
# para la ubicación de instalación seleccionada.
#
```

```
# Ubicación de la instalación predeterminada de AIX
#
#  -OPT installLocation="<inicio del usuario>/IBM/WebSphere/eXtremeScale"
#
# Ubicación de la instalación predeterminada de HP-UX, Solaris o Linux:
#
#  -OPT installLocation="<inicio del usuario>/IBM/WebSphere/eXtremeScale"
#
# Ubicación de la instalación predeterminada de Windows:
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"


###############################################################################
#  Instalación de características opcionales
#
# Especifique cuál de las características opcionales desea instalar definiendo
# con el valor "true" la característica deseada.  Defina con el valor "false"
# las características opcionales que no desee instalar.
#
# Las opciones selectServer, selectClient, selectPF y selectXSStreamQuery sólo
# son válidas cuando la opción installLocation anterior contiene una instalación
# de WebSphere Application Server.  Las opciones se ignoran en una instalación
# autónoma de WebSphere eXtreme Scale.
#
# En la instalación autónoma de WebSphere eXtreme Scale, el servidor y el cliente
# de eXtreme Scale se instalan automáticamente.  Las opciones de característica
# para la instalación autónoma de eXtreme Scale son selectXSConsoleOther y
# selectXSStreamQueryOther.

#
# Esta opción, cuando se selecciona, instala los componentes necesarios para
# ejecutar servidores WebSphere eXtreme Scale y el proveedor del servicio de
# memoria caché dinámica de eXtreme Scale. Si esta opción se selecciona, el
# también se debe seleccionar quitándole la marca de comentario y
# definiendo un valor de "true" para esta opción. De lo contrario,
# la instalación silenciosa FALLARÁ.
#
-OPT selectServer="true"

#
# Esta opción, cuando se selecciona, instala los componentes necesarios para
# ejecutar las aplicaciones clientes de WebSphere eXtreme Scale. Si se selecciona
# la opción de servidor anterior, esta opción también se debe seleccionar quitándole
# la marca de comentario y definiendo un valor de "true" para esta opción o la
# instalación silenciosa FALLARÁ.
#
-OPT selectClient="true"

#
# Esta opción, cuando se selecciona, instala los componentes necesarios para
# la consola de WebSphere eXtreme Scale. Si se selecciona este opción, la ubicación
# de instalación especificada antes, debe ser un directorio nuevo o vacío porque la
# opción de consola solo es válida para el despliegue autónomo de WebSphere eXtreme
# Scale. Para
instalar esta opción, se debe quitar la marca de comentario
# a la línea de la opción siguiente y se debe definir
# con un valor de "true".
#-OPT selectXSConsoleOther="false"

#
# Las opciones siguientes, si se seleccionan, instalarán un funcionalidad en
# desuso.
#
# Esta opción selecciona WebSphere Partition Facility para la instalación.
# Esta funcionalidad está EN DESUSO.  Para instalar esta opción, se debe
# quitar la marca de comentario a la línea de la opción siguiente y se
```

```
# debe definir un valor de "true".
#
#-OPT selectPF="false"

#
# Esta opción selecciona WebSphere eXtreme Scale StreamQuery for WAS para
# su instalación.  Esta funcionalidad está EN DESUSO.  Para instalar esta
# opción, se debe quitar la marca de comentario a la línea de la opción
# siguiente y se debe definir un valor de "true".
# Si esta opción se selecciona, el cliente de WebSphere eXtreme Scale
# también se debe seleccionar quitándole la marca de comentario y
# definiendo un valor de "true" para esta opción. De lo contrario,
# la instalación silenciosa FALLARÁ.
#
#-OPT selectXSStreamQuery="false"

#
# Esta opción selecciona WebSphere eXtreme Scale StreamQuery for J2SE para
# su instalación.  Esta funcionalidad está EN DESUSO.  Para instalar esta
# opción, se debe quitar la marca de comentario a la línea de la opción
# siguiente y se debe definir un valor de "true".
# Si esta opción se selecciona, el cliente de WebSphere eXtreme Scale
# también se debe seleccionar quitándole la marca de comentario y
# definiendo un valor de "true" para esta opción. De lo contrario,
# la instalación silenciosa FALLARÁ.
#
#-OPT selectXSStreamQueryOther="false"


###############################################################################
#  Lista de perfiles para el aumento
#
# Especifique cuál de los perfiles existentes desea aumentar o añada una marca de
# comentario a la línea para aumentar todos los perfiles existentes detectados
# por la instalación.
#
# Para especificar varios perfiles, utilice comas para separar los nombres de
# los distintos perfiles. Por ejemplo, "AppSrv01,Dmgr01,Custom01". La lista
# no debe contener espacios.
#
-OPT profileAugmentList=""


###############################################################################
# Control de rastreo
#
# El formato de salida de rastreo se puede controlar mediante la opción
# -OPT traceFormat=ALL
#
# Las opciones para el formato son 'text' y 'XML'. De forma predeterminada,
# se producirán los dos formatos, en dos archivos de rastreo diferentes.
#
# Si sólo se necesita un formato, utilice la opción traceFormat para
# especificarlo, tal como se indica a continuación:
#
# Valores válidos:
#
#  text -  Las líneas del archivo de rastreo estarán en texto sin formato para
#          que se puedan leer fácilmente.
#  XML  -  Las líneas del archivo de rastreo estarán en el formato XML de registro
#          Java estándar, que se puede ver utilizando cualquier editor de texto o
#          XML o utilizando la herramienta Chainsaw de Apache en el URL siguiente:
#          (http://logging.apache.org/log4j/docs/chainsaw.html).
#
# La cantidad de capturar de rastreo captada se puede controlar utilizando la
# opción siguiente:
# -OPT traceLevel=INFO
```

```
#
# Valores válidos:
#
# Nivel de   Nivel
# rastreo    numérico Descripción
# -------  ---------  -----------------------------------------------------------
# OFF         0       No se produce ningún archivo de rastreo
# SEVERE      1       Sólo se proporcionan en el archivo de rastreo los errores
#                     graves
# WARNING     2       Se añaden al archivo de rastreo los mensajes referentes a
#                     excepciones no graves y avisos
# INFO        3       Se añaden al archivo de rastreo los mensajes informativos
#                     (este es el nivel de rastreo predeterminado)
# CONFIG      4       Se añaden al archivo de rastreo los mensajes relacionados
#                     con la configuración
# FINE        5       Llamadas de método de rastreo para métodos públicos
# FINER       6       Llamadas de método de rastreo para métodos no públicos
#                     excepto métodos de obtención y métodos de establecimiento.
# FINEST      7       Rastrear todas las llamadas de método, la entrada/salida
#                     del rastreo incluirá parámetros y el valor de retorno
```

Archivo de respuestas para la instalación de WebSphere eXtreme Scale Client

```
################################################################################
#
# Archivo de opciones de InstallShield de IBM WebSphere eXtreme Scale Cliente V7.1.1
#
# Nombre del asistente: Instalación
# Origen del asistente: setup.jar
#
# Este archivo se puede utilizar la Instalación con las opciones especificadas a
# continuación cuando se ejecuta el asistente con la opción de la línea de mandatos
# "-options". Lea la documentación sobre cada valor para obtener información sobre
# cómo cambiar su valor. Incluya todos los valores entre un solo par de comillas
# dobles.
#
# Un uso común de un archivo de opciones es ejecutar el asistente en modalidad
# silenciosa. Esto permite al autor del archivo de opciones especificar valores
# del asistente sin tener que ejecutar el asistente en modalidad gráfica o de
# consola. Para utilizar este archivo de opciones para la ejecución en modalidad
# silenciosa, utilice los siguientes argumentos de la línea de mandatos al
# ejecutar el asistente:
#
#    -options "D:\installImage\WXS_Client\wxssetup.response" -silent
#
# Observe que se debe utilizar el nombre de archivo de respuestas totalmente
# calificado.
#
################################################################################

################################################################################
#
# Aceptación de la licencia
#
# Valores válidos:
#  true - Acepta la licencia. Instalará el producto.
#  false - Rechaza la licencia. La instalación no se llevará a cabo.
#
# Si no se realiza la instalación, este hecho se registrará en un archivo de
# registro temporal en el directorio temporal del usuario.
#
# Al cambiar el valor de la propiedad silentInstallLicenseAcceptance de
# este archivo de respuestas por "true", afirma que ha revisado y acepta
# las condiciones del Acuerdo Internacional de Programas Bajo Licencia
# ubicado en
# CD_ROOT\WXS_Cleint\wxs.client.primary.pak\repository\legal.xs.client\license.xs.
# Si no acepta estos términos, no cambie el valor ni descargue, instale, use
```

```
# o acceda al programa y devuelva rápidamente el programa y la prueba
# de titularidad al vendedor para obtener un reintegro por el importe
# que haya pagado.
#
-OPT silentInstallLicenseAcceptance="false"


################################################################################
# Comprobación de requisitos previo que no causen bloqueos
#
# Si desea inhabilitar la comprobación de requisitos previos que no causen
# bloqueos, elimine la marca de comentario de la línea siguiente. Esto
# notificará al instalador que debe continuar con la instalación y registrar
# los avisos aunque la comprobación de requisitos previos haya fallado.
#
#-OPT disableNonBlockingPrereqChecking="true"


################################################################################
#
# Ubicación de instalación
#
# La ubicación de instalación del producto. Especifique un directorio válido en el
# que se instalará el producto. Si el directorio contiene espacios, inclúyalos entre
# comillas dobles tal como se muestra en el ejemplo de Windows siguiente. Tenga en
# cuenta que los espacios en la ubicación de instalación sólo se permiten en los
# sistemas operativos Windows. La longitud máxima de la vía de acceso es de 60
# caracteres para Windows.
#
# A continuación está la lista de ubicaciones de instalación predeterminadas
# para cada sistema operativo soportado cuando se instala como usuario root.  De
# forma predeterminada, en este archivo de respuestas, se utiliza la  ubicación
# de instalación de Windows. Si desea utiliza la ubicación de instalación
# predeterminada para otro sistema operativo, quite las marcas de comentario
# de la entrada de la ubicación de instalación predeterminada apropiada
# (eliminando '#') y luego comente (añadiendo '#') la entrada del sistema
# operativo Windows a continuación.
#
# La ubicación de instalación se utiliza para determinar si WebSphere eXtreme
# Scale se debe instalar como despliegue autónomo o si se debe integrar con
# una instalación existente de WebSphere Application Server.
#
# Si la ubicación especificada es una instalación existente de WebSphere
# Application Server o WebSphere Network Deployment, eXtreme Scale se
# integrará con el WebSphere Application Server existente.  Si la ubicación
# especificada es un directorio nuevo o vacío, WebSphere eXtreme Scale se
# instalará como despliegue autónomo.
#
# Nota: Si la ubicación de instalación especificada contiene una
# instalación anterior de WebSphere eXtreme Scale, WebSphere eXtended
# Deployment DataGrid u ObjectGrid, la instalación fallará.
#
# Ubicación de la instalación predeterminada de AIX
#
#  -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Ubicación de la instalación predeterminada de HP-UX, Solaris o Linux:
#
#  -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Ubicación de la instalación predeterminada de Windows:
#
-OPT installLocation="C:\Archivos de programa\IBM\WebSphere\eXtremeScale"

#
# Si está realizando la instalación como usuario no root en Unix o sin ser
```

```
# administrador en Windows, se recomiendan las siguientes ubicaciones de
# instalación predeterminadas. Asegúrese de tener permiso de grabación
# para la ubicación de instalación seleccionada.
#
# Ubicación de la instalación predeterminada de AIX
#
#  -OPT installLocation="<inicio del usuario>/IBM/WebSphere/eXtremeScale"
#
# Ubicación de la instalación predeterminada de HP-UX, Solaris o Linux:
#
#  -OPT installLocation="<inicio del usuario>/IBM/WebSphere/eXtremeScale"
#
# Ubicación de la instalación predeterminada de Windows:
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"


################################################################################
#  Lista de perfiles para el aumento
#
# Especifique cuál de los perfiles existentes desea aumentar o añada una marca de
# comentario a la línea para aumentar todos los perfiles existentes detectados
# por la instalación.
#
# Para especificar varios perfiles, utilice comas para separar los nombres de
# los distintos perfiles. Por ejemplo, "AppSrv01,Dmgr01,Custom01". La lista
# no debe contener espacios.
#
-OPT profileAugmentList=""


################################################################################
# Control de rastreo
#
# El formato de salida de rastreo se puede controlar mediante la opción
# -OPT traceFormat=ALL
#
# Las opciones para el formato son 'text' y 'XML'. De forma predeterminada,
# se producirán los dos formatos, en dos archivos de rastreo diferentes.
#
# Si sólo se necesita un formato, utilice la opción traceFormat para
# especificarlo, tal como se indica a continuación:
#
# Valores válidos:
#
#  text -  Las líneas del archivo de rastreo estarán en texto sin formato para
#          que se puedan leer fácilmente.
#  XML  -  Las líneas del archivo de rastreo estarán en el formato XML de registro
#          Java estándar, que se puede ver utilizando cualquier editor de texto o
#          XML o utilizando la herramienta Chainsaw de Apache en el URL siguiente:
#          (http://logging.apache.org/log4j/docs/chainsaw.html).
#
# La cantidad de capturar de rastreo captada se puede controlar utilizando la
# opción siguiente:
# -OPT traceLevel=INFO
#
# Valores válidos:
#
# Nivel de   Nivel
# rastreo    numérico Descripción
# -------  ---------  ------------------------------------------------------
# OFF         0       No se produce ningún archivo de rastreo
# SEVERE      1       Sólo se proporcionan en el archivo de rastreo los errores
#                     graves
# WARNING     2       Se añaden al archivo de rastreo los mensajes referentes a
#                     excepciones no graves y avisos
# INFO        3       Se añaden al archivo de rastreo los mensajes informativos
```

```
#                       (este es el nivel de rastreo predeterminado)
# CONFIG      4         Se añaden al archivo de rastreo los mensajes relacionados
#                       con la configuración
# FINE        5         Llamadas de método de rastreo para métodos públicos
# FINER       6         Llamadas de método de rastreo para métodos no públicos
#                       excepto métodos de obtención y métodos de establecimiento.
# FINEST      7         Rastrear todas las llamadas de método, la entrada/salida
#                       del rastreo incluirá parámetros y el valor de retorno
```

Creación y aumento de perfiles para WebSphere eXtreme Scale

Después de instalar el producto, cree tipos exclusivos de perfiles y aumente los existentes para WebSphere eXtreme Scale.

Antes de empezar

Instale WebSphere eXtreme Scale. Si desea más información, consulte "Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server" en la página 163.

El aumento de perfiles para utilizarlos con WebSphere eXtreme Scale es opcional, pero es necesario en los siguientes escenarios de uso:

- Para iniciar automáticamente un servicio de catálogo o contenedor en un proceso de WebSphere Application Server. Sin aumentar los perfiles de servidor, los servidores solo se pueden iniciar mediante programación utilizando la API ServerFactory o como procesos aparte utilizando los scripts **startOgServer**.
- Para utilizar Performance Monitoring Infrastructure (PMI) para supervisar métricas de WebSphere eXtreme Scale.
- Para visualizar la versión de WebSphere eXtreme Scale en la consola de administración de WebSphere Application Server.

Acerca de esta tarea

Ejecución en WebSphere Application Server versión 6.1 o versión 7.0

Si el entorno contiene WebSphere Application Server versión 6.1 o versión 7.0, puede utilizar el plug-in Profile Management Tool o el mandato **manageprofiles** para crear y aumentar perfiles.

Qué hacer a continuación

En función de la tarea que elija completar, inicie la consola Primeros pasos para recibir ayuda en la configuración y la pruebas del entorno del producto. La consola Primeros pasos se encuentra en el directorio *raíz_intal_wxs*`\firststeps\wxs\firststeps.bat`. También puede crear o aumentar perfiles adicionales repitiendo cualquiera de las tareas anteriores.

Utilización de la interfaz gráfica de usuario para crear perfiles:

Utilice la interfaz gráfica de usuario (GUI), que proporciona el plug-in de la herramienta de gestión de perfiles para crear perfiles para WebSphere eXtreme Scale. Un perfil es un conjunto de archivos que define el entorno de ejecución.

Antes de empezar

No puede utilizar la GUI para aumentar perfiles en el escenario siguiente:

- **Instalaciones de 64 bits de WebSphere Application Server**:

La herramienta de gestión de perfiles no existe para instalaciones de 64 bits de WebSphere Application Server. Utilice el script **manageprofiles** desde la línea de mandatos para estas instalaciones.

Acerca de esta tarea

Para utilizar las características del producto, el plug-in de Herramienta de gestión de perfiles permite a la GUI ayudarle a configurar perfiles como un perfil de WebSphere Application Server, un perfil de gestor de despliegue, un perfil de célula y un perfil personalizado. Puede aumentar perfiles durante o después de la instalación de WebSphere eXtreme Scale.

Procedimiento

Utilice la GUI de la herramienta de gestión de perfiles para crear perfiles. Elija de una de las opciones siguientes para iniciar el asistente:

- Seleccione **Herramienta de gestión de perfiles** en la consola Primeros pasos.
- Acceda a la herramienta de gestión de perfiles desde el menú **Inicio**.
- Ejecute el script ./pmt.sh|bat desde el directorio *raíz_instalación*/bin/ProfileManagement.

Qué hacer a continuación

Puede crear perfiles adicionales o aumentar los perfiles existentes. Para reiniciar la herramienta de gestión de perfiles, ejecute el mandato **./pmt.sh|bat** desde el directorio *raíz_was*/bin/ProfileManagement, o seleccione **Herramienta de gestión de perfiles** en la consola Primeros pasos.

Inicie un servicio de catálogos, inicie contenedores y configure los puertos TCP en el entorno WebSphere Application Server. Para obtener más información, consulte el apartado "Configuración de WebSphere eXtreme Scale con WebSphere Application Server" en la página 256.

Utilización de la interfaz gráfica de usuario para aumentar perfiles:

Después de instalar el producto, podrá aumentar un perfil existente para que sea compatible con WebSphere eXtreme Scale.

Acerca de esta tarea

Cuando aumente un perfil existente, cambie el perfil aplicando una plantilla de aumento específica del producto. Por ejemplo, los servidores WebSphere eXtreme Scale no se inician automáticamente, a menos que el perfil de servidor se aumente con la plantilla xs_augment.

- Aumente el perfil con la plantilla xs_augment si ha instalado el cliente eXtreme Scale o el cliente y el servidor.
- Aumente el perfil con la plantilla pf_augment sólo si ha instalado el recurso de particionamiento.
- Aplique ambas plantillas, si el entorno contiene el cliente eXtreme Scale y el recurso de particionamiento.

Procedimiento

Utilice la GUI de la herramienta de gestión de perfiles para aumentar los perfiles para eXtreme Scale. Elija de una de las opciones siguientes para iniciar el asistente:

- Seleccione **Herramienta de gestión de perfiles** en la consola Primeros pasos.
- Acceda a la herramienta de gestión de perfiles desde el menú **Inicio**.
- Ejecute el script ./pmt.sh|bat desde el directorio *raíz_was*/bin/ProfileManagement.

Qué hacer a continuación

Puede aumentar los perfiles adicionales. Para reiniciar la herramienta de gestión de perfiles, ejecute el mandato **./pmt.sh|bat** desde el directorio *raíz_was*/bin/ProfileManagement, o seleccione **Herramienta de gestión de perfiles** en la consola Primeros pasos.

Inicie un servicio de catálogos, inicie contenedores y configure los puertos TCP en el entorno WebSphere Application Server. Si desea más información, consulte "Configuración de WebSphere eXtreme Scale con WebSphere Application Server" en la página 256.

Mandato manageprofiles:

Puede utilizar el programa de utilidad **manageprofiles** para crear perfiles con la plantilla de WebSphere eXtreme Scale, y aumentar y reducir los perfiles existentes del servidor de aplicaciones con las plantillas de aumento de eXtreme Scale. Para utilizar las características del producto, el entorno debe contener, como mínimo, un perfil aumentado para el producto.

- Antes de poder crear y aumentar los perfiles, debe instalar eXtreme Scale . Si desea más información, consulte "Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server" en la página 163.

Finalidad

El mandato **manageprofiles** crea el entorno de ejecución para un proceso de producto en un conjunto de archivos llamado perfil. El perfil define el entorno de ejecución. Puede realizar las siguientes acciones con el mandato **manageprofiles**:

- Crear y aumentar un perfil de gestor de despliegue
- Crear y aumentar un perfil personalizado
- Crear y aumentar un perfil de servidor de aplicación autónomo
- Crear y aumentar un perfil de célula
- Reducir cualquier tipo de perfil

Cuando aumente un perfil existente, cambie el perfil aplicando una plantilla de aumento específica del producto.

- Aumente el perfil con la plantilla xs_augment si ha instalado el cliente de eXtreme Scale, o el cliente y también el servidor.
- Aumente el perfil con la plantilla pf_augment si ha instalado sólo el recurso de particionamiento.
- Aplique ambas plantillas si el entorno contiene el cliente de eXtreme Scale y el recurso de particionamiento.

Ubicación

El archivo de mandato está en el directorio *raíz_instalación*/bin.

Uso

Si desea ayuda detallada, utilice el parámetro **-help**:

```
./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/xs_augment/dmgr -help
```

En las siguientes secciones, se describen las tareas que puede realizar utilizando el mandato **manageprofiles**, junto una lista de los parámetros necesarios. Si desea detalles sobre los parámetros opcionales para cada tarea, consulte el mandato **manageprofiles** en el centro de información de WebSphere Application Server.

Crear un perfil de gestor de despliegue

Puede utilizar el mandato **manageprofiles** para crear un perfil de gestor de despliegue. El gestor de despliegue administra los servidores de aplicaciones que se han federado en la célula.

Parámetros

-create
Crea un perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de archivo de la plantilla. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/
profileTemplates/tipo_plantilla/dmgr
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

  ```
  ./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/xs_augment/dmgr
  ```

- Utilización de la plantilla pf_augment:

  ```
  ./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/pf_augment/dmgr
  ```

Crear un perfil personalizado

Puede utilizar el mandato **manageprofiles** para crear un perfil personalizado. Un perfil personalizado es un nodo vacío que puede personalizar a través del gestor de despliegue para incluir servidores de aplicaciones, clústeres u otros procesos Java.

Parámetros

-create
Crea un perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de archivo de la plantilla. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/managed
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

  ```
  ./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/xs_augment/managed
  ```

- Utilización de la plantilla pf_augment:

  ```
  ./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/pf_augment/managed
  ```

Crear un perfil de servidor de aplicaciones autónomo

Puede utilizar el mandato **manageprofiles** para crear un perfil de servidor de aplicaciones autónomo.

Parámetros

-create
Crea un perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de archivo de la plantilla. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/default
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

  ```
  ./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/xs_augment/default
  ```

- Utilización de la plantilla pf_augment:

  ```
  ./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/pf_augment/default
  ```

Crear un perfil de célula

Puede utilizar el mandato **manageprofiles** para crear un perfil de célula, que está formada por un gestor de despliegue y un servidor de aplicaciones.

Parámetros

Especifique los siguientes parámetros en la plantilla del gestor de despliegue:

-create
Crea un perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de archivo de la plantilla. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantillatemplate_type/cell/dmgr
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Especifique los siguientes parámetros con la plantilla del servidor de aplicaciones:

-create
Crea un perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de archivo de la plantilla. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/cell/default
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

```
./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/xs_augment/cell/dmgr
-nodeProfilePath nombre de nodo/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/xs_augment/cell/default
-dmgrProfilePath raíz_instalación/profiles/Dmgr01 -portsFile
raíz_instalación/profiles/Dmgr01/properties/portdef.props -nodePortsFile
raíz_instalación/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

- Utilización de la plantilla pf_augment:

```
./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/pf_augment/cell/dmgr
-nodeProfilePath nombre de nodo/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr
-appServerNodeName node01

./manageprofiles.sh|bat -create -templatePath raíz_instalación/profileTemplates/pf_augment/cell/default
-dmgrProfilePath raíz_instalación/profiles/Dmgr01 -portsFile
raíz_instalación/profiles/Dmgr01/properties/portdef.props -nodePortsFile
raíz_instalación/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr
-nodeName node01dmgr -appServerNodeName node01
```

Aumentar un perfil de gestor de despliegue

Puede utilizar el mandato **manageprofiles** para aumentar un perfil de gestor de despliegue.

Parámetros

-augment
Aumenta el perfil existente. (Necesario)

-profileName
Especifica el nombre del perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de los archivos de plantilla que se encuentran en el directorio raíz de la instalación. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/dmgr
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath raíz_instalación/profileTemplates/xs_augment/dmgr
```

- Utilización de la plantilla pf_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath raíz_instalación/profileTemplates/pf_augment/dmgr
```

Aumentar un perfil personalizado

Puede utilizar el mandato **manageprofiles** para aumentar un perfil personalizado.

Parámetros

-augment
Aumenta el perfil existente. (Necesario)

-profileName
Especifica el nombre del perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de los archivos de plantilla que se encuentran en el directorio raíz de la instalación. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/managed
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath raíz_instalación/profileTemplates/xs_augment/managed
```

- Utilización de la plantilla pf_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath raíz_instalación/profileTemplates/pf_augment/managed
```

Aumentar un perfil de servidor de aplicaciones autónomo

Puede utilizar el mandato **manageprofiles** para aumentar un perfil de servidor de aplicaciones autónomo.

Parámetros

-augment
Aumenta el perfil existente. (Necesario)

-profileName
Especifica el nombre del perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de los archivos de plantilla que se encuentran en el directorio raíz de la instalación. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/default
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath raíz_instalación/profileTemplates/xs_augment/default
```

- Utilización de la plantilla pf_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01
-templatePath raíz_instalación/profileTemplates/pf_augment/default
```

Aumentar un perfil de célula

Puede utilizar el mandato **manageprofiles** para aumentar un perfil de célula

Parámetros

Especifique los siguientes parámetros para el perfil de gestor de despliegue:

-augment
Aumenta el perfil existente. (Necesario)

-profileName
Especifica el nombre del perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de los archivos de plantilla que se encuentran en el directorio raíz de la instalación. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/cell/dmgr
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Especifique los siguientes parámetros para el perfil de servidor de aplicaciones:

-augment
Aumenta el perfil existente. (Necesario)

-profileName
Especifica el nombre del perfil. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de los archivos de plantilla que se encuentran en el directorio raíz de la instalación. (Necesario)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/cell/default
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment`.

Ejemplo

- Utilización de la plantilla xs_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath raíz_instalación
    /profileTemplates/xs_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath raíz_instalación
    /profileTemplates/xs_augment/cell/default
```

- Utilización de la plantilla pf_augment:

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath raíz_instalación
    /profileTemplates/pf_augment/cell/dmgr

./manageprofiles.sh|bat -augment -profileName profile01 -templatePath raíz_instalación
    /profileTemplates/pf_augment/cell/default
```

Reducir un perfil

Para reducir un perfil, especifique el parámetro **-ignoreStack** con el parámetro **-templatePath** además de especifica los parámetros **-unaugment** y **-profileName** necesarios.

Parámetros

-unaugment
Reduce un perfil aumentado previamente. (Necesario)

-profileName
Especifica el nombre del perfil. El parámetro se emite de forma predeterminada, si no se especifica ningún valor. (Necesario)

-templatePath *vía_acceso_plantilla*
Especifica la vía de acceso de los archivos de plantilla que se encuentran en el directorio raíz de la instalación. (Opcional)

Utilice el siguiente formato:

```
-templatePath raíz_instalación/profileTemplates/tipo_plantilla/tipo_perfil
```

donde *tipo_plantilla* es `xs_augment` o `pf_augment` y *tipo_perfil* adopta uno de estos cuatro tipos:

- `dmgr`: perfil de gestor de despliegue
- `managed`: perfil personalizado
- `default`: perfil de servidor de aplicaciones autónomo
- `cell`: perfil de célula

-ignoreStack
Se utiliza con el parámetro **-templatePath** para reducir un perfil determinado que ha sido aumentado. (Opcional)

Ejemplo

- Utilización de la plantilla xs_augment:

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath raíz_instalación/profileTemplates/xs_augment/tipo_perfil
```

- Utilización de la plantilla pf_augment:

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack
-templatePath raíz_instalación/profileTemplates/pf_augment/tipo_perfil
```

Perfiles que no son root:

Otorgue a un usuario que no es root los permisos para los archivos y directorios de forma que el usuario no root pueda crear un perfil para el producto. El usuario no root también puede aumentar un perfil que fue creado por un usuario root, un usuario no root diferente o el mismo usuario no root.

En un entorno WebSphere Application Server, los usuarios no root (no administradores) están limitados a poder crear y utilizar los perfiles en su entorno. Dentro del plug-in de la herramienta de gestión de perfiles, los nombres y valores de puerto exclusivos están inhabilitaddos para los usuarios no root. El usuario no root debe cambiar los valores de campo predeterminados en la herramienta de gestión de perfiles para el nombre de perfil, nombre de nodo, nombre de célula y asignaciones de puerto. Considere asignar a los usuarios no root un rango de valores para cada uno de los campos. Puede asignar responsabilidad a los usuarios no root para adherir los rangos de valores adecuados y para mantener la integridad de sus propias definiciones.

El término *instalador* hace referencia a un usuario root y, también, a un usuario no root. Como instalador, puede otorgar a los usuarios no root los permisos para crear perfiles y establecer sus propios entornos de producto. Por ejemplo, un usuario no root podría crear un entorno de producto para probar el despliegue de aplicaciones con un perfil que es suyo. Las tareas específicas que puede completar para permitir la creación de un perfil no root incluyen los siguientes elementos:

- La creación de un perfil y la asignación de propiedad del directorio de perfil a un usuario no root, de forma que el usuario no root pueda iniciar WebSphere Application Server para un perfil específico.
- El otorgamiento de permisos de escritura de los archivos y directorios apropiados a un usuario no root, que permite al usuario no root crear el perfil.

Con esta tarea, puede crear un grupo para usuarios que están autorizados para crear perfiles, o proporcionar a los usuarios individuales la capacidad de crear perfiles.

- La instalación de los paquetes de mantenimiento para el producto, que incluye los servicios necesarios para los perfiles existentes que son propiedad de un usuario no root. Como instalador, es el propietario de los archivos nuevos que crea el paquete de mantenimiento.

Para obtener más información acerca de la creación de perfiles para usuarios no root, consulte Creación de perfiles para usuarios no root.

Como instalador, también puede otorgar permisos para un usuario no root para aumentar perfiles. Por ejemplo, un usuario no root puede aumentar un perfil creado por un instalador o aumentar un perfil creado por él mismo. Siga el proceso de aumento del usuario no root WebSphere Application Server Network Deployment.

Sin embargo, cuando un usuario no root aumenta un perfil creado por el instalador, el usuario no root no tiene que crear los archivos siguientes antes del aumento. Los archivos siguientes se establecieron durante el proceso de creación del perfil:

- *raíz_was*/logs/manageprofiles.xml
- *raíz_was*/properties/fsdb.xml
- *raíz_was*/properties/profileRegistry.xml

Cuando un usuario no root aumenta un perfil que crea, el usuario no root debe modificar los permisos para los documentos que se encuentran dentro de las plantillas del perfil eXtreme Scale.

Atención: También puede utilizar un perfil no raíz (no administrador) para WebSphere eXtreme Scale en un entorno autónomo, uno fuera de WebSphere Application Server. Debe cambiar el propietario del directorio de ObjectGrid al perfil no raíz. Entonces puede iniciar este perfil no raíz describen y utilizar eXtreme Scale tal como lo haría normalmente para un perfil raíz (administrador).

Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client autónomos

Puede instalar WebSphere eXtreme Scale o WebSphere eXtreme Scale Client autónomos en un entorno que no contenga WebSphere Application Server o WebSphere Application Server Network Deployment.

Antes de empezar

- Verifique que el directorio de instalación de destino está vacío o que no existe.

  Importante: Si existe una versión anterior de WebSphere eXtreme Scale o del componente ObjectGrid en el directorio que especifica para instalar la versión 7.1.1, el producto no se instala. Por ejemplo, es posible que disponga de una carpeta *raíz_intal_wxs*/ObjectGrid existente. Puede seleccionar un directorio de instalación diferente o cancelar la instalación. A continuación, desinstale la instalación anterior y vuelva a ejecutar el asistente.
- Se instala un entorno de ejecución de IBM como parte de la instalación autónoma en la carpeta *raíz_intal_wxs*/java.

- Si sólo está instalando el cliente: Descargue WebSphere eXtreme Scale Client para la plataforma adecuada del Sitio de soporte.

Acerca de esta tarea

Cuando instale el producto como autónomo, instale de forma conjunta el servidor y el cliente de WebSphere eXtreme Scale. Con la instalación de WebSphere eXtreme Scale Client en modalidad autónoma, se instala un cliente para acceder a los datos de las cuadrícula de datos. Los procesos de servidor y de cliente, por lo tanto, acceden a todos los recursos necesarios de forma local. También puede incorporar WebSphere eXtreme Scale en aplicaciones Java Platform, Standard Edition (J2SE) existentes utilizando scripts y archivos JAR (Java Archive).

Atención: También puede utilizar un perfil no raíz (no administrador) para WebSphere eXtreme Scale en un entorno autónomo. Para utilizar un perfil no raíz, debe cambiar el propietario del directorio de `ObjectGrid` al perfil no raíz. Entonces puede iniciar este perfil no raíz describen y utilizar eXtreme Scale tal como lo haría normalmente para un perfil raíz (administrador).

Procedimiento

1. Utilice el asistente para instalar el servidor y el cliente desde el DVD.
   - Ejecute el siguiente script para iniciar el asistente de la instalación completa de WebSphere eXtreme Scale:
     - Linux UNIX *raíz_dvd*`/install`
     - Windows *raíz_dvd*`\install.bat`
   - Ejecute el siguiente script para iniciar el asistente de la instalación de WebSphere eXtreme Scale Client. Los archivos de instalación están en el archivo zip que ha descargado del Sitio de soporte:
     - Linux UNIX `root/WXS_Client/install`
     - Windows `root\WXS_Client\install.bat`

   Atención: Si utiliza el convenio de denominación universal (UNC) para identificar vías de acceso en el mandato de instalación, es posible que los elementos cuya instalación anticipa no se instalen una vez ejecutado el mandato. Para evitar problemas, correlacione la vía de acceso del archivo con una unidad de red. Ejecute el mandato **`install`** contra la unidad correlacionada. La utilización de una unidad de red correlacionada permite asegurarse de que se instalan todos los elementos.
2. Siga las indicaciones del asistente y pulse **Finalizar**.

   Restricción: El panel de características opcionales lista las características que puede seleccionar para instalarlas. Sin embargo, las características no se pueden añadir de forma incremental en el entorno del producto después de que se instale el producto. Si elige no instalar una característica con la instalación inicial del producto, deberá desinstalar y volver a instalar el producto para añadir la característica.

Resultados

Windows Si va a instalar WebSphere eXtreme Scale Client en Windows, es posible que aparezca el texto siguiente en los resultados de la instalación:

```
Finalizado con éxito: Se ha instalado de forma satisfactoria
el siguiente producto:
cliente de WebSphere eXtreme Scale.  Algunos pasos de la configuración
han dado errores. Consulte el siguiente archivo de anotaciones cronológicas
para obtener más información: <raíz de instalación de WebSphere
Application Server>\logs\wxs_client\install\log.txt" Revise el archivo
de registro de instalación (log.txt) y revise el registro de aumento del
gestor de despliegue.
```

Si aparece una anomalía en el archivo `iscdeploy.sh`, puede pasar por alto el error. Este error no causa problemas.

Qué hacer a continuación

- Verifique la instalación. Para obtener más información, consulte "Verificación de la instalación" en la página 206.
- Empiece configurando la instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client. Para obtener más información, consulte "Primeros pasos después de la instalación" en la página 207.

Archivos de ejecución para una instalación autónoma de WebSphere eXtreme Scale

Se incluyen archivos de archivado Java (JAR) en la instalación. Puede ver los archivos JAR que se incluyen y la ubicación en la que se instalan.

Tabla 7. Archivos de tiempo de ejecución para la instalación completa de WebSphere eXtreme Scale. WebSphere eXtreme Scale se basa en los procesos de ObjectGrid y en las API relacionadas. La tabla siguiente lista los archivos JAR que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
`wxsdynacache.jar`	Cliente y servidor	`dynacache/lib`	El archivo `wxsdynacache.jar` contiene las clases necesarias para utilizar con el proveedor de la memoria caché dinámica. El archivo se incluye automáticamente en el entorno de ejecución de servidor cuando se utilizan los scripts proporcionados.
`wxshyperic.jar`	programa de utilidad	`hyperic/lib`	El plug-in de detección de servidor de WebSphere eXtreme Scale para el agente de supervisión SpringSource Hyperic.
`objectgrid.jar`	Local, cliente y servidor	`lib`	El archivo `objectgrid.jar` es un paquete OSGi que utiliza el entorno de ejecución del servidor de Java SE 5.0 y posteriores. El archivo se incluye automáticamente en el entorno de ejecución de servidor cuando se utilizan los scripts proporcionados.
`ogagent.jar`	Local, cliente y servidor	`lib`	El archivo `ogagent.jar` contiene las clases de tiempo de ejecución necesarias para ejecutar el agente de instrumentación Java que se utiliza con la API EntityManager.
`ogclient.jar`	Local y cliente	`lib`	El archivo `ogclient.jar` es un paquete OSGi que contiene sólo los entornos de ejecución local y de cliente. Puede utilizar este archivo con Java SE 5.0 y posteriores.
`ogspring.jar`	Local, cliente y servidor	`lib`	El archivo `ogspring.jar` contiene clases de soporte para la integarción de la infraestructura de Spring SpringSource.
`wsogclient.jar`	Local y cliente	`lib`	El archivo `wsogclient.jar` instalado cuando se utiliza un entorno que contiene WebSphere Application Server versión 6.0.2 y posterior. Este archivo sólo contiene los entornos de ejecución local y de cliente.
`wxssizeagent.jar`	Local, cliente y servidor	`lib`	El archivo `wxssizeagent.jar` se utiliza para proporcionar información sobre dimensionamiento de entradas de memoria caché más precisa al utilizar el entorno de tiempo de ejecución Java (JRE) Versión 1.5 o posterior.

`ibmcfw.jar`
`ibmorb.jar`
`ibmorbapi.jar`
	Cliente y servidor	`lib/endorsed`	Este conjunto de archivos incluye un módulo de tiempo de ejecución de intermediario de solicitud de objetos (ORB) que se utiliza para ejecutar las aplicaciones en los procesos Java SE.

Tabla 7. Archivos de tiempo de ejecución para la instalación completa de WebSphere eXtreme Scale (continuación). WebSphere eXtreme Scale se basa en los procesos de ObjectGrid y en las API relacionadas. La tabla siguiente lista los archivos JAR que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
`restservice.ear`	Cliente	`restservice/lib`	El archivo `restservice.ear` contiene el archivador empresarial de la aplicación de servicio de datos REST de eXtreme Scale para los entornos de WebSphere Application Server.
`restservice.war`	Cliente	`restservice/lib`	El archivo `restservice.war` contiene el archivo web del servicio de datos REST de eXtreme Scale para los servidores de aplicaciones adquiridos de otro proveedor.
`xsadmin.jar`	programa de utilidad	`samples`	El archivo `xsadmin.jar` contiene el programa de utilidad de ejemplo de administración de eXtreme Scale.
`sessionobjectgrid.jar`	Cliente y servidor	`session/lib`	El archivo `sessionobjectgrid.jar` contiene el tiempo de ejecución de gestión de sesiones HTTP de eXtreme Scale.
`splicerlistener.jar`	programa de utilidad	`session/lib`	El archivo `splicerlistener.jar` contiene el programa de utilidad splicer para el escucha de sesiones HTTP de eXtreme Scale Versión 7.1 y posteriores.
`xsgbean.jar`	Servidor	`wasce/lib`	El archivo `xsgbean.jar` contiene el GBean para incluir los servidores eXtreme Scale en los servidores de aplicaciones WebSphere Application Server Community Edition.
`splicer.jar`	programa de utilidad	`legacy/session/lib`	El programa de utilidad splicer para el filtro del gestor de sesiones HTTP de WebSphere eXtreme Scale Versión 7.0.

Tabla 8. Archivos de tiempo de ejecución de WebSphere eXtreme Scale Client. WebSphere eXtreme Scale Client se basa en los procesos de ObjectGrid y en las API relacionadas. La tabla siguiente lista los archivos JAR que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
`wxsdynacache.jar`	Cliente y servidor	`dynacache/lib`	El archivo `wxsdynacache.jar` contiene las clases necesarias para utilizar con el proveedor de la memoria caché dinámica. El archivo se incluye automáticamente en el entorno de ejecución de servidor cuando se utilizan los scripts proporcionados.
`wxshyperic.jar`	programa de utilidad	`hyperic/lib`	El plug-in de detección de servidor de WebSphere eXtreme Scale para el agente de supervisión SpringSource Hyperic.
`ogagent.jar`	Local, cliente y servidor	`lib`	El archivo `ogagent.jar` contiene las clases de tiempo de ejecución necesarias para ejecutar el agente de instrumentación Java que se utiliza con la API EntityManager.
`ogclient.jar`	Local y cliente	`lib`	El archivo `ogclient.jar` es un paquete OSGi que contiene sólo los entornos de ejecución local y de cliente. Puede utilizar este archivo con Java SE 5 y posterior.
`ogspring.jar`	Local, cliente y servidor	`lib`	El archivo `ogspring.jar` contiene clases de soporte para la integración de la infraestructura de Spring SpringSource.
`wsogclient.jar`	Local y cliente	`lib`	El archivo `wsogclient.jar` instalado cuando se utiliza un entorno que contiene WebSphere Application Server versión 6.0.2 y posterior. Este archivo sólo contiene los entornos de ejecución local y de cliente.
`wxssizeagent.jar`	Local, cliente y servidor	`lib`	El archivo `wxssizeagent.jar` se utiliza para proporcionar información sobre dimensionamiento de entradas de memoria caché más precisa al utilizar el entorno de tiempo de ejecución Java (JRE) Versión 1.5 o posterior.

`ibmcfw.jar`
`ibmorb.jar`
`ibmorbapi.jar`
	Cliente y servidor	`lib/endorsed`	Este conjunto de archivos incluye un módulo de tiempo de ejecución de intermediario de solicitud de objetos (ORB) que se utiliza para ejecutar las aplicaciones en los procesos Java SE.

`restservice.ear`	Cliente	`restservice/lib`	El archivo `restservice.ear` contiene el archivador empresarial de la aplicación de servicio de datos REST de eXtreme Scale para los entornos de WebSphere Application Server.
`restservice.war`	Cliente	`restservice/lib`	El archivo `restservice.war` contiene el archivo web del servicio de datos REST de eXtreme Scale para los servidores de aplicaciones adquiridos de otro proveedor.

Tabla 8. Archivos de tiempo de ejecución de WebSphere eXtreme Scale Client (continuación). WebSphere eXtreme Scale Client se basa en los procesos de ObjectGrid y en las API relacionadas. La tabla siguiente lista los archivos JAR que se incluyen en la instalación. La ubicación de instalación es relativa al directorio *inicio_wxs* que elige durante la instalación.

Nombre de archivo	Entorno	Ubicación de la instalación	Descripción
`xsadmin.jar`	programa de utilidad	`samples`	El archivo `xsadmin.jar` contiene el programa de utilidad de ejemplo de administración de eXtreme Scale.
`sessionobjectgrid.jar`	Cliente y servidor	`session/lib`	El archivo `sessionobjectgrid.jar` contiene el tiempo de ejecución de gestión de sesiones HTTP de eXtreme Scale.
`splicerlistener.jar`	programa de utilidad	`session/lib`	El archivo `splicerlistener.jar` contiene el programa de utilidad splicer para el escucha de sesiones HTTP de eXtreme Scale Versión 7.1 y posteriores.
`splicer.jar`	programa de utilidad	`legacy/session/lib`	El programa de utilidad splicer para el filtro del gestor de sesiones HTTP de WebSphere eXtreme Scale Versión 7.0.

Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con modalidad silenciosa

Utilice un archivo de respuestas completo, que puede configurar de forma específica según sus necesidades, o pasar parámetros a la línea de mandatos para instalar en modalidad silenciosa WebSphere eXtreme Scale o WebSphere eXtreme Scale Client.

Antes de empezar

- Detenga todos los procesos que se están ejecutando en el entorno WebSphere Application Server o WebSphere Application Server Network Deployment. Consulte Programas de utilidad de línea de mandatos para obtener más información sobre los mandatos **`stopManager`**, **`stopNode`** y **`stopServer`**.

  PRECAUCIÓN:
  asegúrese de que todos los procesos en ejecución se hayan detenido. Si no se detienen los procesos en ejecución, la instalación sigue, creando resultados imprevisibles y dejando la instalación en un estado indeterminado en algunas plataformas.
- Verifique que el directorio de instalación de destino está vacío o que no existe.

  Importante: Si existe una versión anterior de WebSphere eXtreme Scale o del componente ObjectGrid en el directorio que especifica para instalar la versión 7.1.1, el producto no se instala. Por ejemplo, es posible que disponga de una carpeta *raíz_intal_wxs*`/ObjectGrid` existente. Puede seleccionar un directorio de instalación diferente o cancelar la instalación. A continuación, desinstale la instalación anterior y vuelva a ejecutar el asistente.

Acerca de esta tarea

Una instalación silenciosa utiliza el mismo programa de instalación que utiliza la versión de la interfaz gráfica de usuario (GUI). Sin embargo, en lugar de visualizar una interfaz de asistente, la instalación silenciosa lee todas las respuestas de un archivo que personaliza, o de los parámetros que pase a la línea de mandatos. Consulte un ejemplo de un "Archivo `wxssetup.response.txt`" en la página 176, que incluye una descripción de cada opción.

Procedimiento

1. Opcional: Si decide instalar WebSphere eXtreme Scale o WebSphere eXtreme Scale Client utilizando un archivo de respuestas, primero personalice el archivo `wxssetup.response.txt`.

   Recuerde: Debe especificar el nombre de archivo de respuestas completo. La especificación de la vía de acceso relativa provoca que la instalación falles sin ninguna indicación de que se haya producido un error.

   a. Haga una copia del archivo de respuestas que debe personalizarse.

      Para realizar una instalación completa de WebSphere eXtreme Scale, copie el archivo de respuestas del DVD del producto en la unidad de disco.

      Para el WebSphere eXtreme Scale Client, desempaquete el archivo zip WebSphere eXtreme Scale Client en el disco duro y busque el archivo de respuestas.

   b. Abra y edite el archivo de respuestas en el editor de texto que elija. El archivo de respuestas de ejemplo anterior proporciona detalles de cómo especificar cada uno de los parámetros. Debe especificar los siguientes parámetros:

      - El acuerdo de licencia
      - El directorio de instalación

      Consejo: Cuando se instala WebSphere eXtreme Scale o WebSphere eXtreme Scale Client en un entorno de WebSphere Application Server, el instalador utiliza el directorio de instalación para determinar donde se ha instalado la instancia de WebSphere Application Server existente. Si realiza la instalación en un nodo que contiene varias instancias de WebSphere Application Server, defina claramente la ubicación.

   c. Ejecute el siguiente script para iniciar la instalación.

      Para la instalación completa de WebSphere eXtreme Scale:

      ```
      ./install.sh|bat -options C:/vía_acceso_unidad/archivo_respuestas.txt -silent
      ```

      Para la instalación de WebSphere eXtreme Scale Client:

      ```
      ./WXS_Client/install.sh|bat -options C:/víaAcceso_unidad/archivo_respuestas.txt -silent
      ```

      También puede utilizar el archivo de respuestas cuando ejecute una instalación de la interfaz gráfica de usuario. Puede utilizar el archivo de respuestas con una instalación de la interfaz gráfica de usuario para depurar problemas ocultos con la instalación silenciosa. Cuando especifique el archivo `wxssetup.response` para instalaciones de interfaz gráfica de usuario o silenciosas, deberá utilizar la vía de acceso completa. Ejecute el siguiente script para ejecutar la instalación de la interfaz gráfica de usuario con el archivo de respuestas:

      - Linux UNIX `<install_home>/install.sh -options <víaAcceso_instalación_completa_necesaria>/wxssetup.response`
      - Windows `<install_home>\install.exe -options c:\<víaAcceso_instalación_completa_necesaria>\wxssetup.response`

2. Opcional: Si elige instalar eXtreme Scale pasando determinados parámetros a la línea de mandatos, ejecute el siguiente script para iniciar la instalación:

   Para la instalación completa de WebSphere eXtreme Scale:

   ```
   ./install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=ubicación_instalación
   ```

   Para la instalación de WebSphere eXtreme Scale Client:

   ```
   ./WXS_Client/install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=ubicación_instalación
   ```

Archivo de respuestas para la instalación silenciosa

Especifique los parámetros en la línea de mandatos para personalizar y configurar la instalación del producto.

Nota: Debe especificar el nombre de archivo de respuestas completo. La especificación de la vía de acceso relativa provoca que la instalación falles sin ninguna indicación de que se haya producido un error.

Parámetros

Puede pasar los siguientes parámetros durante una instalación del producto a través de la línea de mandatos o de un archivo de opciones:

-silent
Elimina la interfaz gráfica de usuario (GUI). Especifique el parámetro **-options** para indicar que el instalador completa la instalación de acuerdo con un archivo de opciones personalizadas. Si no especifica el parámetro **-options**, en su lugar se utilizan los valores predeterminados.

Ejemplo de uso

```
./install.sh|bat -silent -options archivo_opciones.txt
```

-options *nombre_vía/nombre_archivo*
Especifica un archivo de opciones que utiliza el instalador para completar una instalación silenciosa. Las propiedades de la línea de mandatos tienen preferencia.

Ejemplo de uso

```
./install.sh|bat -options c:/nombre_vía/archivo_opciones.txt
```

-log # !nombre_archivo @*tipo_suceso*
Genera un archivo de registro de instalación que anota los siguientes tipos de sucesos:

- err
- wrn
- msg1
- msg2
- dbg
- ALL

Ejemplo de uso

```
./install.sh|bat -log # !c:/temp/logfiles.txt @ALL
```

-is:log *nombre_vía/nombre_archivo*
Crea un archivo de registro que contiene las búsquedas de la máquina virtual Java (JVM) del instalador mientras intenta iniciar la GUI. El archivo de registro no se crea, a menos que se especifique.

Ejemplo de uso

```
./install.sh|bat -is:log c:/logs/javalog.txt
```

-is:javaconsole
Visualiza una ventana de consola durante el proceso de instalación.

Ejemplo de uso

```
./install.sh|bat -is:javaconsole
```

-is:silent
Elimina la ventana de inicialización de Java que se visualiza cuando se inicia el instalador.

Ejemplo de uso

```
./install.sh|bat -is:silent
```

-is:tempdir *nombre_vía*
Especifica el directorio temporal que utiliza el instalador durante la instalación.

Ejemplo de uso

```
./install.sh|bat -is:tempdir c:/temp
```

Instalación del servicio de datos REST

Este tema describe cómo instalar el servicio de datos REST de WebSphere eXtreme Scale en un servidor web.

Antes de empezar

Requisitos de software

El servicio de datos REST de WebSphere eXtreme Scale es una aplicación web Java que se puede desplegar en cualquier servidor de aplicaciones que soporte la especificación de servlet Java, Versión 2.3 y un entorno de tiempo de ejecución Java, Versión 5 o posterior.

Se necesita el software siguiente:

- Java Standard Edition 5 o posterior
- Contenedor de servlet web, Versión 2.3 o posterior, que incluye uno de los siguientes:
  - WebSphere Application Server Versión 6.1.0.25 o posterior
  - WebSphere Application Server Versión 7.0.0.5 o posterior
  - WebSphere Community Edition Versión 2.1.1.3 o posterior
  - Apache Tomcat Versión 5.5 o posterior
- WebSphere eXtreme Scale, Versión 7.1 o posterior, incluyendo la versión de prueba.

Acerca de esta tarea

El servicio de datos REST de WebSphere eXtreme Scale incluye un único archivo `wxsrestservice.war`. El archivo `wxsrestservice.war` incluye un único servlet que actúa como una pasarela entre las aplicaciones cliente de WCF Data Services o cualquier otro cliente HTTP REST y una cuadrícula de datos.

El servicio de datos REST incluye un ejemplo que le permite crear rápidamente una cuadrícula de datos e interactuar con ella utilizando un cliente eXtreme Scale o el servicio de datos REST. Consulte “Configuración de servicios de datos REST” en la página 357 para obtener detalles sobre la utilización del ejemplo.

Cuando se instala WebSphere eXtreme Scale 7.1 o se extrae la versión de evaluación 7.1 de eXtreme Scale, se incluyen los directorios y los archivos siguientes:

- `inicio_restservice/lib`

  El directorio lib contiene estos archivos:

- `wxsrestservice.ear` – Es el archivo de aplicación empresarial del servicio de datos REST que se debe utilizar con WebSphere Application Server y WebSphere Application Server CE.
- `wxsrestservice.war` – Es el módulo web del servicio de datos REST que se debe utilizar con Apache Tomcat.

El archivo `wxsrestservice.ear` incluye el archivo `wxsrestservice.war` y los dos están estrechamente vinculados con el tiempo de ejecución de WebSphere WebSphere eXtreme Scale. Si se actualiza WebSphere eXtreme Scale a una versión nueva o se aplica un fixpack, los archivos `wxsrestservice.war` o `wxsrestservice.ear` se deberán actualizar manualmente a la versión instalada en este directorio.

- `inicio_restservice/gettingstarted`

  El directorio `gettingstarted` contiene un ejemplo simple que muestra cómo utilizar el servicio de datos REST de WebSphere eXtreme Scale con una cuadrícula de datos.

Procedimiento

Empaquete y despliegue el servicio de datos REST.
El servicio de datos REST está diseñado como un módulo WAR autocontenido. Para configurar el servicio de datos REST, en primer lugar debe empaquetar la configuración del servicio de datos REST y los archivos de configuración de WebSphere eXtreme Scale opcionales en un directorio o archivo JAR. El tiempo de ejecución del servidor de contenedores web hace referencia entonces a este empaquetado de aplicaciones. En el diagrama siguiente se ilustran los archivos utilizados por el servicio de datos REST de eXtreme Scale.
El JAR o el directorio de configuración del servicio REST deben contener el archivo

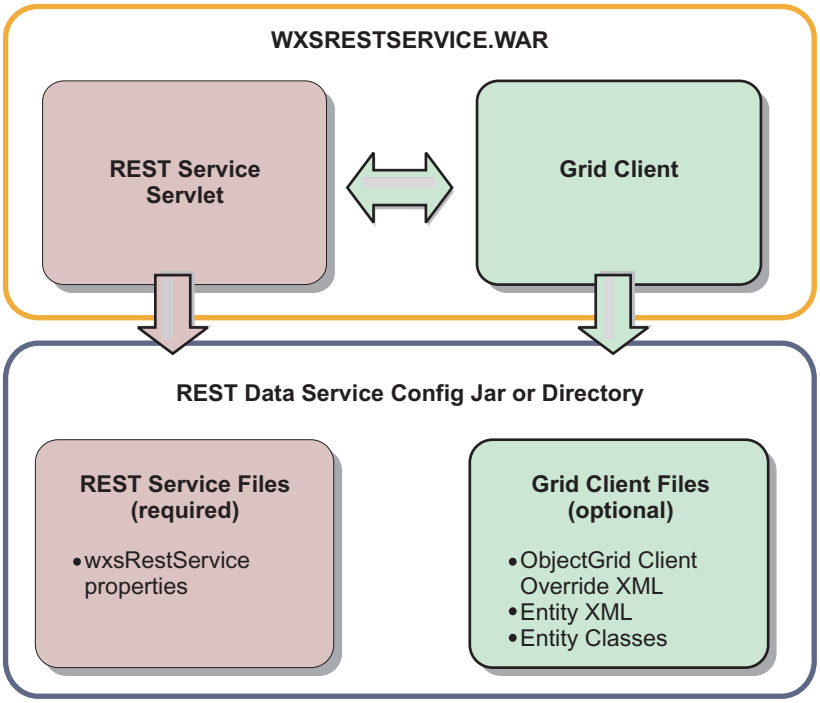


Figura 26. Archivos del servicio de datos REST de WebSphere eXtreme Scale

siguiente:
`wxsRestService.properties`: El archivo `wxsRestService.properties` incluye las opciones de configuración del servicio de datos REST. Incluye los puntos finales de

servicio de catálogo, nombres de ObjectGrid que se deben exponer, opciones de rastreo, etc. Consulte Archivo de propiedades del servicio de datos REST.
Los archivos de cliente de ObjectGrid siguientes son opcionales:

- `META-INF/objectGridClient.xml`: se utiliza el cliente XML de sustitución de cliente de ObjectGrid para conectarse a la cuadrícula de datos remota. De forma predeterminada, este archivo no es necesario. Si no se encuentra presente este archivo, el servicio REST utilizará la configuración del servidor, inhabilitando la memoria caché cercana.

  El nombre del archivo se puede sustituir utilizando la propiedad de configuración del servicio de datos REST nombreobjectGridClientXML. Si se proporciona, este archivo XML debe incluir:

  1. Los ObjectGrids que desee exponer al servicio de datos REST.
  2. Las referencias al archivo XML de descriptor de entidad asociado con la configuración de cada ObjectGrid.
- `META-INF/`*`archivos XML de descriptor de entidad`*: se necesitan uno o varios archivos XML de descriptor de entidad sólo si el cliente necesita sustituir la definición de entidad del cliente. El archivo XML descriptor de entidad se debe utilizar junto con el archivo descriptor XML de sustitución del cliente ObjectGrid.
- **Clases de entidad** Se pueden utilizar clases de entidad anotadas o un archivo XML descriptor de entidad para describir los metadatos de entidad. El servicio REST sólo requiere clases de entidad en la classpath si los servidores de eXtreme Scale están configurados con clases de metadatos de entidad y no se utiliza un descriptor XML de entidad de sustitución de cliente.

  Un ejemplo con el archivo de configuración mínimo necesario, donde las entidades están definidas en XML en los servidores:

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  El archivo de propiedades contiene:

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid
  ```

  Un ejemplo con una entidad, archivos XML de sustitución y clases de entidad:

  ```
  restserviceconfig.jar:
  wxsRestService.properties
  ```

  El archivo de propiedades contiene:

  ```
  catalogServiceEndPoints=localhost:2809
  objectGridNames=NorthwindGrid

  com/acme/entities/Customer.class
  META-INF/objectGridClient.xml
  ```

  El archivo XML de descriptor de ObjectGrid de cliente contiene:

  ```
  <objectGrid name="CustomerGrid" entityMetadataXMLFile="emd.xml"/>
  META-INF/emd.xml
  ```

  El archivo XML de descriptor de metadatos de entidad contiene:

  ```
  <entity class-name="com.acme.entities.Customer" name="Customer"/>
  ```

Instalación de la infraestructura OSGi de Eclipse Equinox con Eclipse Gemini para clientes y servidores

Si desea desplegar WebSphere eXtreme Scale en una infraestructura OSGi, debe configurar el entorno de Eclipse Equinox.

Acerca de esta tarea

La tarea requiere que descargue e instale la infraestructura Blueprint, lo que le permite configurar posteriormente JavaBeans y exponerlos como servicios. El uso de los servicios es importante porque puede exponer plug-ins como servicios OSGi de forma que los pueda utilizar el entorno de ejecución de eXtreme Scale. El producto da soporte a dos contenedores blueprint en la infraestructura OSGi principal de Eclipse Equinox: Eclipse Gemini y Apache Aries. Utilice este procedimiento para configurar el contenedor Eclipse Gemini.

Procedimiento

1. Descargue Eclipse Equinox SDK Versión 3.6.1 o posterior del sitio web de Eclipse. Cree un directorio para la infraestructura Equinox, por ejemplo: `/opt/equinox`. Estas instrucciones hacen referencia a este directorio como `raíz_equinox`. Extraiga el archivo comprimido en el directorio `raíz_equinox`.
2. Descargue el archivo comprimido de gemini-blueprint incubation 1.0.0 del sitio web de Eclipse. Extraiga el contenido del archivo en un directorio temporal y copie los siguientes archivos extraídos en el directorio `raíz_equinox/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. Descargue la infraestructura Spring versión 3.0.5 de la siguiente página web de SpringSource: http://www.springsource.com/download/community. Extráigala en un directorio temporal y copie los siguientes archivos extraídos en el directorio `raíz_equinox/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. Descargue el archivo de archivado Java archive (JAR) de AOP Alliance de la página web de SpringSource. Copie el archivo com.springsource.org.aopalliance-1.0.0.jar en el directorio `raíz_equinox/plugins`.
5. Descargue el archivo JAR de Apache Commons Logging 1.1.1 de la página web de SpringSource. Copie el archivo `com.springsource.org.apache.commons.logging-1.1.1.jar` en el directorio `raíz_equinox/plugins`.
6. Descargue el cliente de línea de mandatos de Luminis OSGi Configuration Admin. Utilice este paquete para gestionar las configuraciones administrativas de OSGi. Puede descargar el archivo JAR de la siguiente página web: https://opensource.luminis.net/wiki/display/SITE/OSGi+Configuration+Admin+command+line+client. Copie el archivo `net.luminis.cmc-0.2.5.jar` en el directorio `raíz_equinox/plugins`.
7. Descargue el paquete de instalación de archivos de la Versión 3.0.2 de Apache Felix de la siguiente página web: http://felix.apache.org/site/index.html. Copie el archivo `org.apache.felix.fileinstall-3.0.2.jar` en el directorio `raíz_equinox/plugins`.
8. Cree un directorio de configuración en el directorio `equinox_root/plugins`, por ejemplo:

   ```
   mkdir equinox_root/plugins/configuration
   ```

9. Cree el archivo `config.ini` siguiente en el directorio `equinox_root/plugins/configuration`, sustituyendo *equinox_root* por la vía de acceso absoluta al directorio `equinox_root` y eliminando todos los espacios de cola después de la barra inclinada invertida de cada línea. Debe incluir una línea en blanco al final del archivo; por ejemplo:

```
osgi.noShutdown=true
osgi.java.profile.bootdelegation=none
org.osgi.framework.bootdelegation=none
eclipse.ignoreApp=true
osgi.bundles=\
org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
com.springsource.org.apache.commons.logging-1.1.1.jar@1:start, \
com.springsource.org.aopalliance-1.0.0.jar@1:start, \
org.springframework.aop-3.0.5.RELEASE.jar@1:start, \
org.springframework.asm-3.0.5.RELEASE.jar@1:start, \
org.springframework.beans-3.0.5.RELEASE.jar@1:start, \
org.springframework.context-3.0.5.RELEASE.jar@1:start, \
org.springframework.core-3.0.5.RELEASE.jar@1:start, \
org.springframework.expression-3.0.5.RELEASE.jar@1:start, \
org.apache.felix.fileinstall-3.0.2.jar@1:start, \
net.luminis.cmc-0.2.5.jar@1:start, \
gemini-blueprint-core-1.0.0.jar@1:start, \
gemini-blueprint-extender-1.0.0.jar@1:start, \
gemini-blueprint-io-1.0.0.jar@1:start
```

   Si ya ha configurado el entorno, puede limpiar el repositorio de plug-ins de Equinox eliminando el directorio siguiente: `raíz_equinox\plugins\configuration\org.eclipse.osgi`.

10. Ejecute los mandatos siguientes para iniciar la consola de equinox.

    Si está ejecutando una versión distinta de Equinox, el nombre de archivo JAR será distinto al del ejemplo siguiente:

    ```
    java -jar plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
    ```

Instalación de paquetes de eXtreme Scale

WebSphere eXtreme Scale incluye paquetes que se pueden instalar en una infraestructura OSGi de Eclipse Equinox. Estos paquetes son necesarios para iniciar los servidores eXtreme Scale o utilizar clientes de eXtreme Scale en OSGi.

Antes de empezar

En esta tarea se supone que se han instalado los productos siguientes:

- Infraestructura OSGi de Eclipse Equinox
- Cliente o servidor autónomo de eXtreme Scale

Acerca de esta tarea

eXtreme Scale incluye dos paquetes. Sólo se necesita uno de los paquetes siguientes en una infraestructura OSGi:

objectgrid.jar
: El paquete de servidor es el archivo `objectgrid.jar` que se instala con la instalación de servidor autónomo de eXtreme Scale, es necesario para ejecutar servidores eXtreme Scale y también se puede utilizar para ejecutar clientes eXtreme Scale o cachés locales en memoria. El ID de paquete para el archivo `objectgrid.jar` es com.ibm.websphere.xs.server_<versión>, donde la versión tiene el formato: <Versión>.<Release>.<Modificación>. Por ejemplo, el paquete de servidor para eXtreme Scale versión 7.1.1 es com.ibm.websphere.xs.server_7.1.1.

ogclient.jar
: El paquete `ogclient.jar` se instala con las instalaciones autónomas y de cliente de eXtreme Scale y se utiliza para ejecutar clientes de eXtreme Scale o cachés locales en memoria. El ID de paquete para el archivo

ogclient.jar es com.ibm.websphere.xs.client_<versión>, donde la versión está en el formato: <Versión>_<Release>_<Modificación. Por ejemplo, el paquete de cliente para eXtreme Scale Versión 7.1.1 es com.ibm.websphere.xs.client_7.1.1.

Para obtener más información sobre el desarrollo de plug-ins de eXtreme Scale, consulte el tema Plug-ins y API del sistema.

Procedimiento

Para instalar el paquete de cliente o servidor de eXtreme Scale en la infraestructura OSGi de Eclipse Equinox utilizando la consola de OSGi:

1. Inicie la infraestructura de Eclipse Equinox con la consola habilitada; por ejemplo:

   ```
   inicio_java/bin/java -jar <raíz_equinox>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```

2. Instale el paquete de cliente o servidor de eXtreme Scale en la consola de Equinox:

   ```
   osgi> install file:///<vía_acceso_archivo>
   ```

3. Equinox visualiza el ID de paquete para el paquete recién instalado:

   ```
   El ID de paquete es 25
   ```

4. Inicie el paquete en la consola de Equinox, donde <id> es el ID de paquete asignado al instalar el paquete:

   ```
   osgi>  start <id>
   ```

5. Recupere el estado de servicio en la consola de Equinox para verificar que el paquete se ha iniciado; por ejemplo:

   ```
   osgi> ss
   ```

   Cuando el paquete se ha iniciado satisfactoriamente, visualiza el estado ACTIVO; por ejemplo:

   ```
   25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
   ```

Instale el paquete de cliente o servidor de eXtreme Scale en la infraestructura OSGi de Eclipse Equinox utilizando el archivo config.ini:

6. Copie el paquete de cliente o servidor de eXtreme Scale (objectgrid.jar o ogclient.jar) del directorio <raíz_instalación_wxs>/ObjectGrid/lib en el directorio de plug-ins de Eclipse Equinox; por ejemplo: <raíz_equinox>/plugins
7. Edite el archivo de configuración config.ini de Eclipse Equinox y añada el paquete a la propiedad osgi.bundles; por ejemplo:

   ```
   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
   ```

   Importante: Verifique que haya una línea en blanco después del último nombre de paquete. Cada paquete está separado por una coma.

8. Inicie la infraestructura de Eclipse Equinox con la consola habilitada; por ejemplo:

   ```
   inicio_java/bin/java -jar <raíz_equinox>/plugins/
   org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```

9. Recupere el estado de servicio en la consola de Equinox para verificar que el paquete se ha iniciado:

```
osgi> ss
```

Cuando el paquete se haya iniciado satisfactoriamente, visualizará el estado ACTIVO; por ejemplo:

```
25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
```

Resultados

El paquete de servidor o cliente de eXtreme Scale se ha instalado e iniciado en la infraestructura OSGi de Eclipse Equinox.

Verificación de la instalación

Después de que se haya completado el asistente de instalación, puede verificar la instalación comprobando diversos aspectos de la instalación.

Procedimiento

- **Para una instalación autónoma o una instalación integrada con WebSphere Application Server:**

  Utilice uno de los métodos siguientes para verificar que la instalación se haya completado satisfactoriamente:

  – Ejecute el mandato de información de versión para WebSphere eXtreme Scale:

    ```
    raíz_was/lib/> java -jar wsobjectgrid.jar version
    ```

    Como resultado se visualizará el número de versión y el número de build del nombre del producto.

  – Compruebe los archivos de propiedades para ver el número de versión adecuado.

    - Archivos de firmas: los archivos de firmas se encuentran en el directorio *raíz_was*/properties/version. Si se ha instalado un fixpack, se incluyen también archivos fxtg adicionales. A continuación se muestran algunos ejemplos de nombres de archivo de firma:

      ```
      WebSphere_eXtreme_Scale.7.1.1..swtag
      WebSphere_eXtreme_Scale.7.1.0.2.fxtag
      WebSphere_eXtreme_Scale.7.1.0.3.fxtag
      ```

    - Archivo del producto de WebSphere eXtreme Scale:

      El archivo del producto se encuentra en el directorio *raíz_was*/properties/version. Busque el archivo WXS.product. A continuación se muestra un ejemplo del contenido del archivo:

      ```
      <?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>
      ```

  – Compruebe que se han instalado los archivos de tiempo de ejecución. Las listas de los archivos de tiempo de ejecución para cada tipo de instalación se documentan en los temas siguientes:

    - "Archivos de ejecución para una instalación autónoma de WebSphere eXtreme Scale" en la página 195

- "Archivos de ejecución de WebSphere eXtreme Scale integrado con WebSphere Application Server" en la página 165

- **Para una instalación integrada con WebSphere Application Server, tiene las siguientes maneras adicionales de comprobar si la instalación se ha completado satisfactoriamente:**
  - Ejecutar el mandato de información de versión para WebSphere Application Server:

    ```
    raíz_was/bin/> versionInfo.sh|.bat
    ```

    La salida visualiza una lista de los productos instalados, incluidos los directorios de instalación, los productos instalados, las versiones, el nivel de build, la fecha de build, etc.

    Consejo: Añada el parámetro **-maintenancePackages** para ver detalles adicionales:

    ```
    raíz_was/bin/> versionInfo.sh|.bat -maintenancePackages
    ```
  - Compruebe el panel de bienvenida de la consola administrativa de WebSphere Application Server. Vaya a `http://localhost:9060/ibm/console`. Inicie la sesión en la consola. La versión de WebSphere eXtreme Scale se visualiza en el panel de bienvenida.
  - Utilice la consola Primeros pasos para aumentar la instalación de WebSphere Application Server con WebSphere eXtreme Scale:

    ```
    raíz_was/firststeps/WXS> firststeps.sh|.bat
    ```

    Para obtener más información, consulte "Creación y aumento de perfiles para WebSphere eXtreme Scale" en la página 184.

Qué hacer a continuación

Si ve que la instalación no se ha completado, tal como creía, debe solucionar los problemas de la instalación. Para obtener más información, consulte "Resolución de problemas de la instalación" en la página 208.

Primeros pasos después de la instalación

Después de completar y verificar la instalación, puede empezar a utilizar WebSphere eXtreme Scale para crear los datos de la cuadrícula de datos.

Procedimiento

1. Actualice la instalación mediante la aplicación de mantenimiento.

   Más información: "Actualización de servidores eXtreme Scale" en la página 211.
2. Si utiliza WebSphere eXtreme Scale por primera vez, puede utilizar la información de Cómo empezar para obtener más información sobre cómo utilizar el producto.

   Más información: Capítulo 1, "Cómo empezar", en la página 1
3. Configure el producto. Cree archivos de propiedades y XML para definir la configuración para cuadrículas de datos, servidores y clientes. También puede configurar la integración de memoria caché o base de datos, los servicios de datos REST o plug-ins OSGi.

   Más información: Capítulo 6, "Configuración", en la página 223
4. Desarrolle una aplicación que acceda a la cuadrícula de datos.

Más información: Desarrollo de aplicaciones

5. Inicie y administre los servidores de contenedor y catálogo con los archivos de configuración y la aplicación de la cuadrícula de datos.

   Más información: Capítulo 7, “Administración”, en la página 395

6. Supervise el rendimiento de la configuración con diversas herramientas de supervisión.

   Más información: Capítulo 8, “Supervisión”, en la página 441

Resolución de problemas de la instalación

Utilice esta información para resolver problemas de la instalación.

Procedimiento

- **Problema:** al ejecutar el mandato de instalación desde un sistema remoto como, por ejemplo, `\\mymachine\downloads\`, se visualiza el mensaje siguiente: `CMD.EXE se ha iniciado con la vía de acceso anterior como directorio actual. No se da soporte a vías de acceso UNC. Se toma el directorio predeterminado de Windows.` Como resultado, la instalación no se completa correctamente.

  Solución: correlacione el sistema remoto con una unidad de red. Por ejemplo, en Windows, puede pulsar con el botón derecho del ratón en **Mi sistema** y elegir **Correlacionar unidad de red** e incluir la vía de acceso de convenio de nomenclatura uniforme (UNC) del sistema remoto. A continuación, puede ejecutar el script de instalación satisfactoriamente desde la unidad de red, por ejemplo: `y:\mymachine\downloads\WXS\install.bat`.

- **Problema:** la instalación no se completa satisfactoriamente.

  Solución: consulte los archivos de registro para ver dónde ha fallado la instalación. Cuando la instalación no se completa satisfactoriamente, los registros se encuentran en el directorio *raíz_intal_wxs*`/logs/wxs`.

- **Problema:** se produce una anomalía extremadamente grave durante la instalación.

  Solución: consulte los archivos de registro para ver dónde ha fallado la instalación. Cuando la instalación falla cuando se ha completado parcialmente, los registros normalmente se pueden encontrar en el directorio *raíz_usuario*`/wxs_install_logs/`.

- Windows **Problema:** si va a instalar WebSphere eXtreme Scale Client en Windows, podría ver el texto siguiente en los resultados de la instalación:

  ```
  Finalizado con éxito: Se ha instalado de forma satisfactoria el siguiente
  producto: cliente de WebSphere eXtreme Scale. Algunos pasos de la configuración
  han dado errores. Consulte el siguiente archivo de anotaciones cronológicas para
  obtener más información: <raíz de instalación de WebSphere Application
  Server>\logs\wxs_client\install\log.txt" Revise el archivo de registro
  de instalación (log.txt) y revise el registro de aumento del
  gestor de despliegue.
  ```

  Solución: si ve una anomalía del archivo `iscdeploy.sh`, puede ignorar el error. Este error no causa problemas.

Desinstalación de WebSphere eXtreme Scale

Para eliminar WebSphere eXtreme Scale del entorno, puede utilizar el asistente o puede desinstalar de forma silenciosa el producto.

Antes de empezar

Atención: El desinstalador elimina todos los archivos binarios y todo el mantenimiento como, por ejemplo, fixpacks y arreglos temporales, a la vez.

Procedimiento

1. Detenga todos los procesos que están ejecutando eXtreme Scale.

   PRECAUCIÓN:
   asegúrese de que todos los procesos en ejecución se hayan detenido. Si no se han detenido los procesos en ejecución, continúa la desinstalación, con lo que los resultados que se crean son imprevisibles y se deja la desinstalación en un estado indeterminado en algunas plataformas.

   - Si ha instalado eXtreme Scale autónomo, lea la información sobre cómo detener servidores autónomos para detener procesos.
   - Si ha instalado eXtreme Scale con una instalación existente de WebSphere Application Server, lea el apartado sobre los programas de utilidad de línea de mandatos si desea más información sobre cómo detener procesos WebSphere Application Server.
   - Si está ejecutando la consola web, utilice el mandato `stopConsoleServer` para detener el servidor de la consola web. El script `stopConsoleServer` se encuentra en el directorio *raíz_intal_wxs*`/ObjectGrid/bin`. Si no detiene este servidor antes de ejecutar la desinstalación, el proceso se detiene automáticamente durante el proceso de desinstalación.

2. Desinstale el producto. Puede ejecutar la desinstalación en una GUI o de forma silenciosa.

   Nota: Al especificar el archivo de respuestas `wxssetup.response` para la instalación o desinstalación silenciosa o en la GUI, se debe especificar siempre la vía de acceso completa. El archivo de respuestas es opcional para la desinstalación de la GUI.

   - **Para ejecutar la desinstalación con la GUI:**
     - Linux UNIX `<inicio_instalación>/uninstall_wxs/uninstall`
     - Windows `<inicio_instalación>\uninstall_wxs\uninstall.exe`

     Si desea ejecutar la desinstalación con la GUI y el archivo `wxssetup.response`, utilice uno de estos mandatos:

     - Linux UNIX

       ```
       <inicio_instalación>/uninstall_wxs/uninstall -options
       <vía_de_acceso_instalación_completa_necesaria>/wxssetup.response
       ```

     - Windows

       ```
       <inicio_instalación>\uninstall_wxs\uninstall.exe -options
       <vía_de_acceso_instalación_completa_necesaria>\wxssetup.response
       ```

   - **Para ejecutar la desinstalación silenciosa con el script de archivo de respuestas `wxssetup.response`:**
     - Linux UNIX

       ```
       <inicio_instalación>/uninstall_wxs/uninstall -options
       <vía_de_acceso_instalación_completa_necesaria>/wxssetup.response -silent
       ```

     - Windows

       ```
       <inicio_instalación>\uninstall_wxs\uninstall.exe -options
       <vía_de_acceso_instalación_completa_necesaria>\wxssetup.response -silent
       ```

Resultados

Ha eliminado eXtreme Scale del entorno.

Capítulo 5. Actualización y migración de WebSphere eXtreme Scale

Puede migrar a la Versión 7.1.1 desde versiones anteriores y puede aplicar paquetes de mantenimiento a la Versión 7.1.1. Para evitar paradas, debe tener en cuenta el orden en el que aplicar las actualizaciones a los servidores de la configuración.

- Para actualizar una instalación de la versión 7.1.0.x, consulte "Actualización de servidores eXtreme Scale" y "Utilización del instalador de actualización para instalar los paquetes de mantenimiento" en la página 215.
- Para actualizar una instalación de la versión 7.0.x, consulte "Actualización de servidores eXtreme Scale" y "Migración a WebSphere eXtreme Scale Versión 7.1.1" en la página 214.

Actualización de servidores eXtreme Scale

Puede actualizar WebSphere eXtreme Scale a una nueva versión, aplicando mantenimiento o instalando una nueva versión, sin interrupción del servicio.

Antes de empezar

Debe tener el archivo binario del release de versión principal o mantenimiento que desee aplicar. Puede obtener la información más reciente sobre los paquetes de release y mantenimiento disponibles en el portal de soporte de IBM para WebSphere eXtreme Scale.

Acerca de esta tarea

Para actualizar sin interrupción del servicio, actualice en primer lugar los servidores de catálogo. A continuación, actualice los servidores de contenedor y los clientes.

Procedimiento

1. Actualice el nivel de servicio de catálogo, repitiendo los pasos siguientes para cada servidor de catálogo de la cuadrícula de datos. Actualice el nivel de servicio de catálogo antes de actualizar los clientes o servidores de contenedor. Los servidores de catálogo individuales pueden interoperar con la compatibilidad de versión, así que puede aplicar actualizaciones a un servidor de catálogo cada vez sin interrumpir el servicio.
   a. Compruebe que el estado de quórum sea correcto. Ejecute el siguiente mandato:

      ```
      xsadmin -quorumStatus
      xscmd -c showQuorumStatus
      ```

      Este resultado indica que todos los servidores de catálogo están conectados.
   b. Si utiliza réplica multimaestro entre dos dominios de servicio de catálogo, deseche el enlace entre los dos dominios de servicio de catálogo mientras actualiza los servidores de catálogo.

      ```
      xsadmin –ch host -p 1099 -dismissLink nombre_dominio
      ```

      7.1.1+

```
xscmd –c dismissLink –cep host:2809 -fd nombre_dominio
```

Sólo necesita ejecutar este mandato desde uno de los dominios de servicio de catálogo para eliminar el enlace entre los dos dominios de servicio de catálogo.

c. Concluya uno de los servidores de catálogo. Puede utilizar el mandato **stopOgserver**, el mandato **xscmd -c teardown**, o concluir el servidor de aplicaciones en ejecución en el servicio de catálogo en WebSphere Application Server. No hay requisitos para el orden de detención de los servidores de catálogo, pero si se concluye el servidor de catálogo primario en último lugar se reducirá la rotación. Para determinar qué servidor de catálogo es el primario, busque el mensaje CWOBJ8106 en los archivos de registro. En condiciones normales, el quórum se mantiene cuando se concluye un servidor de catálogo, pero se recomienda consultar el estado del quórum después de cada conclusión mediante el mandato **xscmd -c showQuorumStatus**.

   Si utiliza el mandato **xscmd -c teardown**, puede filtrar los nombres de servidor. El mandato **stopOgServer** requiere que se especifique un nombre de servidor exacto o una lista de nombres de servidor para detener en paralelo. Debe agrupar el proceso de conclusión en lugar de llamar al proceso stop o teardown para muchos servidores en paralelo. Agrupando los servidores que se deben concluir, la cuadrícula de datos puede reaccionar a los servidores que se están concluyendo moviendo los fragmentos por la cuadrícula de datos. Puede utilizar uno de los mandatos siguientes para concluir los servidores:

   Puede proporcionar una lista específica de servidores para detener a los mandatos **stopOgServer** o **xscmd -c teardown**:

   ```
   stopOgServer <nombre_servidor>[,<nombre_servidor>]
   xsadmin –teardown <nombre_servidor>[,<nombre_servidor>]
   ```

   7.1.1+

   ```
   xscmd –c teardown -sl <nombre_servidor>[,<nombre_servidor>]
   ```

   Con los ejemplos anteriores, los mandatos **stopOgServer** o **xscmd -c teardown** completan las mismas tareas de conclusión. Sin embargo, puede filtrar los servidores para detener con el mandato **xscmd -c teardown**. Consulte “Detención correcta de servidores con el programa de utilidad **xscmd**” en la página 409 para obtener más información sobre cómo filtrar los servidores por zona o nombre de host. El mandato teardown filtra los servidores coincidentes y le solicita si los servidores seleccionados son correctos.

d. Instale las actualizaciones en el servidor de catálogo. Puede migrar el servidor de catálogo a un release principal nuevo del producto o aplicar un paquete de mantenimiento. Consulte los temas siguientes para obtener más información:

   - Para migrar desde una instalación de la versión 7.0.x: “Migración a WebSphere eXtreme Scale Versión 7.1.1” en la página 214
   - Para actualizar desde una instalación de la versión 7.1.0.x: “Utilización del instalador de actualización para instalar los paquetes de mantenimiento” en la página 215

e. Reinicie el servidor de catálogo.

   Si está utilizando un entorno autónomo, consulte “Inicio de un servicio de catálogo autónomo” en la página 395 para obtener más información. Si está utilizando un entorno de WebSphere Application Server, consulte “Inicio y

detención de servidores en un entorno de WebSphere Application Server" en la página 409 para obtener más información.

El servidor de catálogo se ejecuta en modalidad de compatibilidad hasta que todos los servidores de catálogo se mueven al mismo nivel. La modalidad de compatibilidad se aplica principalmente a migraciones de release principales ya que las nuevas funciones no están disponibles en los servidores que no se migran. No existen restricciones sobre cuánto tiempo los servidores de catálogo pueden ejecutarse en modalidad de compatibilidad, pero el método recomendado es migrar todos los servidores de catálogo al mismo nivel lo antes posible.

f. Aplique actualizaciones a los demás servidores de catálogo de la configuración.

2. Actualice los servidores de contenedor, repitiendo los pasos siguientes para cada servidor de contenedor de la cuadrícula de datos. Puede actualizar los servidores de contenedor en cualquier orden. Sin embargo, considere actualizar en primer lugar los servidores y a continuación los clientes si utiliza nuevas funciones en la actualización.

   a. Detenga los servidores de contenedor que desea actualizar. Puede detener el nivel de servidor de contenedor en los grupos con el mandato **stopOgserver** o el mandato **teardown**. Mediante el proceso por lotes de las operaciones teardown y la ejecución de operaciones de inicio de servidor en paralelo, el mecanismo de colocación puede mover los fragmentos en grupos más grandes.

   ```
   xsadmin –teardown -fz DefaultZone
   ```

   7.1.1+

   ```
   xscmd -c teardown -z DefaultZone

   Conexión al servicio de catálogo en localhost:1099

   Procesando opciones de filtro para desmontaje del servidor

   Se ejecutará tear down en los servidores siguientes:

     container00
     container01
     container02
     container03
     container04


   ¿Desea desmontar los servidores listados? (S/N)
   ```

   b. Instale las actualizaciones en los servidores de contenedor. Puede migrar los servidores de contenedor a un nuevo release principal del producto o aplicar un paquete de mantenimiento. Consulte los temas siguientes para obtener más información:

      - Para migrar desde una instalación de la versión 7.0.x: "Migración a WebSphere eXtreme Scale Versión 7.1.1" en la página 214
      - Para actualizar desde una instalación de la versión 7.1.0.x: "Utilización del instalador de actualización para instalar los paquetes de mantenimiento" en la página 215

   c. Reinicie los servidores de contenedor.

   d. Actualice los demás servidores de contenedor de la configuración.

3. Si utiliza réplica multimaestro, vuelva a conectar los dominios de servicio de catálogo. Utilice el mandato **xscmd -c establishLink** para volver a conectar los dominios de servicio de catálogo. **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
```

Qué hacer a continuación

También puede utilizar estos pasos para volver a una versión anterior o para desinstalar paquetes de mantenimiento. Sin embargo, si vuelve a la versión 7.1.0 cuando utiliza réplica multimaestro, es posible que la réplica bidireccional no funcione correctamente al volver a establecer los enlaces. En esta situación, reinicie ambos dominios de servicio de catálogo y vuelva a enlazar los dominios de servicio de catálogo con el mandato **establishLink**.

Migración a WebSphere eXtreme Scale Versión 7.1.1

Con el instalador de WebSphere eXtreme Scale, no puede actualizar ni modificar una instalación anterior. Debe desinstalar la versión anterior antes de instalar la nueva versión. No tiene que migrar los archivos de configuración porque son compatibles con versiones anteriores. No obstante, si ha cambiado cualquiera de los archivos de script que se envían con el producto, debe volver a aplicar estos cambios en los archivos de script actualizados.

Antes de empezar

Verifique que los sistemas cumplen los requisitos mínimos para las versiones de producto que tiene previsto migrar e instalar. Consulte “Requisitos de hardware y software” en la página 49 para obtener más información.

Acerca de esta tarea

Fusione los archivos de script de producto modificados con los nuevos archivos de script de producto en el directorio `/bin` para mantener los cambios.

Consejo: Si no ha modificado los archivos de script que se instalan con el producto, no es necesario completar los pasos de migración siguientes. En su lugar, puede actualizar a la versión 7.1.1 desinstalando la versión anterior e instalando la nueva versión en el mismo directorio.

Procedimiento

1. Detenga todos los procesos que utilizan eXtreme Scale.
   - Lea la información sobre cómo detener servidores autónomos para detener todos los procesos que se ejecutan en el entorno de eXtreme Scale.
   - Lea la información sobre los programas de utilidad de línea de mandatos para detener todos los procesos que se ejecutan en el entorno WebSphere Application Server o WebSphere Application Server Network Deployment.
2. Guarde los scripts modificados del directorio de instalación actual en un directorio temporal.
3. Desinstale el producto.
4. Instale eXtreme Scale Versión 7.1.1. Si desea más información, consulte “Instalación de WebSphere eXtreme Scale con el asistente de instalación” en la página 163.
5. Fusione los cambios de los archivos en el directorio temporal con los nuevos archivos de script de producto en el directorio `/bin`.

6. Inicie todos los procesos de eXtreme Scale para empezar a utilizar el producto. Consulte Capítulo 7, “Administración”, en la página 395 para obtener más información.

Utilización del instalador de actualización para instalar los paquetes de mantenimiento

Utilice IBM Update Installer para actualizar el entorno de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con distintos tipos de mantenimiento como, por ejemplo, arreglos temporales, fixpacks y paquetes de renovación.

Acerca de esta tarea

Utilice IBM Update Installer para instalar y aplicar distintos tipos de paquetes de mantenimiento para WebSphere eXtreme Scale o WebSphere eXtreme Scale Client. Puesto que el instalador de actualización realiza mantenimientos regulares, debe utilizar la versión más actual de la herramienta.

Procedimiento

1. Detenga todos los procesos que se están ejecutando en el entorno.
   - Para detener todos los procesos que se están ejecutando en el entorno autónomo de eXtreme Scale, consulte “Detención de servidores autónomos” en la página 406.
   - Para detener todos los procesos que se están ejecutando en el entorno de WebSphere Application Server, consulte Programas de utilidad de línea de mandatos.
2. Descargue la versión más reciente del instalador de actualización. Consulte los Arreglos recomendados si desea más información.
3. Instale el instalador de actualización. Consulte Instalación de Update Installer para software WebSphere en el Information Center de WebSphere Application Server para obtener más información.
4. Descargue los paquetes de mantenimiento en el directorio *raíz_updi*/`maintenance` en el que tiene previsto realizar la instalación. Consulte el Sitio de soporte si desea más información.
5. Utilice el instalador de actualización para instalar el arreglo temporal, el fixpack o el paquete de renovación. Puede instalar el paquete de mantenimiento ejecutando la interfaz gráfica de usuario (GUI), o ejecutando el instalador de actualización en la modalidad silenciosa.

   Ejecute el siguiente mandato desde el directorio *raíz_updi* para iniciar la GUI:

   - Linux UNIX `update.sh`
   - Windows `update.bat`

   Ejecute el siguiente mandato desde el directorio *raíz_updi* para ejecutar el instalador de actualización en la modalidad silenciosa:

   - Linux UNIX `./update.sh -silent -options` *responsefile/nombre_archivo*
   - Windows `update.bat -silent -options` *responsefile\nombre_archivo*

   Si falla el proceso de instalación, consulte el archivo de registro temporal, que está en el directorio *raíz_updi*`/logs/update/tmp`. El instalador de actualización crea el directorio *raíz_instalación*`/logs/update/`*paquete_mantenimiento*`.install` en el que se encuentran los archivos de registro de la instalación.

Migración de la herramienta xsadmin a la herramienta xscmd

En releases anteriores, la herramienta **xsadmin** era un programa de utilidad de línea de mandatos de ejemplo para supervisar el estado del entorno. La herramienta **xscmd** se ha presentado como una herramienta de línea de mandatos soportada oficialmente de supervisión y administración. Si utilizaba anteriormente la herramienta **xsadmin**, considere migrar los mandatos a la nueva herramienta **xscmd**.

xsadmin y mandatos xscmd equivalentes

Tabla 9. Argumentos del programa de utilidad **xsadmin** *y de los mandatos* **xscmd** *equivalentes.* Algunos mandatos **xscmd** tienen un formato abreviado y una formato largo. Los mandatos con formato abreviado tienen un guión (-) y los mandatos con formato largo tienen dos guiones (--). Puede utilizar cualquiera de los dos formatos indistintamente.

Argumento de la línea de mandatos de xsadmin	Mandato xscmd equivalente	Parámetros del mandato xscmd
-bp	• **-cep** *nombre_host:puerto_escucha* • **--catalogEndpoint** *nombre_host:puerto_escucha*	n/d
-ch	• **-cep***nombre_host:puerto_escucha* • **--catalogEndpoint** *nombre_host:puerto_escucha*	n/d
-clear	**-c clearGrid**	**-g, -ms, -v, -m, (-cep)**
-containers	• **-c listCoreGroups** • **-c listCoreGroupMembers -cg** *grupo_principal*	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-continuous	• **-cnt** • **--continuous**	n/d
-coregroups	• **-c listCoreGroups** • **-c listCoreGroupMembers -cg** *grupo_principal*	n/d
-dismissLink *<dominio_servicio_catálogo>*	**-c dismissLink**	• **-fd** *<dominioServicioCatálogoForáneo>* • **--foreignCatalogServiceDomain** *<dominioServicioCatálogoForáneo>*
-dmgr	n/a - este argumento se determina automáticamente con **xscmd**	n/d
-empties	argumento específico de un nuevo mandato	n/d
-establishLink *<nombre_dominio_foráneo> <host1:puerto1,host2:puerto2...>*	**-c establishLink**	• **-fd** *<dominioServicioCatálogoForáneo>* **-fe** *<host1:puerto1,host2:puerto2...>* • **--foreignCatalogServiceDomain** *<dominioServicioCatálogoForáneo>* **-foreignEndPoints** *<host1:puerto1,host2:puerto2...>*
-fc	• **-ct** • **--container**	n/d
-fh	• **-hf** • **--hostFilter**	n/d

Tabla 9. Argumentos del programa de utilidad **xsadmin** *y de los mandatos* **xscmd** *equivalentes (continuación).* Algunos mandatos **xscmd** tienen un formato abreviado y una formato largo. Los mandatos con formato abreviado tienen un guión (-) y los mandatos con formato largo tienen dos guiones (--). Puede utilizar cualquiera de los dos formatos indistintamente.

Argumento de la línea de mandatos de xsadmin	Mandato xscmd equivalente	Parámetros del mandato xscmd
-fm	• **-m** • **--map**	n/d
-fnp	• **-snp** • **--serversWithNoPrimaries**	n/d
-fp	• **-p** • **--partitionId**	n/d
-fs	• **-s** • **--server**	n/d
-fst	• **-st** <*tipo_fragmento*> • **--shardType** <*tipo_fragmento*> Valores de fragmento: P=primario A=réplicaAsín S=réplicaSín	n/d
-fz	• **-z** • **--zone**	n/d
-force	argumento específico de un nuevo mandato	
-g	• **-g** • **--objectGrid**	n/d
-getstatsspec	**-c getStatsSpec**	n/d
-getTraceSpec	**-c getTraceSpec**	n/d
-h	Puede ejecutar la ayuda con o sin un nombre de mandato específico: • **-h** • **--help** • **-h** <*nombre_mandato*> • **--help** <*nombre_mandato*>	n/d
-hosts	**-c listHosts**	**-g, -ms, -st, -c, -s, -hf, -z**
-jmxUrl	• **-cep** *nombre_host*:*puerto_escucha* • **--catalogEndpoint** *nombre_host*:*puerto_escucha*	n/d
-l	**-c listObjectGridNames**	n/d
-m	• **-ms** • **--mapSet**	n/d
-mapsizes	**-c showMapSizes**	**-g, -ms, -cnt, -i, [-ct, -z, -s, -hf, sht [P,A,S], -p]**
-mbeanservers	**-c listAllJMXAddresses**	n/d
-overridequorum	**-c overrideQuorum**	n/d

Tabla 9. Argumentos del programa de utilidad **xsadmin** *y de los mandatos* **xscmd** *equivalentes (continuación).* Algunos mandatos **xscmd** tienen un formato abreviado y una formato largo. Los mandatos con formato abreviado tienen un guión (-) y los mandatos con formato largo tienen dos guiones (--). Puede utilizar cualquiera de los dos formatos indistintamente.

Argumento de la línea de mandatos de xsadmin	Mandato xscmd equivalente	Parámetros del mandato xscmd
-password	• **-pwd** • **--password**	n/d
-p	• **-cep** *nombre_host:puerto_escucha* • **--catalogEndpoint** *nombre_host:puerto_escucha*	n/d
-placementStatus	**-c placementServiceStatus**	**-g, -ms**
-primaries	**-c showPlacement -sf P**	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-profile	Para guardar los valores de seguridad actuales como un perfil de seguridad: • **-ssp** *nombre_perfil* • **--saveSecProfile** *nombre_perfil* Para utilizar un perfil de seguridad especificado: • **-sp** *nombre_perfil* • **--securityProfile** *nombre_perfil*	
-quorumstatus	**-c showQuorumStatus**	n/d
-releaseShard <*nombre_servidor_contenedor*> <*nombre_objectgrid*> <*nombre_conjunto_correlaciones*> <*nombre_partición*>	**-c releaseShard**	**-c, -g, -ms, -p**
-reserved	• **-sf** *[R,U]* • **--shardFilter** *[R,U]* R=reservado, U=desasignado	n/d
-reserveShard <*nombre_servidor_contenedor*> <*nombre_objectgrid*> <*nombre_conjunto_correlaciones*> <*nombre_partición*>	**-c reserveShard**	**-c, -g, -ms, -p**
-resumeBalancing <*nombre_objectgrid*> <*nombre_conjunto_correlaciones*>	**-c resumeBalancing**	**-g, -ms**
-revisions	**-c revisions**	**-s,-p,-g,-m**
-routetable	**-c routetable**	**-z, -hf,-p,-g,-ms**
-settracespec <*serie_rastreo*>	**-c setTraceSpec**	**-spec** <*serie_rastreo*>
-swapShardWithPrimary <*nombre_servidor_contenedor*> <*nombre_objectgrid*> <*nombre_conjunto_correlaciones*> <*nombre_partición*>	**-c swapShardWithPrimary**	**-c -g, -ms, -p**
-setstatsspec <*espec_estad*>	**-c setStatsSpec**	**-spec** <*espec_estad*>

Tabla 9. Argumentos del programa de utilidad **xsadmin** *y de los mandatos* **xscmd** *equivalentes (continuación).* Algunos mandatos **xscmd** tienen un formato abreviado y una formato largo. Los mandatos con formato abreviado tienen un guión (-) y los mandatos con formato largo tienen dos guiones (--). Puede utilizar cualquiera de los dos formatos indistintamente.

Argumento de la línea de mandatos de xsadmin	Mandato xscmd equivalente	Parámetros del mandato xscmd
-suspendBalancing *<nombre_objectgrid> <nombre_conjunto_correlaciones>*	**-c suspendBalancing**	**-g, -ms**
-ssl	• **-ssl** • **--enableSSL**	n/d
-teardown	**-c teardown**	**-f, , -st, -snp, -c, -s, -p, -hf, -z, -g, -ms, -m**
-triggerPlacement	**-c triggerPlacement**	**-g, -ms**
-trustPass	• **-tsp** • **--trustStorePassword**	n/d
-trustPath	• **-ts** • **--trustStore**	n/d
-trustType	• **-tst** • **--trustStoreType**	n/d
-unassigned	**-c showPlacement -sf U**	**-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms**
-username	• **-user** • **--username**	n/d
-v	• **-v** • **--verbose**	n/d
-xml	**-c showPlacement**	n/d

Propiedades y API en desuso

La lista siguiente muestra las propiedades y API que quedaron en desuso en el release de la Versión 7.1.1. Utilice la acción de migración recomendada para determinar cómo se debe actualizar la configuración.

7.1.1+ Elementos en desuso en la Versión 7.1.1

Tabla 10. Propiedades y API en desuso

En desuso	Acción de migración recomendada
Clase com.ibm.websphere.objectgrid.plugins.builtins.TranPropListener Esta clase se utilizaba para propagar procesos de confirmación de transacción de ObjectGrid satisfactorios a otros servidores de aplicaciones WebSphere que alojaran la misma instancia de ObjectGrid, según el nombre del ObjectGrid.	7.1.1+ La interfaz TranPropListener ha sido sustituida por la interfaz JMSObjectGridEventListener, que es una implementación basada en JMS de la interfaz ObjectGridEventListener. Da soporte a la réplica de igual a igual y a la invalidación de memoria caché cercana del lado del cliente.
Clase com.ibm.websphere.objectgrid.plugins.OptimisticCallback Esta clase se utilizaba para proporcionar operaciones de comparación optimista para los valores de una correlación.	7.1.1+ El plug-in OptimisticCallback ha sido sustituido por la interfaz ValueDataSerializer.Versionable, que puede implementar al utilizar el plug-in DataSerializer con la modalidad de copia COPY_TO_BYTES o al utilizar la anotación @Version con la API EntityManager. Consulte la documentación de la API para obtener más información.

Tabla 10. Propiedades y API en desuso (continuación)

En desuso	Acción de migración recomendada
Plug-in com.ibm.websphere.objectgrid.plugins.NoVersioningOptimisticCallback Este plug-in se utilizaba para bloqueo optimista sin realizar la comprobación de versión. Con este manejador OptimisticCallback incorporado, el cargador manejaba la comprobación de versión pero se utilizaba el bloqueo optimista para garantizar que los datos confirmados siempre se devolvían en una lectura.	7.1.1+ La interfaz NoVersioningOptimisticCallback amplía la interfaz OptimisticCallback. Por lo tanto, utilice la estrategia de bloqueo pesimista con un aislamiento de transacción predeterminado de READ_COMMITTED o inferior. Consulte Ajuste del rendimiento de bloqueo para obtener más información.
Clase com.ibm.websphere.objectgrid.plugins.ObjectTransformer Se utilizaba este plug-in para serializar, deserializar y copiar objetos en la memoria caché.	7.1.1+ La interfaz ObjectTransformer ha sido sustituida por los plug-ins DataSerializer, que se pueden utilizar para almacenar de forma eficiente datos arbitrarios en WebSphere eXtreme Scale de forma que las API existentes del producto puedan interactuar eficientemente con los datos.
Método com.ibm.websphere.objectgrid.BackingMap.setMapEventListeners Este método se utilizaba para establecer la lista de objetos MapEventListener.	7.1.1+ Utilice los métodos addMapEventListener(EventListener) o removeMapEventListener(EventListener) para añadir o eliminar escuchas de suceso desde una correlación de respaldo.
Método com.ibm.websphere.objectgrid.ObjectGrid.setEventListeners Este método se utilizaba para sobrescribir la lista actual de objetos ObjectGridEventListener y sustituirla por la lista proporcionada de objetos ObjectGridEventListeners.	7.1.1+ Utilice los métodos addEventListener(EventListener) o removeEventListener(EventListener) para añadir o eliminar escuchas de sucesos o escuchas de ciclo de vida en la cuadrícula de datos.

7.1.1+ Características estabilizadas en la versión 7.1.1

Si una característica aparece como estabilizada, IBM no tiene previsto actualmente dejar en desuso o eliminar esta prestación en un release posterior del producto; pero las inversiones futuras se centrarán en la función alternativa. Los usuarios no necesitan cambiar las aplicaciones y los scripts existentes que utilizan una función estabilizada; pero deberán tener en cuenta la posibilidad de utilizar la alternativa estratégica para aplicaciones nuevas.

Tabla 11. Propiedades y API en desuso

Característica estabilizada	Acción de migración recomendada
xsadmin El programa de utilidad xsadmin se proporciona como un ejemplo de cómo crear programas de utilidad personalizados para el despliegue.	7.1.1+ Utilice el programa de utilidad **xscmd** para completar tareas administrativas en el entorno, por ejemplo: establecer enlaces de réplica multimaestro, alterar temporalmente el quórum y detener grupos de servidores con el mandato teardown.

Elementos en desuso en la Versión 7.1

Tabla 12. Propiedades y API en desuso

En desuso	Acción de migración recomendada
Propiedad de célula o servidor catalog.services.cluster: esta propiedad personalizada se ha utilizado para definir un grupo de servidores de catálogo en la configuración de WebSphere Application Server.	Esta propiedad personalizada está en desuso a partir del release de la Versión 7.1. Cree un dominio de servicio de catálogo en la consola administrativa de WebSphere Application Server, que crea la misma configuración que si se utiliza la propiedad personalizada. Si desea más información, consulte .
MBean e interfaz CoreGroupServicesMBean	Este MBean está en desuso a partir del release de la Versión 7.1. Utilice CatalogServiceManagementMBean en su lugar.

Tabla 12. Propiedades y API en desuso (continuación)

En desuso	Acción de migración recomendada
Operación de MBean ServerMBean.updateTraceSpec()	Esta operación está en desuso a partir del release de la Versión 7.1. Utilice el atributo TraceSpec en DynamicServerMBean en su lugar.
MBean CoreGroupServicesMBean	Este MBean está en desuso a partir del release de la Versión 7.1. Utilice en su lugar el MBean CatalogServiceManagementMbean.
Excepción ServiceUnavailableException	Esta excepción está en desuso a partir del release de la Versión 7.1. Utilice la excepción TargetNotAvailableException en su lugar.
Recurso de partición (WPF): el recurso de partición es un conjunto de API de programación que permiten a las aplicaciones Java EE dar soporte a la agrupación en clúster asimétrico.	Las prestaciones de WPF se pueden realizar de forma alternativa en WebSphere eXtreme Scale.
StreamQuery: una consulta continua sobre los datos en curso almacenados en correlaciones ObjectGrid.	Ninguna
Configuración de cuadrícula estática: una topología estática basada en clúster que utiliza el archivo XML de despliegue de clúster.	Se sustituye por la topología mejorada de despliegue dinámico para la gestión de grandes cuadrículas de datos.
Propiedades del sistema en desuso: las propiedades del sistema para especificar los archivos de propiedades de servidor y de clientes están en desuso.	Puede seguir utilizando estos argumentos, pero cambie las propiedades del sistema por los valores nuevos. **-Dcom.ibm.websphere.objectgrid.CatalogServerProperties** La propiedad estaba en desuso en WebSphere eXtreme Scale versión 7.0. Utilice la propiedad **-Dobjectgrid.server.props**. **-Dcom.ibm.websphere.objectgrid.ClientProperties** La propiedad estaba en desuso en WebSphere eXtreme Scale versión 7.0. Utilice la propiedad **-Dobjectgrid.client.props**. **-Dobjectgrid.security.server.prop** La propiedad estaba en desuso en WebSphere eXtreme Scale versión 6.1.0.3. Utilice la propiedad **-Dobjectgrid.server.prop**. **-serverSecurityFile** Este argumento estaba en desuso en WebSphere eXtreme Scale versión 6.1.0.3. Esta opción se pasa en el script `startOgServer`. Utilice el argumento **-serverProps**.

Capítulo 6. Configuración

Puede configurar WebSphere eXtreme Scale para ejecutarse en un entorno autónomo, o puede configurar eXtreme Scale para ejecutarse en un entorno con WebSphere Application Server o WebSphere Application Server Network Deployment. Para que un despliegue de WebSphere eXtreme Scale adopte los cambios de configuración en el lado del servidor de la cuadrícula de datos, debe reiniciar los procesos para que estos cambios entren en vigor, en lugar de aplicarlos de forma dinámica. Sin embargo, en el cliente, aunque no puede alterar los valores de configuración para una instancia de cliente existente, puede crear un nuevo cliente con los valores que necesite utilizando un archivo XML o mediante programas. Al crear un cliente, puede alterar temporalmente los valores predeterminados que proceden de la configuración de servidor actual.

Métodos de configuración

Puede configurar la mayoría de los aspectos del producto con archivos XML y de propiedades. También puede utilizar métodos de programación, incluidas las interfaces de programación del sistema y de aplicaciones, plug-ins y beans gestionados.

Acerca de esta tarea

Utilice los archivos siguientes para crear una configuración básica:

Archivo de propiedades de servidor
Utilice el archivo de propiedades de servidor para definir los valores para los servidores de catálogo y contenedor, por ejemplo rastreo, registro, seguridad, puertos, etc. Puede pasar un archivo de propiedades de servidor al script **startOgServer**, poner el archivo en la vía de acceso de clase o definir el archivo con las propiedades de sistema. Para

Archivo de propiedades de cliente
Utilice el archivo de propiedades de cliente para establecer las propiedades en los clientes, incluidos los puertos y los valores de seguridad. Puede especificar el archivo de propiedades de cliente a utilizar con una propiedad de sistema, colocando el archivo en la vía de acceso de clase o utilizando el método ClientClusterContext.getClientProperties.

Archivo XML de descriptor ObjectGrid
El archivo XML de descriptor de ObjectGrid describe la configuración de correlación y cuadrícula de datos. Especifique el archivo que se debe utilizar con el script **startOgServer** para configuraciones autónomas o añada el archivo al módulo de aplicación para configuraciones de WebSphere Application Server.

Archivo XML de descriptor de política de despliegue
El archivo XML de política de despliegue controla el fragmento y la colocación de los datos en los diversos servidores de contenedor de la configuración. Especifique el archivo que se debe utilizar con el script **startOgServer** para configuraciones autónomas o añada el archivo al módulo de aplicación para configuraciones de WebSphere Application Server.

Configuración de cuadrículas de datos

Utilice un archivo XML de descriptor de ObjectGrid para configurar cuadrículas de datos, correlaciones de respaldo, plug-ins, etc. Para configurar WebSphere eXtreme Scale, utilice el archivo XML de descriptor de ObjectGrid y la API ObjectGrid. Para una topología distribuida, necesita un archivo XML de descriptor de ObjectGrid y un archivo XML de política de despliegue.

Configuración de despliegues locales

Se puede crear una configuración de eXtreme Scale en memoria local mediante las API o un archivo XML de descriptor de ObjectGrid.

Acerca de esta tarea

Para crear un despliegue local, crea un archivo XML de descriptor de ObjectGrid y, a continuación, pasa el archivo a los métodos createObjectGrid de la interfaz ObjectGridManager.

Como alternativa, puede también crear todo el despliegue mediante programación mediante la interfaz ObjectGridManager.

Procedimiento

1. Cree un archivo XML de descriptor de ObjectGrid.

   El archivo `companyGrid.xml` siguiente es un ejemplo de un XML de descriptor de ObjectGrid. Las primeras líneas del archivo incluyen la cabecera necesaria para cada archivo XML de ObjectGrid. El archivo define una instancia de ObjectGrid denominada "CompanyGrid" y varias BackingMaps denominadas "Customer," "Item," "OrderLine" y "Order".

   Archivo 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. Pase el archivo XML a uno de los métodos createObjectGrid en la interfaz ObjectGridManager.

   El siguiente código de ejemplo valida el archivo `companyGrid.xml` respecto al esquema XML y crea la instancia de ObjectGrid denominada "CompanyGrid". La instancia de ObjectGrid recién creada no se almacena en memoria caché.

   ```
   ObjectGridManager objectGridManager = ObjectGridManagerFactory.getObjectGridManager();
   ObjectGrid companyGrid = objectGridManager.createObjectGrid("CompanyGrid",
    new URL("file:etc/test/companyGrid.xml"), true, false);
   ```

Qué hacer a continuación

Consulte Creación de instancia de ObjectGrid con la interfaz ObjectGridManager para obtener más información sobre cómo definir todas las correlaciones mediante

programación con los métodos createObjectGrid en la interfaz ObjectGridManager.

Habilitación de desalojadores con configuración XML

En lugar de utilizar la interfaz BackingMap para establecer mediante programación los atributos de BackingMap que pueden ser utilizados por el desalojador TTL, puede utilizar un archivo XML para configurar cada instancia de BackingMap. El siguiente código demuestra cómo establecer estos atributos para tres correlaciones de BackingMap distintas:

Antes de empezar

Antes de empezar, decida sobre el tipo de desalojador que va a utilizar:

- **El desalojador TTL basado en tiempo predeterminado:** el desalojador predeterminado utiliza una política de desalojo de tiempo de vida (TTL) para cada instancia de BackingMap.
- **Un mecanismo de desalojador conectable:** los desalojadores conectables normalmente utilizan una política de desalojo basada en el número de entradas en lugar de en el tiempo.

La mayoría de los valores de configuración de los desalojadores deben establecerse antes de inicializar ObjectGrid.

Procedimiento

- Para establecer el desalojador TTL predeterminado, añada el atributo ttlEvictorType al archivo XML de descriptor de ObjectGrid.

  El ejemplo siguiente muestra que la instancia de BackingMap de map1 utiliza un tipo de desalojador TTL NONE. La instancia de BackingMap de map2 utiliza un tipo de desalojador TTL TTL LAST_ACCESS_TIME o LAST_UPDATE_TIME. Especifique solo uno u otro de estos valores. La instancia de BackingMap de map2 tiene un valor de vida de 1800 segundos, o 30 minutos. La instancia de BackingMap map3 está definida para que utilice un tipo de desalojador TTL CREATION_TIME y tiene un valor de tiempo de vida de 1200 segundos o 20 minutos.

```
<?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>
```

Figura 27. Habilitar el desalojador TimeToLive con XML

- Para establecer un desalojador conectable, utilice el ejemplo siguiente.

```
<?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>
```

Figura 28. Conexión de un desalojador mediante XML

Configuración de una estrategia de bloqueo

Puede definir una estrategia de bloqueo optimista, pesimista o sin bloqueo en cada BackingMap en la configuración de WebSphere eXtreme Scale.

Acerca de esta tarea

Cada instancia de BackingMap se puede configurar para utilizar una de las siguientes estrategias de bloqueo:

1. Modalidad de bloqueo optimista
2. Modalidad de bloqueo pesimista
3. Ninguna

La estrategia de bloqueo predeterminada es OPTIMISTIC. Utilice el bloqueo optimista cuando los datos no se modifican frecuentemente. Los bloqueos sólo se mantienen durante un tiempo breve mientras los datos se leen de la memoria caché y se copian en la transacción. Cuando la memoria caché de la transacción se sincroniza con la memoria caché principal, los objetos de la memoria caché actualizados se comprueban contra la versión original. Si la comprobación falla, la transacción se retrotrae y se produce la excepción OptimisticCollisionException.

La estrategia de bloqueo PESSIMISTIC adquiere bloqueos para las entradas de memoria caché y debe utilizarse cuando los datos se cambian con frecuencia. Cada vez que se lee una entrada de la memoria caché, se adquiere un bloqueo, que puede mantenerse condicionalmente hasta que se complete la transacción. La duración de algunos de los bloqueos pueden ajustarse mediante el uso de niveles de aislamiento para la sesión.

Si el bloqueo no es necesario porque los datos nunca se actualizan o sólo se actualizan durante períodos tranquilos, puede inhabilitar el bloqueo mediante el uso de la estrategia de bloqueo NONE. Esta estrategia es muy rápida porque no se

necesita ningún gestor de bloqueos. La estrategia de bloqueo NONE es ideal en tablas de búsqueda o en correlaciones de sólo lectura.

Para obtener más información sobre las estrategias de bloqueo, consulte Estrategias de bloqueola información sobre las estrategias de bloqueo en la *Visión general del producto*.

Procedimiento

- **Configure una estrategia de bloqueo optimista**
  - Mediante programación utilizando el método setLockStrategy:

```
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 );
```

  - Utilizando el atributo lockStrategy en la Archivo XML de descriptor ObjectGrid:

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

- **Configure una estrategia de bloqueo pesimista**
  - Mediante programación utilizando el método setLockStrategy:

```
especifique la estrategia pesimista a través de programas
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);
```

  - Utilizando el atributo lockStrategy en la Archivo XML de descriptor ObjectGrid.

```
especifique la estrategia pesimista mediante XML
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="pessimisticMap"
                lockStrategy="PESSIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configure una estrategia sin bloqueo**

– Mediante programación utilizando el método setLockStrategy:

```
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);
```

– Utilizando el atributo lockStrategy en la Archivo XML de descriptor ObjectGrid:

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

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="noLockingMap"
                lockStrategy="NONE"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

Qué hacer a continuación

Para evitar que se genere una excepción java.lang.IllegalStateException, debe llamar al método setLockStrategy antes de llamar a los métodos initialize o getSession en la instancia de ObjectGrid.

Configuración de réplica de igual a igual con JMS

El mecanismo de réplica de igual a igual basada en JMS (Java Message Service) se utiliza en ambos entorno de WebSphere eXtreme Scale, el local y el distribuido. JMS es un proceso de réplica de núcleo a núcleo y permite a las actualizaciones de datos fluir entre los ObjectGrids locales y los ObjectGrids distribuidos. Por ejemplo, con este mecanismo podrá mover actualizaciones de datos de una cuadrícula de datos de eXtreme Scale distribuida a una cuadrícula de eXtreme Scale local, o desde una cuadrícula a otra en un dominio de sistema distinto.

Antes de empezar

El mecanismo de réplica de igual a igual basado en JMS se basa en el ObjectGridEventListener basado en JMS incorporado, com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener. Si desea información detallada relacionada con la habilitación del mecanismo de réplica de igual a igual, consulte “Receptor de sucesos JMS” en la página 232.

Si desea más información, consulte “Habilitación del mecanismo de invalidación de clientes” en la página 296.

Lo que aparece a continuación es un ejemplo de configuración XML para habilitar el mecanismo de réplica de igual a igual en una configuración de eXtreme Scale:

Configuración de la réplica de igual a igual - ejemplo de XML

```
<bean id="ObjectGridEventListener"
className="com.ibm.websphere.objectgrid.plugins.JMSObjectGridEventListener">
<property name="replicationRole" type="java.lang.String" value="DUAL_ROLES" description="" />
  <property name="replicationStrategy" type="java.lang.String" value="PUSH" description="" />
      <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
  value="defaultTCF" description="" />
```

```
 <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
 <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
 <property name="jms_userid" type="java.lang.String" value="" description="" />
 <property name="jms_password" type="java.lang.String" value="" description="" />
 <property name="jndi_properties" type="java.lang.String"
value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;
java.naming.provider.url=tcp://localhost:61616;connectionFactoryNames=defaultTCF;
topic.defaultTopic=defaultTopic"
description="jndi properties" />
    </bean>
```

Distribución de cambios entre JVM de igual

Los objetos LogSequence y LogElement distribuyen cambios entre JVM de igual y comunican los cambios que se han producido en una transacción de eXtreme Scale con un plug-in ObjectGridEventListener.

Si desea más información sobre cómo se puede utilizar JMS (Java Message Service) para distribuir los cambios transaccionales, consulte Distribución de transacciones.

Como requisito previo, ObjectGridManager debe almacenar en memoria caché la instancia de ObjectGrid. Si desea más información, consulte los métodos createObjectGrid. El valor booleano de cacheInstance debe establecerse en true.

No es necesario implementar este mecanismo. Existe un mecanismo de réplica de igual a igual incorporado para que utilice esta función. Consulte “Configuración de réplica de igual a igual con JMS” en la página 228.

Los objetos proporcionan una forma sencilla para que una aplicación publique los cambios producidos en un ObjectGrid mediante el transporte de un mensaje a ObjectGrids de igual en Mäquinas virtuales Java remotas y después los aplique en dicha JVM . La clase LogSequenceTransformer es fundamental para habilitar este soporte. Este artículo examina cómo escribir un escucha mediante el sistema de mensajería JMS (Java Message Service) para propagar los mensajes. Con ese fin, eXtreme Scale soporta la transmisión de LogSequences que genera una operación de confirmación de la transacción de eXtreme Scale entre varios miembros del clúster de WebSphere Application Server con un plug-in proporcionado por IBM. Esta función no está habilitada de manera predeterminada, pero puede configurarse. Sin embargo, si el consumidor o el productor no es un WebSphere Application Server, podría ser necesario utilizar un sistema de mensajería JMS externo.

Implementación del mecanismo

La clase LogSequenceTransformer y las API ObjectGridEventListener, LogSequence y LogElement permiten el uso de operaciones fiables de publicar y suscribir para distribuir los cambios y filtrar las correlaciones que desea distribuir. Los fragmentos de código de este tema muestran cómo utilizar estas API con JMS para crear un ObjectGrid de igual a igual compartido por aplicaciones que se alojan en un conjunto diverso de plataformas que comparten un transporte de mensajes común.

Inicializar el plug-in

ObjectGrid llama al método initialize del plug-in, parte del contrato de la interfaz ObjectGridEventListener, cuando ObjectGrid se inicia. El método initialize debe obtener sus recursos JMS, incluidos las conexiones, sesiones y editores, e iniciar la hebra que es la escucha JMS.

En los ejemplos siguientes se muestra el método initialize:

Ejemplo del método 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();

            // debe iniciarse la conexión para recibir mensajes.
            connection.start();

            // la sesión jms no es transaccional (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);

// iniciar la hebra de la escucha.
            listenerRunning = true;
            listenerThread = new Thread(this);
            listenerThread.start();
        } catch (Throwable e) {
            throw new ObjectGridRuntimeException("Cannot initialize", e);
        }
    }
```

El código para iniciar la hebra utiliza una hebra Java 2 Platform, Standard Edition (Java SE). Si ejecuta un servidor WebSphere Application Server versión 6.x o un servidor WebSphere Application Server versión 5.x Enterprise, utilice la interfaz de programación de aplicaciones (API) del bean asíncrono para iniciar esta hebra de daemon. También puede utilizar las API comunes. A continuación se muestra un ejemplo de fragmento de código de sustitución que muestra la misma acción mediante el uso de un gestor de trabajo:

```
// iniciar la hebra de la escucha.
listenerRunning = true;
workManager.startWork(this, true);
```

El plug-in también debe implementar la interfaz Work en lugar de la interfaz Runnable. Debe además añadir un método release para establecer la variable listenerRunning en false. El plug-in debe proporcionarse con una instancia WorkManager en su constructor y mediante inyección si se utiliza un contenedor IoC (Inversión de control).

Transmitir los cambios

A continuación se muestra un método transactionEnd de ejemplo para publicar los cambios locales realizados en un ObjectGrid. En este ejemplo se utiliza un JMS, aunque puede utilizarse cualquier transporte de mensajes capaz de producir una mensajería de publicar y suscribir fiable.

Ejemplo del método transactionEnd

```
// Este método se sincroniza para garantizar que
    // los mensajes se publican en el orden en que se
```

```
// confirmó la transacción. Si se empieza publicando los mensajes
// en paralelo, los receptores podrían dañar la correlación
// ya que las operaciones de supresión pueden llegar antes que las de inserción, etc.
public synchronized void transactionEnd(String txid, boolean isWriteThroughEnabled, boolean committed,
        Collection changes) {
    try {
        // debe utilizarse la modalidad write through y confirmarse.
        if (isWriteThroughEnabled && committed) {
            // escribir las secuencias en un byte []
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            ObjectOutputStream oos = new ObjectOutputStream(bos);
            if (publishMaps.isEmpty()) {
                // serializar toda la colección
                LogSequenceTransformer.serialize(changes, oos, this, mode);
            } else {
                // filtrar LogSequences basado en el contenido de publishMaps
                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);
            }
            // realizar un mensaje de objeto para los cambios
            oos.flush();
            ObjectMessage om = jmsSession.createObjectMessage(bos.toByteArray());
            // establecer propiedades
            om.setStringProperty(PROP_TX, txid);
            om.setStringProperty(PROP_GRIDNAME, myGrid.getName());
            // transmitirlo.
            publisher.publish(om);
        }
    } catch (Throwable e) {
        throw new ObjectGridRuntimeException("Cannot push changes", e);
    }
}
```

Este método utiliza diversas variables de instancia:

- Variable `jmsSession`: sesión JMS que se utiliza para publicar mensajes. Se crea al inicializarse el plug-in.
- Variable `mode`: modo de distribución.
- Variable `publishMaps`: conjunto que contiene el nombre de cada correlación con los cambios que se van a publicar. Si la variable está vacía, se publicarán todas las correlaciones.
- Variable `publisher`: objeto TopicPublisher que se crea durante el método initialize del plug-in.

Recibir y aplicar mensajes de actualización

A continuación se muestra el método run. Este método se ejecuta en un bucle hasta que la aplicación detiene el bucle. Cada repetición del bucle intenta recibir un mensaje JMS y aplicarlo a ObjectGrid.

Ejemplo del método de ejecución del mensaje JMS

```
private synchronized boolean isListenerRunning() {
        return listenerRunning;
    }

    public void run() {
        try {
            System.out.println("Listener starting");
            // obtener una sesión jms para recibir los mensajes.
            // No transaccional.
            TopicSession myTopicSession;
            myTopicSession = connection.createTopicSession(false, javax.jms.Session.AUTO_ACKNOWLED

            // obtener un suscriptor para el tema, true indica no recibir
```

```
            // mensajes transmitidos mediante editores
            // en esta conexión. De lo contrario, recibiríamos nuestras propias actualizaciones.
            TopicSubscriber subscriber = myTopicSession.createSubscriber(topic, null, true);
            System.out.println("Listener started");
            while (isListenerRunning()) {
                ObjectMessage om = (ObjectMessage) subscriber.receive(2000);
                if (om != null) {
                    // Usar objeto Session pasado en el método initialize...
                    // es muy importante utilizarlo, sin write through
                    mySession.beginNoWriteThrough();
                    byte[] raw = (byte[]) om.getObject();
                    ByteArrayInputStream bis = new ByteArrayInputStream(raw);
                    ObjectInputStream ois = new ObjectInputStream(bis);
                    // inflar LogSequences
                    Collection collection = LogSequenceTransformer.inflate(ois, myGrid);
                    Iterator iter = collection.iterator();
                    while (iter.hasNext()) {
                        // procesar los cambios de las correlaciones de acuerdo con la modalidad
                        // una vez serializado LogSequence
                        LogSequence seq = (LogSequence) iter.next();
                        mySession.processLogSequence(seq);
                    }
                    mySession.commit();
                } // if there was a message
            } // while loop
            // detener la conexión
            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");
    }
```

Receptor de sucesos JMS

JMSObjectGridEventListener se ha diseñado para soportar la invalidación de la memoria caché cercada del cliente y un mecanismo de réplica de igual a igual. Se trata de una implementación JMS (Java Message Service) de la interfaz ObjectGridEventListener.

El mecanismo de invalidación del cliente se puede utilizar en un entorno distribuido de eXtreme Scale para asegurarse de que los datos de la memoria caché cercana del cliente estén sincronizados con los servidores y otros clientes. Sin esta función, la memoria caché cercana del cliente podría albergar datos obsoletos. Sin embargo, incluso con este mecanismo de invalidación de cliente basado en JMS, debe tener en cuenta la ventana de temporización para actualizar una memoria caché cercana debido al retardo para el tiempo de ejecución en la publicación de actualizaciones.

El mecanismo de réplica de igual a igual se puede utilizar en entornos distribuidos y también locales de eXtreme Scale. Se trata de un proceso de réplica de núcleo a núcleo y permite a las actualizaciones de datos fluir entre los ObjectGrids locales y los ObjectGrids distribuidos. Por ejemplo, con este mecanismo puede mover las actualizaciones de datos de una cuadrícula distribuida a un ObjectGrid local, o de una cuadrícula a otra en un distinto dominio de sistema.

JMSObjectGridEventListener requiere que el usuario configure la información de JMS y JNDI (Java Naming and Directory Interface) para poder obtener los recursos JMS necesarios. De forma adicional, las propiedades relacionadas con la réplica se deben definir de forma correcta. En un entorno JEE, JNDI debe estar disponibles en los contenedores web y también en los contenedores EJB (Enterprise JavaBean). En este caso, la propiedad JNDI es opcional, a menos que desee obtener recursos JMS externos.

Este receptor de sucesos tiene propiedades que puede configurar mediante XML o con enfoques programáticos, que se pueden utilizar sólo para la invalidación de cliente, sólo para la réplica de igual a igual, o ambos. La mayoría de propiedades son opcionales para personalizar el comportamiento para conseguir la funcionalidad necesaria.

.

Si desea más información, consulta la API JMSObjectGridEventListener.

Ampliación del plug-in JMSObjectGridEventListener

El plug-in JMSObjectGridEventListener permite a las instancias de ObjectGrid iguales recibir actualizaciones cuando los datos de la cuadricula se han modificado o desalojado. También permite notificar a los clientes cuando se actualizan o desalojan entradas de una cuadrícula de eXtreme Scale. Este tema describe cómo ampliar el plug-in JMSObjectGridEventListener para permitir notificar a las aplicaciones cuando se recibe un mensaje JMS. Esto es más práctico cuando se utiliza el valor CLIENT_SERVER_MODEL para la invalidación de clientes.

Cuando se ejecuta en el rol de receptor, la ejecución de eXtreme Scale llama automáticamente al método JMSObjectGridEventListener.onMessage alterado temporalmente cuando la instancia de JMSObjectGridEventListener recibe actualizaciones de mensaje JMS de la cuadrícula. Estos mensajes recortan una colección de objetos LogSequence. Los objetos LogSequence se pasan en el método onMessage y la aplicación utiliza el LogSequence para identificar qué entradas de memoria caché se han insertado, suprimido, actualizado o invalidado.

Para utilizar el punto de ampliación onMessage, las aplicaciones realizan los siguientes pasos.

1. Cree una nueva clase, ampliando la clase JMSObjectGridEventListener, alterando temporalmente el método onMessage.
2. Configure el JMSObjectGridEventListener ampliado del mismo modo que el ObjectGridEventListener para ObjectGrid.

El JMSObjectGridEventListener ampliado es una clase hija de JMSObjectGridEventListener y sólo puede alterar temporalmente dos métodos: los métodos initialize (opcional) y onMessage. Si una clase hija de la clase JMSObjectGridEventListener debe utilizar cualquier artefacto de ObjectGrid como, por ejemplo, ObjectGrid o Session en el método onMessage, puede obtener estos artefactos en el método initialize y almacenarlos en la memoria caché como variables de instancia. Además, en el método onMessage, los artefactos de ObjectGrid almacenados en la memoria caché se pueden utilizar para procesar una colección pasada de LogSequences.

Nota: El método initialize alterado temporalmente debe invocar el método super.initialize para poder inicializar un JMSObjectGridEventListener padre de forma apropiada.

A continuación, aparece un ejemplo de una clase JMSObjectGridEventListener ampliada.

```
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;

    /**
     * Esta es la cuadrícula asociada al receptor.
     */
     ObjectGrid grid;

    /**
     * Esta es la sesión asociada a este receptor.
     */
     Session session;

    String objectGridType;

    public List receivedLogSequenceList = new ArrayList();


 /* (no Javadoc)
  * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
    #initialize(com.ibm.websphere.objectgrid.Session)
  */
public void initialize(Session session) {
  // Nota: si debe utilizar algún artefacto de ObjectGrid, este clase necesita obtener el ObjectGrid
  //  de la instancia pasada de Session y obtener el ObjectMap de la instancia de la sesión
  // para cualquier operación de correlación de ObjectGrid transaccional.

  super.initialize(session);  // debe invocar el método initialize de super.
  this.session = session; // almacene en la memoria caché la instancia de la sesión, en caso
  //   que necesite utilizarla para realizar la operación de correlación.
  this.grid = session.getObjectGrid(); // obtenga el ObjectGrid, en caso de
  //   necesitar obtener la información de ObjectGrid.

  if (grid.getObjectGridType() == ObjectGrid.CLIENT)
   objectGridType = "CLIENT";
  else if (grid.getObjectGridType() == ObjectGrid.SERVER)
   objectGridType = "Server";

  if (debug)
   System.out.println("ExtendedJMSObjectGridEventListener[" +
     objectGridType + "].initialize() : grid = " + this.grid);
 }

 /* (no 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 + "]";
 }
}
```

Configuración

La clase JMSObjectGridEventListener ampliada se debe configurar del mismo modo para la invalidación de cliente y, también, para el mecanismo de réplica de igual a igual. Lo que aparece a continuación es el ejemplo de configuración de XML.

```
<objectGrid name="PRICEGRID">
<bean id="ObjectGridEventListener"
    className="com.ibm.websphere.samples.objectgrid.jms.
      price.ExtendedJMSObjectGridEventListener">
<property name="invalidationModel" type="java.lang.String" value=
"CLIENT_SERVER_MODEL" description="" />
    <property name="invalidationStrategy" type="java.lang.String"
     value="INVALIDATE" description="" />
    <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
     value="jms/TCF" description="" />
    <property name="jms_topicJndiName" type="java.lang.String"
     value="GRID.PRICEGRID" description="" />
    <property name="jms_topicName" type="java.lang.String"
     value="GRID.PRICEGRID" description="" />
<property name="jms_userid" type="java.lang.String" value="" description="" />
<property name="jms_password" type="java.lang.String" value="" description="" />
   </bean>
   <backingMap name="PRICE" pluginCollectionRef="PRICE"></backingMap>
       </objectGrid>
```

Nota: El className del bean ObjectGridEventListener se ha configurado con la clase JMSObjectGridEventListener ampliada con las mismas propiedades que el JMSObjectGridEventListener genérico.

Configuración de las políticas de despliegue

Utilice el archivo XML de descriptor de la política de despliegue y el archivo XML de descriptor de objectgrid para gestionar una topología distribuida. La política de despliegue está codificada como un archivo XML que se proporciona al servidor de contenedor. La política de despliegue proporciona información sobre correlaciones, conjuntos de correlaciones, particiones, réplicas, etc. Controla también los comportamientos de colocación de fragmentos.

Configuración de despliegues distribuidos

Utilice el archivo XML de descriptor de política de despliegue y el archivo XML de descriptor ObjectGrid para gestionar la topología.

La política de despliegue se codifica como archivo XML que se proporciona al servidor de contenedor eXtreme Scale. El archivo XML especifica la siguiente información:

- Las correlaciones que pertenecen a cada conjunto de correlaciones
- El número de particiones
- El número de réplicas síncronas y asíncronas

La política de despliegue también controla los siguientes comportamientos de colocación.

- El número mínimo de servidores de contenedor activos antes de que se produzca la colocación
- Sustitución automática de fragmentos perdidos
- Colocación de cada fragmento desde una sola partición en otra máquina

La información de punto final no está preconfigurada en el entorno dinámico. En la política de despliegue no hay ningún nombre de servidor ni información de topología física. El servicio de catálogo coloca automáticamente todos los fragmentos de una cuadrícula de datos en servidores de contenedor. El servicio de catálogo utiliza las restricciones definidas por la política de despliegue para gestionar automáticamente la colocación de fragmentos. Esta colocación automática de fragmentos permite una fácil configuración para cuadrículas de datos grandes. También puede añadir servidores al entorno según sea necesario.

Restricción: En un entorno WebSphere Application Server, no está soportado un tamaño de grupo principal de más de 50 miembros.

Se pasa un archivo XML de política de despliegue al servidor de contenedor durante el inicio. Se debe utilizar una política de despliegue junto con un archivo XML de ObjectGrid. La política de despliegue no es necesaria para iniciar un servidor de contenedor, aunque se recomienda. La política de despliegue debe ser compatible con el archivo XML de ObjectGrid que se utiliza con ella. Para cada elemento objectgridDeployment de la política de despliegue, debe incluir un elemento objectGrid correspondiente en el archivo XML de ObjectGrid. Las correlaciones en objectgridDeployment deben ser coherentes con los elementos backingMap encontrados en el XML de ObjectGrid. Se debe hacer referencia a cada backingMap dentro de un solo un elemento mapSet.

En el siguiente ejemplo, se intenta emparejar el archivo `companyGridDpReplication.xml` con el correspondiente archivo `companyGrid.xml`.

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>
```

El archivo `companyGridDpReplication.xml` tiene un elemento mapSet dividido en 11 particiones. Cada partición debe tener, exactamente, una réplica síncrona. El número de réplicas síncronas viene especificado por los atributos minSyncReplicas y maxSyncReplicas. Puesto que el atributo minSyncReplicas está establecido en 1, cada partición del elemento mapSet debe tener, como mínimo, una réplica síncrona disponible para procesar transacciones de escritura. Dado que el atributo maxSyncReplicas se establece en 1, cada partición no puede sobrepasar de una réplica síncrona. Las particiones de este elemento mapSet no tiene réplicas asíncronas.

El atributo numInitialContainers indica al servicio de catálogo que difiera la colocación hasta que estén disponibles cuatro servidores de contenedor para soportar esta instancia de ObjectGrid. El atributo numInitialContainers se ignora después que se haya alcanzado el número especificado de servidores de contenedor.

7.1.1+ También puede utilizar la propiedad **placementDeferralInterval** y el mandato **xscmd -c suspendBalancing** para retardar la colocación de fragmentos en los servidores de contenedor.

Aunque el archivo `companyGridDpReplication.xml` es un ejemplo básico, una política de despliegue le puede ofrecer control total sobre el entorno.

Topología distribuida

Las memorias caché coherentes distribuidas ofrecen un mayor rendimiento, disponibilidad y escalabilidad que puede configurar el usuario.

WebSphere eXtreme Scale equilibra automáticamente los servidores. Puede incluir servidores adicionales sin reiniciar WebSphere eXtreme Scale. La adición de servidores adicionales sin tener que reiniciar eXtreme Scale le permite tener despliegues sencillos y, también, despliegues grandes de terabytes en los que son necesarios cientos de servidores.

Esta topología de despliegue es flexible. Mediante el servicio de catálogo, puede añadir y eliminar servidores para utilizar mejor los recursos sin eliminar toda la memoria caché. Puede utilizar los mandatos **startOgServer** y **stopOgServer** para iniciar y detener los servidores de contenedor. Estos dos parámetros requieren que especifique la opción **-catalogServiceEndPoints**. Todos los clientes de la topología distribuida se comunican con el servicio de catálogo a través del protocolo IIOP (Internet Interoperability Object Protocol). Todos los clientes utilizan la interfaz de ObjectGrid para comunicarse con los servidores.

La prestación de la configuración dinámica de WebSphere eXtreme Scale facilita la adición de recursos al sistema. Los contenedores alojan los datos y el servicio de catálogo permite a los clientes comunicarse con la cuadrícula de servidores de contenedor. El servicio de catálogo reenvía las solicitudes, asigna espacio en los servidores de contenedor de host y gestiona el estado y la disponibilidad del sistema en general. Los clientes se conectan a un servicio de catálogo, recuperan una descripción de la topología de servidor de contenedor y, a continuación, se comunican directamente con cada servidor según sea necesario. Cuando la topología del servidor cambia debido a la adición de nuevos servidores, o debido a la anomalía de otros, el servicio de catálogo direcciona automáticamente las solicitudes del cliente al servidor apropiado que aloja los datos.

Normalmente, un servicio de catálogo existe en su propia cuadrícula de Máquinas virtuales Java. Un solo servidor de catálogo puede gestionar varios servidores. Puede iniciar un servidor de contenedor en una JVM solo o cargar el servidor de contenedor en una JVM arbitraria con otros servidores de contenedor para distintos servidores. Un cliente puede existir en cualquier JVM y comunicarse con uno o más servidores. Un cliente también puede existir en la misma JVM que un servidor de contenedor.

También puede crear una política de despliegue mediante el programa al incorporar un servidor de contenedor en una aplicación o un proceso Java existente. Para obtener más información, consulte la documentación de la API DeploymentPolicy.

Control de la colocación de fragmentos con zonas

Utilizar la política de despliegue para definir zonas. Las zonas le ofrecen control sobre la colocación de fragmentos en WebSphere eXtreme Scale. Las zonas son un concepto lógico definido por el usuario utilizado para representar agrupaciones lógicas de servidores físicos.

Configuración de zonas para la colocación de réplicas

El soporte de zonas permite realizar configuraciones sofisticadas para la colocación de réplicas en centros de datos. Con esta prestación, las cuadrículas de miles de particiones se pueden gestionar fácilmente utilizando un puñado de reglas de colocación opcionales. Un centro de datos puede estar en varias plantas de un edificio, distintos edificios, o incluso en distintas ciudades u otras distinciones, tal como se haya configurado con las reglas de zonas.

Flexibilidad de zonas

Puede colocar fragmentos en zonas. Esta función le permite tener más control sobre cómo eXtreme Scale coloca los fragmentos en una cuadrícula. Máquinas virtuales Java que aloja un servidor eXtreme Scale se puede marcar con un identificador de zona. El archivo de despliegue ahora puede incluir una o más reglas de zona y estas reglas de zona están asociadas con un tipo de fragmento. La siguiente sección proporciona una visión general del uso de la zona. Para obtener

más detalles, consulte la información sobre cómo controlar la colocación de fragmentos con zonas en la *Guía de administración*.

El control de zonas de colocación de cómo eXtreme Scale asigna primarios y réplicas para configurar topologías avanzadas.

Una Máquina virtual Java puede tener varios contenedores pero sólo 1 servidor. Un contenedor puede alojar varios fragmentos de un solo ObjectGrid.

Esta prestación es útil para asegurarse de que los fragmentos primarios y los fragmentos réplicas se colocan en distintas ubicaciones o zonas para obtener una mejor alta disponibilidad. Normalmente, eXtreme Scale no coloca un fragmento primario y de réplica en la Mäquinas virtuales Java con la misma dirección IP. Esta regla simple normalmente impide que dos servidores eXtreme Scale se coloquen en el mismo sistema físico. No obstante, quizás necesite un mecanismo más flexible. Por ejemplo, es posible que esté utilizando dos chasis blade y desee que los primarios se *extiendan* por los dos chasis y la réplica de cada primario pueda colocarse en el otro chasis desde el primario.

Primarios *extendidos* significa que los primarios se colocan en cada zona y la réplica de cada primario se coloca en la zona opuesta. Por ejemplo, el primario 0 estaría en zoneA, y la réplica sinc 0 estaría en zoneB. El primario 1 estaría en zoneB, y la réplica sinc 1 estaría zoneA.

En este caso el nombre del chasis sería el nombre de zona. De forma alternativa, puede especificar nombres para zonas según sus plantas de un edificio y utilizar las zonas para asegurarse de que los primarios y las réplicas para los mismos datos estén en distintas plantas. También es posible edificios y centros de datos. Se han realizado comprobaciones en centros de datos utilizando zonas como mecanismo para asegurarse de que los datos se dupliquen de forma correcta entre los centros de datos. Si utiliza el gestor de sesiones HTTP para eXtreme Scale, también puede utilizar zonas. Con esta característica puede desplegar una sola aplicación web por tres centros de datos y asegurarse de que las sesiones HTTP para los usuarios se dupliquen a lo largo de los centros de datos para que las sesiones se puedan recuperar incluso si falla todo un centro de datos.

WebSphere eXtreme Scale reconoce la necesidad de gestionar una cuadrícula de gran tamaño por varios centros de datos. Si es necesario, puede asegurarse de que los fragmentos primarios y de copia de seguridad para la misma partición estén ubicados en distintos centros de datos. Puede colocar todos los primarios en el centro de datos 1 y todas las réplicas en el centro de datos 2, o puede aplicar un sistema de redondeo en los primarios y las réplicas entre los dos centros de datos. Las reglas son flexibles por lo que hay muchos escenarios posibles. eXtreme Scale también puede gestionar miles de servidores, que junto con la colocación totalmente automática con reconocimiento de centro de datos, hace que esas cuadrículas de gran tamaño sean asequibles desde un punto de vista administrativo. Los administradores pueden especificar lo que desean de forma simple y eficiente.

Como administrador, utiliza zonas de colocación para controlar donde se colocan los fragmentos primarios y de réplica, lo que permite configurar topologías avanzadas de alta disponibilidad y alto rendimiento. Puede definir una zona para cualquier agrupación lógica de procesos de eXtreme Scale, tal como se ha indicado anteriormente: estas zonas pueden corresponder a ubicaciones de estaciones de trabajo físicas, como un centro de datos, una planta de un centro de datos o un chasis de blade. Puede extender datos a través de zonas, que proporciona una

mayor disponibilidad, o puede partir los primarios y las réplicas en distintas zonas cuando es necesario una parada activa.

Asociación de un servidor eXtreme Scale a una zona que no utiliza WebSphere Extended Deployment

Si se utiliza eXtreme Scale con Java Standard Edition o un servidor de aplicaciones que no se basa en WebSphere Extended Deployment versión 6.1, se puede asociar una JVM que es un contenedor de fragmento con una zona si utiliza las siguientes técnicas.

Aplicaciones que utilizan el script startOgServer

El script startOgServer se utiliza para iniciar una aplicación de eXtreme Scale cuando no se ha incrustado en un servidor existente. El parámetro **-zone** se utiliza para especificar la zona para utilizar todos los contenedores dentro del servidor.

Especificación de la zona al iniciar un contenedor utilizando las API

Asociación de nodos de WebSphere Extended Deployment con zonas

Si utiliza eXtreme Scale con aplicaciones de WebSphere Extended Deployment Java EE, puede aprovechar los grupos de nodos de WebSphere Extended Deployment para colocar servidores en zonas específicas.

En eXtreme Scale, una JVM puede ser miembro de una sola zona. No obstante, WebSphere permite que un nodo forme parte de varios grupos de nodos. Puede utilizar esta funcionalidad de zonas de eXtreme Scale si se asegura de que cada uno de sus nodos sólo esté en un grupo de nodos de zona.

Utilice la sintaxis siguiente para nombrar su grupo de nodos para declararlo una zona: `ReplicationZone<SufijoExclusivo>`. Los servidores que se ejecutan en un nodo que forma parte de dicho grupo de nodos se incluyen en la zona especificada por el nombre del grupo de nodos. A continuación se ofrece una descripción de una topología de ejemplo.

En primer lugar, debe configurar 4 nodos: node1, node2, node3 y node4. Cada uno de estos nodos tiene 2 servidores. Luego debe crear un grupo de nodos denominado ReplicationZoneA y un grupo de nodos denominado ReplicationZoneB. A continuación, debe añadir node1 y node2 a ReplicationZoneA y añadir node3 y node4 a ReplicationZoneB.

Cuando se inicien los servidores en node1 y node2, éstos pasarán a formar parte de ReplicationZoneA y, del mismo modo, los servidores en node3 y node4 pasarán a formar parte de ReplicationZoneB.

Una JVM miembro de la cuadrícula comprueba la pertenencia a la zona sólo durante el inicio. Si se añade un nuevo grupo de nodos o se cambia la pertenencia sólo afecta a las JVM recién iniciadas o reiniciadas.

Reglas de zonas

Una partición deeXtreme Scale tiene un fragmento primario y cero o más fragmentos de réplica. Para este ejemplo, considere el siguiente convenio de denominación para estos fragmentos. `P` es el fragmento primario, `S` es una réplica síncrona y `A` es una réplica asíncrona. Una regla de zonas tiene tres componentes:

- Un nombre de regla
- Una lista de zonas
- Un distintivo inclusivo o exclusivo

El nombre de la zona de un contenedor se puede especificar tal como se describe en la documentación de "API de servidor incorporado" en la página 413. Una regla de zonas especifica el conjunto de zonas posible en el que se puede colocar el fragmento. El distintivo inclusivo indica que tras colocar un fragmento en una zona de la lista, los demás fragmentos también se colocan en esa zona. Un valor exclusivo indica que cada fragmento correspondiente a una partición se coloca en una zona distinta en la lista de zonas. Por ejemplo, el uso de un valor exclusivo significa que si hay tres fragmentos (primario y dos réplicas síncronas), la lista de zonas debe tener tres zonas.

Cada fragmento puede asociarse a una regla de zonas. Una regla de zonas puede compartirse entre dos fragmentos. Cuando una regla se comparte, el distintivo inclusivo o exclusivo se extiende a través de fragmentos de todos los tipos que comparten una sola regla.

Ejemplos

A continuación se proporciona un conjunto de ejemplos que muestran distintos casos de ejemplo y la configuración de despliegue para implementar los casos de ejemplo.

Escritura en bandas de primarios y réplicas a través de zonas

Dispone de tres chasis blade y desea que los primarios se distribuyan a lo largo de los tres, con una sola réplica síncrona colocada en un chasis distinto al primario. Defina cada chasis como una zona con los nombres de chasis `ALPHA`, `BETA` y `GAMMA`. A continuación de proporciona un XML de despliegue de ejemplo:

```
<?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>
```

Un XML de despliegue contiene una cuadrícula denominada library con una cola correlación (map) denominada book. Utiliza cuatro particiones con una sola réplica síncrona. La cláusula de metadatos de zona muestra la definición de una sola regla de zonas y la asociación de reglas de zonas a fragmentos. Los fragmentos primario y síncrono están asociados a la regla de zonas "stripeZone". La regla de zonas tiene todas las tres zonas en ellas y utiliza la colocación exclusiva. Esta regla indica que si el primario de la partición 0 se coloca en `ALPHA`, la réplica de la partición 0 se colocará en `BETA` o `GAMMA`. De forma parecida, los primarios para otras particiones se colocan en otras zonas y las réplicas se colocarán.

Réplica asíncrona en una zona distinta a la réplica primaria y síncrona

En este ejemplo, existen dos edificios con una alta conexión de latencia entre ellos. Desea que no se pierdan datos de alta disponibilidad para todos los casos de ejemplo. No obstante, el impacto en el rendimiento de la réplica síncrona entre edificios le lleva a un compromiso. Desea un primario con una réplica síncrona en un edificio y una réplica asíncrona en el otro edificio. Normalmente, las anomalías son cuelgues de la JVM o anomalías en el sistema en lugar de problemas a gran escala. Con esta topología, puede superar anomalías normales sin pérdida de datos. La pérdida de un edificio es tan rato que alguna pérdida de datos es aceptable en ese caso. Puede crear dos zonas, una para cada edificio. A continuación se muestra el archivo XML de despliegue:

```
<?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>
```

La réplica primaria y síncrona comparten la regla de zonas primaySync con un valor de distintivo exclusivo de false. Por lo tanto, después de colocar el primario o síncrono en una zona, el otro también se coloca en la misma zona. La réplica asíncrona utiliza una segunda regla de zonas con las mismas zonas que la regla de zonas primarySync pero utiliza el atributo **exclusivePlacement** establecido en true. Este atributo indica que un fragmento no se puede colocar en una zona con otro fragmento de la misma partición. Como resultado, la réplica asíncrona no se coloca en la misma zona que el primario o las réplicas síncronas.

Colocar todos los primarios en una zona y todas las réplicas en otra zona

Aquí, todos los primarios están en una zona específica y todas las réplicas en una zona distinta. Tendremos un primario y una sola réplica asíncrona. Todas las réplicas estarán en la zona A y los primarios en 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>
```

Aquí, puede ver dos reglas, una para los primarios (P) y otra para la réplica (A).

Zonas en redes de área amplia (WAN)

Es posible que desee desplegar un solo eXtreme Scale en varios edificios o centros de datos con interconexiones de red más lentas. Las conexiones de red más lentas llevan a un ancho de banda más bajo y a conexiones de latencia más alta. La posibilidad de particiones de red también aumenta en esta modalidad debido a la congestión de la red y otros factores. eXtreme Scale aborda este entorno duro de las siguientes formas:

Pulsaciones limitadas entre zonas

Las Máquinas virtuales Java que están agrupadas en grupos principales se envían pulsaciones entre sí. Cuando el servicio de catálogo organiza las Mäquinas virtuales Java en grupos, estos grupos no abarcan zonas. Un líder dentro de este grupo pasa información de pertenencia al servicio de catálogo. El servicio de catálogo verifica todas las anomalías notificadas antes de realizar alguna acción. Lo lleva a cabo intentando conectarse a las Máquinas virtuales Java sospechosas. Si el servicio de catálogo ve una detección de anomalía falsa no realizará ninguna acción ya que la partición del grupo principal se arreglará en un corto periodo de tiempo.

El servicio de catálogo también enviará pulsaciones a líderes de grupo principal de forma periódica a velocidad baja para manejar el caso de aislamiento de grupo principal.

Direccionamiento a zonas según preferencias

Con el direccionamiento a zonas según preferencias, puede definir cómo WebSphere eXtreme Scale direcciona las transacciones a las zonas.

Tiene control sobre dónde se colocan los fragmentos de una cuadrícula de datos. Consulte “Configuración de zonas para la colocación de réplicas” en la página 238 para obtener más información sobre algunos escenarios básicos y cómo configurar la política de despliegue de la forma correspondiente.

El direccionamiento a zonas según preferencias permite a los clientes de WebSphere eXtreme Scale especificar una preferencia para una zona determinada o un conjunto de zonas. Como resultado, las transacciones de cliente se direccionan a las zonas de preferencia antes de intentar direccionarse a cualquier otra zona.

Requisitos para el direccionamiento a zonas según preferencias

Antes de intentar el direccionamiento a zonas según preferencias, asegúrese de que la aplicación pueda satisfacer los requisitos del escenario.

La colocación de partición por contenedor es necesaria para utilizar el direccionamiento a zonas según preferencias. Esta estrategia de colocación es muy adecuada para las aplicaciones que almacenan datos de sesión en ObjectGrid. La estrategia de colocación de particiones predeterminada de WebSphere eXtreme Scale es `fixed-partition`. Las claves utilizan el código hash en el momento de confirmar una transacción para determinar qué partición alberga el par de clave-valor de la correlación cuando se utiliza la ubicación de partición fija.

La colocación por contenedor asigna los datos a una partición aleatoria cuando la transacción confirma tiempo mediante el objeto SessionHandle. Debe poder reconstruir el objeto SessionHandle para recuperar los datos de la cuadrícula de datos.

Puede utilizar zonas para tener más control sobre dónde se colocan los fragmentos primarios y de réplica en el dominio. La utilización de varias zonas en el despliegue tiene ventajas cuando los datos se encuentran en varias ubicaciones físicas de datos. La separación geográfica de los fragmentos primarios y de réplica es una forma de asegurarse de que la pérdida catastrófica de un centro de datos no afectará a la disponibilidad de los datos.

Cuando los datos se distribuyen entre varias zonas, es probable que los clientes también se distribuyan a lo largo de la topología. El direccionamiento de clientes a su centro de datos o zona local tiene la obvia ventaja de rendimiento de una menor latencia de red. Direccione los clientes a centros de datos o zonas locales cuando sea posible.

Configuración de la topología para el direccionamiento a zonas según preferencias

Considere el siguiente escenario. Tiene dos centros de datos: Chicago y Londres. Para minimizar el tiempo de respuesta de los clientes, desea que los clientes lean y escriban los datos en su centro de datos local.

Los fragmentos primarios se deben colocar en cada centro de datos de forma que se puedan escribir las transacciones localmente desde cada ubicación. Los clientes deben conocer las zonas para poder direccionar a la zona local.

La colocación por contenedor localiza nuevos fragmentos primarios en cada contenedor que se inicia. Las réplicas se colocan según la zona y las reglas de colocación especificadas por la política de despliegue. De forma predeterminada, una réplica se coloca en una zona distinta a su fragmento primario. Tenga en cuenta la siguiente política de despliegue para este escenario.

```
<?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>
```

Cada contenedor que se inicia con la política de despliegue recibe tres nuevos fragmentos primarios. Cada fragmento primario tiene una réplica asíncrona. Inicie cada contenedor con el nombre de zona apropiado. Utilice el parámetro **-zone** si está iniciando los contenedores con el script **startOgServer**.

Para un servidor de contenedor de 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
```

Si los contenedores se ejecutan en WebSphere Application Server, debe crear un grupo de nodos y nombrarlo con el prefijo `ReplicationZone`. Los servidores que se ejecutan en los nodos en estos grupos de nodos se colocan en la zona correspondiente. Por ejemplo, los servidores en ejecución en un nodo de Chicago podrían estar en un grupo de nodos denominado `ReplicationZoneChicago`.

Si desea más información, consulte "Configuración de zonas para la colocación de réplicas" en la página 238.

Los fragmentos primarios de la zona de Chicago tienen réplicas en la zona de Londres. Los fragmentos primarios de la zona de Londres tienen réplicas en la zona de Chicago.

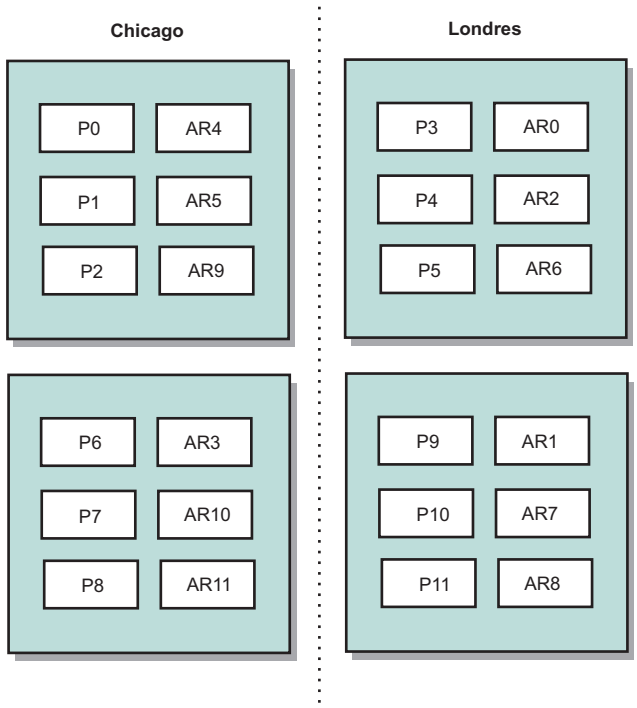


Figura 29. Primarios y réplicas en las zonas

Establezca las zonas preferidas para los clientes. Proporcione un archivo de propiedades de cliente a la máquina virtual Java (JVM) de cliente. Cree un archivo denominado `objectGridClient.properties` y asegúrese de que este archivo esté en la classpath.

Incluya la propiedad **`preferZones`** en el archivo. Establezca el valor de propiedad en la zona apropiada. Los clientes de Chicago deben tener el siguiente valor en el archivo `objectGridClient.properties`:

```
preferZones=Chicago
```

El archivo de propiedades de los clientes de Londres debe contener el siguiente valor:

```
preferZones=London
```

Esta propiedad indica a cada cliente que direccione las transacciones a su zona local, si es posible. La topología replica de forma asíncrona los datos insertados en un fragmento primario de la zona local en la zona foránea.

Utilización de la interfaz SessionHandle para direccionar a la zona local

La estrategia de colocación por contenedor no utiliza un algoritmo basado en hash para determinar la ubicación de los pares clave-valor de la cuadrícula de datos. Debe utilizar objetos SessionHandle para asegurarse de que las transacciones se direccionan a la ubicación correcta cuando se utiliza esta estrategia de colocación. Cuando se confirma una transacción, se enlaza un objeto SessionHandle a la sesión, si no se ha establecido aún uno. También se puede enlazar el objeto SessionHandle a la sesión llamando al método Session.getSessionHandle antes de confirmar la transacción. El fragmento de código siguiente muestra el enlace de un SessionHandle antes de confirmar la transacción.

```
Session ogSession = objectGrid.getSession();

// enlazando SessionHandle
SessionHandle sessionHandle = ogSession.getSessionHandle();

ogSession.begin();
ObjectMap map = ogSession.getMap("planet");
map.insert("planet1", "mercury");

// la transacción se direcciona a la partición especificada por SessionHandle
ogSession.commit();
```

Supongamos que el código anterior se ejecutaba en un cliente en el centro de datos de Chicago. El atributo **preferZones** se establece en `Chicago` para este cliente. Como resultado, el despliegue direccionaría las transacciones a una de las particiones primarias de la zona de Chicago: partición 0, 1, 2, 6, 7 u 8.

El objeto SessionHandle proporciona una vía de acceso a la partición que está almacenando estos datos confirmados. El objeto SessionHandle se debe reutilizar o reconstruir y establecer en la sesión para volver a la partición que contiene los datos confirmados.

```
ogSession.setSessionHandle(sessionHandle);
ogSession.begin();

// el valor devuelto será "mercury "
String value = map.get("planet1");
ogSession.commit();
```

La transacción de este código reutiliza el objeto SessionHandle creado durante la transacción de inserción. A continuación, la transacción get direcciona a la partición que contiene los datos insertados. Sin el objeto SessionHandle, la transacción no puede recuperar los datos insertados.

Cómo afectan los errores de contenedor y zona en el direccionamiento basado en zonas

Normalmente, un cliente con la propiedad **preferZones** establecida direcciona todas las transacciones a la zona o zonas especificadas. Sin embargo, la pérdida de un contenedor causa que un fragmento de réplica pase a ser un fragmento primario. Un cliente que direccionaba anteriormente a las particiones de la zona local debe recuperar los datos insertados anteriormente de la zona remota.

Considere el siguiente escenario. Se ha perdido un contenedor de la zona de Chicago. Previamente, contenía los primarios para las particiones 0, 1 y 2. A continuación, los nuevos fragmentos primarios de estas particiones se colocan en la zona de Londres ya que la zona de Londres alojaba las réplicas de estas particiones.

Cualquier cliente de Chicago que utilice un objeto SessionHandle que apunte a una de las particiones que se ha migrado tras el error ahora se direccionará a Londres. Los clientes de Chicago que utilizan nuevos objetos SessionHandle se direccionarán a fragmentos primarios basados en Chicago.

De forma similar, si se pierde toda la zona de Chicago, todas las réplicas de la zona de Londres pasarán a ser fragmentos primarios. En este escenario, todos los clientes de Chicago direccionan sus transacciones a Londres.

Definición de zonas para servidores de contenedor

Las zonas son colecciones de servidores de contenedor. Un servidor de contenedor puede pertenecer sólo a una zona. Un servidor de contenedor se asigna a una zona cuando se inicia.

Acerca de esta tarea

Debe planificar las zonas antes de iniciar los servidores de contenedor porque los servidores de contenedor definen su pertenencia a zona al iniciarse. Si desea cambiar la pertenencia de zona de un servidor de contenedor, debe reiniciar el servidor con la nueva información de zona.

Procedimiento

- **Defina zonas para servidores de contenedor autónomos.**
  1. Utilice el parámetro **-zone** del script **startOgServer** para especificar la zona para todos los contenedores del servidor iniciado. Para obtener más información sobre cómo iniciar los servidores, consulte “Script **startOgServer**” en la página 401.
  2. También puede asignar nombres de zona al iniciar servidores de contenedor mediante programación con la API del servidor incorporada. Para obtener más información, consulte “Utilización de la API de servidor incorporado para iniciar y detener servidores” en la página 410.
- **Defina zonas para servidores de contenedor que se ejecutan en WebSphere Application Server.**

  Puede utilizar grupos de nodos para colocar servidores de contenedor en zonas específicas. Utilice la sintaxis siguiente para nombrar su grupo de nodos para asignarlo a una zona: `ReplicationZone<identificador>`. Al definir zonas en el despliegue de la política, debe dar a las zonas exactamente el mismo nombre que los grupos de nodos. El nombre de grupo de nodos y el nombre de zona del archivo XML de descriptor de política de despliegue deben ser idénticos.

  Importante: WebSphere Application Server no prohíbe que haya nodos que estén en varios grupos de nodos. Puesto que los servidores de contenedor sólo puede ser una zona, asegúrese de que los nodos están exactamente en un grupo de nodos ReplicationZone.

  Por ejemplo, divida cuatro nodos en dos zonas, A y B.

  1. Configure cuatro nodos: node1, node2, node3 y node4, donde cada nodo tiene dos servidores.

2. Cree un grupo de nodos denominado ReplicationZoneA y un grupo de nodos denominado ReplicationZoneB.
3. Añada node1 y node2 a ReplicationZoneA y añada node3 y node4 a ReplicationZoneB.
4. Defina ReplicationZoneA y ReplicationZoneB en el archivo XML de descriptor de política de despliegue. Consulte “Ejemplo: Zonas en un entorno de WebSphere Application Server” en la página 250 para ver un ejemplo.
5. Cuando se inician los servidores en node1 y node2, estos se unen a ReplicationZoneA, o a la zona A de la configuración de WebSphere eXtreme Scale. Los servidores en el node3 y el node4 se unen a ReplicationZoneB, como zona B en la configuración de WebSphere eXtreme Scale.

Ejemplo: Definiciones de zona en el archivo XML de descriptor de política de despliegue

Puede especificar zonas y reglas de zonas con el archivo XML de descriptor de política de despliegue.

Ejemplo: Fragmentos primarios y de réplica en distintas zonas

Este ejemplo coloca los fragmentos primarios en una zona, y los fragmentos de réplica en una zona distinta, con una única réplica asíncrona. Todos los fragmentos primarios se inician en la zona DC1. Los fragmentos de réplica se inician en la zona DC2.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
 ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
 <objectgridDeployment objectgridName="library">
<mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
maxSyncReplicas="0" maxAsyncReplicas="1">
<map ref="book" />
  <zoneMetadata>
<shardMapping shard="P" zoneRuleRef="primaryRule"/>
<shardMapping shard="A" zoneRuleRef="replicaRule"/>
<zoneRule name ="primaryRule">
     <zone name="DC1" />
    </zoneRule>
<zoneRule name="replicaRule">
    </zoneRule>
  </zoneMetadata>
      </mapSet>
</objectgridDeployment>
 </deploymentPolicy>
```

Se define una réplica asíncrona en el elemento de mapSet ms1. Por lo tanto, existen dos fragmentos en cada partición: un fragmento primario y una réplica asíncrona. En el elemento zoneMetadata, se define un elemento shardMapping para cada fragmento: P para el primario y DC1 para la réplica asíncrona. El atributo primaryRule define la zona establecida para los fragmentos primarios, que es solo la zona DC1, y esta regla se utilizará para la colocación de fragmentos primarios. Las réplicas asíncronas se colocan en la zona DC2.

Si embargo, si se pierde la zona DC2, los fragmentos de réplica dejarán de estar disponibles. La pérdida o anomalía de un servidor de contenedor en la zona DC1 puede producir pérdida de datos, aunque se haya especificado una réplica.

Para tratar esta posibilidad, puede añadir una zona o añadir una réplica, tal como se describe en las secciones siguientes.

Ejemplo: Añadir una zona, fragmentos de escritura en bandas

El código siguiente configura una nueva zona:

```
<?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>
```

Se han definido tres zonas en total en este código: A, B y C. En lugar de reglas de zona de primario y de réplica distintas, se define una regla de zona compartida denominada stripeRule. Esta regla incluye todas las zonas, con el atributo exclusivePlacement establecido en `true`. La política de colocación de eXtreme Scale garantiza que los fragmentos primario y de réplica se encuentran en zonas distintas. Esta escritura en bandas de la colocación hace que los fragmentos primario y de réplica se distribuyan entre ambas zonas para adecuarse a esta política. La adición de una tercera zona C garantiza que la pérdida de cualquiera de las zonas no producirá pérdida de datos, y sigue dejando los fragmentos primario y de réplica para cada partición. Una anomalía de zona produce pérdida del fragmento primario, del fragmento de réplica, o de ninguno de ellos. Cualquier fragmento perdido se sustituye desde el fragmento superviviente en una zona superviviente, colocándolo en la otra zona superviviente.

Ejemplo: Añadir una réplica y definir varios centros de datos

El clásico escenario de dos centros de datos tiene redes de baja latencia y alta velocidad en cada centro de datos, pero alta latencia entre los centros de datos. Las réplicas síncronas se utilizan en cada centro de datos donde la baja latencia minimiza el impacto de la réplica en los tiempos de respuesta. Se utiliza la réplica asíncrona entre los centros de datos, de forma que la red de latencia alta no tiene ningún impacto en el tiempo de respuesta.

```
<?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>
```

La réplica primaria y la réplica síncrona comparten la regla primarySync con un valor de atributo exclusivePlacement de `false`. El atributo exclusivePlacement establecido en false crea una configuración con los fragmentos de réplica primaria y síncrona de cada partición colocada en la misma zona. El fragmento de réplica asíncrona utiliza una segunda regla de zona con básicamente las mismas zonas que la regla de zona primarySync. Sin embargo, la réplica asíncrona utiliza el atributo exclusivePlacement establecido en `true`. El atributo exclusivePlacement, cuando se establece en true, indica que un fragmento no se puede colocar en una zona con otro fragmento de la misma partición. Como resultado, el fragmento de réplica asíncrona no se coloca en la misma zona que el fragmento de réplica primaria o síncrona. Hay tres fragmentos por partición en esta mapSet: una réplica primaria, y una réplica síncrona y asíncrona, de forma que hay tres elementos shardMapping, uno para cada fragmento.

Si se pierde una zona, las réplicas asíncronas se perderán, y no se volverán a generar, ya que no tienen ninguna zona aparte. Si se pierden los fragmentos primario y de réplica, la réplica asíncrona superviviente pasará a ser la primaria, y se creará una nueva réplica síncrona en la zona. Los primarios y las réplicas se fragmentan en cada zona.

Con la colocación exclusiva, cada fragmento tiene su propia zona: debe tener suficientes zonas para todos los fragmentos que desea colocar en sus propias zonas. Si una regla tiene una zona, solo se puede colocar un único fragmento en la zona. Con dos zonas, puede tener hasta dos fragmentos en la zona.

Ejemplo: Zonas en un entorno de WebSphere Application Server

El código siguiente configura una nueva zona:

```
<?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>
```

Para este ejemplo, se definen tres grupos de nodos en el entorno de WebSphere Application Server: ReplicationZoneA, ReplicationZoneB y ReplicationZoneC. El nombre de grupo de nodos y el nombre de zona en el archivo XML de descriptor de política de despliegue deben ser idénticos, y deben contener el texto `ReplicationZone<identificador>`. Este archivo define una configuración similar al ejemplo de fragmentos de escritura en bandas, pero muestra la denominación necesaria para una configuración de WebSphere Application Server.

Visualización de la información de zona con el programa de utilidad `xscmd`

Puede utilizar el programa de utilidad de ejemplo **`xscmd`** para visualizar información sobre el despliegue de la zona actual, incluidos los datos de colocación de fragmentos.

Antes de empezar

- Despliegue una cuadrícula de datos distribuida con varios centros de datos. Si desea más información, consulte "Direccionamiento a zonas según preferencias" en la página 243.

Acerca de esta tarea

Puede determinar la información sobre la configuración relacionada con los valores de zona mediante el programa de utilidad **`xscmd`** que se proporciona con el producto.

Procedimiento

Utilice el programa de utilidad **`xscmd`** para determinar la información sobre los fragmentos de datos. Ejecute el siguiente mandato:

```
xscmd -c showPlacement -z nombre_zona
```

Ejemplo

También puede ejecutar un escenario más simple utilizando el ejemplo de iniciación: *raíz_intal_wxs*`/ObjectGrid/gettingstarted`. Si desea más información, consulte "Guía de aprendizaje: Cómo empezar con WebSphere eXtreme Scale" en la página 1.

1. Inicie un servidor de catálogo:

   ```
   runcat.bat
   ```

2. Determine el número necesario de réplicas, reglas de zona, contenedores y otros valores como por ejemplo con el mandato siguiente: `startOgServer.bat serverA0 -objectgridFile xml\objectgrid.xml -deploymentPolicyFile xml\deployment.xml -zone zoneA`
3. Puede detener los procesos de contenedor para simular una anomalía en la cuadrícula de datos: `stopOgServer.bat serverA0,serverA1,serverB0 -catalogServiceEndPoints localhost:2809`.

   Si se detiene el servidor que contiene el último fragmento de una partición, eXtreme Scale asigna un nuevo fragmento primario. Puede comprobar si hay pérdida de datos:

   - El script **`runclient`** inserta y lee elementos en la cuadrícula de datos.
   - El mandato **`xscmd -c showMapSizes`** muestra el número de elementos de la cuadrícula de datos.

4. Muestre los servidores de contenedor activos con el mandato siguiente:

   ```
   xscmd -c showPlacement -z nombre_zona
   ```

Configuración de los servidores de catálogo y de contenedor

WebSphere eXtreme Scale tiene dos tipos de servidores: servidores de catálogo y servidores de contenedor. Los servidores de catálogos controlan la colocación de fragmentos y descubren y supervisan los servidores de contenedor. Varios servidores de catálogo comprimen de forma conjunta el servicio de catálogo. Un servidor de contenedor es una Máquina virtual Java (JVM) que almacena los datos de la aplicación para la cuadrícula de datos.

Acerca de esta tarea

Los servidores de catálogo y de contenedor pueden iniciar procesos de WebSphere Application Server, como procesos Java SE autónomos, o incorporando los servidores en aplicaciones Java SE. La manera de configurar los servidores de catálogo y contenedor dependerá de su topología.

Servidores de catálogo

- **Servidores de catálogo autónomos**:

  Configure los servidores de catálogo autónomos con un archivo de propiedades de servidor. Controle el ciclo de vida de un servidor de catálogo con los scripts **startOgServer** y **stopOgServer** o mediante la API del servidor incorporada.
- **Servidores de catálogo que se inician en WebSphere Application Server:**

  Configure los servidores de catálogo que se ejecuten en WebSphere Application Server mediante la consola administrativa, las tareas administrativas y el archivo de propiedades de servidor de WebSphere Application Server. El ciclo de vida del servidor lo controla el ciclo de vida del proceso en WebSphere Application Server. Cuando se inician o detienen procesos en WebSphere Application Server, los servidores de catálogo que se ejecutan en estos procesos también se inician o detienen.

Servidores de contenedor

- **Servidores de contenedor autónomos**:

  Configure los servidores de contenedor autónomos con un archivo de propiedades de servidor y un archivo XML de política de despliegue. Controle el ciclo de vida de un servidor de contenedor mediante los scripts **startOgServer** y **stopOgServer** o mediante la API del servidor incorporada.
- **Servidores de contenedor que se inician en WebSphere Application Server:**

  Configure los servidores de contenedor en WebSphere Application Server mediante un archivo de propiedades de servidor y un archivo XML de política de despliegue que está incorporado en un módulo de aplicación Java EE. El ciclo de vida de los servidores de contenedor lo controla la aplicación. Los servidores de contenedor se inician y detienen con la aplicación.

Utilice los temas siguientes para configurar los servidores de catálogo y contenedor:

Procedimiento recomendado: Agrupación en clúster del servicio de catálogo con dominios de servicio de catálogo

Cuando se utiliza el servicio de catálogo, se requiere un mínimo de dos servidores de catálogo para evitar un punto único de anomalía. En función del número de nodos en el entorno, puede crear distintas configuraciones para garantizar que como mínimo haya dos servidores de catálogo siempre en ejecución.

Número de servidores de catálogo

El procedimiento recomendado para evitar un punto único de anomalía del dominio de servicio de catálogo es iniciar un mínimo de tres servidores de catálogo en tres nodos distintos.

Si está utilizando solo dos nodos, configure dos servidores de catálogo en cada uno de los dos nodos para tener un total de cuatro procesos de servidor de catálogo. La creación de esta configuración garantiza que cuando solo está iniciado uno de los nodos, los dos servidores de catálogo necesarios están en ejecución. Debe iniciar como mínimo dos servidores de catálogo simultáneamente. Cuando se inician los servidores de catálogo, estos buscan otros servidores de catálogo en la configuración, y no se inician satisfactoriamente hasta que cómo mínimo se encuentra uno de los servidores de catálogo.

Ejemplo: Inicio de cuatro servidores de catálogo en cuatro nodos en un entorno autónomo

El script siguiente inicia los servidores de catálogo cs0 y cs1 en el nodo host1, e inicia los servidores de catálogo cs2 y cs3 en el nodo 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
```

Recuerde: Debe utilizar la opción **-listenerPort** porque cada uno de los servidores de catálogo que está en ejecución en un nodo requiere un número de puerto exclusivo.

Ejemplo: Inicio de varios servidores de catálogo en un entorno de WebSphere Application Server

Los servidores de catálogo se inician automáticamente en un entorno de WebSphere Application Server. Puede definir varios servidores de catálogo para su inicio creando un dominio de servicio de catálogo. Después de especificar varios puntos finales en el dominio de servicio de catálogo, reinicie los servidores de aplicaciones incluidos de forma que los servidores de catálogo se inicien en paralelo.

- **WebSphere Application Server Network Deployment**: puede seleccionar varios servidores de aplicaciones existentes desde la célula para que sean miembros de su dominio de servicio de catálogo.
- **WebSphere Application Server** base: puede iniciar el servicio de catálogo en varios nodos autónomos. Definiendo varios perfiles en la misma imagen de instalación con la herramienta de gestión de perfiles, puede crear un conjunto de nodos autónomos cada uno de los cuales tenga puertos exclusivos asignados. En cada servidor de aplicaciones, defina el dominio de servicio de catálogo. Puede especificar cualquier otro servidor de aplicaciones añadiendo servidores remotos

a la configuración. Después de crear esta configuración en todos los servidor autónomos, puede iniciar el conjunto de servidores de aplicaciones base en paralelo ejecutando el script **startServer** o utilizando un servicio Windows para iniciar los servidores.

Ajuste del valor de intervalo de pulsación para la detección de migración tras error

Puede configurar la cantidad de tiempo entre las comprobaciones de sistema para los servidores que han fallado con el valor de intervalo de pulsaciones.

Acerca de esta tarea

La configuración de la migración tras error varía en función del tipo de entorno que utiliza. Si utiliza un entorno autónomo, puede configurar una migración tras error con la línea de mandatos. Si utiliza un entorno WebSphere Application Server Network Deployment, debe configurar la migración tras error en la consola de administración de WebSphere Application Server Network Deployment.

Procedimiento

- Configure la migración tras error para los entornos autónomos.

  Puede configurar los intervalos de pulsación en la línea de mandatos utilizando el parámetro **-heartbeat** en el archivo de script **startOgServer**. Establezca este parámetro en uno de los siguientes valores:

  Tabla 13. Intervalos de pulsaciones

Valor	Acción	Descripción
0	Típica (valor predeterminado)	Las migraciones tras error se detectan normalmente en 30 segundos.
-1	Agresiva	Las migraciones tras error se detectan normalmente en 5 segundos.
1	Relajada	Las migraciones tras error se detectan normalmente en 180 segundos.

  Un intervalo de pulsaciones agresivo puede ser útil cuando los procesos y la red son estables. Si la red o los procesos no se han configurado de forma óptima, es posible que las pulsaciones se pierdan, lo que comportará en una detección de anomalía falsa.

- Configure la migración tras error para los entornos WebSphere Application Server.

  Puede configurar WebSphere Application Server Network Deployment versión 6.0.2 y posterior para permitir a WebSphere eXtreme Scale que realice la migración tras error muy rápidamente. El tiempo de migración tras error predeterminado para las anomalías graves es aproximadamente de 200 segundos. Una anomalía grave es un bloqueo del servidor o sistema físico, una desconexión del cable de red o un error del sistema operativo. Las anomalías debidas a cuelgues del proceso o a anomalías leves normalmente realizan la migración tras error en menos de un segundo. La detección de anomalías correspondientes a anomalías leves sucede cuando el sistema operativo cierra automáticamente los sockets de red del proceso inactivo para el servidor que aloja el proceso.

  Configuración de pulsaciones de grupo principal

  WebSphere eXtreme Scale que se ejecuta en un proceso WebSphere Application Server hereda las características de migración tras error de los valores del grupo

principal del servidor de aplicaciones. Las siguientes secciones describen cómo configurar los valores de pulsación del grupo principal para distintas versiones de WebSphere Application Server Network Deployment:

– **Actualice los valores de grupo principal para WebSphere Application Server Network Deployment versión 6.x y 7.x:**

  Especifique el intervalo de pulsación en segundos en las versiones de WebSphere Application Server de la versión 6.0 a la versión 6.1.0.12 o en milisegundos a partir de la versión 6.1.0.13. También debe especificar el número de pulsaciones que faltan. Este valor indica cuántas pulsaciones pueden perderse antes de que se considere anómala una Máquina virtual Java (JVM) de igual. El tiempo de detección de anomalías graves es aproximadamente el producto del intervalo de pulsaciones y el número de pulsaciones perdidas.

  Estas propiedades se especifican utilizando las propiedades personalizadas en el grupo principal a través de la consola administrativa de WebSphere. Consulte Propiedades personalizadas del grupo principal para obtener detalles sobre la configuración. Estas propiedades deben especificarse para todos los grupos principales que la aplicación utiliza:

  - El intervalo de pulsación se especifica utilizando la propiedad personalizada IBM_CS_FD_PERIOD_SEC para segundos o la propiedad personalizada IBM_CS_FD_PERIOD_MILLIS para milisegundos (requiere la Versión 6.1.0.13 o posterior).
  - El número de pulsaciones perdidas se especifica utilizando la propiedad personalizada IBM_CS_FD_CONSECUTIVE_MISSED.

  El valor predeterminado para la propiedad IBM_CS_FD_PERIOD_SEC es 20 y para la propiedad IBM_CS_FD_CONSECUTIVE_MISSED es 10. Si se especifica la propiedad IBM_CS_FD_PERIOD_MILLIS, altera temporalmente cualquier conjunto de propiedades personalizadas IBM_CS_FD_PERIOD_SEC. Los valores de estas propiedades son valores enteros positivos.

  Utilice los siguientes valores para conseguir un tiempo de detección de anomalías de 1500 ms para los servidores WebSphere Application Server Network Deployment versión 6.x:

  - Establezca `IBM_CS_FD_PERIOD_MILLIS = 750` (WebSphere Application Server Network Deployment V6.1.0.13 y posterior)
  - Establezca `IBM_CS_FD_CONSECUTIVE_MISSED = 2`

– **Actualice los valores de grupo principal para WebSphere Application Server Network Deployment versión 7.0**

  WebSphere Application Server Network Deployment versión 7.0 proporciona dos valores de grupo principal que se pueden ajustar para aumentar o reducir la detección de migración tras error:

  - **Periodo de transmisión de pulsación.** El valor predeterminado es 30000 milisegundos.
  - **Periodo de tiempo de espera de pulsación.** El valor predeterminado es 180000 milisegundos.

  Si desea más detalles sobre cómo cambiar estos valores, consulte el centro de información de WebSphere Application Server Network Deployment: Valores de descubrimiento y detección de errores.

  Utilice los valores siguientes para conseguir un tiempo de detección de anomalías de 1500 ms para los servidores WebSphere Application Server Network Deployment versión 7:

  - Establezca el periodo de transmisión de pulsaciones en 750 milisegundos.

- Establezca el periodo de tiempo de espera de pulsaciones en 1500 milisegundos.

Qué hacer a continuación

Cuando estos valores se modifican para proporcionar tiempos de migración tras error cortos, se debe tener en cuenta algunas cuestiones relativas al ajuste del sistema. En primer lugar, Java no es un entorno de tiempo real. Es posible que las hebras se demoren si JVM está sufriendo tiempos de recogida de basura de larga duración. Las hebras también podrían demorarse si la máquina que aloja la JVM tiene mucha carga (debido a la propia JVM o a otros procesos que se ejecutan en la máquina). Si las hebras se retrasan, es posible que las pulsaciones no se envíen a tiempo. En el peor de los casos, podrían demorarse el tiempo de migración tras error necesario. Si las hebras se demoran, se producen detecciones de anomalías falsas. El sistema se debe ajustar y se debe modificar su tamaño para asegurarse de que las detecciones de anomalías falsas no se producen en un entorno de producción. La mejor manera de garantizarlo es utilizando una carga adecuada durante la fase de prueba.

Nota: La versión actual de eXtreme Scale soporta WebSphere Real Time.

Configuración de WebSphere eXtreme Scale con WebSphere Application Server

Puede ejecutar los procesos de servicio de catálogo y de servidor de contenedor en WebSphere Application Server. El proceso para configurar estos servidores es diferente que una configuración autónoma. El servicio de catálogo se puede iniciar automáticamente en los servidores o los gestores de despliegue de WebSphere Application Server. El proceso de contenedor se inicia cuando se despliega una aplicación eXtreme Scale en el entorno WebSphere Application Server.

Acerca de esta tarea

Atención: No ubique conjuntamente los servidores de contenedor y los servidores de catálogo en un entorno de producción. Incluya el servicio de catálogo en varios procesos de agente de nodo o en un servidor de aplicaciones que no contenga ninguna aplicación eXtreme Scale.

Configuración del servicio de catálogo en WebSphere Application Server

Los procesos de servicio de catálogo se pueden ejecutar en WebSphere Application Server. El ciclo de vida del servidor en WebSphere Application Server determina cuándo se inicia y detiene el servicio de catálogo.

Procedimiento

1. Elija uno o más procesos de WebSphere Application Server para aumentar con el perfil de WebSphere eXtreme Scale. Si desea más información, consulte “Creación y aumento de perfiles para WebSphere eXtreme Scale” en la página 184. Si desea que el servicio de catálogo se inicie automáticamente en WebSphere Application Server Network Deployment en el gestor de despliegue, instale WebSphere eXtreme Scale en el nodo del gestor de despliegue y aumente el perfil del gestor de despliegue.
2. Configure los archivos de propiedades del servidor para los procesos de WebSphere Application Server y añádalo a la classpath del nodo. Consulte el apartado Archivo de propiedades de servidor para obtener más información.

3. Configure un dominio de servicio de catálogo. El dominio de servicio de catálogo es un grupo de servidores de catálogo del entorno. Consulte el apartado “Creación de dominios de servicio de catálogo en WebSphere Application Server” para obtener más información.
4. Inicie los procesos de WebSphere Application Server que alojan los servidores de catálogo. Si desea más información, consulte “Inicio y detención de servidores en un entorno de WebSphere Application Server” en la página 409.

Creación de dominios de servicio de catálogo en WebSphere Application Server:

Los dominios de servicio de catálogo definen un grupo de servidores de catálogo que gestionan la colocación de fragmentos y supervisan el estado de los servidores de contenedor de la cuadrícula de datos.

Antes de empezar

- Instale WebSphere eXtreme Scale en WebSphere Application Server. Si desea más información, consulte “Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server” en la página 163.

Acerca de esta tarea

Al crear un dominio de servicio de catálogo está definiendo una colección de alta disponibilidad de servidores de catálogo.

Estos servidores de catálogo se pueden ejecutar en WebSphere Application Server dentro de una única célula y grupo principal. El dominio de servicio de catálogo también puede definir un grupo remoto de servidores que se ejecutan en distintos procesos de Java SE o células de WebSphere Application Server.

Para los servidores de catálogo que se ejecutan en servidores de aplicaciones existentes dentro de la célula: El usuario define un dominio de servicio de catálogo que ubica servidores de catálogo en servidores de aplicaciones dentro de la célula y se utilizan los mecanismos del grupo principal de WebSphere Application Server. El servicio de catálogo se inicia automáticamente en los servidores de aplicaciones de la célula. En consecuencia, los miembros de un único dominio de servicio de catálogo no pueden abarcar los límites de un grupo principal y, por ello, el dominio de servicio de catálogo no puede abarcar células. No obstante, los clientes y servidores de contenedor de WebSphere eXtreme Scale pueden abarcar células conectándose a un servidor de catálogo entre los límites de célula como, por ejemplo, un dominio de servicio de catálogo autónomo o un dominio de servicio de catálogo integrado en otra célula.

Para los servidores de catálogo remotos: puede conectar contenedores y clientes de WebSphere eXtreme Scale a un dominio de servicio de catálogo que se esté ejecutando en otra célula de WebSphere Application Server o que se esté ejecutando como proceso autónomo. Puesto que los servidores de catálogo configurados remotamente no se inician automáticamente en la célula, debe iniciar manualmente los servidores de catálogo configurados remotamente. Cuando configure un dominio de servicio de catálogo remoto, el nombre de dominio debe coincidir con el nombre de dominio especificado al iniciar los servidores de catálogo remotos. El nombre de dominio de servicio de catálogo predeterminado para los servidores de catálogo autónomos es `DefaultDomain`. Especifique un nombre de dominio de servicio de catálogo con el parámetro **`-domain`** del mandato **`startOgServer`**, un archivo de propiedades de servidor o con la API de servidor incluida. Debe iniciar cada proceso de servidor de catálogo remoto en el dominio

remoto con el mismo nombre de dominio. Consulte “Inicio de un servicio de catálogo autónomo” en la página 395 para obtener más información sobre cómo iniciar los servidores de catálogo.

Atención: No coloque los servicios de catálogo con servidores de contenedor de WebSphere eXtreme Scale en un entorno de producción. Incluya el servicio de catálogo en varios procesos de nodo de agente o en un servidor de aplicaciones que no esté alojando una aplicación WebSphere eXtreme Scale.

Procedimiento

1. Cree el dominio de servicio de catálogo.
   a. En la consola administrativa de WebSphere Application Server, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo** > **Nuevo**.
   b. Defina un nombre, un valor predeterminado y credenciales de autenticación JMX para el dominio de servicio de catálogo. Si está configurando puntos finales remotos para el dominio de servicio de catálogo, el nombre del dominio de servicio de catálogo debe coincidir con el nombre del dominio de servicio de catálogo que especifique cuando inicie los servidores de catálogo.
   c. Añada puntos finales de servidor de catálogo. Puede seleccionar servidores de aplicaciones existentes o añadir servidores remotos que ejecuten un servicio de catálogo.
2. Compruebe la conexión con los servidores de catálogo del dominio de servicio de catálogo. Para los servidores de aplicación existentes, los servidores de catálogo se inician cuando se inicia el servidor de aplicación asociado. Para los servidores de aplicación remotos, debe iniciar los servidores manualmente mediante el mandato **startOgServer** o la API de servidor incluida.
   a. En la consola administrativa de WebSphere Application Server, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo**.
   b. Seleccione el dominio de servicio de catálogo que desee probar y pulse **Probar conexión**. Cuando se pulsa este botón, se consultan todos los puntos finales de dominio de servicio de catálogo uno por uno (si es que hay algún punto final disponible) y el sistema devuelve un mensaje indicando que la conexión con el dominio de servicio de catálogo ha sido satisfactoria.

Tareas administrativas del dominio de servicio de catálogo:

Puede utilizar los lenguajes de script Jacl o Jython para gestionar dominios de servicio de catálogo en la configuración de WebSphere Application Server.

Requisitos

Debe haber instalado WebSphere eXtreme Scale Client en el entorno de WebSphere Application Server.

Lista de todas las tareas administrativas

Para obtener una lista de todas las tareas administrativas asociadas con los dominios de servicio de catálogo, ejecute el siguiente mandato con wsadmin:

```
wsadmin>$AdminTask help XSDomainManagement
```

Mandatos

Las tareas administrativas para dominios de servicio de catálogo incluyen los siguientes mandatos:

- "createXSDomain"
- "deleteXSDomain" en la página 262
- "getDefaultXSDomain" en la página 262
- "listXSDomains" en la página 263
- "modifyXSDomain" en la página 263
- "testXSDomainConnection" en la página 268
- "testXSServerConnection" en la página 268

createXSDomain

El mandato **createXSDomain** registra un nuevo dominio de servicio de catálogo.

Tabla 14. Argumentos del mandato createXSDomain

Argumento	Descripción
-name (necesario)	Especifica el nombre del dominio de servicio de catálogo que el usuario desea crear.
-default	Especifica si el dominio de servicio de catálogo es el predeterminado para la célula. El valor predeterminado es `true`. (Booleano: establecido en `true` o`false`)
-properties	Especifica las propiedades personalizadas del dominio de servicio de catálogo.

Tabla 15. Argumentos del paso defineDomainServers

Argumento	Descripción
nombre_de_punto_final	Especifica el nombre del punto final de servicio de catálogo. • **Para los servidores de aplicación existentes**: el nombre del punto final debe tener el formato siguiente: *nombre_de_célula**nombre_de_nodo*\ *nombre_de_servidor* • **Para los servidores remotos**: especifica el nombre de host del servidor remoto. Puede tener el mismo nombre para varios puntos finales pero los valores de puerto de cliente deben ser exclusivos para cada punto final.
propiedades_personalizadas	Especifica las propiedades personalizadas del punto final del dominio de servicio de catálogo. Si no tiene propiedades personalizadas, utilice un juego de comillas ("") para este argumento.

Tabla 15. Argumentos del paso defineDomainServers (continuación)

Argumento	Descripción
puertos_de_punto_final	Especifica los números de puerto para el punto final del dominio de servicio de catálogo. Los puertos se deben especificar por el orden siguiente: <*puerto_de_cliente*>,<*puerto_de_escucha*> **Puerto de cliente** Especifica el puerto que se utiliza para la comunicación entre los servidores de catálogo en el dominio de servicio de catálogo. Este valor es necesario para los servidores de catálogo que se están ejecutando sólo en procesos de WebSphere Application Server y se puede establecer en cualquier puerto que no se esté utilizando en otro sitio. **Puerto de escucha** Especifica el puerto que se utiliza para la comunicación con clientes. Este valor es necesario para los puntos finales remotos y debe coincidir con el valor utilizado cuando se inició el servicio de catálogo. El puerto de escucha lo utilizan los clientes y los contenedores para comunicarse con el servicio de catálogo. **Para los puntos finales remotos de WebSphere eXtreme Scale**: define el puerto de escucha de intermediario para solicitudes de objetos (ORB) para que los contenedores y clientes se comuniquen con el servicio de catálogo a través del ORB. Para puntos finales de WebSphere Application Server, el valor de puerto de escucha es opcional porque el valor se hereda de la configuración del puerto BOOTSTRAP_ADDRESS.

Tabla 16. Argumentos del paso configureClientSecurity

Argumento	Descripción
-securityEnabled	Especifica que está habilitada la seguridad de cliente para el servidor de catálogo. El archivo de propiedades del servidor que está asociado al servidor de catálogo seleccionado debe tener un valor **securityEnabled** coincidente en el archivo de propiedades del servidor. Si estos valores no coinciden, se produce una excepción. (Booleano: establecido en `true` o`false`)

Tabla 16. Argumentos del paso configureClientSecurity (continuación)

Argumento	Descripción
-credentialAuthentication (opcional)	Indica si se fuerza o si se da soporte a la autenticación de credenciales. **Nunca** No se fuerza la autenticación de certificados. **Necesario** La autenticación de credenciales se fuerza siempre. Si el servidor no da soporte a la autenticación de credenciales, el cliente no puede establecer conexión con el servidor. **Soportado** (Valor predeterminado) Sólo se fuerza la autenticación de credenciales si tanto el cliente como el servidor dan soporte a la misma.
-authenticationRetryCount (opcional)	Especifica el número de veces que se vuelve a intentar la autenticación si la credencial ha caducado. Si no desea volver a intentar la autenticación, establezca el valor a `0`. El valor predeterminado es `0`.
-credentialGeneratorClass	Indica la clase de implementación com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator, de forma que el cliente recupera las señales de seguridad de la hebra.
-credentialGeneratorProps	Especifica las propiedades para la clase de implementación CredentialGenerator. Las propiedades se envían al objeto con el método setProperties(String). El valor propiedades del generador de credenciales se utiliza únicamente cuando se especifica un valor en el campo **Clase del generador de credenciales**.

Valor de retorno:

Uso de ejemplo de modalidad de proceso por lotes

La modalidad de proceso necesita un formato correcto de la entrada de mandatos. Puede utilizar la modalidad interactiva para asegurarse de que los valores especificados se procesan correctamente. Cuando utiliza la modalidad de proceso por lotes, debe definir los argumentos de paso de **-defineDomainServers** mediante una matriz de propiedades específica. Esta matriz de propiedades tiene el formato *nombre_de_punto_final propiedades_personalizadas puertos_de_punto_final*. El valor *puertos_de_punto_final* es una lista de puertos que se deben especificar por el orden siguiente: *<puerto_de_cliente>*,*<puerto_de_escucha>*.

- Crear un dominio de servicio de catálogo de puntos finales remotos mediante Jacl:

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
{{xhost1.ibm.com "" ,2809}} -configureClientSecurity {-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"}}}
```

- Crear un dominio de servicio de catálogo de puntos finales remotos mediante la serie Jython:

```
AdminTask.createXSDomain('[-name TestDomain -default true
-defineDomainServers [[xhost1.ibm.com "" ,2809]
[xhost2.ibm.com "" ,2809]] -configureClientSecurity [-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"] ]')
```

- Crear un dominio de servicio de catálogo de puntos finales de servidor de aplicaciones mediante Jacl:

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
 {{cellName/nodeName/serverName "" 1109}}}
```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

```
$AdminTask createXSDomain {-interactive}
```

- Utilizando la serie de Jython:

```
AdminTask.createXSDomain ('[-interactive]')
```

deleteXSDomain

El mandato **deleteXSDomain** suprime un dominio de servicio de catálogo.

Parámetros necesarios:

-name
Especifica el nombre del dominio de servicio de catálogo que debe suprimirse.

Valor de retorno:

Uso de ejemplo de modalidad de proceso por lotes

- Utilizando Jacl:

```
$AdminTask deleteXSDomain {-name TestDomain }
```

- Utilizando la serie de Jython:

```
AdminTask.deleteXSDomain('[-name TestDomain ]')
```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

```
$AdminTask deleteXSDomain {-interactive}
```

- Utilizando la serie de Jython:

```
AdminTask.deleteXSDomain ('[-interactive]')
```

getDefaultXSDomain

El mandato **getDefaultXSDomain** devuelve el dominio de servicio de catálogo de la célula.

Parámetros necesarios: ninguno

Valor de retorno: el nombre del dominio de servicio de catálogo predeterminado.

Uso de ejemplo de modalidad de proceso por lotes

- Utilizando Jacl:

```
$AdminTask getDefaultXSDomain
```

- Utilizando la serie de Jython:

```
AdminTask.getDefaultXSDomain
```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

```
$AdminTask getDefaultXSDomain {-interactive}
```

- Utilizando la serie de Jython:

```
AdminTask.getDefaultXSDomain ('[-interactive]')
```

listXSDomains

El mandato **listXSDomains** devuelve una lista de los dominios de servicio de catálogo existentes.

Parámetros necesarios: ninguno

Valor de retorno: una lista de todos los dominios de servicio de catálogo de la célula.

Uso de ejemplo de modalidad de proceso por lotes

- Utilizando Jacl:

```
$AdminTask listXSDomains
```

- Utilizando la serie de Jython:

```
AdminTask.listXSDomains
```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

```
$AdminTask listXSDomains {-interactive}
```

- Utilizando la serie de Jython:

```
AdminTask.listXSDomains ('[-interactive]')
```

modifyXSDomain

El mandato **modifyXSDomain** modifica un dominio de servicio de catálogo existente.

La modalidad de proceso necesita un formato correcto de la entrada de mandatos. Puede utilizar la modalidad interactiva para asegurarse de que los valores especificados se procesan correctamente. Cuando utiliza la modalidad de proceso por lotes, debe definir los argumentos de paso de **-modifyEndpoints**, **-addEndpoints** y **-removeEndpoints** mediante una matriz de propiedades específica. Esta matriz de propiedades tiene el formato *nombre_de_punto_final nombre_de_host propiedades_personalizadas puertos_de_punto_final*. El valor *puertos_de_punto_final* es una lista de puertos que se deben especificar por el orden siguiente: *<puerto_de_cliente>*,*<puerto_de_escucha>*.

Tabla 17. Argumentos del mandato modifyXSDomain

Argumento	Descripción
-name (necesario)	Especifica el nombre del dominio de servicio de catálogo que el usuario desea editar.
-default	Si se establece en `true`, especifica que el dominio de servicio de catálogo seleccionado es el valor predeterminado de la célula. (Booleano)
-properties	Especifica las propiedades personalizadas del dominio de servicio de catálogo.

Tabla 18. Argumentos del paso modifyEndpoints

Argumento	Descripción
nombre_de_punto_final	Especifica el nombre del punto final de servicio de catálogo. • **Para los servidores de aplicación existentes**: el nombre del punto final debe tener el formato siguiente: *nombre_de_célula**nombre_de_nodo**nombre_de_servidor* • **Para los servidores remotos**: especifica el nombre de host del servidor remoto. Puede tener el mismo nombre para varios puntos finales pero los valores de puerto de escucha deben ser exclusivos para cada punto final.
puertos_de_punto_final	Especifica los números de puerto para el punto final del dominio de servicio de catálogo. Los puntos finales se deben especificar por el orden siguiente: *<puerto_de_cliente>*,*<puerto_de_escucha>* **Puerto de cliente** Especifica el puerto que se utiliza para la comunicación entre los servidores de catálogo en el dominio de servicio de catálogo. Este valor es necesario para los servidores de catálogo que se están ejecutando sólo en procesos de WebSphere Application Server y se puede establecer en cualquier puerto que no se esté utilizando en otro sitio. **Puerto de escucha** Especifica el puerto que se utiliza para la comunicación con clientes. Este valor es necesario para los puntos finales remotos y debe coincidir con el valor utilizado cuando se inició el servicio de catálogo. El puerto de escucha lo utilizan los clientes y los contenedores para comunicarse con el servicio de catálogo. **Para los puntos finales remotos de WebSphere eXtreme Scale**: define el puerto de escucha de intermediario para solicitudes de objetos (ORB) para que los contenedores y clientes se comuniquen con el servicio de catálogo a través del ORB. Para puntos finales de WebSphere Application Server, la especificación del valor de puerto de escucha es opcional porque el valor se hereda de la configuración del puerto BOOTSTRAP_ADDRESS.

Tabla 19. Argumentos del paso addEndpoints

Argumento	Descripción
nombre_de_punto_final	Especifica el nombre del punto final de servicio de catálogo. • **Para los servidores de aplicación existentes**: el nombre del punto final debe tener el formato siguiente: *nombre_de_célula**nombre_de_nodo*\ *nombre_de_servidor* • **Para los servidores remotos**: especifica el nombre de host del servidor remoto. Puede tener el mismo nombre para varios puntos finales pero los valores de puerto de escucha deben ser exclusivos para cada punto final.
propiedades_personalizadas	Especifica las propiedades personalizadas del punto final del dominio de servicio de catálogo. Si no tiene propiedades personalizadas, utilice un juego de comillas ("") para este argumento.

Tabla 19. Argumentos del paso addEndpoints (continuación)

Argumento	Descripción
puertos_de_punto_final	Especifica los números de puerto para el punto final del dominio de servicio de catálogo. Los puntos finales se deben especificar por el orden siguiente: *<puerto_de_cliente>*,*<puerto_de_escucha>* **Puerto de cliente** Especifica el puerto que se utiliza para la comunicación entre los servidores de catálogo en el dominio de servicio de catálogo. Este valor es necesario para los servidores de catálogo que se están ejecutando sólo en procesos de WebSphere Application Server y se puede establecer en cualquier puerto que no se esté utilizando en otro sitio. **Puerto de escucha** Especifica el puerto que se utiliza para la comunicación con clientes. Este valor es necesario para los puntos finales remotos y debe coincidir con el valor utilizado cuando se inició el servicio de catálogo. El puerto de escucha lo utilizan los clientes y los contenedores para comunicarse con el servicio de catálogo. **Para los puntos finales remotos de WebSphere eXtreme Scale**: define el puerto de escucha de intermediario para solicitudes de objetos (ORB) para que los contenedores y clientes se comuniquen con el servicio de catálogo a través del ORB. Para puntos finales de WebSphere Application Server, la especificación del valor de puerto de escucha es opcional porque el valor se hereda de la configuración del puerto BOOTSTRAP_ADDRESS.

Tabla 20. Argumentos del paso removeEndpoints

Argumento	Descripción
nombre_de_punto_final	Especifica el nombre del punto final de servicio de catálogo que debe suprimirse.

Tabla 21. Argumentos del paso configureClientSecurity

Argumento	Descripción
-securityEnabled	Especifica que está habilitada la seguridad de cliente para el servidor de catálogo. El archivo de propiedades del servidor que está asociado al servidor de catálogo seleccionado debe tener un valor **securityEnabled** coincidente en el archivo de propiedades del servidor. Si estos valores no coinciden, se produce una excepción. (Booleano: establecido en `true` o`false`)
-credentialAuthentication (opcional)	Indica si se fuerza o si se da soporte a la autenticación de credenciales. **Nunca** No se fuerza la autenticación de certificados. **Necesario** La autenticación de credenciales se fuerza siempre. Si el servidor no da soporte a la autenticación de credenciales, el cliente no puede establecer conexión con el servidor. **Soportado** (Valor predeterminado) Sólo se fuerza la autenticación de credenciales si tanto el cliente como el servidor dan soporte a la misma.
-authenticationRetryCount (opcional)	Especifica el número de veces que se vuelve a intentar la autenticación si la credencial ha caducado. Si no desea volver a intentar la autenticación, establezca el valor a `0`. El valor predeterminado es `0`.
-credentialGeneratorClass	Indica la clase de implementación com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator, de forma que el cliente recupera las señales de seguridad de la hebra.
-credentialGeneratorProps	Especifica las propiedades para la clase de implementación CredentialGenerator. Las propiedades se envían al objeto con el método setProperties(String). El valor propiedades del generador de credenciales se utiliza únicamente cuando se especifica un valor en el campo **Clase del generador de credenciales**.

Valor de retorno:

Uso de ejemplo de modalidad de proceso por lotes

- Utilizando Jacl:

```
$AdminTask modifyXSDomain {-name TestDomain -default true -modifyEndpoints
{{xhost1.ibm.com "" ,2809}} -addEndpoints {{xhost2.ibm.com "" ,2809}}}
-removeEndpoints {{xhost3.ibm.com}}}
```

- Utilizando la serie de Jython:

```
AdminTask.modifyXSDomain('[-name TestDomain
-default false -modifyEndpoints [[xhost1.ibm.com "" ,2809]]
-addEndpoints [[xhost3.ibm.com "" ,2809]]
-removeEndpoints [[xhost2.ibm.com]]]')
```

- Uso de la especificación de la seguridad de cliente durante el mandato modify:

```
$AdminTask modifyXSDomain {-name myDomain -default false
-configureClientSecurity {-securityEnabled true -
Supported -authenticationRetryCount 1 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"}}
```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

  ```
  $AdminTask modifyXSDomain {-interactive}
  ```

- Utilizando la serie de Jython:

  ```
  AdminTask.modifyXSDomain ('[-interactive]')
  ```

testXSDomainConnection

El mandato **testXSDomainConnection** prueba la conexión a un dominio de servicio de catálogo.

Parámetros necesarios:

-name
Especifica el nombre del dominio de servicio de catálogo al que se va a probar la conexión.

Parámetros opcionales

-timeout
Especifica la cantidad máxima de tiempo que debe esperarse para que se establezca la conexión en segundos.

Valor de retorno: si se puede establecer una conexión, devuelve `true` y, si no se puede, devuelve información sobre el error de conexión.

Uso de ejemplo de modalidad de proceso por lotes

- Utilizando Jacl:

  ```
  $Admintask testXSDomainConnection
  ```

- Utilizando la serie de Jython:

  ```
  AdminTask.testXSDomainConnection
  ```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

  ```
  $AdminTask testXSDomainConnection {-interactive}
  ```

- Utilizando la serie de Jython:

  ```
  AdminTask.testXSDomainConnection ('[-interactive]')
  ```

testXSServerConnection

El mandato **testXSServerConnection** prueba la conexión a un servidor de catálogo. Este mandato funciona tanto para servidores autónomos como para servidores que forman parte de un dominio de servicio de catálogo.

Parámetros necesarios:

host
Especifica el host en el que reside el servidor de catálogo.

listenerPort
Especifica el puerto de escucha del servidor de catálogo.

Parámetros opcionales

tiempo de espera
Especifica la duración máxima de tiempo que se va a esperar para una conexión al servidor de catálogo, en segundos.

domain
Especifica el nombre de un dominio de servicio de catálogo. Si define un valor para este parámetro, se utilizan las propiedades de la seguridad de cliente del dominio de servicio de catálogo especificado para probar la conexión. De lo contrario, se produce una búsqueda para encontrar el dominio de servicio de catálogo para el host y el puerto de escucha especificados. Si se encuentra un dominio de servicio de catálogo, se utilizan las propiedades de seguridad de cliente definidas para ese dominio de servicio de catálogo para probar el servidor. De lo contrario no se utilizan propiedades de seguridad de cliente durante la prueba.

Valor de retorno:

Uso de ejemplo de modalidad de proceso por lotes

- Utilizando Jacl:

```
$Admintask testXSServerConnection {-host xhost1.ibm.com -listenerPort 2809}
```

- Utilizando la serie de Jython:

```
AdminTask.testXSServerConnection('[-host xshost3.ibm.com -listenerPort 2809]')
```

Uso de ejemplo de modalidad interactiva

- Utilizando Jacl:

```
$AdminTask testXSServerConnection {-interactive}
```

- Utilizando la serie de Jython:

```
AdminTask.testXSServerConnection ('[-interactive]')
```

Colección del dominio de servicio de catálogo:

Utilice esta página para gestionar los dominios de servicio de catálogo. Los dominios de servicio de catálogo definen un grupo de servidores de catálogo que gestionan la colocación de fragmentos y supervisan el estado de los servidores de contenedor de la cuadrícula de datos.

Para visualizar esta página de la consola administrativa, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo**. Para crear un nuevo dominio de servicio de catálogo, pulse **Nuevo**. Para suprimir un dominio de servicio de catálogo, seleccione el dominio de servicio de catálogo que desea eliminar y pulse **Suprimir**.

Conexión de prueba:

Al pulsar el botón **Conexión de prueba**, se consultan todos los puntos finales de dominio de servicio de catálogo definido uno por uno (si es que hay algún punto final disponible) y el sistema devuelve un mensaje indicando que la conexión con el dominio de servicio de catálogo ha sido satisfactoria. Puede utilizar este botón para probar que ha configurado la conexión y la seguridad de la información correctamente.

Establecer predeterminado:

Define el dominio de servicio de catálogo que se utiliza como el valor predeterminado. Seleccione un dominio de servicio de catálogo como valor predeterminado y pulse **Establecer predeterminado**. Sólo se puede seleccionar un dominio de servicio de catálogo como el valor predeterminado.

Nombre:

Especifica el nombre del dominio de servicio de catálogo.

Valor predeterminado:

Especifica qué dominio de servicio de catálogo de la lista es el valor predeterminado. El dominio de servicio de catálogo predeterminado se indica con el icono siguiente:  .

Valores del dominio de servicio de catálogo:

Utilice esta página para gestionar los valores de un dominio de servicio de catálogo específico. Los dominios de servicio de catálogo definen un grupo de servidores de catálogo que gestionan la colocación de fragmentos y supervisan el estado de los servidores de contenedor de la cuadrícula de datos. Puede definir un dominio de servicio de catálogo que esté en la misma célula que el gestor de despliegue. Puede definir también dominios de servicio de catálogo remotos si la configuración de WebSphere eXtreme Scale está en una célula distinta o si la cuadrícula de datos se compone de procesos Java SE.

Para visualizar esta página de la consola administrativa, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios del servicio de catálogo** > ***nombre_dominio_servicio_catálogo***.

Conexión de prueba:

Al pulsar el botón **Conexión de prueba**, se consultan todos los puntos finales de dominio de servicio de catálogo definido uno por uno (si es que hay algún punto final disponible) y el sistema devuelve un mensaje indicando que la conexión con el dominio de servicio de catálogo ha sido satisfactoria. Puede utilizar este botón para probar que ha configurado la conexión y la seguridad de la información correctamente.

Nombre:

Especifica el nombre del dominio de servicio de catálogo.

Habilite este dominio de servicio de catálogo como valor predeterminado a menos que se especifique explícitamente otro dominio de servicio de catálogo:

Si selecciona este recuadro de selección, el dominio de servicio de catálogo seleccionado pasa a ser el dominio de servicio de catálogo predeterminado de la célula. Cada perfil de servidor de la célula se aumenta con el perfil de WebSphere eXtreme Scale al dominio de servicio de catálogo seleccionado.

Para WebSphere eXtreme Scale, todos los contenedores de eXtreme Scale incorporados en módulos de aplicación Java EE se conectan al dominio predeterminado. Los clientes se pueden conectar al dominio predeterminado

mediante la API ServerFactory.getServerProperties().getCatalogServiceBootstrap() para recuperar los puntos finales de servicio de catálogo que se utilizarán al llamar a la API ObjectGridManager.connect().

Si cambia el dominio predeterminado para que apunte a un conjunto distinto de servidores de catálogo, todos los contenedores y clientes hacen referencia al nuevo dominio una vez que se han reiniciado.

Servidores de catálogo:

Especifica una lista de servidores de catálogo que pertenecen a este dominio de servicio de catálogo.

Pulse **Nuevo** para añadir un servidor de catálogo a la lista. Este servidor de catálogo ya debe existir en la configuración de eXtreme Scale. También puede editar o suprimir un servidor de la lista seleccionando el punto final y a continuación pulsando **Editar** o **Suprimir**. Defina las propiedades siguientes para cada punto final de servidor de catálogo:

Punto final de servidor de catálogo
Especifica el nombre del servidor de aplicaciones existente o servidor remoto en el que se ejecuta el servicio de catálogo. Un dominio de servicio de catálogo no puede contener una combinación de servidores de aplicaciones existentes y puntos finales de servidor remoto.

- **Servidor de aplicaciones existente**: especifica la vía de acceso a un servidor de aplicaciones, agente de nodo o gestor de despliegue en la célula. Un servicio de catálogo se inicia automáticamente en el servidor seleccionado. Seleccione en la lista de servidores de aplicaciones existentes. Todos los servidores de aplicaciones que define en el dominio de servicio de catálogo deben estar en el mismo grupo principal.
- **Servidor remoto**: especifica el nombre de host del servidor de catálogo remoto.

  Para puntos finales remotos de WebSphere eXtreme Scale: especifica el nombre de host del proceso de servidor de catálogo remoto. Debe iniciar los servidores remotos con el script **`startOgServer`** o la API de servidor incorporado.

Puerto de cliente
Especifica el puerto que se utiliza para la comunicación entre los servidores de catálogo en el dominio de servicio de catálogo. Este valor es necesario para servidores de catálogo que se ejecutan en procesos de WebSphere Application Server. Puede establecer el valor en cualquier puerto que no utilice otro proceso.

Puerto de escucha
Especifica el puerto que se utiliza para la comunicación con clientes. Este valor es necesario para los puntos finales remotos y debe coincidir con el valor utilizado cuando se inició el servicio de catálogo. El puerto de escucha lo utilizan los clientes y los contenedores para comunicarse con el servicio de catálogo.

Para los puntos finales remotos de WebSphere eXtreme Scale: define el puerto de escucha de intermediario para solicitudes de objetos (ORB) para que los contenedores y clientes se comuniquen con el servicio de catálogo a través del ORB. Para puntos finales de WebSphere Application Server, el valor de puerto de escucha se hereda de la configuración del puerto BOOTSTRAP_ADDRESS.

Estado

Tabla 22. Estado de punto final de servidor de catálogo

Icono	Definición
	Desconocido
	Iniciado
	Detenido

Propiedades de la seguridad de cliente:

Utilice esta página para configurar la seguridad de cliente de un dominio de servicio de catálogo. Estos valores se aplican a todos los servidores del dominio de servicio de catálogo. Se pueden sustituir estas propiedades especificando un archivo `splicer.properties` con la propiedad personalizada com.ibm.websphere.xs.sessionFilterProps o uniendo el archivo EAR de la aplicación.

Para visualizar esta página de la consola administrativa, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo** > ***nombre_dominio_servicio_catálogo*** > **Propiedades de seguridad de cliente**.

Habilitar la seguridad de cliente:

Especifica que está habilitada la seguridad de cliente para el servidor de catálogo. El archivo de propiedades del servidor que está asociado al servidor de catálogo seleccionado debe tener un valor **`securityEnabled`** coincidente en el archivo de propiedades del servidor. Si estos valores no coinciden, se produce una excepción.

Autenticación de credenciales:

Indica si se fuerza o si se da soporte a la autenticación de credenciales.

Nunca

No se aplica ninguna autenticación de credenciales de cliente.

Necesario

La autenticación de credenciales se fuerza siempre. Si el servidor no da soporte a la autenticación de credenciales, el cliente no puede establecer conexión con el servidor.

Soportado

Se aplica autenticación de credenciales solo si tanto el cliente como el servidor dan soporte a la autenticación de credenciales.

Recuento de reintentos de autenticación:

Especifica el número de veces que se vuelve a intentar la autenticación si la credencial ha caducado.

Si no desea volver a intentar la autenticación, establezca el valor a `0`.

Clase de generador de credenciales:

Indica la clase de implementación com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator, de forma que el cliente recupera la credencial del objeto CredentialGenerator.

Puede elegir entre dos clases de generador de credenciales predefinidas, o bien puede especificar un generador de credenciales personalizado. Si elige un generador de credenciales personalizado, debe indicar el nombre de la clase de generador de credenciales.

- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins. UserPasswordCredentialGenerator
- Generador de credenciales personalizado

Tipo de asunto:

Especifica si utiliza el tipo de asunto de emisor J2EE o J2EE runAs. Debe especificar este valor al elegir el generador de credenciales WSTokenCredentialGenerator.

- **runAs**: el asunto contiene el principal de la identidad de J2EE runas y la credencial J2EE runas.
- **emisor**: el asunto contiene el principal del emisor J2EE y la credencial del emisor J2EE.

ID de usuario:

Especifique un ID de usuario cuando utilice la implementación de generador de credenciales UserPasswordCredentialGenerator.

Contraseña:

Especifique una contraseña cuando utilice la implementación de generador de credenciales UserPasswordCredentialGenerator.

Propiedades del generador de credenciales:

Especifica las propiedades de la clase de implementación CredentialGenerator personalizada. Las propiedades se establecen en el objeto con el método setProperties(String). El valor propiedades del generador de credenciales se utiliza únicamente cuando se especifica un valor en el campo **Clase del generador de credenciales**.

Propiedades personalizadas del dominio de servicio de catálogo:

Puede editar adicionalmente la configuración del dominio de servicio de catálogo definiendo propiedades personalizadas.

Para visualizar esta página de la consola administrativa, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo** > **Propiedades personalizadas**. Para crear una nueva propiedad personalizada, pulse **Nueva**.

Nombre:

Especifica el nombre de la propiedad personalizada del dominio de servicio de catálogo.

Valor:

Especifica un valor para la propiedad personalizada para el dominio de servicio de catálogo.

Configuración de servidores de contenedor en WebSphere Application Server

Configure servidores de contenedor en WebSphere Application Server utilizando un archivo de propiedades del servidor y el archivo XML de política de despliegue que está incorporado en un módulo de aplicación Java EE. Los servidores de contenedor se detienen e inician cuando la aplicación se detiene e inicia.

Antes de empezar

Configure un dominio de servicio de catálogo. Si desea más información, consulte "Creación de dominios de servicio de catálogo en WebSphere Application Server" en la página 257.

Acerca de esta tarea

Para crear servidores de contenedor en WebSphere Application Server, debe incorporar los archivos XML de configuración de WebSphere eXtreme Scale para crear los servidores de contenedor en el módulo de aplicación.

Procedimiento

1. Identifique los servidores de aplicaciones en los que desea desplegar la aplicación Java EE que contiene las definiciones de servidor de contenedor de WebSphere eXtreme Scale. Verifique que los perfiles de servidor de aplicaciones de destino se hayan aumentado con el perfil de WebSphere eXtreme Scale. En un entorno de producción, no coloque los servidores que utiliza para servidores de contenedor junto con los servidores de catálogo. Si desea más información, consulte "Creación y aumento de perfiles para WebSphere eXtreme Scale" en la página 184.
2. Configure un archivo de propiedades del servidor y añada el archivo de propiedades del servidor en la classpath para cada nodo de servidor de aplicaciones de destino. Consulte el apartado Archivo de propiedades de servidor para obtener más información.
3. Añada el archivo XML de descriptor de ObjectGrid y el archivo XML de política de despliegue al módulo de aplicación. Consulte el apartado "Configuración de aplicaciones WebSphere Application Server para el inicio automático de servidores de contenedor" para obtener más información.

Configuración de aplicaciones WebSphere Application Server para el inicio automático de servidores de contenedor:

Los servidores de contenedor de un entorno de WebSphere Application Server se inician automáticamente cuando se inicia un módulo que tiene incluidos los archivos XML de eXtreme Scale.

Antes de empezar

WebSphere Application Server y WebSphere eXtreme Scale deben estar instalados y debe poder acceder a la consola administrativa de WebSphere Application Server.

Acerca de esta tarea

Las aplicaciones Java Platform, Enterprise Edition tienen reglas de cargador de clases complejas que complican considerablemente la carga de clases cuando se utiliza una cuadrícula de datos compartida en un servidor Java EE. Una aplicación Java EE normalmente es un único archivo EAR (Enterprise Archive). El archivo EAR contiene uno o más módulos EJB Enterprise JavaBeans (EJB) o de archivado web (WAR).

WebSphere eXtreme Scale observa con atención que cada módulo se inicie y busca los archivos XML de eXtreme Scale. Si el servicio de catálogo detecta que un módulo se inicia con los archivos XML, el servidor de aplicaciones se registra como una Máquina virtual Java (JVM) de servidor de contenedor. Registrando los servidores de contenedor con el servicio de catálogo, la misma aplicación se puede desplegar en distintas cuadrículas de datos, pero es utilizada como una única cuadrícula de datos por el servicio de catálogo. El servicio de catálogo no se ocupa de células, cuadrículas o cuadrículas dinámicas. Una única cuadrícula de datos puede distribuirse en varias células, si es necesario.

Procedimiento

1. Empaquete el archivo EAR para que tenga varios módulos que incluyan los archivos XML de eXtreme Scale en la carpeta `META-INF`. WebSphere eXtreme Scale detecta la presencia de los archivos `objectGrid.xml` y `objectGridDeployment.xml` en la carpeta `META-INF` de los módulos EJB y WEB cuando se inician. Si solo se encuentra un archivo `objectGrid.xml`, se asume que la JVM es cliente. De lo contrario, se asume que esta JVM actúa como un contenedor para la cuadrícula de datos definida en el archivo `objectGridDeployment.xml`.

   Debe utilizar los nombres correctos para estos archivos XML. Los nombres de archivo son sensibles a las mayúsculas y minúsculas. Si los archivos no existen, el contenedor no se inicia. Puede comprobar el archivo `systemout.log` para obtener mensajes que indican que se han colocado los fragmentos. Un módulo EJB o módulo WAR que utiliza eXtreme Scale debe tener archivos XML de eXtreme Scale en su directorio `META-INF`.

   Los archivos XML de eXtreme Scale incluyen:

   - Un archivo XML de descriptor de ObjectGrid, denominado `objectGrid.xml`. Si desea más información, consulte Archivo XML de descriptor ObjectGrid.
   - Un archivo XML de descriptor de despliegue denominado `objectGridDeployment.xml`. Si desea más información, consulte Archivo XML de descriptor de política de despliegue .
   - (Opcional) Un archivo XML de descriptor de metadatos de entidad, si se utilizan entidades. El nombre de archivo `entity.xml` debe coincidir con el nombre especificado en el archivo `objectGrid.xml`. Si desea más información, consulte Archivo XML de descriptor de metadatos de entidad .

   El tiempo de ejecución detecta estos archivos y, a continuación, contacta con el servicio de catálogo para informarle de que otro contenedor está disponible para alojar fragmentos para ese eXtreme Scale.

Consejo: Si la aplicación tiene entidades y está planificando utilizar un servidor de contenedor, establezca el valor **minSyncReplicas** en `0` en el archivo XML de descriptor de despliegue. De lo contrario, podría aparecer uno de estos mensajes en el archivo `SystemOut.log` debido a que no se puede producir la colocación hasta que otro servidor empieza a cumplir la política minSyncReplica:

```
CWPRJ1005E: Se ha producido un error al resolver la asociación de entidad.
Entity=nombre_entidad, association=nombre_asociación.

CWOBJ3013E: El depósito de EntityMetadata no está disponible. Se ha alcanzado
el umbral del tiempo de espera excedido al intentar registrar la entidad:
nombre_entidad.
```

2. Despliegue e inicie la aplicación,

   El contenedor se inicia automáticamente cuando se inicia el módulo. El servicio de catálogo empieza a colocar primarios y réplicas (fragmentos) de particiones lo antes posible. Esta colocación se produce inmediatamente a menos que se configure el entorno para retardar la colocación. Para obtener más información, consulte "Control de la colocación" en la página 427.

Qué hacer a continuación

Las aplicaciones de la misma célula que los contenedores pueden conectarse a estas cuadrículas de datos utilizando un método ObjectGridManager.connect(null, null) y a continuación llamar al método getObjectGrid(ccc, "nombre de cuadrícula de objetos"). Los métodos connect o getObjectGrid podrían bloquearse hasta que los contenedores hayan colocado los fragmentos, pero este bloqueo es solo un problema cuando se inicia la cuadrícula de datos.

Cargadores de clases

Todos los plug-ins u objetos almacenados en un eXtreme Scale se cargan en un cargador de clases determinado. Dos módulos EJB en el mismo archivo EAR pueden incluir estos objetos. Los objetos son los mismos pero se cargan con distintos cargadores de clases. Si la aplicación A almacena un objeto Person en una correlación que es local al servidor, la aplicación B recibe una `ClassCastException` si intenta leer ese objeto. Esta excepción se produce porque la aplicación B ha cargado el objeto Person en un cargador de clases distinto.

Un método para resolver este problema es hacer que un módulo root contenga los plug-ins y objetos necesarios que están almacenados en el eXtreme Scale. Cada módulo que utiliza eXtreme Scale debe hacer referencia a dicho módulo para sus clases. Otra resolución es colocar estos objetos compartidos en un archivo JAR de programa de utilidad que se encuentra en un cargador de clases común compartido por módulos y aplicaciones. Los objetos también se pueden colocar en las clases WebSphere o en el directorio `lib/ext`. Sin embargo, esta colocación complica el despliegue.

Los módulos EJB en un archivo EAR normalmente comparten el mismo ClassLoader y no se ven afectados por este problema. Cada módulo WAR tiene su propio ClassLoader y se ve afectado por este problema.

Conexión a una cuadrícula de datos de solo cliente

Si la propiedad **catalog.services.cluster** se define en las propiedades de célula, nodo o servidor, cualquier módulo del archivo EAR puede llamar al método ObjectGridManager.connect (ServerFactory.getServerProperties().getCatalogServiceBootstrap(), null, null) para

obtener un ClientClusterContext. El módulo también puede llamar al método ObjectGridManager.getObjectGrid(ccc, "nombre de cuadrícula") para obtener una referencia a la cuadrícula de datos. Si los objetos de aplicación se almacenan en correlaciones, verifique que estos objetos estén presentes en un cargador de clases común.

Los clientes Java o clientes externos a la célula se pueden conectar al puerto IIOP de programa de arranque del servicio de catálogo. En WebSphere Application Server, el gestor de despliegue aloja el servicio de catálogo de forma predeterminada. A continuación, el cliente puede obtener un ClientClusterContext y la cuadrícula de datos con nombre.

Gestor de entidades

Con el gestor de entidades, los tuples se almacenan en las correlaciones en lugar de hacerlo en los objetos de aplicación, lo que genera menos problemas de cargador de clases. Sin embargo, los plug-ins pueden ser un problema. Además, tenga en cuenta que el archivo XML de descriptor de ObjectGrid de sustitución de cliente siempre es necesario al conectar a una cuadrícula de datos que tenga entidades definidas: ObjectGridManager.connect("host:puerto[,host:puerto], null, objectGridOverride) o ObjectGridManager.connect(null, objectGridOverride).

Configuración de IBM eXtremeMemory e IBM eXtremeIO

Configurando eXtremeMemory, puede almacenar objetos en memoria nativa en lugar de hacerlo en el almacenamiento dinámico Java. La configuración de eXtremeMemory habilita eXtremeIO, un nuevo mecanismo de transporte. Si mueve los objetos fuera del almacenamiento dinámico de Java, evitará las pausas de recogida de basura, lo que hará que el rendimiento sea más constante y los tiempos de respuesta sean predecibles.

Antes de empezar

- Linux eXtremeIO y eXtremeMemory están soportados en sistemas Linux x86 de 64 bits que utilizan solo un SDK de 64 bits.
- Debe utilizar conjuntos de correlaciones que tengan todas las correlaciones configuradas con modalidad de copia `COPY_TO_BYTES` o `COPY_TO_BYTES_RAW`. Si las correlaciones del conjunto de correlaciones no utilizan una de estas modalidades de copia, los objetos se almacenan en el almacenamiento dinámico Java y se utiliza el intermediario de solicitud de objetos (ORB).
- No puede utilizar eXtremeIO ni eXtremeMemory en los siguientes escenarios de configuración:
  - Cuando se utilizan servidores de contenedor que se ejecutan en un entorno de WebSphere Application Server.
  - Cuando se utilizan plug-ins de desalojador personalizados.
  - Cuando se utilizan índices compuestos.
  - Cuando se utilizan cargadores de grabación diferida incorporados.
  - Cuando se utiliza la interfaz ReplicationMapListener para crear una implementación de un escucha de sucesos para correlaciones del lado del cliente que están en modalidad de réplica.

Acerca de esta tarea

La JVM se basa en la heurística de uso para recopilar, compactar y ampliar la memoria de proceso. La recogida de basura completa estas operaciones. Sin

embargo, la ejecución de la recogida de basura tiene un coste asociado. El coste de ejecutar la recogida de basura aumenta a medida que aumenta el tamaño del almacenamiento dinámico Java y el número de objetos de la cuadrícula de datos. La JVM proporciona heurísticas diferentes para distintos casos de uso y objetivos: rendimiento óptimo, tiempo de pausa óptimo, tiempo de generación, equilibrado y real. Si no hay heurística, es perfecto. Una heurística individual no puede ajustarse a todas las configuraciones posibles.

WebSphere eXtreme Scale utiliza almacenamiento caché de datos, con correlaciones distribuidas que tienen entradas con un ciclo de vida conocido públicamente. Est ciclo de vida incluye las operaciones siguientes: GET, INSERT, DELETE y UPDATE. Al utilizar estos ciclos de vida de correlación conocidos públicamente, eXtremeMemory y eXtremeIO pueden utilizar la memoria de forma más eficaz que la heurística de uso de JVM.

El diagrama siguiente muestra cómo el uso de eXtremeMemory lleva a tiempos de respuesta relativos más coherentes en el entorno. A medida que los tiempos de respuesta relativos alcanzan los percentiles más altos, las solicitudes que están utilizando eXtremeMemory tienen tiempos de respuesta relativos más bajos. El diagrama muestra los percentiles 95 a 100.

.

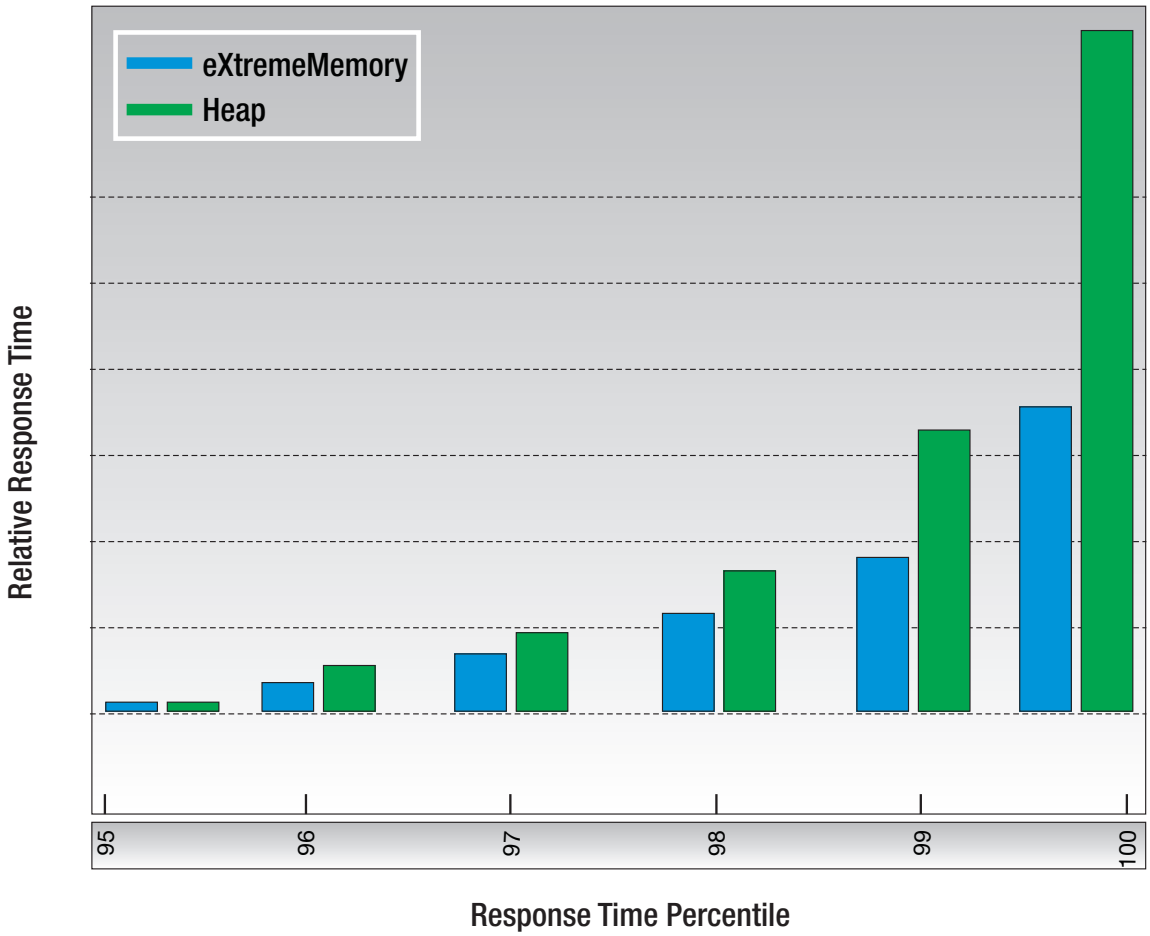


Figura 30. Comparación de tiempos de respuesta de almacenamiento dinámico y de eXtremeMemory

Cuando se utiliza eXtremeMemory, eXtremeIO se utiliza para la comunicación entre servidores de contenedor. Los objetos se serializan en bytes en el servidor de contenedor. Para habilitar eXtremeIO y eXtremeMemory, establece las propiedades de servidor necesarias en todos los servidores de contenedor en la cuadrícula de datos y reinicia los servidores.

Procedimiento

1. Opcional: Determinar el valor de propiedad de **maxXMSize** apropiado a utilizar.
   a. En la configuración existente, determinar el tamaño por entrada. Ejecute el mandato **xscmd -c showMapSizes** para determinar este tamaño.
   b. Calcule el valor de **maxXMSize**. Para obtener el tamaño total máximo de las entradas (*tamaño_total_máximo*), multiplique el *tamaño_por_entrada* * *número_máximo_de_entradas*. No utilice más de 60% de **maxXMSize** para justificar el proceso de metadatos. Multiplique *tamaño_total_máximo* * 1,65 para obtener el valor de **maxXMSize**.
2. Actualice el archivo de propiedades del servidor para cada servidor de contenedor de la configuración para habilitar el nuevo transporte. Las siguientes propiedades de servidor habilitan el nuevo transporte:

   Propiedades necesarias

7.1.1+ **enableXM**
Cuando se establece en `true`, habilita IBM eXtremeMemory en el servidor y configura el servidor para utilizar IBM eXtremeIO para la réplica síncrona y asíncrona. Las entradas de memoria caché se almacenan en la memoria nativa en lugar de hacerlo en el almacenamiento dinámico Java. Todos los servidores de contenedor de la cuadrícula de datos deben utilizar el mismo valor para la propiedad **`enableXM`**.

Valor predeterminado: `false`

Propiedades sugeridas

7.1.1+ **maxXMSize**
Establece la cantidad máxima de memoria, en megabytes, utilizada por el servidor para el almacenamiento eXtremeMemory.

Valor predeterminado: 25% del total de memoria del sistema

Propiedades opcionales

7.1.1+ **maxXIONetworkThreads**
Establece el número máximo de hebras que se asignarán en la agrupación de hebras de red de transporte de eXtremeIO.

Valor predeterminado:`50`

7.1.1+ **minXIONetworkThreads**
Establece el número mínimo de hebras que se asignarán en la agrupación de hebras de red de transporte de eXtremeIO.

Valor predeterminado:`50`

7.1.1+ **maxXIOWorkerThreads**
Establece el número máximo de hebras que se asignarán en la agrupación de hebras de proceso de solicitud de transporte de eXtremeIO.

Valor predeterminado:`128`

7.1.1+ **minXIOWorkerThreads**
Establece el número mínimo de hebras que se asignarán en la agrupación de hebras de proceso de solicitudes de transporte de eXtremeIO.

Valor predeterminado:`128`

7.1.1+ **xioChannel.xioContainerTCPNonSecure.Port**
Especifica el número de puerto de escucha no seguro de eXtremeIO en el servidor. Si no establece el valor, se utiliza un puerto efímero. Esta propiedad sólo se utiliza cuando la propiedad **`transportType`** se establece en `TCP/IP`.

7.1.1+ **xioChannel.xioContainerTCPSecure.Port**
Especifica el número de puerto SSL de eXtremeIO en el servidor. Esta propiedad sólo se utiliza cuando la propiedad **`transportType`** se establece en `SSL-Supported` o `SSL-Required`.

3. Reinicie los servidores de contenedor para empezar a utilizar el nuevo mecanismo de transporte. Si desea más información, consulte “Inicio y detención de los servidores autónomos” en la página 395 y “Inicio y detención de servidores en un entorno de WebSphere Application Server” en la página 409.

Configuración de topologías de varios centros de datos

Con la réplica asíncrona multimaestro, enlaza un conjunto de dominios de servicio de catálogo. A continuación, los dominios de servicio de catálogo conectados se sincronizan mediante réplica a través de los enlaces. Puede definir los enlaces utilizando archivos de propiedades, en tiempo de ejecución con programas JMX (Java Management Extensions) o con programas de utilidad de línea de mandatos. El conjunto de enlaces actuales de un dominio se almacena en el servicio de catálogo. Puede añadir y eliminar enlaces sin reiniciar el dominio de servicio de catálogo que aloja la cuadrícula de datos.

Antes de empezar

- Consulte "Planificación de topologías de varios centros de datos" en la página 36 para obtener más información sobre las topologías de réplica multimaestro y las consideraciones sobre diseño. Puede configurar enlaces entre dominios de servicio de catálogo con el archivo de propiedades del servidor para formar la topología durante el inicio del servidor. También puede configurar enlaces durante la ejecución.
- Si utiliza cargadores en la topología de réplica multimaestro, debe planificar cómo va a mantener datos precisos entre los centros de datos. Los enfoques que puede utilizar varían según la topología que utilice. Para obtener más información, consulte "Consideraciones sobre el cargador en una topología multimaestro" en la página 41.

Procedimiento

- Defina enlaces en el archivo de propiedades del servidor para el servidor de catálogo de cada dominio de servicio de catálogo de la topología, con fines del programa de arranque.

  Consulte Archivo de propiedades de servidor para obtener más información sobre cómo definir este archivo para el servidor de catálogo.

  Importante: Los nombres de propiedad son sensibles a las mayúsculas y minúsculas.

  Nombre de dominio local:
  Especifique el nombre del dominio de servicio de catálogo para el servicio de catálogo actual, por ejemplo, el dominio A:

  ```
  domainName=A
  ```

  Una lista opcional de nombres de dominios foráneos:

  Especifique los nombres de dominios de servicio de catálogo a los que desea enlazar en la topología de réplica multimaestro, por ejemplo, el dominio B:

  ```
  foreignDomains=B
  ```

  Una lista opcional de puntos finales para los nombres de dominio foráneos:
  Especifica la información de conexión de los servidores de catálogo de los dominios foráneos, por ejemplo, el dominio B:

  ```
  B.endPoints=hostB1:2809, hostB2:2809
  ```

  Si un dominio foráneo tiene varios servidores de catálogo, especifique todos ellos.
- Utilice el programa de utilidad **xscmd** o la programación de JMX para añadir o eliminar enlaces en tiempo de ejecución.

Los enlaces de un dominio se conservan en el servicio de catálogo en la memoria replicada. El administrador puede cambiar este conjunto de enlaces cuando desee sin necesidad de reiniciar este dominio o ningún otro dominio. El programa de utilidad **xscmd** incluye varias opciones para trabajar con enlaces.

El programa de utilidad **xscmd** se conecta a un servicio de catálogo y por lo tanto a un único dominio de servicio de catálogo. En consecuencia, se puede utilizar el programa de utilidad **xscmd** para crear y destruir enlaces entre el dominio al que se conecta y otros dominios.

Utilice la línea de mandatos para crear un enlace, por ejemplo:

```
xscmd –c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
```

El mandato establece un nuevo enlace entre el dominio local y el dominio foráneo denominado `dname`. El servicio de catálogo `dname` se ejecuta en `fdHostA:2809` y `fdHostB:2809`. El dominio de servicio de catálogo local tiene un host y un puerto de escucha de servicio de catálogo de `host:2809`. Especifique todos los puntos finales de servicio de catálogo desde el dominio foráneo de forma que sea posible la conectividad al dominio con tolerancia a errores. No utilice un único par `host:puerto` para el servicio de catálogo del dominio de servicio de catálogo foráneo.

Puede utilizar cualquier JVM de servicio de catálogo local con **xscmd** y usando la opción **-cep**. Si el servidor de catálogo se aloja en el gestor de despliegue de WebSphere Application Server, el puerto normalmente es 9809.

Los puertos especificados para el dominio foráneo no son puertos JMX. Suelen ser puertos que utilizaría para los clientes de eXtreme Scale.

Después de que se emite el mandato para añadir un nuevo enlace, el servicio de catálogo indica a todos los contenedores bajo su gestión que comiencen a replicar en el dominio foráneo. No es necesario un enlace en los dos extremos. Solo es necesario crear un enlace en un extremo.

Utilice la línea de mandatos para eliminar un enlace, por ejemplo:

```
xscmd –c dismissLink -cep host:2809 -fd dname
```

El mandato se conecta al servicio de catálogo para un dominio y le indica que deje de replicar en un dominio concreto. Un enlace sólo se tiene que desechar de un extremo.

Enlace entre dos dominios de servicio de catálogo

Supongamos que desea definir una configuración de dos dominios que implica los dominios de servidor de catálogo A y B.

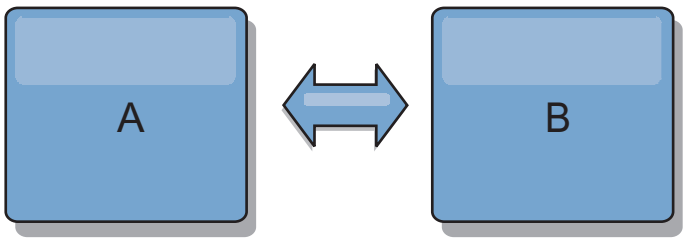


Figura 31. Enlace entre dominios de servicio de catálogo

A continuación se muestra el archivo de propiedades de servidor para el servidor de catálogo en el dominio A:

```
domainName=A
foreignDomains=B
B.endPoints=hostB1:2809, hostB2:2809
```

A continuación se muestra el archivo de propiedades de servidor para el servidor de catálogo en el dominio B. Tenga en cuenta la similitud entre los dos archivos de propiedades.

```
domainName=B
foreignDomains=A
A.endPoints=hostA1:2809,hostA2:2809
```

Una vez que se han iniciado los dos dominios, las cuadrículas de datos que tengan las características siguientes se replicarán entre los dominios.

- Tener un servicio de catálogo privado con un nombre de dominio único
- Tener el mismo nombre de cuadrícula de datos que otras cuadrículas del dominio
- Tener el mismo número de particiones que otras cuadrículas de datos del dominio
- Ser una cuadrícula de datos FIXED_PARTITION (las cuadrículas de datos PER_CONTAINER no se pueden replicar)
- Tener el mismo número de particiones (podrían tener o no tener el mismo número y tipos de réplicas)
- Tener los mismos tipos de datos replicados que otras cuadrículas de datos del dominio
- Tener el mismo nombre de conjunto de correlaciones, los mismos nombres de correlación y las mismas plantillas de correlación dinámica que otras cuadrículas de datos del dominio

La política de réplica de un dominio de servicio de catálogo se ignora.

En el ejemplo anterior se muestra cómo configurar cada dominio para que tenga un enlace con el otro dominio, pero es necesario solo definir un enlace en un sentido. Este hecho es de especial utilidad en las topologías de hub y radio, con lo que se permite una configuración mucho más sencilla. El archivo de propiedades de hub no requiere actualizaciones a medida que se añaden los radios y cada archivo de radio necesita solo incluir la información del hub. De forma similar, una topología de anillo requiere que cada dominio tenga solo un enlace con el dominio anterior y el siguiente en el anillo.

Ejemplo: Topología de hub y radio

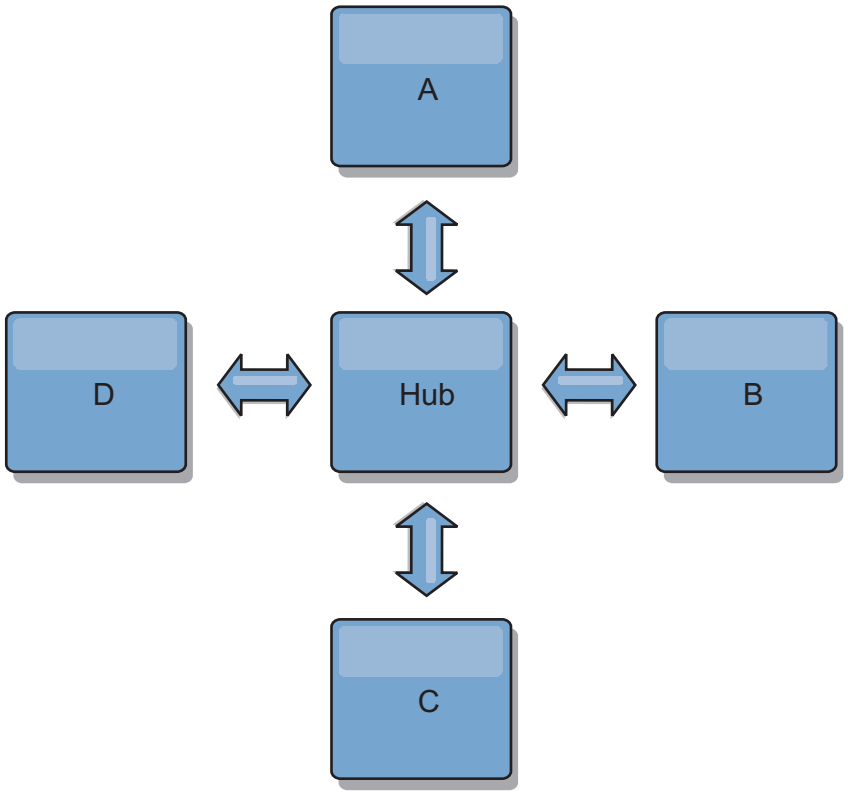


Figura 32. Topología de hub y radio

El hub y cuatro radios (los dominios A, B, C y D) tienen los archivos de propiedades de servidor como en los ejemplos siguientes.

```
domainName=Hub
```

El radio A tiene las siguientes propiedades de servidor:

```
domainName=A
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

El radio B tiene las siguientes propiedades de servidor:

```
domainName=B
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

El radio C tiene las siguientes propiedades de servidor:

```
domainName=C
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

El radio D tiene las siguientes propiedades:

```
domainName=D
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

Qué hacer a continuación

Puede proporcionar un árbitro de colisión personalizado para resolver colisiones entre los dominios de servicio de catálogo. Si desea más información, consulte Desarrollo de árbitros personalizados para la réplica con varios maestros.

Configuración de los puertos

WebSphere eXtreme Scale es una memoria caché distribuida que requiere abrir puertos para comunicarse con el intermediario de solicitud de objetos (ORB) y la pila del protocolo de control de transmisiones (TCP) entre Máquina virtual Java (JVM) y otros servidores.

Configuración de puertos en modalidad autónoma

Puede configurar los puertos necesarios de servidores y clientes en un despliegue de eXtreme Scale mediante parámetros de línea de mandatos, archivos de propiedades o mediante programación. La mayoría de los ejemplos incluidos en las secciones siguientes describen los parámetros de línea de mandatos en el script **startOgServer**. Se pueden establecer también opciones de configuración equivalentes en los archivos de propiedades, mediante la API de servidor incorporado o la API de cliente.

Procedimiento

1. Inicie los puntos finales de servicio de catálogo

   WebSphere eXtreme Scale utiliza IIOP para la comunicación entre máquinas virtuales Java. Las JVM del servicio de catálogo son los únicos procesos que requieren la configuración explícita de puertos para los puertos de servicios IIOP y los servicios de grupo. Otros procesos asignan los puertos dinámicamente.

   El puerto de cliente y el puerto de igual se utilizan para la comunicación entre servicios de catálogo de un dominio de servicio de catálogo. Para especificar el puerto de cliente y el puerto de igual, utilice la siguiente opción de línea de mandatos:

   -catalogServiceEndPoints
   <nombreServidor:nombreHost:puertoCliente:puertoIgual>

   En el contenedor, hace referencia al host y puerto ORB (Object Request Broker - Intermediario de solicitud de objetos) en el servicio de catálogo. Cada atributo se define de la manera siguiente:

   serverName
   Especifica un nombre para identificar el proceso que está iniciando.

   hostName
   Especifica el nombre de host para el sistema donde se inicia el servidor.

   clientPort
   Especifica el puerto que se utiliza para la comunicación de servicio de catálogo de igual.

   peerPort
   Este valor es el mismo que el de haManagerPort. Especifica el puerto que se utiliza para la comunicación de servicio de catálogo de igual.

   El ejemplo siguiente inicia el servidor de catálogo cs1, que está en el dominio de servicio de catálogo que los servidores cs2 y cs3:

   ```
   startOgServer.bat|sh cs1 -catalogServiceEndPoints
   cs1:MyServer1.company.com:6601:6602,
   cs2:MyServer2.company.com:6601:6602,
   cs3:MyServer3.company.com:6601:6602
   ```

Los puntos finales de servicio de catálogo también se pueden establecer con la propiedad de servidor catalogServiceEndPoints. El puerto de escucha del intermediario de solicitud de objetos (ORB) se utiliza para la comunicación entre los servicios de catálogo de un dominio de servicio de catálogo y para la comunicación entre servicios de catálogo y servidores de contenedor y clientes. Para especificar el puerto de escucha y el host de escucha, utilice las siguientes opciones de línea de mandatos:

-listenerHost <nombre de host>

Especifique el nombre de host al que se enlaza el Intermediario para solicitudes de objetos (ORB) para la comunicación con el protocolo Inter-ORB de Internet (IIOP). El valor debe ser un nombre de dominio totalmente calificado o una dirección IP. Si la configuración incluye varias tarjetas de red, establezca el puerto y el host de escucha para permitir que el Intermediario para solicitudes de objetos de la JVM conozca la dirección IP a la que enlazarse. Si no especifica qué dirección IP se debe utilizar, se pueden producir síntomas como tiempos de espera de conexión excedidos, anomalías de API inusuales y clientes que parece que se cuelgan.**Valor predeterminado:** `localhost`

-listenerPort <puerto>

Especifica el número de puerto al que se enlaza el intermediario para solicitudes de objetos (ORB). Este valor configura los contenedores y clientes para que se comuniquen con el servicio de catálogo mediante el ORB. En WebSphere Application Server, el listenerPort se hereda de la configuración de puerto BOOTSTRAP_ADDRESS.Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.**Valor predeterminado:** `2809`

El puerto de escucha y el host de escucha también se pueden establecer utilizando las propiedades de servidor listenerHost y listenerPort.

El puerto de servicio JMX se utiliza para la comunicación desde clientes JMX. Para especificar el puerto de servicio JMX, utilice la siguiente opción de línea de mandatos:

-JMXServicePort <puerto>

Especifica el número de puerto en el que el servidor MBean escucha las comunicaciones con Java Management Extensions (JMX). Debe utilizar un número de puerto distinto para cada JVM de la configuración. Si desea utilizar JMX/RMI, especifique explícitamente **JMXServicePort** y el número de puerto, incluso si desea utilizar el valor de puerto predeterminado. Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.

Valor predeterminado: `1099`

El puerto de servicio JMX también se puede establecer utilizando la propiedad de servidor JMXServicePort. Cuando la seguridad está habilitada, también se requiere un puerto SSL (Secure Socket Layer). Para especificar el puerto SSL, utilice la siguiente opción de línea de mandatos:

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<puertoSSL>
```

```
./startOgServer.sh cs1 -listenerHost hostA -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

Figura 33. Ejemplo utilizando la línea de mandatos. Inicie el primer servidor de catálogo en el hostA. A continuación, se muestra un ejemplo del mandato:

Inicie el segundo servidor de catálogo en el hostB.A continuación, se muestra un ejemplo del mandato:

```
./startOgServer.sh cs2 -listenerHost hostB -listenerPort 2809
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

2. Inicie los puntos finales de servidor de contenedor.

   El mandato siguiente inicia una JVM de contenedor para utilizar con el servicio de catálogo de ejemplo:

   ```
   ./startOgServer.sh c0 -catalogServiceEndPoints hostA:2809,hostB:2809
   ```

   Las Mäquinas virtuales Java de servidor de contenedor utilizan dos puertos. El puerto del gestor de alta disponibilidad se utiliza para la comunicación interna entre los servidores de catálogo y los servidores de contenedor de igual. El puerto de escucha se utiliza para la comunicación IIOP entre servidores de contenedor iguales, servidores de catálogo y clientes. El host de escucha se utiliza para enlazar el intermediario de solicitud de objetos (ORB) a un adaptador de red específico.Si no lo especifica, ambos puertos se seleccionan dinámicamente. Sin embargo, si desea configurar explícitamente puertos, por ejemplo en un entorno de cortafuegos, puede utilizar opciones de línea de mandatos para especificar el puerto de ORB. Para especificar el puerto de escucha y el host de escucha, utilice las siguientes opciones de línea de mandatos:

   ```
   -listenerHost <nombre_host>
   -listenerPort <puerto>
   ```

   El puerto de escucha y el host de escucha se pueden establecer también mediante las propiedades de servidor listenerHost y listenerPort.

   Para especificar el puerto del gestor de alta disponibilidad, utilice la siguiente opción de línea de mandatos:

   -haManagerPort <puerto>
   Sinónimo con puerto de igual. Especifica el número de puerto que utiliza el High Availability Manager. Si esta propiedad no está establecida, el servicio de catálogos genera un puerto disponible de forma automática. Esta propiedad se aplica tanto al servidor de contenedor, como al servicio catálogos. (Sólo necesario para entornos de WebSphere Application Server.)

   El puerto de gestor HA también se puede establecer utilizando la propiedad de servidor HAManagerPort.

   Cuando la seguridad está habilitada, también se requiere un puerto SSL (Secure Socket Layer). Para especificar el puerto SSL, utilice la siguiente opción de línea de mandatos:

   ```
   -jvmArgs -Dcom.ibm.CSI.SSLPort=<puertoSSL>
   ```

3. Inicie los puntos finales de cliente.

   Los clientes sólo deben conocer los puntos finales de escucha de servicio de catálogo. Los clientes recuperan los puntos finales para las Mäquinas virtuales Java de servidor de contenedor, que son las Mäquinas virtuales Java que contienen los datos, automáticamente desde el servicio de catálogo. Para conectarse al servicio de catálogo del ejemplo anterior, el cliente debe pasar la lista siguiente de pares `host:puerto` a la API de conexión:

   ```
   hostA:2809,hostB:2809
   ```

   El cliente también puede recibir devoluciones de llamada de los servidores de contenedor cuando se utiliza la API DataGrid. Estas devoluciones de llamada se comunican utilizando IIOP con el puerto de escucha de intermediario de solicitud de objetos (ORB). Para especificar el puerto y el adaptador de red para recibir devoluciones de llamada, establezca las propiedades **listenerHost** y **listenerPort** en el archivo de propiedades de cliente.

Cuando la seguridad está habilitada, también se requiere un puerto SSL (Secure Socket Layer). Para especificar el puerto SSL, utilice la siguiente propiedad del sistema al iniciar el proceso cliente:

```
-Dcom.ibm.CSI.SSLPort=<puertoSSL>
```

Configuración de puertos en un entorno de WebSphere Application Server

Los servicios de catálogo, servidores de contenedor y clientes de WebSphere eXtreme Scale, cuando se ejecutan en procesos de WebSphere Application Server, utilizan puertos y servicios ya definidos para el proceso.

Acerca de esta tarea

Las secciones siguientes explican detalles relacionados con la utilización de puertos en el despliegue.

1. Puntos finales de servicio de catálogo

   Los servicios de catálogo de WebSphere eXtreme Scale se ejecutan en cualquier proceso de WebSphere Application Server y se configuran mediante la consola administrativa o mediante las tareas administrativas. El proceso hereda todos los puertos, excepto el puerto del cliente, que se configura explícitamente. Para obtener detalles sobre qué puertos utiliza el servicio de catálogo, consulte “Planificación de puertos de red” en la página 64. Para obtener detalles sobre la configuración de un dominio de servicio de catálogo, consulte Servicio de catálogo de alta disponibilidad .

2. Puntos finales de servidor de contenedor

   Los servidores de contenedor de WebSphere eXtreme Scale se alojan en módulos Java EE. Los servidores de contenedor utilizan los puertos definidos para el proceso del servidor de aplicaciones. Para obtener detalles sobre qué puertos utiliza el servicio de contenedor, consulte “Planificación de puertos de red” en la página 64. Para obtener detalles sobre cómo iniciar un contenedor en un módulo Java EE como un módulo Enterprise JavaBeans™ (EJB) o módulo web, consulte “Configuración de aplicaciones WebSphere Application Server para el inicio automático de servidores de contenedor” en la página 274.

3. Puntos finales de cliente

   Los clientes de WebSphere eXtreme Scale se alojan en módulos EJB o web Java EE.

   Los clientes se conectan mediante programación al dominio de servicio de catálogo mediante la API ObjectGridManager.connect(). Al conectarse a un dominio de servicio de catálogo alojado dentro de la misma célula, la conexión de cliente encontrará automáticamente el dominio de servicio de catálogo predeterminado mediante la siguiente llamada a la API del ObjectGridManager:

   ```
   connect(securityProps, overRideObjectGridXML)
   ```

   Si el dominio de servicio de catálogo predeterminado está alojado remotamente (externamente a la célula), se deben especificar los puntos finales de servicio de catálogo mediante el método siguiente en la API de ObjectGridManager:

   ```
   connect(catalogServerAddresses, securityProps, overRideObjectGridXml)
   ```

   Si el dominio de servicio de catálogo predeterminado se define en la célula, se puede utilizar la API de CatalogServerProperties para recuperar las direcciones del servidor de catálogo. La tarea administrativa de XSDomainManagement también se puede utilizar para recuperar puntos finales de dominio de servicio de catálogo configurados.

Servidores con varias tarjetas de red

Puede ejecutar procesos de eXtreme Scale en un servidor que tenga más de una tarjeta de red.

Si un servidor o cliente se ejecuta en un servidor que contiene más de una tarjeta de red, debe especificar el puerto de red y el nombre de host en la configuración de eXtreme Scale para enlazar con una tarjeta de red especificada. Si no se especifica esta configuración, la ejecución de eXtreme Scale elegirá automáticamente una, lo que puede producir anomalías de conexión o un rendimiento más lento.

Para los servidores de catálogo o contenedor, debe establecer el host de escucha y el puerto de escucha de una de las formas siguientes:

- propiedades de servidor
- Parámetro de línea de mandatos en el script startOgServer.sh | bat.

Para los clientes, no puede utilizar la línea de mandatos y debe utilizar propiedades de cliente.

Configuración de transportes

Los transportes habilitan el intercambio de objetos y datos entre distintos procesos de servidor en la configuración.

Acerca de esta tarea

El mecanismo de transporte principal es el intermediario de solicitud de objetos (ORB). Este mecanismo almacena entradas de memoria caché en el almacenamiento dinámico Java.

7.1.1+ Se requiere la utilización del ORB como mecanismo de transporte en los siguientes escenarios de configuración:

- Cuando se utiliza un sistema que no sea Linux de 64 bits x86.
- Cuando se utilizan servidores de contenedor que se ejecutan en un entorno de WebSphere Application Server.
- Cuando se utilizan plug-ins de desalojador o índices compuestos.

7.1.1+ Si está utilizando eXtremeMemory, se utiliza un nuevo transporte denominado eXtremeIO. Con eXtremeMemory, las entradas de memoria caché se almacenan en la memoria nativa. En la memoria nativa se realiza la recogida de basura, lo que produce tiempos de respuestas predecibles y de rendimiento constante. Los objetos se serializan en bytes en el servidor de contenedor. Para obtener más información, consulte “Configuración de IBM eXtremeMemory e IBM eXtremeIO” en la página 277.

Configuración de intermediarios de solicitud de objetos

El intermediario de solicitud de objetos (ORB) lo utiliza WebSphere eXtreme Scale para comunicarse en una pila TCP. Utilice el archivo `orb.properties` para pasar las propiedades utilizadas por el intermediario de solicitud de objetos (ORB) para modificar el comportamiento de transporte de la cuadrícula de datos. No es necesaria ninguna acción para utilizar el intermediario de solicitud de objetos (ORB) proporcionado por WebSphere eXtreme Scale o por WebSphere Application Server para los servidores WebSphere eXtreme Scale.

Configuración del intermediario de solicitud de objetos (ORB) en un entorno de WebSphere Application Server

Puede utilizar WebSphere eXtreme Scale con aplicaciones que utilizan el intermediario de solicitud de objetos (ORB) directamente en entornos de WebSphere Application Server o WebSphere Application Server Network Deployment.

Procedimiento

1. Dé nombre a los servidores de aplicaciones según corresponda.

   No puede tener servidores en un entorno de WebSphere Application Server con el mismo nombre cuando los servidores utilizan el intermediario de solicitud de objetos ORB para comunicarse entre ellos. Puede resolver esta restricción especificando la propiedad del sistema **-Dcom.ibm.websphere.orb.uniqueServerName=true** para los procesos que tengan el mismo nombre. Por ejemplo, cuando se utilizan servidores con el nombre `server1` en cada nodo como un dominio de servicio de catálogo, o donde se utilizan varios agentes de nodo para formar un dominio de servicio de catálogo.
2. Ajuste las propiedades del intermediario de solicitud de objetos (ORB) en la configuración de WebSphere Application Server.

   Consulte "Propiedades ORB" en la página 492 para obtener más información sobre las propiedades que puede ajustar. En función de la propiedad, podría cambiar un valor en la consola administrativa o en el archivo *raíz_was*`properties/orb.properties`.
3. Si está utilizando varias tarjetas de interfaz de red, debe establecer el valor ORB_LISTENER_ADDRESS en el panel de puertos en la consola administrativa de WebSphere Application Server. Repita este paso para cada servidor de aplicaciones de la configuración.
   a. Para un servidor de aplicaciones, pulse **Servidores** > **Servidores de aplicaciones** > *nombre_servidor*. En Comunicaciones, pulse **Puertos**. El panel Puertos se visualiza para el servidor especificado.
   b. Pulse **Detalles** y edite el valor ORB_LISTENER_ADDRESS.
   c. Especifique la dirección IP en el campo **Host**. Este valor debe ser una dirección privada para un entorno de varias interfaces de red.

      Nota: No se da soporte a nombres de host DNS para el valor ORB_LISTENER_ADDRESS.
   d. Especifique el número de puerto en el campo **Puerto**. El número de puerto especifica el puerto para el que el servicio está configurado para aceptar solicitudes de cliente. El valor de puerto se utiliza con el nombre de host.

Qué hacer a continuación

7.1.1+ Puede utilizar la herramienta **wxsLogAnalyzer** para verificar los valores de ORB en todo el entorno. Si desea más información, consulte "Análisis de datos de registro y rastreo" en la página 540.

Configuración del intermediario de solicitud de objetos con procesos de WebSphere eXtreme Scale autónomos

Puede utilizar WebSphere eXtreme Scale con las aplicaciones que utilizan el intermediario de solicitud de objetos (ORB) directamente en los entornos que no contienen WebSphere Application Server o WebSphere Application Server Network Deployment.

Antes de empezar

Si utiliza el ORB dentro del mismo proceso que eXtreme Scale cuando ejecuta aplicaciones, u otros componentes e infraestructuras, que no están incluidos con eXtreme Scale, es posible que tenga que completar tareas adicionales para asegurarse de que eXtreme Scale se ejecuta correctamente en el entorno.

Acerca de esta tarea

Añada la propiedad **ObjectGridInitializer** al archivo `orb.properties` para inicializar el uso del ORB en el entorno. Utilice el ORB para habilitar la comunicación entre los procesos eXtreme Scale y otros procesos que están en el entono.

Procedimiento

1. La instalación autónoma no incluye ningún archivo `orb.properties`. Debe colocar un archivo `orb.properties` en el directorio `java/jre/lib`. Si desea descripciones de las propiedades y los valores, consulte "Propiedades ORB" en la página 492.
2. En el archivo `orb.properties`, especifique la línea siguiente y guarde los cambios:

```
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer
```

Resultados

eXtreme Scale inicializa correctamente el ORB y coexiste con otras aplicaciones para las que el ORB está habilitado.

Para utilizar una versión personalizada del ORB con eXtreme Scale, consulte "Configuración de un intermediario de solicitud de objetos personalizado".

Qué hacer a continuación

7.1.1+ Puede utilizar la herramienta **xsLogAnalyzer** para verificar los valores de ORB en todo el entorno. Si desea más información, consulte "Análisis de datos de registro y rastreo" en la página 540.

Configuración de un intermediario de solicitud de objetos personalizado

WebSphere eXtreme Scale utiliza el intermediario de solicitud de objetos (ORB) para habilitar la comunicación entre procesos. No es necesaria ninguna acción para utilizar el intermediario de solicitud de objetos (ORB) proporcionado por WebSphere eXtreme Scale o por WebSphere Application Server para los servidores WebSphere eXtreme Scale. Es necesario poco esfuerzo para utilizar los mismos ORB para los clientes WebSphere eXtreme Scale. Si en lugar de ello debe utilizar un ORB personalizado, el ORB proporcionado con el IBM SDK es una buena opción, aunque debe configurar el ORB. Se pueden utilizar los ORB de otros proveedores, también con la configuración.

Antes de empezar

Determine si está utilizando el ORB proporcionado con WebSphere eXtreme Scale o WebSphere Application Server, el ORB proporcionado con el IBM SDK o un ORB de proveedor externo.

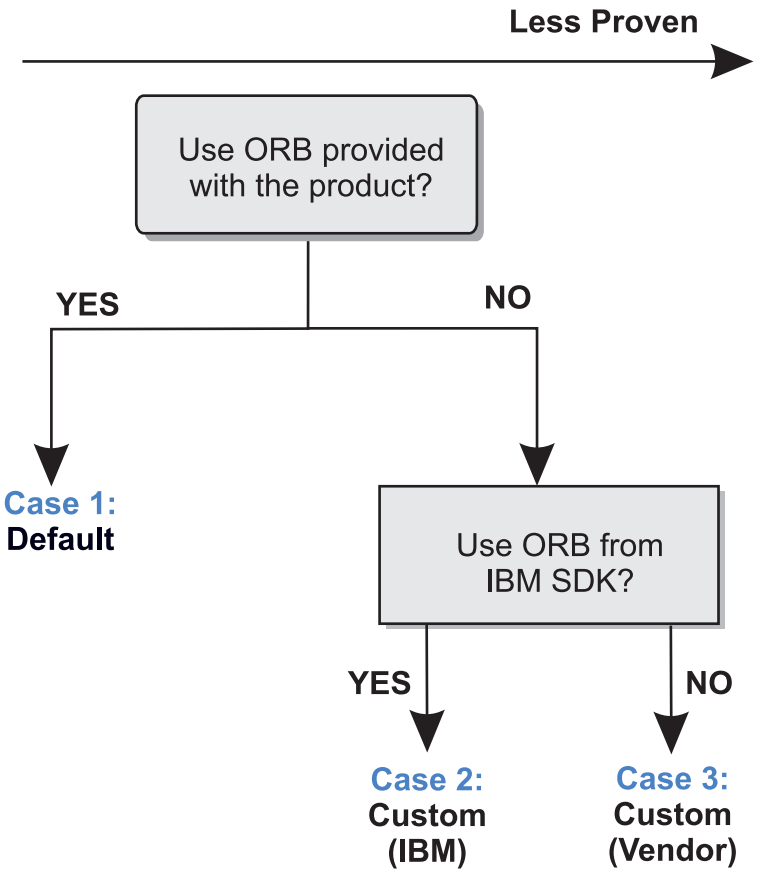


Figura 34. Selección de un ORB

Puede crear decisiones individuales para los procesos servidor WebSphere eXtreme Scale y para los procesos cliente WebSphere eXtreme Scale. Mientras que eXtreme Scale soporta los kits de desarrollador de la mayoría de los proveedores, se recomienda que utilice el ORB proporcionado con eXtreme Scale para los procesos servidor y cliente.eXtreme Scale no soporta el ORB que se suministra con Sun Microsystems Java Development Kit (JDK).

Acerca de esta tarea

Familiarícese con la configuración necesaria para utilizar el ORB seleccionado.

Caso 1: ORB predeterminado

- Para sus procesos servidor WebSphere eXtreme Scale, no es necesaria ninguna configuración para utilizar el ORB proporcionado con WebSphere eXtreme Scale o WebSphere Application Server.
- Para sus procesos cliente WebSphere eXtreme Scale, es necesaria una configuración de vía de acceso de clases mínima para utilizar el ORB proporcionado con WebSphere eXtreme Scale o WebSphere Application Server.

Caso 2: ORB personalizado (IBM)

Para configurar los procesos cliente WebSphere eXtreme Scale para utilizar el ORB proporcionado con IBM SDK, consulte las instrucciones que figuran más adelante en este tema. Puede utilizar IBM ORB si utiliza IBM SDK u otro kit de desarrollo. Puede utilizar IBM SDK Versión 5 o posterior.

Caso 3: ORB personalizado (proporcionado por un proveedor externo)
La utilización de un ORB de proveedor para los procesos de cliente WebSphere eXtreme Scale es la opción menos probada. Cualquier problema que encuentre al utilizar los ORB de proveedores de software independientes debe ser reproducible con el ORB de IBM y compatible con JRE antes de ponerse en contacto con el equipo de soporte.

No se admite el ORB suministrado con Sun Microsystems Java Development Kit (JDK).

Procedimiento

- Configure los procesos de cliente para utilizar uno de los ORB predeterminados (**Caso 1**). Utilice los siguientes argumentos de JVM :

```
-jvmArgs –Djava.endorsed.dirs=directorio_ORB_predeterminado${pathSeparator}JRE_HOME/lib/endorsed
```

  El directorio de ORB predeterminado es: *inicio_wxs*`/lib/endorsed`. Es posible que también sea necesario actualizar las propiedades siguientes en el archivo `orb.properties`:

  ```
  org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
  org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton
  ```

- Configure los procesos cliente o servidor para utilizar IBM SDK, Versión 5 **(Caso 2)**.
  1. Copie los archivos de archivado Java (JAR) del ORB en un directorio vacío, o en el *directorio_ORB_personalizado*.
     - `ibmorb.jar`
     - `ibmorbapi.jar`
  2. Especifique el directorio *directorio_ORB_personalizado* como un directorio "endorsed" en los scripts que inician el mandato Java.

     Consejo: Si los mandatos Java ya especifican un directorio "endorsed", otra opción consiste en poner el directorio *directorio_ORB_personalizado* bajo el directorio "endorsed" existente. Si se pone el directorio *directorio_ORB_personalizado* bajo el directorio "endorsed" existente, no es necesario actualizar los scripts. Si de todas formas decide actualizar los scripts, asegúrese de añadir el directorio *directorio_ORB_personalizado* como prefijo del argumento `–Djava.endorsed.dirs=` existente, en lugar de sustituir completamente el argumento existente.
     - Actualice los scripts para un entorno eXtreme Scale autónomo.

       Edite la vía de acceso de la variable *OBJECTGRID_ENDORSED_DIRS* en el archivo `setupCmdLine.bat|sh` para especificar el *directorio_ORB_personalizado*. Guarde los cambios.
     - Actualice los scripts cuando se incorpora eXtreme Scale en un entorno de WebSphere Application Server.

       Añada los siguientes parámetros y propiedad del sistema al script `startOgServer`:

       ```
       -jvmArgs –Djava.endorsed.dirs=directorio_ORB_personalizado
       ```
     - Actualice los scripts personalizados que utiliza para iniciar un proceso de aplicación cliente o un proceso de servidor.

       ```
       -Djava.endorsed.dirs=directorio_ORB_personalizado
       ```

Configuración de clientes

Puede configurar WebSphere eXtreme Scale para ejecutarse en un entorno autónomo, o puede configurar eXtreme Scale para ejecutarse en un entorno con WebSphere Application Server. Para que un despliegue de WebSphere eXtreme Scale adopte los cambios de configuración en la cuadrícula del servidor, debe reiniciar los procesos para que estos cambios entren en vigor, en lugar de aplicarlos de forma dinámica. Sin embargo, en el cliente, aunque no puede alterar los valores de configuración para una instancia de cliente existente, puede crear una nueva instancia de cliente con los valores que necesite utilizando un archivo XML o mediante programación. Al crear un cliente, puede alterar temporalmente los valores predeterminados que proceden de la configuración de servidor actual.

Puede configurar un cliente eXtreme Scale de las formas siguientes, que se pueden llevar a cabo con un archivo XML de sustitución del cliente o mediante programación:

- Configuración XML
- Configuración mediante programación
- Configuración de la infraestructura Spring
- Inhabilitación de la memoria caché cercana

Puede alterar temporalmente los siguientes plug-ins en un cliente:

- **Plug-ins ObjectGrid**
  - Plug-in TransactionCallback
  - Plug-in ObjectGridEventListener
- **Plug-ins de BackingMap**
  - Plug-in Evictor
  - Plug-in MapEventListener
  - Atributo numberOfBuckets
  - Atributo ttlEvictorType
  - Atributo timeToLive

Configuración de clientes con configuración XML

Puede utilizar un archivo de configuración XML de ObjectGrid para modificar los valores en el lado del cliente.

Acerca de esta tarea

Para modificar los valores de un cliente de WebSphere eXtreme Scale, debe crear un archivo XML de ObjectGrid que sea similar en estructura al archivo que se ha utilizado en el servidor de contenedor.

Puede sustituir los valores siguientes en el cliente:

1. Cree una instancia de ObjectGrid específica del cliente.
2. Copie el archivo XML de ObjectGrid que se utilizó para abrir el servidor.
3. Edite el nuevo archivo para realizar la personalización en el lado del cliente:
   - Para establecer o actualizar uno de los atributos en el cliente, especifique un valor nuevo existente cambiar el valor existente.
   - Para eliminar un plug-in del cliente, utilice la serie vacía como el valor para el atributo className.

- Para cambiar un plug-in existente, especifique un nuevo valor para el atributo className.
- También puede añadir cualquier plug-in soportado para una alteración temporal de cliente: TRANSACTION_CALLBACK, OBJECTGRID_EVENT_LISTENER, EVICTOR, MAP_EVENT_LISTENER.

4. Cree un cliente con el archivo XML de alteración temporal del cliente recién creado.

Procedimiento

1. Cree un archivo XML de configuración de ObjectGrid para el cliente que sea similar en estructura al archivo para el servidor de contenedor.

   Supongamos que el siguiente archivo XML se ha emparejado con un archivo XML de política de despliegue, y que estos archivos se han utilizado para iniciar un servidor de contenedor.

   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>
```

   En un servidor de contenedor, la instancia de ObjectGrid denominada CompanyGrid se comporta tal como se ha definido mediante el archivo `companyGridServerSide.xml`. De forma predeterminada, el cliente de CompanyGrid tiene los mismos valores que los de la instancia CompanyGrid que se ejecuta en el servidor.

   El siguiente archivo XML de ObjectGrid se puede utilizar para especificar algunos de los atributos y plug-ins en el cliente de CompanyGrid.

   `companyGridClientSide.xml`

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

    <objectGrids>
        <objectGrid name="CompanyGrid">
            <bean id="TransactionCallback"
                classname="com.company.MyClientTxCallback" />
            <bean id="ObjectGridEventListener" classname="" />
            <backingMap name="Customer" 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>
```

A continuación se muestra un resumen de las sustituciones que se definen:

- La TransactionCallback en el cliente es com.company.MyClientTxCallback en lugar del valor del lado del servidor de com.company.MyTxCallback.
- El cliente no tiene un ObjectGridEventListener porque el valor de className es la serie vacía.
- El cliente define 1429 como numberOfBuckets para la backingMap del cliente, retiene su plug-in Evictor y elimina el plug-in MapEventListener.
- Los atributos numberOfBuckets y timeToLive de la backingMap OrderLine han cambiado
- Aunque se especifique un atributo de lockStrategy diferente, no hay ningún efecto porque el atributo lockStrategy no se soporta para una alteración temporal de cliente.

2. Cree el cliente utilizando el archivo XML.

   Para crear el cliente CompanyGrid mediante el archivo `companyGridClientSide.xml`, pase el archivo XML de ObjectGrid como URL a uno de los métodos de conexión en la interfaz ObjectGridManager:

```
ObjectGridManager ogManager =
 ObjectGridManagerFactory.ObjectGridManager();
ClientClusterContext clientClusterContext =
 ogManager.connect("MyServer1.company.com:2809", null, new URL(
                "file:xml/companyGridClientSide.xml"));
```

Habilitación del mecanismo de invalidación de clientes

En un entorno de WebSphere eXtreme Scale distribuido, el cliente tiene una memoria caché cercana de manera predeterminada al utilizar la estrategia de bloqueo optimista o al inhabilitar el bloqueo. La memoria caché cercana tiene sus propios datos locales almacenados en memoria caché. Si un cliente de eXtreme Scale confirma una actualización, la actualización se produce en el servidor y la memoria caché cercana del cliente. Sin embargo, otros clientes de eXtreme Scale no reciben la información de actualización y podrían tener datos desfasados.

Memoria caché cercana

Las aplicaciones deben estar al tanto del problema de los datos obsoletos en el cliente de eXtreme Scale. Puede utilizar la clase com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener del ObjectGridEventListener basado en JMS (Java Message Service) incorporado para habilitar el mecanismo de invalidación de cliente en un entorno distribuido deeXtreme Scale.

El mecanismo de invalidación de clientes es la solución al problema de los datos obsoletos de la memoria caché cercana del cliente en el entorno distribuido de eXtreme Scale. Este mecanismo garantiza que la memoria caché cercana del cliente se sincronice con los servidores u otros clientes. Sin embargo, a pesar de este mecanismo de invalidación de clientes basado en JMS, la memoria caché cercana del cliente no se actualiza inmediatamente. Se produce un retardo cuando el tiempo de ejecución de eXtreme Scale publica actualizaciones.

Están disponibles dos modelos para el mecanismo de invalidación de cliente en un entorno de eXtreme Scale distribuido:

- Modelo cliente/servidor: en este modelo, todos los procesos de servidor desempeñan el rol de editor que publica todos los cambios de las transacciones en el destino JMS designado. Todos los procesos de cliente desempeñan los roles de receptor y reciben todos los cambios transaccionales del destino JMS designado.
- Cliente como modelo de roles duales: en este modelo, los procesos de servidor no tienen nada que ver con el destino JMS. Todos los procesos de cliente desempeñan los roles tanto de editor JMS como de receptor. Los cambios transaccionales que se producen en el cliente se publican en el destino JMS y todos los clientes reciben estos cambios transaccionales.

Para obtener más información, consulte la información sobre el "Receptor de sucesos JMS" en la página 232.

Modelo cliente/servidor

En un modelo cliente/servidor, los servidores desempeñan un rol de editor JMS y el cliente desempeña un rol de receptor JMS.

Ejemplo de XML del modelo cliente-servidor

```
<?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>
```

Cliente como modelo de roles duales

En un modelo de cliente como roles duales, cada cliente desempeña los roles tanto de editor JMS como de receptor. El cliente publica cada cambio transaccional confirmado en un destino JMS designado y recibe todos los cambios transaccionales confirmados de otros clientes. En este modelo el servidor no tiene nada que ver con JMS.

Ejemplo de XML de modelo de roles duales

```
<?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>
```

Configuración de los valores de tiempo de espera de reintento de solicitud

Con las correlaciones fiables, puede proporcionar un valor de tiempo de espera de reintento en milisegundos a WebSphere eXtreme Scale para las solicitudes de transacción.

Acerca de esta tarea

Puede configurar el valor de tiempo de espera en el archivo de propiedades de cliente o en una sesión. El valor de sesión sustituye el valor de propiedades de cliente. Si el valor se establece en un valor mayor que cero, la solicitud se intenta hasta que se cumple una condición de tiempo de espera o se produce una anomalía permanente. Una anomalía permanente puede ser una excepción `DuplicateKeyException`. Un valor de cero indica el valor de modalidad fail-fast y eXtreme Scale no intenta de nuevo la transacción después de cualquier tipo de transacción.

Durante la ejecución, el valor de tiempo de espera de transacción se utiliza con el valor de tiempo de espera de reintento, lo que garantiza que el tiempo de espera de reintento no sobrepase el tiempo de espera de transacción.

Existen dos tipos de transacciones: transacciones de confirmación automática y transacciones que utilizan los métodos begin y commit explícitos. Las excepciones válidas de reintento difieren entre los dos tipos de transacciones:

- Para transacciones a las que se llama en una sesión, se intentan de nuevo las transacciones para las excepciones SystemException de CORBA y TargetNotAvailable de eXtreme Scale.
- Para transacciones de confirmación automática, se intentan de nuevo las transacciones para las excepciones de disponibilidad de eXtreme Scale y SystemException de CORBA. Estas excepciones incluyen las excepciones ReplicationVotedToRollbackTransactionException, TargetNotAvailable y AvailabilityException.

Las anomalías de la aplicación y otras anomalías permanentes se devuelven inmediatamente y el cliente no intenta de nuevo la transacción. Estas anomalías permanentes incluyen las excepciones DuplicateKeyException y KeyNotFoundException. Utilice el valor fail-fast para que se devuelvan todas las excepciones sin intentar transacciones de nuevo después de cualquier excepción.

Excepciones donde el cliente intenta de nuevo la transacción:

- ReplicationVotedToRollbackTransactionException (sólo en compromiso automático)
- TargetNotAvailable
- org.omg.CORBA.SystemException
- AvailabilityException (sólo en compromiso automático)
- LockTimeoutException (sólo en compromiso automático)
- UnavailableServiceException (sólo en compromiso automático)

Excepciones permanentes, donde la transacción no se vuelve a intentar:

- DuplicateKeyException
- KeyNotFoundException
- LoaderException
- TransactionAffinityException
- LockDeadlockException
- OptimisticCollisionException

Procedimiento

- Establezca el valor de tiempo de espera en un archivo de propiedades de cliente.

  Para establecer el valor requestRetryTimeout en un cliente, añadir o modificar la propiedad requestRetryTimeout en Archivo de propiedades de cliente . Las propiedades de cliente están en el archivo `objectGridClient.properties` de forma predeterminada. La propiedad requestRetryTimeout se establece en milisegundos. Establézcalo en un valor mayor que cero para la solicitud que se debe reintentar en las excepciones para las que está disponible el reintento. Establezca el valor en `0` para fail sin reintentos en las excepciones. Para utilizar el comportamiento predeterminado, elimine la propiedad o establezca el valor en `-1`. A continuación se muestra un ejemplo del valor del archivo `objectGridClient.properties`:

  ```
  requestRetryTimeout = 30000
  ```

  El valor requestRetryTimeout se especifica en milisegundos. En el ejemplo, si se utiliza el valor en una instancia de ObjectGrid, el valor de requestRetryTimeout es de 30 segundos.

- Establezca el valor de tiempo de espera mediante programación.

  Para establecer las propiedades de cliente a través de un programa, en primer lugar, cree un archivo de propiedades en una `<ubicación>` adecuada para la aplicación. En el siguiente ejemplo, el archivo de propiedades de cliente hace referencia al fragmento de código `objectGridClient.properties` de la sección anterior. Después de conectar a la instancia de ObjectGridManager, establezca las propiedades de cliente tal como se describe. A continuación, cuando tenga una instancia de ObjectGrid, la instancia tendrá las propiedades de cliente que ha definido en el archivo. Si cambia el archivo de propiedades del cliente, deberá obtener explícitamente una nueva instancia de ObjectGrid cada vez.

  ```
  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);
  ```

- Establezca el archivo de sustitución durante la confirmación de la sesión.

  Para establecer el tiempo de espera de reintento de solicitud en una sesión o para alterar temporalmente la propiedad de cliente requestRetryTimeout, llame al método setRequestRetryTimeout(long) en la interfaz Session.

  ```
  Session sessionA = objectGrid.getSession();
  sessionA.setRequestRetryTimeout(30000);
  ObjectMap mapA = sessionA.getMap("payroll");
  String key = "key:" + j;
  mapA.insert(key, "valueA");
  ```

  Ahora esta sesión utiliza un valor requestRetryTimeout de 30000 ms o 30 segundos, independientemente del valor que esté establecido en el archivo de propiedades de cliente. Para obtener más información sobre la interfaz de sesión, consulte Utilización de sesiones para acceder a los datos de la cuadrícula .

Configuración de la integración de la memoria caché

WebSphere eXtreme Scale se puede integrar con otros productos relacionados con la memoria caché. También puede utilizar el proveedor de memoria caché dinámica de WebSphere eXtreme Scale para conectar WebSphere eXtreme Scale en el componente de la memoria caché dinámica en WebSphere Application Server. Otra ampliación para WebSphere Application Server es el gestor de sesiones HTTP de WebSphere eXtreme Scale, que puede ayudar a colocar en la memoria caché las sesiones HTTP.

Configuración de gestores de sesiones HTTP

El gestor de sesiones HTTP proporciona funciones de réplica de sesiones para una aplicación asociada. El gestor de sesiones funciona con el contenedor web para crear y gestionar los ciclos de vida de sesiones HTTP asociadas con la aplicación.

Configuración del gestor de sesiones HTTP con WebSphere Application Server

Mientras que WebSphere Application Server proporciona función de gestión de sesiones, el rendimiento disminuye a medida que el número de solicitudes aumenta. WebSphere eXtreme Scale se entrega empaquetado con una implementación de gestión de sesiones que proporciona réplica de sesiones, mejor escalabilidad y opciones de configuración más potentes.

Antes de empezar

- Se debe instalar WebSphere eXtreme Scale en una célula de WebSphere Application Server o WebSphere Application Server Network Deployment para utilizar el gestor de sesiones de eXtreme Scale. Si desea más información, consulte “Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server” en la página 163.
- La seguridad global debe estar habilitada en la consola administrativa de WebSphere Application Server si los servidores de catálogos de su dominio de servicio de catálogos tienen Secure Sockets Layer (SSL) habilitado o si desea utilizar SSL para un dominio de servicio de catálogo con SSL soportado. SSL es necesario para un servidor de catálogo si se establece el atributo transportType en `SSL-Required`, en Archivo de propiedades de servidor . Para obtener más información sobre cómo configurar la seguridad global, consulte Valores de seguridad globales.

Acerca de esta tarea

El gestor de sesiones HTTP de WebSphere eXtreme Scale soporta ambos servidores, los remotos y los incrustados, para el almacenamiento en la memoria caché.

- **Caso de ejemplo de incrustado**

  En el caso de ejemplo de incrustado, los servidores de WebSphere eXtreme Scale comparten ubicación en los mismos procesos donde se ejecutan los servlets. El gestor de sesiones se puede comunicar directamente con la instancia local de ObjectGrid, lo que evita costosos retardos de red.

  Si utiliza WebSphere Application Server, coloque los archivos *inicio_wxs*`/session/samples/objectGrid.xml` y *inicio_wxs*`/session/samples/objectGridDeployment.xml` proporcionados en los directorios `META-INF` de sus archivos WAR (Web ARchive). eXtreme Scale detecta automáticamente estos archivos cuando se inicia la aplicación e inicia automáticamente los contenedores de eXtreme Scale en el mismo proceso que el gestor de sesiones.

Puede modificar el archivo `objectGridDeployment.xml` en función de si desea utilizar la réplica síncrona o asíncrona y de cuantas réplicas desea configurar.

- **Caso de ejemplo de servidores remotos**

  En el escenario de servidores remotos, los servidores de contenedor se ejecutan en distintos procesos que los servlets. El gestor de sesiones se comunica con un servidor de contenedor remoto. Para utilizar un servidor de contenedor remoto conectado a la red, el gestor de sesiones se debe configurar con los nombres de host y los números de puerto del dominio de servicio de catálogo. El gestor de sesiones utilizará una conexión de cliente de eXtreme Scale para comunicarse con el servidor de catálogo y los servidores de contenedor.

  Si los servidores de contenedor se inician en procesos independientes autónomos, inicie los contenedores de eXtreme Scale con los archivos `objectGridStandAlone.xml` y `objectGridDeploymentStandAlone.xml` proporcionados en el directorio de ejemplos del gestor de sesiones.

Procedimiento

1. Una la aplicación de modo que pueda utilizar el gestor de sesiones. Para utilizar el gestor de sesiones, debe añadir las declaraciones de filtro apropiadas a los descriptores de despliegue web para la aplicación. Además, los parámetros de configuración del gestor de sesiones se pasan al gestor de sesiones en el formato de parámetros de inicialización de contexto de servlet en los descriptores de despliegue. Existen varias formas en las que puede introducir esta información en la aplicación:

   - **Unión automática con WebSphere Application Server**

     Puede configurar la aplicación para utilizar el gestor de sesiones HTTP de WebSphere eXtreme Scale al instalar la aplicación. Puede editar también la configuración de servidor o de la aplicación para utilizar el gestor de sesiones HTTP de WebSphere eXtreme Scale. Si desea más información, consulte “Empalmar automáticamente aplicaciones para la gestión de sesiones HTTP en WebSphere Application Server” en la página 305.

   - **Unión automática de la aplicación con propiedades personalizadas**

     No tiene que unir manualmente las aplicaciones cuando la aplicación se ejecuta en WebSphere Application Server o WebSphere Application Server Network Deployment.

     Añada una propiedad personalizada a una célula o a un servidor para definir el archivo `splicer.properties` para todas las aplicaciones de ese ámbito. Siga los pasos siguientes para configurar la propiedad personalizada:

     a. En la consola administrativa de WebSphere Application Server, navegue a la vía de acceso correcta donde desea definir que la propiedad personalizada para indicar la ubicación del archivo `splicer.properties`.
        - Para establecer la propiedad personalizado para todas las aplicaciones o para una aplicación específica, pulse **Administración del sistema** > **Célula** > **Propiedades personalizadas**.
        - Para establecer la propiedad personalizada para aplicar a todas las aplicaciones en un servidor de aplicaciones específico, pulse **Servidor de aplicaciones** > ***<nombre_servidor>*** > **Administración** > **Propiedades personalizadas**. El nombre de propiedad es `com.ibm.websphere.xs.sessionFilterProps` y su valor es la ubicación del archivo `splicer.properties` que requieren las aplicaciones. A continuación se muestra una vía de acceso de ejemplo de un archivo: `/opt/splicer.properties`.

     b. Añada la propiedad personalizada `com.ibm.websphere.xs.sessionFilterProps`. Este valor de propiedad

personalizada proporciona la ubicación del archivo `splicer.properties` que se va a editar. El archivo existe en el gestor de despliegue. Si desea indicar el archivo `splicer.properties` para una aplicación específica con una propiedad personalizada de nivel de célula, especifique el nombre de la propiedad personalizada como: `<nombre_aplicación>,com.ibm.websphere.xs.sessionFilterProps`, donde *nombre_aplicación* indica el nombre de la aplicación para la que desea aplicar la propiedad personalizada.

Importante: Asegúrese de que el archivo `splicer.properties` actualizado está en la misma vía de acceso en todos los nodos que contienen un servidor de aplicaciones que aloja la aplicación o las aplicaciones que se están uniendo para la réplica de sesión.

Los ámbitos de célula de servidor y de aplicación son los ámbitos disponibles y sólo están disponibles si se trabaja con un gestor de despliegue. Si requiere un ámbito distinto, una manualmente las aplicaciones web.

Recuerde: Observe también que la opción de unión automática funciona únicamente si todos los nodos que ejecutan la aplicación contienen el archivo `splicer.properties` en la misma vía de acceso. Para entornos mixtos que contienen nodos Windows y UNIX, esta opción no es posible, así que debe unir manualmente la aplicación.

- **Puede unir la aplicación con el script `addObjectGridFilter`**

  Utilice un script de línea de mandatos proporcionado junto con eXtreme Scale para unir una aplicación con declaraciones de filtro y configuración con el formato de parámetros de inicialización de contexto de servlet. Para un despliegue de WebSphere Application Server, este script se carga en `<inicio_was>/optionalLibraries/ObjectGrid/session/bin/addObjectGridFilter.bat/sh`. En un despliegue autónomo, este script está en `WXS_HOME/ObjectGrid/session/bin/addObjectGridFilter.sh/bat`. El script **`oddObjectGridFilter`** utiliza dos parámetros:

  – Aplicación: vía de acceso absoluta del archivo de archivado empresarial que se unirá
  – Vía de acceso absoluta del archivo de propiedades de splicer que contiene diversas propiedades de configuración.

  El formato de uso de este script es el siguiente:

  Windows

  ```
  addObjectGridFilter.bat [archivo_ear] [archivo_propiedades_splicer]
  ```

  UNIX

  ```
  addObjectGridFilter.sh [archivo_ear] [archivo_propiedades_splicer]
  ```

  UNIX **Ejemplo utilizando eXtreme Scale instalado en WebSphere Application Server en UNIX:**

  a. `cd inicio_wxs/optionalLibraries/ObjectGrid/session/bin`
  b. `addObjectGridFilter.sh /tmp/mySessionTest.ear raíz_was/optionalLibraries/ObjectGrid/session/samples/splicer.properties`

  UNIX **Ejemplo utilizando eXtreme Scale instalado en un directorio autónomo en UNIX:**

  a. `cd raíz_was/session/bin`
  b. `addObjectGridFilter.sh /tmp/mySessionTest.ear raíz_was/session/samples/splicer.properties`

El filtro de servlet que se une mantiene los valores predeterminados para los valores de configuración. Puede alterar temporalmente estos valores predeterminados con las opciones de configuración que especifique en el archivo de propiedades en el segundo argumento. Para obtener una lista de los parámetros que puede utilizar, consulte "Parámetros de inicialización del contexto del servlet" en la página 321.

Puede modificar y utilizar el archivo `splicer.properties` de ejemplo proporcionado con la instalación de eXtreme Scale. También puede utilizar el script **addObjectGridServlets**, que inserta el gestor de sesiones ampliando cada servlet. No obstante, el script recomendado es el script **addObjectGridFilter**.

- **Una manualmente la aplicación con el script de construcción Ant**

  WebSphere eXtreme Scale se entrega con un archivo `build.xml` que se puede utilizar con Apache Ant, que está incluido en la carpeta *raíz_was*`/bin` de una instalación de WebSphere Application Server. Puede modificar el archivo `build.xml` para cambiar las propiedades de configuración del gestor de sesiones. Las propiedades de configuración son idénticas a los nombres de propiedades del archivo `splicer.properties`. Puede modificar el archivo `build.xml` e invocar el proceso Ant ejecutando el mandato siguiente:

  – UNIX `ant.sh`, `ws_ant.sh`

  – Windows `ant.bat`, `ws_ant.bat`

  (UNIX) o (Windows).

- **Actualice manualmente el descriptor web**

  Edite el archivo `web.xml` que se empaqueta con la aplicación web para incorporar la declaración de filtro, su correlación de servlets y los parámetros de inicialización de contexto de servlet. No utilice este método porque es propenso a errores.

Para obtener una lista de los parámetros que puede utilizar, consulte "Parámetros de inicialización del contexto del servlet" en la página 321.

2. Despliegue la aplicación. Despliegue la aplicación con un conjunto de pasos normales para un servidor o un clúster. Después de desplegar la aplicación, puede iniciarla.
3. Acceda a la aplicación. Ahora puede acceder a la aplicación, que interactúa con el gestor de sesiones y WebSphere eXtreme Scale.

Qué hacer a continuación

Puede cambiar la mayoría de los atributos de configuración para el gestor de sesiones cuando indica a la aplicación que utilice el gestor de sesiones. Estos atributos incluyen: réplica síncrona o asíncrona, tamaño de tabla de sesión en memoria, etc. Aparte de los atributos que pueden cambiarse durante la instrumentación de la aplicación, los otros únicos atributos de configuración que puede cambiar después del despliegue de la aplicación son los atributos relacionados con la topología de clúster de servidores de WebSphere eXtreme Scale y la forma en que sus clientes (gestores de sesiones) se conectan a los mismos.

Comportamiento de escenario remoto: si no se puede acceder a toda la cuadrícula de datos que aloja los datos de sesión de la aplicación desde el cliente de contenedor web, en su lugar el cliente utiliza el contenedor web base en WebSphere Application Server para la gestión de sesiones. Es posible que no se pueda acceder a la cuadrícula de datos en los escenarios siguientes:

- Problema de red entre el contenedor web y los servidores de contenedor remotos.
- Los procesos del servidor de contenedor remoto se han detenido.

El número de referencias de sesión que se mantiene en la memoria, especificado por el parámetro **sessionTableSize**, se sigue manteniendo cuando las sesiones se almacenan en el contenedor web base. Las sesiones utilizadas menos recientemente se invalidan desde la memoria caché de sesión de contenedor web cuando se sobrepasa el valor **sessionTableSize**. Si la cuadrícula de datos remota pasa a estar disponible, las sesiones que se han invalidado desde la memoria caché de contenedor web pueden recopilar datos de la cuadrícula de datos remota y cargar los datos en una nueva sesión. Si toda la cuadrícula de datos remota no está disponible y la sesión se invalida desde la memoria caché de sesión, los datos de sesión de usuario se pierden. Debido a este problema, no concluya toda la cuadrícula de datos remota de producción cuando el sistema se ejecute con carga.

Empalmar automáticamente aplicaciones para la gestión de sesiones HTTP en WebSphere Application Server:

Puede configurar la aplicación WebSphere Application Server para que persista sesiones de una cuadrícula de datos. Esta cuadrícula de datos puede estar en un servidor contenedor que se ejecute en WebSphere Application Server o puede estar en una cuadrícula de datos remota.

Antes de empezar

Antes de cambiar la configuración en WebSphere Application Server, debe tener:

- El nombre de la cuadrícula de datos de sesión que desea utilizar. Consulte "Configuración del gestor de sesiones HTTP con WebSphere Application Server" en la página 301 para obtener información sobre cómo crear una cuadrícula de datos de sesión.
- Si el servicio de catálogo que desea utilizar para gestionar las sesiones está fuera de la célula en la que está instalando la aplicación de sesión, debe crear un dominio de servicio de catálogo. Consulte "Creación de dominios de servicio de catálogo en WebSphere Application Server" en la página 257 para obtener más información.
- Si configura el dominio de un servicio de catálogo, puede que tenga que habilitar la seguridad de cliente en el dominio de servicio de catálogo si los servidores de contenedor requieren autenticación. Estos valores informan al tiempo de ejecución sobre qué implementación CredentialGenerator se debe utilizar. Esta implementación genera una credencial que se pasará a la cuadrícula de datos remota. Consulte "Configuración de la seguridad de cliente en un dominio de servicio de catálogo" en la página 528 para obtener más información sobre cómo configurar estos valores.
- Seguridad global habilitada en la consola administrativa de WebSphere Application Server si desea soportar uno de los escenarios siguientes:
  - Los servidores de catálogo dentro de su dominio de servicio de catálogo tienen la capa de sockets seguros (SSL) habilitada.
  - Desea utilizar SSL para un dominio de servicio de catálogo con SSL soportado.

  Puede requerir que se utilice SSL para un servidor de catálogo definiendo el atributo **transportType** a `SSL-Required` en Archivo de propiedades de servidor . Para obtener más información sobre cómo configurar la seguridad global, consulte Valores de seguridad globales.

- Si utiliza la Versión 7.1.0.3 o posterior, puede persistir las sesiones que utilizan reescritura de URL o cookies como un mecanismo de seguimiento de sesiones a la cuadrícula de datos. Para releases anteriores a la Versión 7.1.0.3, no puede persistir las sesiones que utilizan reescritura de URL como mecanismo de seguimiento de sesiones. Para habilitar la persistencia de sesiones que utilizan reescritura de URL, establezca la propiedad **useURLEncoding** en `true` en el archivo `splicer.properties` después de unir automáticamente la aplicación.
- 7.1.1+ Cuando se unen automáticamente aplicaciones para la gestión de sesiones HTTP en WebSphere Application Server, todos los servidores de aplicaciones que alojan la aplicación web tienen la propiedad personalizada de contenedor web **HttpSessionIdReuse** establecida en `true`. Esta propiedad permite que las sesiones que han migrado de un servidor de aplicaciones a otro tras un error o que se han invalidado desde la memoria caché de sesión en memoria en un escenario remoto conserven el ID de sesión entre solicitudes. Si no desea este comportamiento, establezca la propiedad personalizada de contenedor web en `false` en todos los servidores de aplicaciones aplicables antes de configurar la gestión de sesiones para las aplicaciones. Para obtener más información sobre esta propiedad personalizada, consulte "Resolución de problemas de la integración de la memoria caché" en la página 545.

Procedimiento

- **Para configurar la gestión de sesiones al instalar la aplicación, complete los pasos siguientes:**
  1. En la consola de administración de WebSphere Application Server, pulse **Aplicaciones** > **Aplicación nueva** > **Nueva aplicación empresarial**. Elija la vía de acceso **Detallada** para crear la aplicación y complete los pasos iniciales del asistente.
  2. En el paso **Valores de gestión de sesiones de eXtreme Scale** del asistente, configure la cuadrícula de datos que desea utilizar. Elija **Cuadrícula de datos de eXtreme Scale remota** o **Cuadrícula de datos de eXtreme Scale incorporada**.
     - Para la opción **Cuadrícula de datos remota de eXtreme Scale**, elija el dominio de servicio de catálogo que gestiona la cuadrícula de datos de sesión y elija una cuadrícula de datos en la lista de cuadrículas de datos de sesión activas.
     - Para la opción **Cuadrícula de datos incorporada de eXtreme Scale**, elija la configuración ObjectGrid predeterminada o especifique la ubicación específica de los archivos de configuración de ObjectGrid.
  3. Complete los pasos del asistente para finalizar la instalación de la aplicación.

  También puede instalar la aplicación con un script wsadmin. En el siguiente ejemplo, el parámetro **-SessionManagement** crea la misma configuración que la realizada en la consola de administración:

  Para la configuración de la cuadrícula de datos de eXtreme Scale remota:

```
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]]]')
```

Para el escenario incorporado de eXtreme Scale con la configuración predeterminada:

```
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]]]')
```

Para el escenario incorporado de eXtreme Scale con una configuración personalizada:

```
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]]]')
```

- **Para configurar la gestión de sesiones en una aplicación existente de la consola administrativa de WebSphere Application Server:**
  1. En la consola de administración de WebSphere Application Server, pulse **Aplicaciones** > **Tipos de aplicación** > **Aplicaciones empresariales de WebSphere** > *nombre_aplicación* > **Propiedades de módulo web** > **Gestión de sesiones** > **Valores de gestión de sesiones de eXtreme Scale**.
  2. Actualice los campos para habilitar la persistencia de sesión en una cuadrícula de datos.

  También puede actualizar la aplicación con un script wsadmin. En el siguiente ejemplo, el parámetro **-SessionManagement** crea la misma configuración que la realizada en la consola de administración:

  Para la configuración de cuadrícula de datos remota de eXtreme Scale:

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSRemoteSessionManagement cs0:!:grid0]]]')
  ```

  Los caracteres **:!:** que se pasan se utilizan como delimitadores. Los valores que se pasan son:

  nombre_servicio_catálogo:!:*nombre_cuadrícula*

  Para el escenario incorporado de eXtreme Scale con la configuración predeterminada:

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSEmbeddedSessionManagement :!:!:!:default]]]')
  ```

  Los caracteres **:!:** que se pasan se utilizan como delimitadores. Los valores que se pasan son:

  nombre_servicio_catálogo:!:*nombre_cuadrícula*:!:default:!:
  vía_acceso_absoluta_a_archivo_Xml_cuadrícula_objeto:!:*vía_acceso_absoluta_a_archivo_Xml_despliegue*

  Para el escenario incorporado de eXtreme Scale con una configuración personalizada:

  ```
  AdminApp.edit('DefaultApplication','[-SessionManagement[[true
  XSEmbeddedSessionManagement
  :!:!:!:custom:!:c:\XS\objectgrid.xml:!:c:\XS\objectgriddeployment.xml]]]')
  ```

Los caracteres **:!:** que se pasan se utilizan como delimitadores. Los valores que se pasan son:

nombre_servicio_catálogo:!:*nombre_cuadrícula*:!:custom:!:
vía_acceso_absoluta_a_archivo_Xml_cuadrícula_objeto:!:*vía_acceso_absoluta_a_archivo_Xml_despliegue*

Al guardar los cambios, la aplicación utiliza la cuadrícula de datos configurada para la persistencia de sesión en el dispositivo.

- **Para configurar la gestión de sesiones en un servidor existente:**
  1. En la consola de administración de WebSphere Application Server, pulse **Servidores** > **Tipos de servidor** > **servidores de aplicación de WebSphere** > *nombre_servidor* > **Gestión de sesiones** > **Valores de gestión de sesiones de eXtreme Scale**.
  2. Actualice los campos para habilitar la persistencia de sesión.

  También puede configurar la gestión de sesiones en un servidor existente con los siguientes mandatos de la herramienta wsadmin:

  Para la configuración de la cuadrícula de datos de eXtreme Scale remota:

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSRemoteSessionManagement -XSRemoteSessionManagement
[-catalogService cs0 -csGridName grid0]]')
```

  Para la configuración incorporada de eXtreme Scale:

  – La configuración predeterminada, si utiliza los archivos XML predeterminados:

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement [-embeddedGridType default -objectGridXML -objectGridDeploymentXML ]]')
```

  – La configuración personalizada, si utiliza los archivos XML personalizados:

```
AdminTask.configureServerSessionManagement('[-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement
[-embeddedGridType custom -objectGridXML c:\XS\objectgrid.xml -objectGridDeploymentXML
c:\XS\objectgriddeployment.xml]]')
```

  Cuando se guardan los cambios, el servidor utiliza ahora la cuadrícula de datos configurada para la persistencia de sesión con las aplicaciones que se ejecutan en el servidor.

- Si desea editar otros aspectos de la configuración de sesión HTTP, puede editar el archivo `splicer.properties`. Puede obtener la ubicación de vía de acceso del archivo `splicer.properties` localizando la propiedad personalizada **`sessionFilterProps`**. Si ha configurado la persistencia de sesiones en el nivel de servidor, el nombre de la propiedad personalizada es: `com.ibm.websphere.xs.sessionFilterProps`. Si ha configurado la persistencia de sesión en el nivel de aplicación, el nombre de la propiedad personalizada es: *<nombre_aplicación>*`,com.ibm.websphere.xs.sessionFilterProps`. Estas propiedades personalizadas pueden estar en una de las siguientes ubicaciones:
  – En un entorno WebSphere Application Server Network Deployment: Cambie el archivo `splicer.properties` en la vía de acceso de perfil de gestor de despliegue.
  – En un entorno WebSphere Application Server autónomo: una propiedad personalizada en el servidor de aplicaciones

  Puede abrir el archivo indicado, hacer cambios y sincronizar los nodos de modo que el archivo de propiedades actualizado se propague a los otros nodos de la configuración. Todos los nodos del servidor de aplicaciones requieren que el archivo `splicer.properties` esté en la vía de acceso especificada para persistir las sesiones adecuadamente.

Atención: Si desea habilitar la persistencia para sesiones que utilizan la reescritura de URL, establezca la propiedad **useURLEncoding** en `true` en el archivo `splicer.properties`.

Para obtener más información sobre las propiedades del archivo `splicer.properties`, consulte "Archivo `splicer.properties`" en la página 324.

Resultados

Ha configurado el gestor de sesiones HTTP para que las sesiones persistan en una cuadrícula de datos. Las entradas se eliminan de la cuadrícula de datos cuando las sesiones caducan. Consulte los Valores de gestión de sesión para obtener más información sobre cómo actualizar el valor de tiempo de espera de sesión en la consola administrativa de WebSphere Application Server.

Valores de gestión de sesiones de eXtreme Scale:

Puede configurar las aplicaciones de WebSphere Application Server para utilizar WebSphere eXtreme Scale o un WebSphere DataPower XC10 Appliance para persistencia de sesión.

Puede editar estos valores en el asistente de instalación empresarial o en las páginas de detalle de la aplicación o del servidor:

- Versión 6.1: **Aplicaciones** > **Instalar nueva aplicación**
- Versión 6.1: **Aplicaciones** > **Aplicaciones empresariales** > *nombre_aplicación*
- Versión 6.1: **Servidores** > **Servidores de aplicaciones** > *nombre_servidor* > **Valores de contenedor web** > **Gestión de sesiones**
- Versión 7.0: **Aplicaciones** > **Nueva aplicación** > **Nueva aplicación empresarial** y elija la vía de acceso detallada para crear la aplicación.
- Versión 7.0: **Aplicaciones** > **Tipos de aplicación** > **Aplicaciones empresariales WebSphere** > *nombre_aplicación* > **Propiedades de módulo web** > **Gestión de sesiones** > **Valores de gestión de sesiones**
- Versión 7.0: **Servidores** > **Tipos de servidores** > **Servidores de aplicaciones WebSphere** > *nombre_servidor* > **Valores de contenedor** > **Valores de gestión de sesiones**

Habilite la gestión de sesiones:

Habilita la gestión de sesiones para utilizar cuadrículas de datos incorporadas o remotas de WebSphere eXtreme Scale o un WebSphere DataPower XC10 Appliance para persistencia de sesiones.

Gestionar persistencia de sesión por:

Especifica cómo se gestiona la persistencia de sesión. Puede elegir una de las opciones siguientes:

- WebSphere DataPower XC10 Appliance
- Cuadrícula de datos remota de eXtreme Scale
- Cuadrícula de datos incorporada de eXtreme Scale

Los demás valores que configure dependerán del mecanismo de persistencia de sesión que elija.

Valores específicos de WebSphere DataPower XC10 Appliance:

Los valores siguientes son específicos de la configuración de WebSphere DataPower XC10 Appliance para la persistencia de sesión.

IP o nombre de host de WebSphere DataPower XC10 Appliance:

Especifica la IP o el nombre de host del dispositivo que se utilizará para la persistencia de sesión.

Credenciales administrativas de IBM WebSphere DataPower XC10 Appliance:

Especifica el **Nombre de usuario** y la **Contraseña** que utiliza para iniciar la sesión en la interfaz de usuario de DataPower XC10 Appliance. Pulse **Probar conexión...** para probar la conexión al dispositivo.

Preferencia de persistencia de sesión:

Especifica la cuadrícula de datos en la que persisten las sesiones. Puede elegir una de las opciones siguientes:

- **Persistir sesiones en una nueva cuadrícula de datos en IBM WebSphere DataPower XC10 Appliance**. A continuación, puede especificar un **Nombre de cuadrícula de datos**.
- **Persistir sesiones en una cuadrícula de datos existente en el IBM WebSphere DataPower XC10 Appliance**. A continuación, puede especificar un **Nombre de cuadrícula de datos existente** o ir hasta uno.

Configuración de cuadrícula de datos remota de eXtreme Scale:

Los valores siguientes son específicos de la configuración de la cuadrícula remota de eXtreme Scale para la persistencia de sesión.

Dominio de servicio de catálogo que gestiona la cuadrícula de datos de sesión remota:

Especifica el dominio de servicio de catálogo que desea utilizar para gestionar las sesiones.

Si no se visualiza ningún dominio de servicio de catálogo, o si desea crear un nuevo dominio de servicio de catálogo, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo**.

Cuadrícula de datos remota en la que almacenar información de sesión:

Especifica el nombre de la cuadrícula de datos del dominio de servicio de catálogo en el que desea almacenar la información de sesión. La lista de cuadrículas remotas activas se llena al seleccionar un servicio de catálogo. La cuadrícula de datos remota ya debe existir en la configuración de eXtreme Scale.

Configuración de cuadrícula de datos incorporada de eXtreme Scale:

Los valores siguientes son específicos de la definición de una configuración incorporada de eXtreme Scale. En el escenario de eXtreme Scale incorporado, los procesos de eXtreme Scale los alojan los procesos de WebSphere Application Server.

Configuración de cuadrícula de datos incorporada de Xtreme Scale:

- **Utilice la configuración predeterminada de ObjectGrid**
- **Especifique los archivos de configuración de ObjectGrid personalizados**

Vía de acceso completa del archivo `objectgrid.xml` que se debe copiar en la configuración

Especifica la vía de acceso completa del archivo `objectgrid.xml` para la configuración que desea utilizar.

Vía de acceso completa del archivo `objectgriddeployment.xml` que se debe copiar en la configuración

Especifica la vía de acceso completa del archivo `objectgriddeployment.xml` para la configuración que desea utilizar.

Uso de WebSphere eXtreme Scale para la gestión de sesiones SIP

Puede utilizar WebSphere eXtreme Scale como un mecanismo de réplica SIP (Session Initiation Protocol) como alternativa fiable para el servicio de duplicación de datos (DRS) para la réplica de sesiones SIP.

Configuración de la gestión de sesiones SIP

Para utilizar WebSphere eXtreme Scale como el mecanismo de réplica SIP, establezca la propiedad personalizada com.ibm.sip.ha.replicator.type. En la consola de administración, seleccione **Servidores de aplicaciones** > ***mi_servidor_aplicaciones*** > **Contenedor SIP** > **Propiedades personalizadas** para cada servidor para añadir la propiedad personalizada. Escriba `com.ibm.sip.ha.replicator.type` como nombre (Name) y `OBJECTGRID` como valor (Value).

Utilice las propiedades siguientes para personalizar el comportamiento del objeto ObjectGrid que se usa para almacenar sesiones SIP. En la consola de administración, pulse **Servidores de aplicaciones** > ***mi_servidor_aplicaciones*** > **Contenedor SIP** > **Propiedades personalizadas** para cada servidor para añadir la propiedad personalizada. Escriba el **Nombre** y el **Valor**. Cada servidor debe tener el mismo conjunto de propiedades para funcionar correctamente.

Tabla 23. Propiedades personalizadas para la gestión de sesiones SIP con ObjectGrid

Propiedad	Valor	Valor predeterminado
com.ibm.sip.ha.replicator.type	`OBJECTGRID`: utilizar ObjectGrid como un almacén de sesiones SIP	
min.synchronous.replicas	Número mínimo de réplicas síncronas	0
max.synchronous.replicas	Número máximo de réplicas síncronas	0
max.asynchronous.replicas	Número máximo de réplicas asíncronas	1
auto.replace.lost.shards	Si desea más información, consulte “Configuración de despliegues distribuidos” en la página 236.	true
development.mode	• `true` - permite que las réplicas estén activas en el mismo nodo que los primarios • `false` - las réplicas deben estar en nodos distintos que los primarios	false

Configuración del gestor de sesiones HTTP con WebSphere Portal

Puede hacer persistir sesiones HTTP de WebSphere Portal insertándolas en una cuadrícula de datos.

Antes de empezar

El entorno de WebSphere eXtreme Scale y WebSphere Portal debe cumplir los siguientes requisitos:

- La forma de instalar WebSphere eXtreme Scale dependerá del escenario de despliegue. Puede ejecutar los servidores de contenedores, que alojan las cuadrículas de datos, sea dentro o fuera de la célula de WebSphere Application Server:
  - Si ejecuta servidores de contenedores en la célula de WebSphere Application Server **(escenario incorporado)**: Instale tanto el cliente como el servidor de WebSphere eXtreme Scale en los nodos de WebSphere Application Server y WebSphere Portal.
  - Si ejecuta servidores de contenedores fuera de la célula de WebSphere Application Server**(escenario remoto)**: Instale WebSphere eXtreme Scale Client en los nodos de WebSphere Application Server y WebSphere Portal.

  Si desea más información, consulte "Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server" en la página 163.
- WebSphere Portal Versión 7 o posterior.
- Se deben configurar portlets personalizados dentro de WebSphere Portal. Los portlets administrativos que se suministran con WebSphere Portal actualmente no se pueden integrar en cuadrículas de datos.

Acerca de esta tarea

La introducción de WebSphere eXtreme Scale en un entorno de WebSphere Portal puede ser beneficioso en los siguientes escenarios:

Importante: Aunque en los siguientes escenarios se presentan ventajas, el resultado de introducir WebSphere eXtreme Scale en el entorno también puede ser un aumento de uso del procesador a nivel de WebSphere Portal.

- **Cuando se requiere persistencia de sesión.**

  Por ejemplo, si los datos de sesión de los portlets personalizados deben permanecer disponibles durante un fallo de WebSphere Portal Server, puede hacer persistir las sesiones HTTP en la cuadrícula de datos de WebSphere eXtreme Scale. Los datos se replican en muchos servidores, lo que aumenta la disponibilidad de los datos.
- **En una topología de diversos centros de datos.**

  Si su topología distribuye diversos centros de datos en distintas ubicaciones físicas, puede hacer persistir las sesiones HTTP de WebSphere Portal en la Cuadrícula de datos de WebSphere eXtreme Scale. Las sesiones se replican en las cuadrículas de datos de los distintos centros de datos. Si un centro de datos falla, las sesiones se pasan a otro centro de datos que tenga una copia de los datos de la cuadrícula de datos.
- **Para reducir los requisitos de memoria en el nivel de WebSphere Portal Server.**

  Al descargar los datos de sesión a un nivel remoto de servidores de contenedores, un subconjunto de las sesiones están en los servidores de WebSphere Portal. Esta descarga de datos reduce los requisitos de memoria en el nivel del WebSphere Portal Server.

Procedimiento

1. Una la aplicación `wps` WebSphere Portal y los portlets personalizados para permitir el almacenamiento de las sesiones en la cuadrícula de datos.

   Puede unir la aplicación configurando la gestión de sesiones HTTP al desplegar la aplicación, o puede utilizar propiedades personalizadas para empalmar automáticamente sus aplicaciones. Consulte el apartado "Configuración del gestor de sesiones HTTP con WebSphere Application Server" en la página 301 para obtener más información sobre cómo empalmar la aplicación.
2. Si utiliza el escenario remoto, en el que los servidores de contenedores están fuera de WebSphere Application Server, inicie explícitamente los contenedores remotos de eXtreme Scale para los escenarios de persistencia remota de sesión HTTP. Inicie los contenedores con los archivos de configuración `XS/ObjectGrid/session/samples/objectGridStandAlone.xml` y `objectGridDeploymentStandAlone.xml`. Por ejemplo, puede utilizar el mandato siguiente:

   ```
   startOgServer.sh xsContainer1 -catalogServiceEndPoints <host>:<puerto>
   -objectgridFile XS/ObjectGrid/session/samples/objectGridStandAlone.xml -deploymentPolicyFile
   XS/ObjectGrid/session/samples/objectGridDeploymentStandAlone.xml
   ```

   Para obtener más información sobre cómo iniciar servidores de contenedores, consulte "Inicio de servidores de contenedor" en la página 398. Si utiliza un escenario incluido, consulte "Configuración de servidores de contenedor en WebSphere Application Server" en la página 274 para obtener más información sobre cómo configurar e iniciar servidores de contenedores.
3. Reinicie los servidores de WebSphere Portal. Consulte WebSphere Portal versión 7: Inicio y detención de servidores, gestores de despliegue y agentes de nodo para obtener más información.

Resultados

Puede acceder a WebSphere Portal Server y los datos de sesión HTTP de los portlets personalizados configurados persisten en la cuadrícula de datos.

Si no se puede acceder desde el cliente de contenedor web a toda la cuadrícula de datos que aloja los datos de sesión de la aplicación, el cliente utiliza en su lugar el contenedor web base de WebSphere Application Server para la gestión de sesiones. Es posible que no se pueda acceder a la cuadrícula de datos en los escenarios siguientes:

- Problema de red entre el contenedor web y los servidores de contenedor remotos.
- Los procesos del servidor de contenedor remoto se han detenido.

El número de referencias de sesión que se mantiene en la memoria, especificado por el parámetro **sessionTableSize**, se sigue manteniendo cuando las sesiones se almacenan en el contenedor web base. Las sesiones utilizadas menos recientemente se invalidan desde la memoria caché de sesión de contenedor web cuando se sobrepasa el valor **sessionTableSize**. Si la cuadrícula de datos remota pasa a estar disponible, las sesiones que se han invalidado desde la memoria caché de contenedor web pueden recopilar datos de la cuadrícula de datos remota y cargar los datos en una nueva sesión. Si no está disponible toda la cuadrícula de datos remota y la sesión se invalida desde la memoria caché de sesión, los datos de sesión del usuario se perderán. Debido a este problema, no debe concluir toda la cuadrícula de datos remota de producción cuando el sistema se ejecute con carga.

Configuración del gestor de sesiones HTTP para distintos servidores de aplicaciones

WebSphere eXtreme Scale se empaqueta con una implementación de gestión de sesiones que altera temporalmente el gestor de sesiones predeterminado para un contenedor web. Esta implementación proporciona opciones de réplica de sesiones, alta disponibilidad, mejor escalabilidad y configuración. Puede habilitar el inicio del contenedor de ObjectGrid incorporado genérico y del gestor de réplica de sesiones de WebSphere eXtreme Scale.

Acerca de esta tarea

Puede utilizar el gestor de sesiones HTTP con otros servidores de aplicaciones en los que no se ejecuta WebSphere Application Server, como WebSphere Application Server Community Edition. Para configurar otros servidores de aplicaciones para que utilicen la cuadrícula de datos, debe unir la aplicación e incorporar los archivos de archivado Java (JAR) de WebSphere eXtreme Scale en la aplicación.

Procedimiento

1. Una la aplicación de modo que pueda utilizar el gestor de sesiones. Para utilizar el gestor de sesiones, debe añadir las declaraciones de filtro apropiadas a los descriptores de despliegue web para la aplicación. Además, los parámetros de configuración del gestor de sesiones se pasan al gestor de sesiones en el formato de parámetros de inicialización de contexto de servlet en los descriptores de despliegue. Existen tres formas en las que puede introducir esta información en la aplicación:
   - Script **addObjectGridFilter**:

     Utilice un script de línea de mandatos proporcionado junto con eXtreme Scale para unir una aplicación con declaraciones de filtro y configuración con el formato de parámetros de inicialización de contexto de servlet. El script *inicio_wxs*/session/bin/addObjectGridFilter.sh|bat toma dos parámetros: la vía de acceso absoluta al archivo archivador de empresa (EAR) o al archivo archivador web (WAR) que desea unir y la vía de acceso absoluta al archivo de propiedades splicer que contiene varias propiedades de configuración. El formato de uso de este script es el siguiente:

     Windows

     ```
     addObjectGridFilter.bat <archivo_ear_o_war> <archivo_propiedades_splicer>
     ```

     UNIX

     ```
     addObjectGridFilter.sh <archivo_ear_o_war> <archivo_propiedades_splicer>
     ```

     UNIX Ejemplo de utilización de eXtreme Scale instalado en un directorio autónomo en UNIX:

     a. `cd inicio_wxs/session/bin`
     b. `addObjectGridFilter.sh /tmp/mySessionTest.ear inicio_wxs/session/samples/splicer.properties`

     El filtro de servlet que se une mantiene los valores predeterminados para los valores de configuración. Puede alterar temporalmente estos valores predeterminados con las opciones de configuración que especifique en el archivo de propiedades en el segundo argumento. Para obtener una lista de los parámetros que puede utilizar, consulte "Parámetros de inicialización del contexto del servlet" en la página 321.

     Puede modificar y utilizar el archivo splicer.properties de ejemplo proporcionado con la instalación de eXtreme Scale. También puede utilizar el

script **addObjectGridServlets**, que inserta el gestor de sesiones ampliando cada servlet. No obstante, el script recomendado es el script **addObjectGridFilter**.

- Script de construcción Ant:

  WebSphere eXtreme Scale se entrega con un archivo `build.xml` que se puede utilizar con Apache Ant, que está incluido en la carpeta *raíz_was*`/bin` de una instalación de WebSphere Application Server. Puede modificar el archivo `build.xml` para cambiar las propiedades de configuración del gestor de sesiones. Las propiedades de configuración son idénticas a los nombres de propiedades del archivo `splicer.properties`. Después de modificar el archivo `build.xml`, se invoca el proceso Ant ejecutando `ant.sh`, `ws_ant.sh` (UNIX) o `ant.bat`, `ws_ant.bat` (Windows).

- Actualizar manualmente el descriptor web:

  Edite el archivo `web.xml` que se empaqueta con la aplicación web para incorporar la declaración de filtro, su correlación de servlets y los parámetros de inicialización de contexto de servlet. No utilice este método porque es propenso a errores.

Para obtener una lista de los parámetros que puede utilizar, consulte "Parámetros de inicialización del contexto del servlet" en la página 321.

2. Incorpore los archivos JAR del gestor de réplica de sesión de WebSphere eXtreme Scale en la aplicación. Puede incorporar los archivos en el directorio `WEB-INF/lib` del módulo de aplicación o en la vía de acceso de clases del servidor de aplicaciones. Los archivos JAR varían según el tipo de contenedores que utilice:
   - Servidores de contenedor remotos: `ogclient.jar` y `sessionobjectgrid.jar`
   - Servidores de contenedor incorporados: `objectgrid.jar` y `sessionobjectgrid.jar`
3. Opcional: Si utiliza servidores de contenedor remotos, inicie los servidores de contenedor. Consulte "Inicio de servidores de contenedor" en la página 398 si desea más detalles.
4. Despliegue la aplicación. Despliegue la aplicación con un conjunto de pasos normales para un servidor o un clúster. Después de desplegar la aplicación, puede iniciarla.
5. Acceda a la aplicación. Ahora puede acceder a la aplicación, que interactúa con el gestor de sesiones y WebSphere eXtreme Scale.

Qué hacer a continuación

Puede cambiar la mayoría de los atributos de configuración para el gestor de sesiones cuando indica a la aplicación que utilice el gestor de sesiones. Estos atributos incluyen variaciones en el tipo de réplica (síncrona o asíncrona), el tamaño de la tabla de sesión en memoria, etc. Aparte de los atributos que pueden cambiarse durante la instrumentación de la aplicación, los otros únicos atributos de configuración que puede cambiar después del despliegue de la aplicación son los atributos relacionados con la topología de clúster de servidores de WebSphere eXtreme Scale y la forma en que sus clientes (gestores de sesiones) se conectan a los mismos.

Comportamiento del escenario remoto: Si no se puede acceder a la cuadrícula de datos entera que aloja los datos de sesión de aplicación desde el cliente de contenedor web, en su lugar el cliente utiliza el contenedor web base del servidor de aplicaciones para la gestión de sesiones. Es posible que no se pueda acceder a la cuadrícula de datos en los escenarios siguientes:

- Problema de red entre el contenedor web y los servidores de contenedor remotos.
- Los procesos del servidor de contenedor remoto se han detenido.

El número de referencias de sesión que se mantiene en la memoria, especificado por el parámetro **sessionTableSize**, se sigue manteniendo cuando las sesiones se almacenan en el contenedor web base. Las sesiones utilizadas menos recientemente se invalidan desde la memoria caché de sesión de contenedor web cuando se sobrepasa el valor **sessionTableSize**. Si la cuadrícula de datos remota pasa a estar disponible, las sesiones que se han invalidado desde la memoria caché de contenedor web pueden recopilar datos de la cuadrícula de datos remota y cargar los datos en una nueva sesión. Si toda la cuadrícula de datos remota no está disponible y la sesión se invalida desde la memoria caché de sesión, los datos de sesión de usuario se pierden. Debido a este problema, no concluya toda la cuadrícula de datos remota de producción cuando el sistema se ejecute con carga.

Archivos XML para la configuración del gestor de sesiones HTTP

Cuando inicia un servidor de contenedor que almacena datos de sesión HTTP, puede utilizar los archivos XML predeterminados o puede especificar archivos XML personalizados. Estos archivos crean nombres de ObjectGrid específicos, número de réplicas, etc.

Ubicación de los archivos de ejemplo

Estos archivos XML se empaquetan en *raíz_intal_wxs*/ObjectGrid/session/samples para una instalación autónoma o *raíz_was*/optionalLibraries/ObjectGrid/session/samples para WebSphere eXtreme Scale instalado en una célula de WebSphere Application Server.

Paquete XML incorporado

Si está configurando un escenario incorporado, el servidor de contenedor se inicia en el nivel de contenedor web. Utilice el archivo objectGrid.xml y el archivo objectGridDeployment.xml, que se proporcionan de forma predeterminada. Puede actualizar estos archivos para personalizar el comportamiento del gestor de sesiones HTTP.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd" xmlns="http://ibm.com/ws/objectgrid/config">
 <objectGrids>
  <objectGrid name="session" txTimeout="30">
   <bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
   <backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata" readOnly="false"
     lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
   <backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
     ttlEvictorType="NONE" copyMode="NO_COPY"/>
       <backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
     ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
  </objectGrid>
 </objectGrids>
    <backingMapPluginCollections>
        <backingMapPluginCollection id="objectgridSessionMetadata">
            <bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>
```

Figura 35. Archivo objectGrid.xml

Valores que puede cambiar:

Atributo de nombre ObjectGrid

El valor debe coincidir con los valores siguientes en otros archivos de configuración:

- La propiedad **objectGridName** del archivo `splicer.properties` que se utiliza para unir la aplicación web.
- El atributo **objectgridName** del archivo `objectGridDeployment.xml`.

Si tiene varias aplicaciones y desea que los datos de sesión se almacenen en cuadrículas de datos diferentes, esas aplicaciones deben tener valores de atributo de nombre de ObjectGrid diferentes.

7.1.1+ **Atributo txTimeout de ObjectGrid**

Este valor determina durante cuántos segundos puede estar abierta una transacción antes de que el servidor de contenedor desencadene un tiempo de espera excedido para la transacción. El valor predeterminado es 30 segundos y se puede modificar en función del entorno. Si la persistencia de sesiones HTTP se configura con el parámetro de inicialización de contexto de servlet de **replicationInterval** establecido en un valor mayor que cero, las transacciones se procesarán por lotes en una hebra. La longitud de la transacción la determinan el tamaño y el número de sesiones HTTP. El valor predeterminado es 10 segundos y 30 segundos es un tiempo suficiente para que se completen esas transacciones. Si la propiedad **replicationInterval** se establece en `0`, una transacción normalmente empieza cuando una aplicación web recupera un objeto HttpSession válido. La transacción se confirma al final de la solicitud de aplicación web. Si el entorno tiene solicitudes que tardan más de 30 segundos, establezca este valor como corresponda.

Valores que no puede cambiar:

ObjectGridEventListener

La línea ObjectGridEventListener no se puede cambiar y se utiliza internamente.

objectgridSessionMetadata

La línea objectgridSessionMetadata hace referencia a la correlación donde se almacenan los metadatos de sesión HTTP. Hay una entrada para cada sesión HTTP almacenada en la cuadrícula de datos de esta correlación.

objectgridSessionTTL.*

Este valor no se puede cambiar y es para uso futuro.

objectgridSessionAttribute.*

El texto `objectgridSessionAttribute.*` define una correlación dinámica. Este valor se utiliza para crear la correlación en la que se almacenan los atributos de sesión HTTP cuando el parámetro **fragmentedSession** se establece en `true` en el archivo `splicer.properties`. Esta correlación dinámica se denomina `objectgridSessionAttribute`. Se crea otra correlación basándose en esta plantilla denominada `objectgridSessionAttributeEvicted`, que almacena sesiones que han excedido el tiempo de espera pero que el contenedor web no ha invalidado.

La línea **MapEventListener** es interna y no se puede modificar

```
<?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>
```

Figura 36. Archivo `objectGridDeployment.xml`

Valores que puede cambiar:

Atributo de nombre ObjectGrid
El valor debe coincidir con los valores siguientes en otros archivos de configuración:

- La propiedad **`objectGridName`** del archivo `splicer.properties` que se utiliza para unir la aplicación web.
- El atributo **`name`** de ObjectGrid del archivo `objectGrid.xml`.

Si tiene varias aplicaciones y desea que los datos de sesión se almacenen en cuadrículas de datos diferentes, esas aplicaciones deben tener valores de atributo de nombre de ObjectGrid diferentes.

Atributos de elemento mapSet
Puede cambiar todas las propiedades de mapSet excepto las del atributo placementStrategy.

Name Se puede actualizar con cualquier valor.

numberOfPartitions
Especifica el número de particiones primarias que se han iniciado en cada servidor en que se aloja la aplicación web. A medida que añade las particiones, los datos se extienden más en el caso de una migración tras error. El valor predeterminado es 5 particiones y es correcto para la mayoría de las aplicaciones.

minSyncReplicas, maxSyncReplicas y maxAsyncReplicas
Especifica el número y tipo de réplicas que almacenan los datos de sesión HTTP. El valor predeterminado es 1 réplica asíncrona, que es correcto para la mayoría de las aplicaciones. La réplica síncrona se produce durante la vía de acceso de solicitud, que puede aumentar los tiempos de respuesta de su aplicación web.

developmentMode
Informa al servicio de ubicación de eXtreme Scale si los fragmentos réplica de una partición se pueden ubicar en el mismo nodo que su réplica primaria. Puede establecer el valor en true en un entorno de desarrollo, pero inhabilite esta función en un entorno de producción porque una anomalía en un nodo podría provocar la pérdida de los datos de sesión.

placementStrategy
No cambie el valor de este atributo.

El resto del archivo hace referencia a los mismos nombres de correlación que en el archivo `objectGrid.xml`. No se pueden cambiar estos nombres.

Valores que no puede cambiar:

- El atributo placementStrategy del elemento mapSet.

Paquete XML remoto

Cuando esté utilizando la modalidad remota, donde los contenedores se ejecutan como procesos autónomos, deberá utilizar el archivo `objectGridStandAlone.xml` y el archivo `objectGridDeploymentStandAlone.xml` para iniciar los procesos. Puede actualizar estos archivos para modificar la configuración.

```
<?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>
```

Figura 37. Archivo `objectGridStandAlone.xml`

Valores que puede cambiar:

Atributo de nombre ObjectGrid
El valor debe coincidir con los valores siguientes en otros archivos de configuración:

- La propiedad **`objectGridName`** del archivo `splicer.properties` que se utiliza para unir la aplicación web.
- El atributo **`objectgridName`** del archivo `objectGridStandAlone.xml`.

Si tiene varias aplicaciones y desea que los datos de sesión se almacenen en cuadrículas de datos diferentes, esas aplicaciones deben tener valores de atributo de nombre de ObjectGrid diferentes.

7.1.1+ **Atributo txTimeout de ObjectGrid**
Este valor determina durante cuántos segundos puede estar abierta una transacción antes de que el servidor de contenedor desencadene un tiempo de espera excedido para la transacción. El valor predeterminado es 30 segundos y se puede modificar en función del entorno. Si la persistencia de sesiones HTTP se configura con el parámetro de inicialización de contexto de servlet de **`replicationInterval`** establecido en un valor mayor que cero, las transacciones se procesarán por lotes en una hebra. La longitud de la transacción la determinan el tamaño y el número de sesiones HTTP. El valor predeterminado es 10 segundos y 30 segundos es un tiempo suficiente para que se completen esas transacciones. Si la propiedad **`replicationInterval`** se establece en `0`, una transacción normalmente empieza cuando una aplicación web recupera un objeto HttpSession válido. La transacción se confirma al final de la solicitud de aplicación web. Si el entorno tiene solicitudes que tardan más de 30 segundos, establezca este valor como corresponda.

Valores que no puede cambiar:

ObjectGridEventListener
La línea ObjectGridEventListener no se puede cambiar y se utiliza internamente.

objectgridSessionMetadata
La línea objectgridSessionMetadata hace referencia a la correlación donde se almacenan los metadatos de sesión HTTP. Hay una entrada para cada sesión HTTP almacenada en la cuadrícula de datos de esta correlación.

objectgridSessionTTL.*
Este valor no se puede cambiar y es para uso futuro.

objectgridSessionAttribute.*
El texto `objectgridSessionAttribute.*` define una correlación dinámica. Este valor se utiliza para crear la correlación en la que se almacenan los atributos de sesión HTTP cuando el parámetro **fragmentedSession** se establece en `true` en el archivo `splicer.properties`. Esta correlación dinámica se denomina `objectgridSessionAttribute`. Se crea otra correlación basándose en esta plantilla denominada `objectgridSessionAttributeEvicted`, que almacena sesiones que han excedido el tiempo de espera pero que el contenedor web no ha invalidado.

La línea **MetadataMapListener** es interna y no se puede modificar.

```
<?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>
```

Figura 38. Archivo `objectGridDeploymentStandAlone.xml`

Valores que puede cambiar:

Atributo objectgridName
El valor debe coincidir con los valores siguientes en otros archivos de configuración:

- La propiedad **objectGridName** del archivo `splicer.properties` que se utiliza para unir la aplicación web.
- El atributo **name** de ObjectGrid del archivo `objectGrid.xml`.

Si tiene varias aplicaciones y desea que los datos de sesión se almacenen en cuadrículas de datos diferentes, esas aplicaciones deben tener valores de atributo de nombre de ObjectGrid diferentes.

Atributos de elemento mapSet
Puede cambiar todas las propiedades de mapSet excepto las del atributo placementStrategy.

Name Se puede actualizar con cualquier valor.

numberOfPartitions
Especifica el número de particiones primarias que se han iniciado en cada servidor en que se aloja la aplicación web. A medida que añade las particiones, los datos se extienden más en el caso de una migración tras error. El valor predeterminado es 5 particiones y es correcto para la mayoría de las aplicaciones.

minSyncReplicas, maxSyncReplicas y maxAsyncReplicas
Especifica el número y tipo de réplicas que almacenan los datos de sesión HTTP. El valor predeterminado es 1 réplica asíncrona, que es correcto para la mayoría de las aplicaciones. La réplica síncrona se produce durante la vía de acceso de solicitud, que puede aumentar los tiempos de respuesta de su aplicación web.

developmentMode
Informa al servicio de ubicación de eXtreme Scale si los fragmentos réplica de una partición se pueden ubicar en el mismo nodo que su réplica primaria. Puede establecer el valor en true en un entorno de desarrollo, pero inhabilite esta función en un entorno de producción porque una anomalía en un nodo podría provocar la pérdida de los datos de sesión.

placementStrategy
No cambie el valor de este atributo.

El resto del archivo hace referencia a los mismos nombres de correlación que en el archivo `objectGrid.xml`. No se pueden cambiar estos nombres.

Valores que no puede cambiar:

- El atributo placementStrategy del elemento mapSet.

Parámetros de inicialización del contexto del servlet

La siguiente lista de parámetros de inicialización de contexto de servlet se puede especificar en el archivo de propiedades de splicer como corresponda en el método de unión elegido.

Parámetros

`objectGridType`

Un valor de serie de `REMOTE` o `EMBEDDED`. El valor predeterminado es `REMOTE`.

Si se establece en `REMOTE`, los datos de sesión se almacenan fuera del servidor en el que se ejecuta la aplicación web.

Si se establece en `EMBEDDED`, un contenedor de eXtreme Scale incorporado se inicia en el proceso de servidor de aplicaciones en el que se ejecuta la aplicación web.

`objectGridName`

Un valor tipo serie que define el nombre de la instancia de ObjectGrid que se utiliza para una aplicación web concreta. El valor predeterminado es session.

Esta propiedad debe reflejar el objectGridName en los archivos XML de ObjectGrid y XML de despliegue utilizados para iniciar los servidores de contenedor de eXtreme Scale.

`catalogHostPort`

Se puede contactar con el servidor de catálogo para obtener una instancia de ObjectGrid en el cliente. El valor debe tener el formato

host:puerto<,host:puerto>. El host es el host de escucha en el que se ejecuta el servidor de catálogo. El puerto es el puerto de escucha para ese proceso de servidor de catálogo. Esta lista puede ser arbitrariamente larga y sólo se utiliza para la rutina de carga. Se utiliza la primera dirección viable. Es opcional en WebSphere Application Server si se ha configurado la propiedad **catalog.services.cluster**.

replicationInterval

Valor entero (en segundos) que define el tiempo entre la grabación de las sesiones actualizadas en ObjectGrid. El valor predeterminado es 10 segundos. Los valores posibles están entre 0 y 60. 0 significa que las sesiones actualizadas se graban en el ObjectGrid al final de la llamada al método de servicio para cada solicitud. Un valor **replicationInterval** más alto mejora el rendimiento porque se graban menos actualizaciones en la cuadrícula de datos. No obstante, un valor más alto hace que la configuración sea menos tolerante a errores.

Este valor sólo se aplica cuando objectGridType se establece en REMOTE.

sessionTableSize

Valor entero que define el número de referencias de sesión conservadas en memoria. El valor predeterminado es 1000.

Este valor sólo pertenece a una topología REMOTE porque la topología EMBEDDED ya tiene los datos de sesión en el mismo nivel que el contenedor web.

Las sesiones se desalojan de la tabla en memoria basándose en la lógica utilizada menos recientemente (LRU). Cuando se desaloja una sesión de la tabla en memoria, se invalida del contenedor web. No obstante, los datos no se eliminan de la cuadrícula, por lo que las solicitudes subsiguientes de esa sesión todavía pueden recuperar los datos. Este valor debe ser mayor al valor de agrupación de hebras máximo de contenedor web, lo que reduce la contención en la memoria caché de sesión.

fragmentedSession

Un valor de serie de true o false. El valor predeterminado es true. Utilice este valor para controlar si el producto almacena los datos de sesión como una entrada completa o almacena cada atributo por separado.

Establezca el parámetro fragmentedSession en true si la sesión de aplicación web tiene muchos atributos o atributos con tamaños grandes. Establezca fragmentedSession en false si una sesión tiene menos atributos, porque todos los atributos se almacenan en la misma clave en la cuadrícula de datos.

En la implementación anterior basada en filtro, se hacía referencia a esta propiedad como persistenceMechanism, con los valores posibles de ObjectGridStore (fragmentado) y ObjectGridAtomicSessionStore (sin fragmentar).

securityEnabled

Un valor de serie de true o false. El valor predeterminado es false. Este valor habilita la seguridad del cliente eXtreme Scale. Debe coincidir con el valor **securityEnabled** en el archivo de propiedades de servidor eXtreme Scale. Si no coinciden los valores, se produce una excepción.

credentialGeneratorClass

El nombre de la clase que implementa la interfaz com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator. Esta clase se utiliza para obtener las credenciales de los clientes.

credentialGeneratorProps

Las propiedades de la clase de implementación CredentialGenerator. Las propiedades se establecen en el objeto con el método setProperties(String). El valor credentialGeneratorProps sólo se utiliza si el valor de la propiedad **credentialGeneratorClass** no es nulo.

objectGridXML

La ubicación de archivo del archivo `objectgrid.xml`. El archivo XML incorporado empaquetado en la biblioteca de eXtreme Scale se carga automáticamente si `objectGridType=EMBEDDED` y no se especifica la propiedad **objectGridXML**.

objectGridDeploymentXML

Especifica la ubicación del archivo XML de política de despliegue de objectGrid. El archivo XML incorporado empaquetado en la biblioteca de eXtreme Scale se carga automáticamente si `objectGridType=EMBEDDED` y no se especifica la propiedad **objectGridDeploymentXML**.

traceSpec

Especifica la especificación de rastreo de IBM WebSphere como un valor tipo serie. Utilice este valor para los servidores de aplicaciones que no sean WebSphere Application Server.

traceFile

Especifica la ubicación del archivo de rastreo como un valor tipo serie. Utilice este valor para los servidores de aplicaciones que no sean WebSphere Application Server.

cookieDomain

Especifica si necesita que las sesiones sean accesibles entre hosts. Establezca el valor en el nombre del dominio común de los hosts.

reuseSessionID

Establézcalo en `true` si el contenedor web subyacente vuelve a utilizar los ID de sesión entre solicitudes en hosts diferentes. El valor predeterminado es `false`. El valor de esta propiedad debe ser el mismo que el valor que tenga en el contenedor web. Si utiliza WebSphere Application Server y configura la persistencia de sesión HTTP de eXtreme Scale utilizando la consola administrativa o los scripts de herramienta **wsadmin**, la propiedad personalizada de contenedor web `HttpSessionIdReuse=true` se añade de forma predeterminada. **reuseSessionID** también se establece en `true`. Si no desea que se vuelvan a utilizar los ID de sesión, establezca la propiedad personalizada `HttpSessionIdReuse=false` en la propiedad personalizada de contenedor Web antes de configurar la persistencia de sesión de eXtreme Scale.

shareSessionsAcrossWebApps

Especifica si se comparten sesiones entre las aplicaciones web, se especifica con un valor tipo serie o bien con `true` o `false`. El valor predeterminado es `false`. La especificación de servlet indica que las Sessiones HTTP no se pueden compartir entre aplicaciones web. Para permitir este uso compartido se proporciona una ampliación de la especificación de servlet.

useURLEncoding

Establézcalo en `true` si desea habilitar la reescritura de URL. El valor

predeterminado es `false`, que indica que se utilizan cookies para almacenar datos de sesión. El valor de este parámetro debe ser el mismo que los valores de contenedor web para gestión de sesiones.

Archivo `splicer.properties`

El archivo `splicer.properties` contiene todas las opciones de configuración para configurar un gestor de sesiones basado en filtro de servlets.

Propiedades de splicer de ejemplo

Si elige utilizar cualquiera de las propiedades adicionales descritas en este archivo, asegúrese de eliminar el comentario de las líneas correspondientes a las propiedades que desea habilitar.

```
# Archivo de propiedades que contiene todas las opciones
# de configuración con las que el gestor de secciones de ObjectGrid
# basado en filtro de servlet se puede configurar para el uso.
#
# Este archivo de propiedades se puede crear para que contenga todos los
# valores predeterminados que se deben asignar a estos valores de configuración
# y se pueden alterar temporalmente valores individuales utilizando propiedades
# de tarea ANT, si este archivo de propiedades se utiliza conjuntamente con la
# tarea ANT filtersplicer.

# Un valor de serie de "REMOTE" o "EMBEDDED". El valor predeterminado es REMOTE.
# Si se establece en "REMOTE", los datos de la sesión se almacenarán fuera del
# servidor en el que se ejecuta la aplicación web. Si se establece en
# "EMBEDDED", se iniciará un contenedor incorporado de WebSphere eXtreme Scale
# en el proceso del servidor de aplicaciones en el que se ejecuta la aplicación.

objectGridType = REMOTE

# Un valor de serie que define el nombre del ObjectGrid
# que se debe utilizar para una determinada aplicación web. El nombre
# es session. Esta propiedad debe reflejar el objectGridName en los
# archivos xml de objectgrid y xml de despliegue para iniciar los
# contenedores de eXtreme Scale.

objectGridName = session

# Se puede contactar con el servidor de catálogo para obtener una
# instancia de ObjectGrid del lado del cliente. El valor debe tener el formato
#"host:puerto<,host:puerto>", donde el host es el host de escucha
# en el que se ejecuta el servidor de catálogo, y el puerto es el puerto de
# escucha de ese proceso de servidor de catálogo.
# Esta lista puede ser arbitrariamente larga y sólo se utiliza para la
# rutina de carga. Se utilizará la primera dirección viable. Es opcional
# si la propiedad catalog.services.cluster está configurada.

# catalogHostPort = host:puerto<,host:puerto>

# Valor entero (en segundos) que define el tiempo en segundos entre la
# grabación de sesiones actualizadas en ObjectGrid. El valor predeterminado es 10. Esta propiedad
# sólo se utiliza cuando objectGridType está establecido en REMOTE. Los valores posibles son de 0 a 60.
0 significa que se graban # sesiones actualizadas en el ObjectGrid al final de la llamada al
# método de servicio de servlet para cada solicitud.

replicationInterval = 10

# Valor entero que define el número de sesiones que se mantienen en
# memoria. El valor predeterminado es 1000.Esta propiedad solo se utiliza
# cuando objectGridType se establece en REMOTE. Cuando el número de sesiones
# almacenadas en memoria en el contenedor web excede este valor, la sesión
# a la que se ha accedido menos recientemente se invalida desde el contenedor
# web. Si entra una solicitud
```

```
# para esa sesión después de que ésta se haya invalidado, se creará una nueva sesión
# (con un nuevo ID de sesión si reuseSessionId=false),
# llena con los atributos de la sesión invalidada. Este valor
# se debe establecer siempre en un valor mayor que el tamaño máximo de la agrupación
# de hebras de contenedor web para evitar la contienda en esta memoria caché de sesión.

sessionTableSize = 1000

# Un valor de serie de "true" o "false", el valor predeterminado es "true".
# Sirve para controlar si almacenamos los datos de sesión como una entrada
# o almacenamos cada atributo individualmente.
# Se ha hecho referencia a esta propiedad como persistenceMechanism en la
# implementación anterior basada en filtro, con los valores posibles
# de ObjectGridStore (fragmentado) y ObjectGridAtomicSessionStore
# (no fragmentado).

fragmentedSession = true

# Un valor de serie de "true" o "false", el valor predeterminado es "false".
# Habilita la seguridad de cliente de eXtreme Scale. Este valor debe coincidir
# con el valor securityEnabled en el archivo de propiedades del servidor de
# eXtreme Scale. Si no coinciden los valores, se produce una excepción.

securityEnabled = false

# Especifica el soporte de autenticación de credenciales de cliente.
#   Los valores posibles son:
#   Nunca      - El cliente no da soporte a la autenticación de credenciales.
#   Soportado* - El cliente da soporte a la autenticación de credenciales solo si
#                el servidor también le da soporte.
#   Necesario  - El cliente requiere la autenticación de credenciales.
# El valor predeterminado es Supported.

# credentialAuthentication =

# Especifica el recuento de reintentos de autenticación si la credencial
# ha caducado. Si el valor se establece en 0, no habrá ningún
# reintento de autenticación.

# authenticationRetryCount =

# Especifica el nombre de la clase que implementa la interfaz
# com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator.
# Esta clase se utiliza para obtener credenciales para los clientes.

# credentialGeneratorClass =

# Especifica las propiedades de la clase de implementación CredentialGenerator.
# Las propiedades se establecen en el objeto con el método setProperties(String).
# El valor credentialGeneratorProps se utiliza sólo si el valor de la propiedad
# credentialGeneratorClass no es nulo.

# credentialGeneratorProps =

# La ubicación del archivo XML de objectgrid.
# El archivo XML incorporado empaquetado en la biblioteca de eXtreme Scale
# se cargará automáticamente si esta propiedad no se especifica y si
# objectGridType=EMBEDDED

# objectGridXML =

# La ubicación del archivo XML de política de despliegue de objectGrid.
# El archivo XML incorporado empaquetado en la biblioteca de eXtreme Scale
# se cargará automáticamente si esta propiedad no se especifica y si
# objectGridType=EMBEDDED

# objectGridDeploymentXML =
```

```
# Serie de especificación de rastreo de IBM WebSphere, útil para
# todos los otros servidores de aplicaciones, además de WebSphere.

# traceSpec =

# Serie de ubicación de archivo de rastreo, útil para
# todos los otros servidores de aplicaciones, además de WebSphere.

# traceFile=

# Esta propiedad debe establecerse si se requiere que se pueda acceder a las
# sesiones entre hosts. El valor será el nombre del
# dominio común entre los hosts.

# cookieDomain=

# Establecer en true si el contenedor web subyacente reutilizará
# los ID de sesión entre solicitudes a distintos hosts. El valor predeterminado
# es false. Este valor debe ser el mismo que lo que se ha establecido
# en el contenedor web.

# reuseSessionId=

# Un valor de serie de "true" o "false". El valor predeterminado es
# "false". Según la especificación de servlet, las sesiones HTTP no
# se pueden compartir entre aplicaciones web. Se proporciona una
# ampliación a la especificación de servlet para permitir este
# uso compartido.

# shareSessionsAcrossWebApps = false

# Establézcalo en true si desea habilitar urlRewriting.  El valor predeterminado
# es false, lo que indica que se utilizarán cookies para almacenar datos.
# Este valor debe reflejar lo que se ha establecido en los valores del contenedor
# web para la gestión de sesiones.

# useURLEncoding = false
```

Configuración del proveedor de memoria caché dinámica para WebSphere eXtreme Scale

La instalación y configuración del proveedor de memoria caché dinámica para eXtreme Scale depende de sus requisitos y del entorno que ha configurado.

Antes de empezar

- Para utilizar el proveedor de memoria caché dinámica, WebSphere eXtreme Scale debe estar instalado encima de los despliegues del nodo de WebSphere Application Server, incluido el nodo del gestor de despliegue. Si desea más información, consulte “Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client con WebSphere Application Server” en la página 163.
- La seguridad global debe estar habilitada en la consola administrativa de WebSphere Application Server si los servidores de catálogos de su dominio de servicio de catálogos tienen Secure Sockets Layer (SSL) habilitado o si desea utilizar SSL para un dominio de servicio de catálogo con SSL soportado. SSL es necesario para un servidor de catálogo si se establece el atributo transportType en `SSL-Required`, en Archivo de propiedades de servidor . Para obtener más información sobre cómo configurar la seguridad global, consulte Valores de seguridad globales.

Acerca de esta tarea

Si desea información sobre cómo utilizar el proveedor de memoria caché dinámica de eXtreme Scale con IBM WebSphere Commerce, consulte los temas siguientes en la documentación de IBM WebSphere Commerce:

- Enabling the dynamic cache service and servlet caching (Habilitación del servicio de memoria caché dinámica y el almacenamiento en la memoria caché de servlets)
- Enabling WebSphere Commerce data cache (Habilitación de la memoria caché de datos de WebSphere Commerce)

Si no dirige específicamente el almacenamiento en memoria caché a una instancia de la memoria caché de objetos o de la memoria caché de servlets, es probable que las llamadas a la API de memoria caché dinámica las atienda la memoria caché base. Si desea utilizar el proveedor de memoria caché dinámica de eXtreme Scale para almacenamiento en memoria caché de mandatos, JSP o servicios web, debe establecer la instancia de baseCache para utilizar el proveedor de memoria caché dinámica de eXtreme Scale. Se utilizan las mismas propiedades de configuración para configurar la instancia de baseCache. Recuerde que es necesario establecer estas propiedades de configuración como propiedades personalizadas de máquina virtual Java (JVM). Esta advertencia se aplica a cualquier propiedad de configuración de memoria caché descrita en esta sección, excepto el almacenamiento en memoria caché de servlet. Para utilizar eXtreme Scale con el proveedor de memoria caché dinámica para el almacenamiento en memoria caché de servlet, asegúrese de configurar la habilitación en las propiedades del sistema en lugar de propiedades personalizadas.

Procedimiento

1. Habilite el proveedor de memoria caché dinámica de eXtreme Scale.
   - **WebSphere Application Server Versión 7.0 y posterior:**

     Puede configurar el servicio de memoria caché dinámica para utilizar el proveedor de memoria caché dinámica de eXtreme Scale con la consola administrativa. Después de instalar eXtreme Scale, el proveedor de memoria caché dinámica de eXtreme Scale está inmediatamente disponible como una opción **Proveedor de memoria caché** en la consola administrativa. Para obtener más información, consulte el Information Center de WebSphere Application Server Versión 7.0: Selección de un proveedor de servicio de memoria caché.
   - **WebSphere Application Server Versión 6.1:**

     Utilice una propiedad personalizada para configurar el servicio de memoria caché dinámica para utilizar el proveedor de memoria caché dinámica de eXtreme Scale. También puede utilizar estas propiedades personalizadas en WebSphere Application Server Versión 7.0 y posterior. Para crear una propiedad personalizada en una instancia de memoria caché, pulse **Recursos** > **Instancias de memoria caché** > **tipo_instancia_memoria_caché** > **nombre_instancia_memoria_caché** > **Propiedades personalizadas** > **Nueva**. Si está utilizando la instancia de memoria caché base, cree las propiedades personalizadas en la JVM.

     com.ibm.ws.cache.CacheConfig.cacheProviderName
     Para utilizar el proveedor de memoria caché dinámica de eXtreme Scale, establezca el valor en `com.ibm.ws.objectgrid.dynacache.CacheProviderImpl`. Puede crear esta propiedad personalizada en una instancia de memoria caché dinámica, o en la instancia de memoria caché base. Si elige establecer

la propiedad personalizada en la instancia de memoria caché base, todas las demás instancias de memoria caché en el servidor utilizan de forma predeterminada el proveedor de eXtreme Scale. Cualquier propiedad de configuración del proveedor de memoria caché dinámica de eXtreme Scale establecida para baseCache corresponde a la propiedad de configuración predeterminada para todas las instancias de memoria caché respaldadas por eXtreme Scale. Para sustituir la instancia de memoria caché base y hacer que una instancia de memoria caché dinámica determinada utilice el proveedor de memoria caché dinámica predeterminado, cree la propiedad personalizada com.ibm.ws.cache.CacheConfig.cacheProviderName en la instancia de memoria caché dinámica y establezca el valor en `default`.

2. Opcional: Si utiliza instancias de memoria caché replicadas, configure el valor de réplica de la memoria caché.

   Con el proveedor de memoria caché dinámica de eXtreme Scale, puede tener instancias de memoria caché locales o instancias de memoria caché replicadas. Si está utilizando solo instancias de memoria caché locales, puede saltarse este paso.

   Utilice uno de los métodos siguientes para configurar la memoria caché replicada:

   - Habilitar réplica de memoria caché con la consola administrativa. Puede habilitar la réplica de memoria caché en cualquier momento en WebSphere Application Server Versión 7.0. En WebSphere Application Server Versión 6.1, debe crear un dominio de réplica DRS.
   - Habilitar réplica de memoria caché con la propiedad personalizada com.ibm.ws.cache.CacheConfig.enableCacheReplication para forzar a la memoria caché a notificar que se trata de una memoria caché replicada, aunque no se ha asignado a ella un dominio de réplica DRS. Establezca el valor de esta propiedad personalizada en `true`. Establezca esta propiedad personalizada en la instancia de memoria caché si está utilizando una memoria caché de objetos o una memoria caché de servlet, o en la JVM si está utilizando la instancia de baseCache.

3. Opcional: Si está utilizando eXtreme Scale como una memoria caché de fragmentos JSP, establezca la propiedad personalizada com.ibm.ws.cache.CacheConfig.disableTemplateInvalidation en `true` para inhabilitar las invalidaciones basadas en plantilla durante recargas JSP.

4. Configure la topología para el servicio de memoria caché dinámica.

   El único parámetro de configuración necesario para el proveedor de memoria caché dinámica de eXtreme Scale es la topología de memoria caché. Establezca la propiedad personalizada en la instancia de memoria caché o para el servicio de memoria caché dinámica si está utilizando la instancia baseCache. Entre el nombre de la propiedad personalizada como: `com.ibm.websphere.xs.dynacache.topology`.

   A continuación se muestran los tres valores posibles de esta propiedad. Debe utilizar uno de los valores permitidos:

   - `embedded`
   - `embedded_partitioned`
   - `remote`

   Si está utilizando las topologías incorporadas o incorporadas particionadas, considere establecer la propiedad personalizada `com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent` en `true` para

ahorrar costes de serialización. Establezca esta propiedad personalizada en la instancia de memoria caché o la JVM si está utilizando la instancia de baseCache.

5. Opcional: Si está utilizando una topología incorporada particionada, configure el número de contenedores iniciales para el servicio de memoria caché dinámica.

   Puede maximizar el rendimiento de las memorias caché que están utilizando la topología incorporada con particiones configurando el número de contenedores iniciales. Establezca la variable como una propiedad del sistema en la máquina virtual Java WebSphere Application Server.

   Especifique el nombre de la propiedad como: `com.ibm.websphere.xs.dynacache.num_initial_containers`.

   El valor recomendado de esta propiedad de configuración es un entero que es igual a, o ligeramente inferior a, el número total de instancias de WebSphere Application Server que acceden a esta instancia de memoria caché distribuida. Por ejemplo, si un servicio de memoria caché dinámica se comparte entre miembros de la cuadrícula de datos, el valor se debe establecer en el número de miembros de la cuadrícula de datos.

   Para las topologías embedded o embedded_partitioned, debe utilizar la Versión 7.0 de WebSphere Application Server. Establezca la propiedad personalizada siguiente en el proceso de la JVM para asegurarse de que los contenedores iniciales están disponibles inmediatamente.

   ```
   com.ibm.ws.cache.CacheConfig.createCacheAtServerStartup=true
   ```

6. Configure la cuadrícula de servicio de catálogo de eXtreme Scale.

   Al utilizar eXtreme Scale como el proveedor de memoria caché dinámica para una instancia de memoria caché distribuida, debe configurar un dominio de servicio de catálogo de eXtreme Scale.

   Un único dominio de servicio de catálogo puede prestar servicio a varios proveedores de servicios de memoria caché dinámica respaldados por eXtreme Scale.

   Un servicio de catálogo se puede ejecutar dentro o fuera de los procesos WebSphere Application Server. A partir de eXtreme Scale Versión 7.1, cuando se utiliza la consola administrativa para configurar los dominios de servicio de catálogo, la memoria caché dinámica utiliza estos valores. No es necesario llevar a cabo pasos adicionales para configurar un servicio de catálogo. Para obtener más información, consulte “Creación de dominios de servicio de catálogo en WebSphere Application Server” en la página 257.

7. Configure los objetos de clave personalizados.

   Cuando se utilicen los objetos personalizados como claves, los objetos deben implementar la interfaz Serializable o Externalizable. Cuando se utilizan las topologías particionadas incorporadas o las topologías incorporadas, debe colocar los objetos en la vía de acceso de biblioteca compartida de WebSphere, simplemente como si estuvieran siendo utilizadas con el proveedor de la memoria caché dinámica predeterminada. Consulte Utilización de las interfaces DistributedMap y DistributedObjectCache para la memoria caché dinámica en el centro de información de WebSphere Application Server Network Deployment si desea más detalles.

   Si utiliza la topología remota, debe colocar los objetos de clave personalizados en la CLASSPATH para los contenedores autónomos de eXtreme Scale. Si desea más información, consulte “Inicio de servidores de contenedor” en la página 398.

8. Opcional: Si está utilizando una topología remota, configure los servidores de contenedor de eXtreme Scale.
   - **Topología incorporada o incorporada particionada:**

     Los datos almacenados en memoria caché se almacenan en los servidores de contenedor de WebSphere eXtreme Scale. Los servidores de contenedor se pueden ejecutar dentro o fuera de procesos de WebSphere Application Server. El proveedor de eXtreme Scale crea automáticamente contenedores dentro del proceso WebSphere cuando se utilizan topologías incorporadas o topologías particionadas incorporadas para una instancia de memoria caché. No es necesaria realizar una configuración adicional para estas topologías.
   - **Topología remota:**

     Cuando utiliza la topología remota, debe iniciar servidores de contenedor autónomos de eXtreme Scale para que se inicien las instancias de WebSphere Application Server que acceden a la instancia de memoria caché. Consulte los pasos para iniciar los servidores de contenedor autónomos en la *Guía de administración* para obtener más información. Compruebe que todos los servidores de contenedor para un servicio de memoria caché dinámica específico apunten a los mismos puntos finales del servicio de catálogo.

     Los archivos de configuración XML de los contenedores autónomos de proveedor de memoria caché dinámica de eXtreme Scale están en el directorio *raíz_intal_wxs*/customLibraries/ObjectGrid/dynacache/etc para instalaciones sobre WebSphere Application Server, o en el directorio *raíz_intal_wxs*/ObjectGrid/dynacache/etc para instalaciones autónomas. Los archivos se denominan dynacache-remote-objectgrid.xml y dynacache-remote-definition.xml. Realice copias de estos archivos para editarlos y utilizarlos cuando inicie contenedores autónomos para el proveedor de memoria caché dinámica de eXtreme Scale. El parámetro **numInitialContainers** del archivo **dynacache-remote-deployment.xml** debe coincidir con el número de procesos de contenedor que están en ejecución. Tenga en cuenta que el atributo **numberOfPartitions** del archivo dynacache-remote-objectgrid.xml tiene un valor predeterminado de 47.

     Nota: El conjunto de procesos de servidor de contenedor debe tener suficiente memoria libre para dar servicio a todas las instancias de memoria caché dinámica configuradas para utilizar la topología remota. Cualquier proceso de WebSphere Application Server que comparta los mismos valores, o valores equivalentes, para la propiedad personalizada catalog.services.cluster debe utilizar el mismo conjunto de contenedores autónomos. El número de contenedores y el número de servidores en los que estos residen se deben dimensionar adecuadamente. Consulte “Planificación de la capacidad de memoria caché dinámica” en la página 59 si desea detalles adicionales.

     A continuación se muestra una entrada de línea de mandatos que inicia un contenedor autónomo para el proveedor de memoria caché dinámica de eXtreme Scale:

     UNIX

     ```
     startOgServer.sh contenedor1 -objectGridFile
     ../dynacache/etc/dynacache-remote-objectgrid.xml -deploymentPolicyFile
     ../dynacache/etc/dynacache-remote-deployment.xml -catalogServiceEndPoints
     MyServer1.company.com:2809
     ```

9. Para las topologías distribuidas o incorporadas, habilite el agente de dimensionamiento para mejorar las estimaciones de consumo de memoria.

   El agente de dimensionamiento hace una estimación del consumo de memoria (estadística usedBytes). El agente requiere una JVM Java 5 o superior.

Para cargar el agente añada el argumento siguiente a la línea de mandatos de la JVM:

`-javaagent:`*`directorio lib de WXS`*`/wxssizeagent.jar`

Para una topología incorporada, añada el argumento a la línea de mandatos del proceso de WebSphere Application Server.

Para una topología distribuida, añada el argumento a la línea de mandatos de los procesos (contenedores) de eXtreme Scale y al proceso de WebSphere Application Server.

Plug-in de memoria caché de nivel 2 (L2) JPA

WebSphere eXtreme Scale incluye los plug-ins de memoria caché de nivel (L2) para los proveedores OpenJPA e Hibernate Java Persistence API (JPA). Cuando se utiliza uno de estos plug-ins, la aplicación utiliza la API JPA. Se introduce una cuadrícula de datos entre la aplicación y la base de datos, lo cual mejora los tiempos de respuesta.

EL uso de eXtreme Scale como un proveedor de memoria caché de nivel 2 aumenta el rendimiento al leer y consultar datos y reduce la carga de la base de datos. WebSphere eXtreme Scale tiene ventajas sobre las implementaciones de memoria caché incorporadas porque la memoria caché se duplica automáticamente entre todos los procesos. Cuando un cliente almacena en memoria caché un valor, todos los demás clientes son capaces de utilizar el valor almacenado en memoria caché que está localmente en la memoria.

Puede configurar la topología y propiedades del proveedor de memoria caché L2 en el archivo `persistence.xml`. Para obtener más información sobre cómo configurar estas propiedades, consulte “Propiedades de configuración de la memoria caché JPA” en la página 338.

Consejo: El plug-in de memoria caché L2 JPA requiere una aplicación que utilice las API JPA. Si desea utilizar las API WebSphere eXtreme Scale para acceder a un origen de datos JPA, utilice el cargador JPA. Para obtener más información, consulte Cargadores JPA.

Consideraciones acerca de la topología de memoria caché L2 JPA

Los factores siguientes incluyen en el tipo de topología que se debe configurar:

1. **¿Qué volumen de datos espera que se almacenen en memoria caché?**
   - Si los datos pueden colocarse en un solo almacenamiento dinámico de JVM, utilice la “Topología incorporada” en la página 333 o la “Topología interna del dominio” en la página 332.
   - Si los datos no pueden colocarse en un solo almacenamiento dinámico de JVM, utilice la “Topología incorporada con particiones” en la página 334 o la “Topología remota” en la página 336
2. **¿Cuál es la proporción esperada de lectura respecto a grabación?**

   La proporción de lectura respecto a grabación afecta al rendimiento de la memoria caché L2. Cada topología maneja las operaciones de lectura y grabación de formas diferentes.
   - “Topología incorporada” en la página 333: lectura local, grabación remota
   - “Topología interna del dominio” en la página 332: lectura local, grabación local

- "Topología incorporada con particiones" en la página 334: Particionada: lectura remota, grabación remota
- "Topología remota" en la página 336: lectura remota, grabación remota.

Las aplicaciones que son principalmente de sólo lectura deben utilizar topologías incorporadas e internas de dominio cuando sea posible. Las aplicaciones que realizan más grabación deben utilizar topologías internas de dominio.

3. **¿Cuál es el porcentaje de datos consultados respecto a encontrados por una clave?**

   Cuando están habilitadas, las operaciones de consulta utilizan la memoria caché de consulta de JPA. Habilite la memoria caché de consulta de JPA sólo para aplicaciones con una alta proporción de lectura respecto a grabación, por ejemplo cuando se aproxime al 99% de operaciones de lectura. Si utiliza JPA la memoria caché de consulta de JPA, debe utilizar la "Topología incorporada" en la página 333 o la "Topología interna del dominio".

   La operación de búsqueda por clave capta una entidad de destino si la entidad de destino no tiene ninguna relación. Si la entidad de destino tiene relaciones con el tipo de captación EAGER, se captan estas relaciones junto con la entidad de destino. En la memoria caché de datos JPA, la captación de estas relaciones hace unos pocos aciertos de caché obtengan todos los datos de relación.

4. **¿Cuál es el nivel de obsolescencia tolerado de los datos?**

   En un sistema con pocas JVM, existe la latencia de réplica de datos para operaciones de grabación. El objetivo de la memoria caché es mantener una vista de datos sincronizados final en todas las JVM. Cuando se utiliza la topología interna del dominio, existe un retardo de réplica de datos para las operaciones de grabación. Las aplicaciones que utilizan esta topología deben ser capaces de tolerar lecturas obsoletas y grabaciones simultáneas que pueden sobrescribir los datos.

7.1.1+ Topología interna del dominio

Con una topología interna del dominio, los fragmentos primarios se colocan en cada uno de los servidores de contenedor de la topología. Estos fragmentos internos contiene el conjunto completo de datos para la partición. Cualquiera de estos fragmentos primarios también puede completar operaciones de grabación de memoria caché. Esta configuración elimina el cuello de botella en la topología incorporada donde todas las operaciones de grabación de memoria caché deben pasar por un único fragmento primario.

En una topología interna del dominio, no se crean fragmentos de réplica, incluso si tiene definidas réplicas en los archivos de configuración. Cada fragmento primario redundante contiene una copia completa de los datos, de forma que cada fragmento primario también se puede considerar un fragmento de réplica. Esta configuración utiliza una partición única, similar a la topología incorporada.

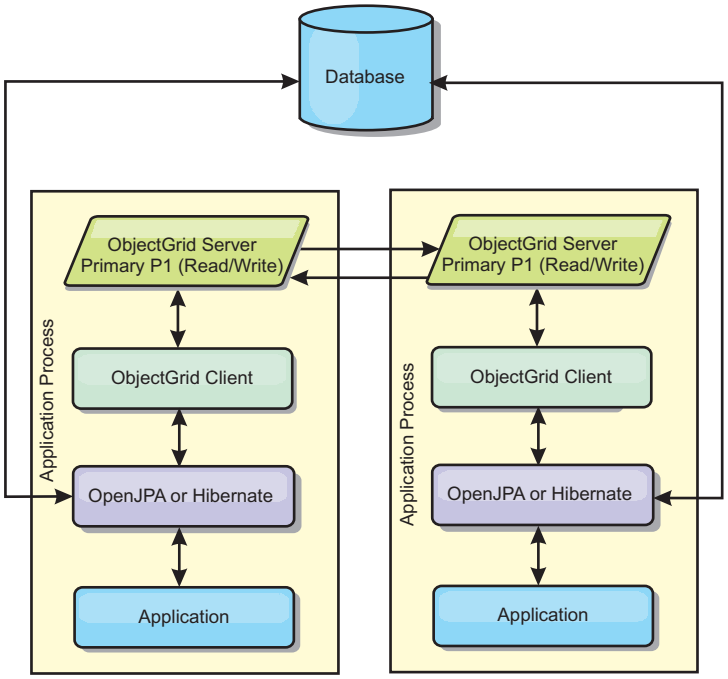


Figura 39. Topología interno del dominio JPA

Propiedades de configuración de memoria caché JPA relacionadas para la topología interna del dominio:

```
ObjectGridName=nombre_objectgridx,ObjectGridType=EMBEDDED,PlacementScope=CONTAINER_SCOPE,PlacementScopeTopology=HUB | RING
```

Ventajas:

- Las lecturas y actualizaciones de memoria caché son locales.
- La configuración es sencilla.

Limitaciones:

- Esta topología es más adecuada cuando los servidores de contenedor pueden contener todo el conjunto de datos de partición.
- Los fragmentos de réplica, incluso si se han configurado, nunca se colocan ya que cada uno de los servidores de contenedor aloja un fragmento primario. Sin embargo, todos los fragmentos primarios se replican con los otros fragmentos primarios, así que estos fragmentos primarios pasan a ser réplicas entre ellos.

Topología incorporada

Consejo: Considere utilizar la topología interna del dominio para obtener el mejor rendimiento.

Una topología incorporada crea un servidor de contenedor en el espacio de proceso de cada aplicación. OpenJPA e Hibernate leen directamente con la copia en memoria de la memoria caché y escriben en todas las demás copias. Puede mejorar el rendimiento de la escritura utilizando la réplica asíncrona. El rendimiento de esta topología predeterminada es mejor cuando el volumen de datos en caché es lo suficientemente pequeño para caber en un solo proceso. Con una topología incorporada, cree una única partición para los datos.

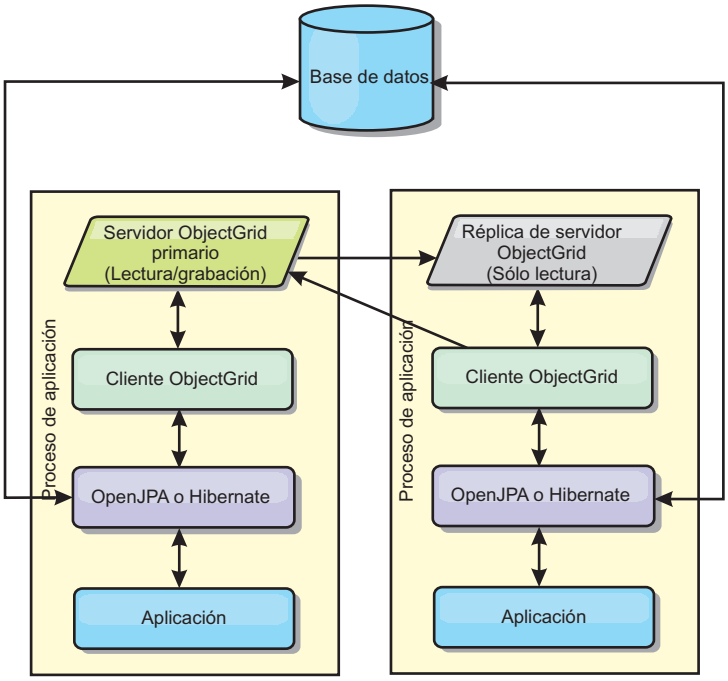


Figura 40. Topología incorporada JPA

Propiedades de configuración de memoria caché JPA relacionadas para la topología incorporada:

```
ObjectGridName=nombre_objectgrid,ObjectGridType=EMBEDDED,MaxNumberOfReplicas=núm_réplicas,ReplicaMode=SYNC | ASYNC | NONE
```

Ventajas:

- Todas las lecturas de memoria caché son acceso locales rápidos.
- La configuración es sencilla.

Limitaciones:

- El volumen de los datos se limita al tamaño del proceso.
- Todas las actualizaciones de memoria caché se envían a través del fragmento primario, lo que crea un cuello de botella.

Topología incorporada con particiones

Consejo: Considere utilizar la topología interna del dominio para obtener el mejor rendimiento.

PRECAUCIÓN:
No utilice la memoria caché de consulta de JPA con una topología particionada incorporada. La memoria caché de consulta almacena los resultados de consulta que son una colección de claves de entidad. La memoria caché de consulta va a la memoria caché de datos para captar todos los datos de entidad. Dado que la memoria caché de datos se divide entre múltiples procesos, estas llamadas adicionales pueden anular las ventajas de la memoria caché L2.

Cuando los datos en memoria caché son demasiado voluminosos para caber en un solo proceso, puede utilizar la topología incorporada particionada. Esta topología divide los datos en varios procesos. Los datos se dividen entre los fragmentos primarios, de modo que cada fragmento primario contiene un subconjunto de los

datos. Aún puede utilizar esta opción cuando la latencia de base de datos sea alta.

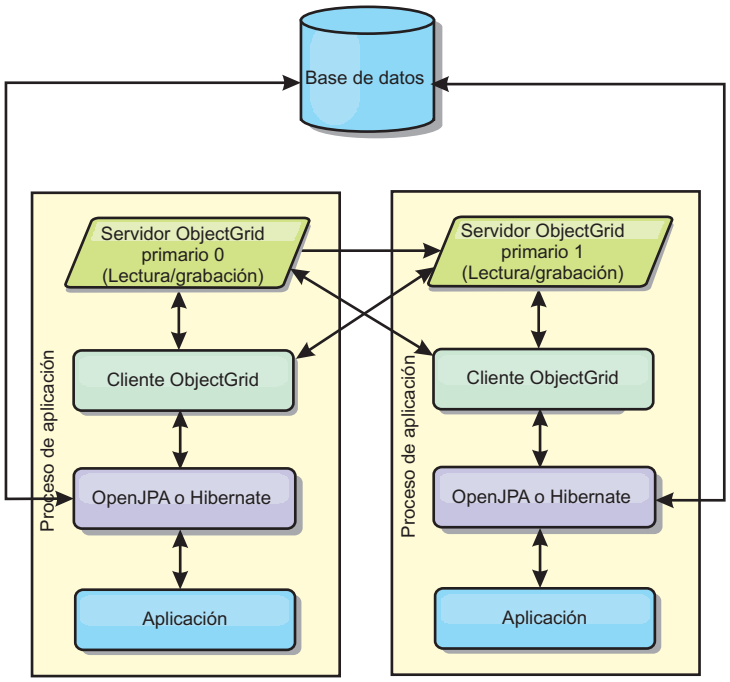


Figura 41. Topología incorporada con particiones JPA

Propiedades de configuración de memoria caché JPA relacionadas para la topología particionada incorporada:

```
ObjectGridName=nombre_objectgrid,ObjectGridType=EMBEDDED_PARTITION,ReplicaMode=SYNC | ASYNC | NONE,
NumberOfPartitions=núm_particiones,ReplicaReadEnabled=TRUE | FALSE
```

Ventajas:

- Almacena grandes cantidades de datos.
- La configuración es sencilla.
- Las actualizaciones de la memoria caché se reparten en diversos procesos.

Limitación:

- La mayoría de las lecturas y actualizaciones de la memoria caché son remotas.

Por ejemplo, para almacenar en memoria caché 10 GB de datos con un máximo de 1 GB por JVM, se necesitan 10 Máquinas virtuales Java. Por lo tanto, el número de particiones debe establecerse en 10 o más. De manera ideal, el número de particiones se debe establecer en un número primo donde cada fragmento almacena una cantidad razonable de memoria. Normalmente, el valor numberOfPartitions es igual al número de Máquinas virtuales Java. Con este valor, cada JVM almacena una partición. Si habilita las réplicas, debe aumentar el número de Máquinas virtuales Java en el sistema. De lo contrario, cada JVM también almacena una partición de réplica, que consume tanta memoria como una partición primaria.

Lea la información sobre el dimensionamiento de la memoria y el cálculo del número de particiones en la *Guía de administración* para maximizar el rendimiento de su configuración elegida.

Por ejemplo, en un sistema con cuatro Máquinas virtuales Java, y el valor del parámetro numberOfPartitions de 4, cada JVM aloja una partición primaria. Una operación de lectura tiene un 25 por ciento de posibilidades de captar datos desde una partición disponible localmente, que es mucho más rápido que obtener los datos de una JVM remota. Si una operación de lectura como, por ejemplo, ejecutar una consulta, debe captar una colección de datos que implican 4 particiones de manera uniforme, el 75 por ciento de las llamadas son remotas y el otro 25 por ciento son locales. Si el valor ReplicaMode se establece en `SYNC` o `ASYNC` y el valor ReplicaReadEnabled está establecido en `true`, se crean las cuatro particiones de réplica y se expanden a lo largo de las cuatro Máquinas virtuales Java. Cada JVM aloja una partición primaria y una partición de réplica. La probabilidad de que la operación de lectura se ejecute de forma local aumenta a un 50 por ciento. La operación de lectura que capta una colección de datos que implican cuatro particiones de manera uniforme tiene un 50 por ciento de llamadas remotas y un 50 por ciento de llamadas locales. Las llamadas locales son mucho más rápidas que las memorias remotas. Siempre que se producen llamadas remotas, baja el rendimiento.

Topología remota

PRECAUCIÓN:
No utilice la memoria caché de consulta de JPA con una topología remota. La memoria caché de consulta almacena los resultados de consulta que son una colección de claves de entidad. La memoria caché de consulta va a la memoria caché de datos para captar todos los datos de entidad. Dado que la memoria caché de datos es remota, estas llamadas adicionales pueden anular las ventajas de la memoria caché L2.

Consejo: Considere utilizar la topología interna del dominio para obtener el mejor rendimiento.

Una topología remota almacena todos los datos almacenados en la memoria caché en uno o más procesos separados, reduciendo el uso de la memoria de los procesos de la aplicación. Puede aprovechar la distribución de los datos en distintos procesos desplegando una cuadrícula de datos particionada y replicada de eXtreme Scale. A diferencia de las configuraciones incorporadas, e incorporadas particionadas, descritas en las secciones anteriores, si desea gestionar la cuadrícula de datos remota debe hacerlo de forma independiente de la aplicación y del proveedor JPA.

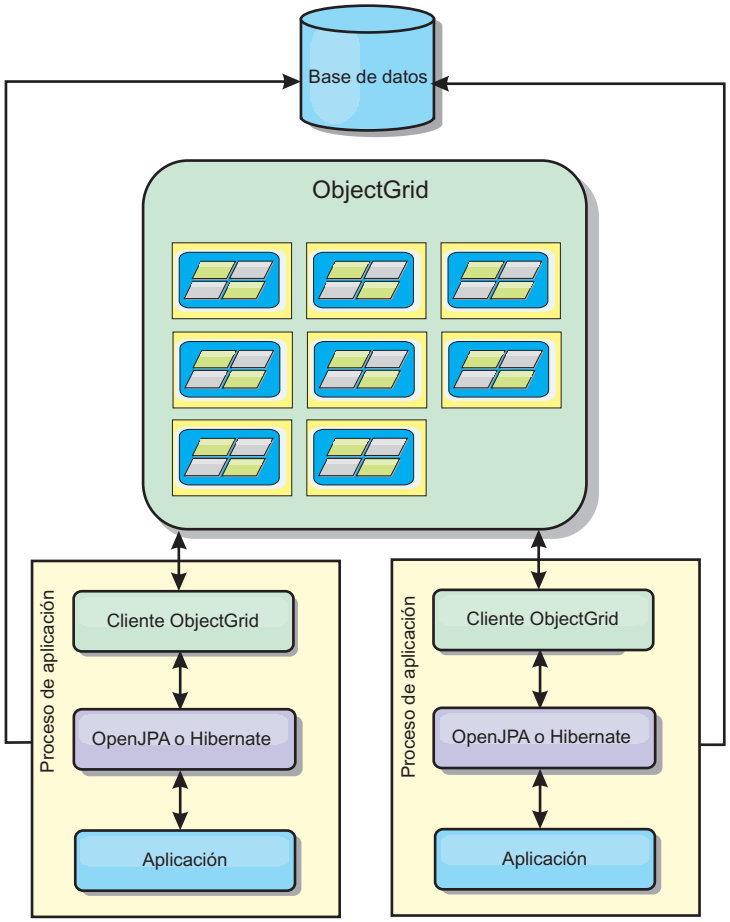


Figura 42. Topología remota JPA

Propiedades de configuración de memoria caché JPA relacionadas para la topología remota:

`ObjectGridName=`*`nombre_objectgrid`*`,ObjectGridType=REMOTE`

El tipo de ObjectGrid REMOTE no requiere ningún valor de propiedad porque el ObjectGrid y la política de despliegue están definidos independientemente de la aplicación JPA. El plug-in de memoria caché JPA se conecta remotamente a un ObjectGrid remoto.

Puesto que toda la interacción con ObjectGrid es remota, esta topología tiene el rendimiento más lento entre todos los tipos ObjectGrid.

Ventajas:

- Almacena grandes cantidades de datos.
- El proceso de aplicaciones está libre de los datos en memoria caché.
- Las actualizaciones de la memoria caché se reparten en diversos procesos.
- Opciones de configuración flexibles.

Limitación:

- Todas las lecturas y actualizaciones de la memoria caché son remotas.

Propiedades de configuración de la memoria caché JPA

WebSphere eXtreme Scale incluye plug-ins de memoria caché de nivel 2 para proveedores de OpenJPA e Hibernate Java Persistence API (JPA). Para configurar el plug-in de memoria caché L2, debe actualizar las propiedades en el archivo `persistence.xml`.

Consejo: El plug-in de memoria caché L2 JPA requiere una aplicación que utilice las API JPA. Si desea utilizar las API WebSphere eXtreme Scale para acceder a un origen de datos JPA, utilice el cargador JPA. Para obtener más información, consulte "Configuración de cargadores JPA" en la página 354.

Ubicación de las propiedades

Puede configurar estas propiedades en el archivo `persistence.xml`. La sintaxis para especificar las propiedades de este archivo varía en función de si utiliza OpenJPA o Hibernate:

- **OpenJPA:** puede establecer las propiedades en DataCache o QueryCache:

  ```
  <property name="openjpa.DataCache"
   value="<object_grid_datacache_class(<property>=<value>,...)"/>
  ```

  o bien

  ```
  <property name="openjpa.QueryCache"
   value="<object_grid_querycache_class(<property>=<value>,...)"/>
  ```

- **Hibernate:**

  ```
  <property name="objectgrid.configuration" value="<property>=<value>,..." />
  ```

Topología y propiedades predeterminadas

Se utilizan los siguientes valores predeterminados de las propiedades si no especifica valores en la configuración:

- **ObjectGridName**: nombre de unidad de persistencia
- **ObjectGridType**: `EMBEDDED`
- **NumberOfPartitions**: 1 (no se puede modificar, si el tipo de ObjectGrid es `EMBEDDED`)
- **ReplicaMode**: `SYNC`
- **ReplicaReadEnabled**: `TRUE` (no se puede modificar cuando el tipo de ObjectGrid es `EMBEDDED`)
- **MaxUsedMemory**: `TRUE`
- **MaxNumberOfReplicas**: `47` (debe ser inferior o igual al número de Máquinas virtuales Java en un sistema distribuido)

Propiedades

Puede configurar los plug-ins de memoria caché JPA con las propiedades siguientes.

ObjectGridName
: Especifica el nombre ObjectGrid exclusivo. El valor predeterminado es el nombre de la unidad de persistencia definida. Si el nombre de la unidad de persistencia no está disponible desde el proveedor JPA, se utiliza un nombre generado.

ObjectGridType
: Especifica el tipo de ObjectGrid.

Valores válidos:

EMBEDDED
El tipo de configuración predeterminado y recomendado. Sus valores predeterminados incluyen: `NumberOfPartitions=1`, `ReplicaMode=SYNC`, `ReplicaReadEnabled=true` y `MaxNumberOfReplicas=47`. Utilice el parámetro **ReplicaMode** para establecer la modalidad de réplica y el parámetro **MaxNumberOfReplicas** para establecer el número máximo de réplicas. Si un sistema tiene más de 47 Máquinas virtuales Java, establezca el valor **MaxNumberOfReplicas** para que sea igual al número de Máquinas virtuales Java.

EMBEDDED_PARTITION
El tipo que se utilizará cuando el sistema necesite almacenar en memoria caché una gran cantidad de datos en un sistema distribuido. El número predeterminado de particiones es `47` con una modalidad de réplica de `NONE`. En un sistema pequeño que sólo tiene unas pocas Máquinas virtuales Java, establezca el valor **NumberOfPartitions** en un valor igual o menor que el número de Máquinas virtuales Java. Puede especificar los valores **ReplicaMode**, **NumberOfPartitions**, y **ReplicaReadEnabled** para ajustar el sistema.

REMOTE La memoria caché intenta conectarse a un ObjectGrid remoto distribuido desde el servicio de catálogo.

MaxNumberOfReplicas
Especifica el número máximo de réplicas que se utilizarán para la memoria caché. Este valor se aplica solo al tipo `EMBEDDED`. Este número debe ser igual o mayor que el número de Máquinas virtuales Java en un sistema. El valor predeterminado es `47`.

Valores válidos: mayor o igual que 1

MaxUsedMemory

Valores válidos: `TRUE` o `FALSE`Habilita el desalojo de las entradas de la memoria caché cuando la memoria se restringe. El valor predeterminado es `TRUE` y desaloja los datos cuando el umbral de uso del almacenamiento dinámico de la JVM supera el 70 por ciento. Puede modificar el porcentaje predeterminado del umbral de utilización del almacenamiento dinámico de la JVM estableciendo la propiedad memoryThresholdPercentage en el archivo `objectGridServer.properties` y colocando este archivo en la classpath. Para obtener más información sobre los desalojadores, consulte Plug-ins para desalojar los objetos de memoria cachéla información sobre los desalojadores de la *Visión general del producto*. Si desea más información sobre el archivo de propiedades de servidor, consulte Archivo de propiedades de servidor .

NumberOfPartitions

Valores válidos: mayor o igual que 1Especifica el número de particiones que se utiliza para la memoria caché. Esta propiedad se aplica cuando el valor ObjectGridType está establecido en `EMBEDDED_PARTITION`. El valor predeterminado es `47`. Para el tipo `EMBEDDED`, el valor de **NumberOfPartitions** siempre es 1.

7.1.1+ **PlacementScope**
Indica la granularidad de una única instancia de un conjunto de correlaciones.

Valores válidos:

DOMAIN_SCOPE
(Valor predeterminado) Coloca un fragmento primario para cada partición en un servidor de contenedor del dominio de servicio de catálogo. Los fragmentos de réplica para cada partición se colocan en otros servidores de contenedor del dominio de servicio de catálogo.

CONTAINER_SCOPE
Coloca un fragmento primario en cada servidor de contenedor del dominio de servicio de catálogo.

7.1.1+ **PlacementScopeTopology**
Define la topología de enlace de los servidores de contenedor del dominio de servicio de catálogo. Este valor solo se utiliza cuando el valor PlacementScope se establece en un valor distinto a `DOMAIN_SCOPE`.

Valores válidos:

HUB (Valor predeterminado) Si se selecciona la topología de hub, se seleccionará una única cuadrícula de datos para que sea el hub. Todas las demás cuadrículas de datos se conectarán al hub. Esta topología es bastante escalable porque los radios tienen una única conexión. El hub puede convertirse en un cuello de botella y un único punto temporal de anomalía. El hub se reubicará a otro servidor de contenedor se falla. La ventaja de esta configuración es que se puede escribir un código de arbitraje más complejo que permite que un único punto, el hub, maneje todas las colisiones.

RING Si se selecciona la topología de anillo, todas las cuadrículas de datos se enlazan con otras dos cuadrículas de datos. No se garantiza el orden de los enlaces. Sin embargo, cada contenedor que se inicie probablemente se enlazará al primer contenedor y al último contenedor añadidos al anillo. Esta topología es la más escalable, pero solo pueden fallar dos enlaces antes de que se produzca una interrupción temporal. Si los servidores de contenedor fallan, se establecerán enlaces entre los supervivientes después de que se haya descubierto la anomalía.

ReplicaMode

Valores válidos: `SYNC/ASYNC/NONE`Especifica el método que se utiliza para copiar la memoria caché en las réplicas. Esta propiedad se aplica cuando el valor de ObjectGridType está establecido en `EMBEDDED` o `EMBEDDED_PARTITION`. El valor predeterminado es `NONE` para el tipo `EMBEDDED_PARTITION` y `SYNC` para el tipo `EMBEDDED`. Si el valor **ReplicaMode** está establecido en `NONE` para el ObjectType `EMBEDDED`, el tipo `EMBEDDED` sigue utilizando una **ReplicaMode** de `SYNC`.

ReplicaReadEnabled

Valores válidos: `TRUE` o `FALSE`Cuando está habilitado, los clientes leen las réplicas. Esta propiedad se aplica al tipo `EMBEDDED_PARTITION`. El valor predeterminado es FALSE para el tipo `EMBEDDED_PARTITION`. El tipo `EMBEDDED` siempre establece el valor **ReplicaReadEnabled** en `TRUE`.

writeBehind

Para proveedores de Hibernate solamente: cuando writeBehind está habilitado, las actualizaciones se almacenan temporalmente en un

almacenamiento de datos con ámbito de JVM hasta que se cumplen las condiciones de writeBehindInterval o writeBehindMaxBatchSize.

Atención: A no ser que esté habilitado writeBehind, se omiten los demás valores de configuración de grabación diferida.

Importante: Tenga cuidado al utilizar la función de grabación diferida. Las configuraciones de grabación diferida presentan una latencia mayor de la sincronización de datos entre todas las JVM y mayores posibilidades de pérdida de actualizaciones. En un sistema que tiene habilitada la configuración de grabación diferida con cuatro o más JVM, la actualización realizada en una JVM tiene un retardo aproximado de 15 segundos antes de que la actualización pase a estar disponible a otras JVM. Si dos JVM actualizan la misma entrada, la que deseche primero la actualización perderá su actualización.

Valores válidos: TRUE o FALSE

Valor predeterminado: FALSE

writeBehindInterval

Para proveedores de Hibernate solamente: especifica el intervalo de tiempo, en milisegundos, para desechar actualizaciones en la memoria caché.

Valores válidos: mayor o igual que 1

Valor predeterminado: 5000 (5 segundos)

writeBehindPoolSize

Para proveedores Hibernate solamente: especifica el tamaño máximo de la agrupación de hebras utilizado para desechar actualizaciones en la memoria caché.

Valores válidos: mayor o igual que 1

Valor predeterminado: 5

writeBehindMaxBatchSize

Para proveedores de Hibernate solamente: especifica el tamaño de lote máximo por memoria caché de región para desechar actualizaciones en la memoria caché. Por ejemplo, si el tamaño se establece en 1000, y las actualizaciones almacenadas en el almacenamiento de grabación diferida de una memoria caché de región sobrepasan las 1000 entradas, las actualizaciones se desecharán en la memoria caché, incluso si no se cumple la condición writeBehindInterval especificada. Las actualizaciones se desechan en la memoria caché con la periodicidad aproximada especificada del número de segundos especificados por el valor writeBehindInterval o siempre que el tamaño del almacenamiento de grabación diferida sobrepase las 1000 entradas. Tenga en cuenta que, en el caso de cumplimiento de la condición writeBehindMaxBatchSize, solo la memoria caché de región que cumpla esta condición desechará sus actualizaciones en almacenamiento de grabación diferida en la memoria caché. Un memoria caché de región normalmente se corresponde con una entidad o una consulta.

Valores válidos: mayor o igual que 1

Valor predeterminado: 1000

Configuración del plug-in de memoria caché OpenJPA

Puede configurar tanto implementaciones DataCache como implementaciones QueryCache para OpenJPA.

Antes de empezar

- Debe determinar la topología de plug-in de memoria caché JPA que desea utilizar. Consulte "Plug-in de memoria caché de nivel 2 (L2) JPA" en la página 331 para obtener más información sobre las distintas configuraciones y las propiedades que se deben establecer para cada topología.
- Debe tener una aplicación que utiliza las API JPA. Si desea utilizar API WebSphere eXtreme Scale para acceder a datos con JPA, utilice el cargador JPA. Para obtener más información, consulte "Configuración de cargadores JPA" en la página 354.

Procedimiento

1. Establezca las propiedades en el archivo `persistence.xml` para configurar el plug-in de memoria caché OpenJPA: Puede establecer estas propiedades en la implementación de memoria caché DataCache o Query.

   Las configuraciones DataCache y QueryCache son independientes entre sí. Puede habilitar cualquiera de estas configuraciones. Sin embargo, si ambas configuraciones están habilitadas, la configuración de QueryCache utiliza la misma configuración que la configuración de DataCache y se descartan sus propiedades de configuración.

   ```
   <property name="openjpa.DataCache"
             value="<object_grid_datacache_class(<property>=<value>,...)"/>
   ```

   o bien

   ```
   <property name="openjpa.QueryCache"
             value="<object_grid_querycache_class(<property>=<value>,...)"/>
   ```

   Nota: Sólo puede habilitar la configuración de QueryCache para topologías incorporadas e internas de dominio incorporadas.

   Puede especificar la propiedad ObjectGridName, la propiedad ObjectGridType y otras propiedades relacionadas con la política de despliegue de la lista de propiedades de la clase de memoria caché de ObjectGrid para personalizar la personalizar de la memoria caché. A continuación se muestra un ejemplo:

   ```
   <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"/>
   ```

   Consulte "Propiedades de configuración de la memoria caché JPA" en la página 338 para ver una lista de las propiedades que puede establecer.

2. En el archivo `persistence.xml`, también debe establecer la propiedad openjpa.RemoteCommitProvider en `sjvm`.

   ```
   <property name="openjpa.RemoteCommitProvider" value="sjvm"/>
   ```

3. Opcional: Para personalizar adicionalmente la cuadrícula de datos utilizada por la memoria caché, puede proporcionar valores adicionales con archivos XML.

   Para la mayoría de los escenarios, definir las propiedades de memoria caché debería ser suficiente. Para personalizar adicionalmente el ObjectGrid utilizado por la memoria caché, puede proporcionar archivos XML de configuración de

ObjectGrid OpenJPA en el directorio META-INF de forma similar al archivo persistence.xml. Durante la inicialización, la memoria caché intenta localizar estos archivos XML y procesarlos si se encuentran.

Existen tres tipos de archivos XML de configuración de ObjectGrid OpenJPA:

- openjpa-objectGrid.xml (configuración de ObjectGrid)

  Vía de acceso del archivo: META-INF/openjpa-objectGrid.xml

  Este archivo se utiliza para personalizar la configuración de ObjectGrid para los tipos EMBEDDED y EMBEDDED_PARTITION. Con el tipo REMOTE, se ignora este archivo. De manera predeterminada, cada clase de entidad se correlaciona con su propia configuración de BackingMap cuyo nombre será un nombre de clase de entidad dentro de la configuración de ObjectGrid. Por ejemplo, la clase de entidad com.mycompany.Employee se correlaciona con el objeto BackingMap com.mycompany.Employee. La configuración predeterminada de BackingMap es readOnly="false", copyKey="false", lockStrategy="NONE" y copyMode="NO_COPY". Puede personalizar algunos BackingMaps con la configuración que elija. Puede utilizar la palabra clave reservada ALL_ENTITY_MAPS para representar todas las correlaciones, excepto otras correlaciones personalizadas listadas en el archivo openjpa-objectGrid.xml. Los BackingMaps que no aparecen listados en este archivo openjpa-objectGrid.xml utilizan la configuración predeterminado. Si los BackingMaps personalizados no especifican las propiedades o el atributo BackingMaps y estos atributos se especifican en la configuración predeterminada, se aplican los valores de atributo de la configuración predeterminada. Por ejemplo, si se anota una clase de entidad con timeToLive=30, la configuración predeterminada de BackingMap para dicha entidad tiene un valor timeToLive=30. Si el archivo personalizado openjpa-objectGrid.xml también incluye dicho BackingMap, pero no especifica ningún valor timeToLive, el BackingMap personalizado tiene un valor timeToLive=30 de forma predeterminada. El archivo openjpa-objectGrid.xml tiene como objetivo alterar temporalmente o ampliar la configuración predeterminada.

- openjpa-objectGridDeployment.xml (política de despliegue)

  Vía de acceso del archivo: META-INF/openjpa-objectGridDeployment.xml

  Este archivo se utiliza para personalizar la política de despliegue. Cuando personalice la política de despliegue, si se proporciona el archivo openjpa-objectGridDeployment.xml, se descarta la política de despliegue predeterminada. Todos los valores de atributo de política de despliegue proceden del archivo openjpa-objectGridDeployment.xml proporcionado.

- openjpa-objectGrid-client-override.xml (configuración de sustitución de ObjectGrid de cliente)

  Vía de acceso del archivo: META-INF/openjpa-objectGrid-client-override.xml

  Este archivo se utiliza para personalizar un ObjectGrid del lado del cliente. De manera predeterminada, la memoria caché ObjectGrid aplica una configuración de alteración temporal de ObjectGrid de cliente predeterminada que inhabilita la memoria caché cercana. Si una aplicación requiere una memoria caché cercana, puede proporcionar este archivo y especificar numberOfBuckets="xxx". La alteración temporal del cliente predeterminado inhabilita la memoria caché cercana estableciendo numberOfBuckets="0". La memoria caché cercana se puede activar al restablecer numberOfBuckets en un valor mayor que 0 con el archivo openjpa-objectGrid-client-override.xml. La forma en la que trabaja el archivo openjpa-objectGrid-client-override.xml es similar al archivo

`openjpa-objectGrid.xml`. Altera temporalmente o amplía la configuración de alteración temporal de ObjectGrid de cliente predeterminada.

En función de la topología configurada de eXtreme Scale, puede proporcionar cualquier de es tos tres archivos XML para personalizar dicha topología.

Para ambos tipos, EMBEDDED y EMBEDDED_PARTITION, puede proporcionar cualquiera de estos tres archivos XML para personalizar el ObjectGrid, la política de despliegue y la configuración de alteración temporal de ObjectGrid de cliente.

Para un ObjectGrid REMOTE, la memoria caché de ObjectGrid no crea un ObjectGrid dinámico. En lugar de esto, la memoria caché sólo obtiene un ObjectGrid del cliente en el servicio de catálogo. Sólo puede proporcionar el archivo `openjpa-objectGrid-client-override.xml` para personalizar la configuración de alteración temporal de ObjectGrid de cliente.

4. Opcional: (Solo configuraciones remotas) Configure un sistema eXtreme Scale externo si desea configurar una memoria caché con un tipo de ObjectGrid REMOTE.

   Debe configurar un sistema eXtreme Scale externo si desea configurar una memoria caché con un tipo de ObjectGrid REMOTE. Necesita ambos archivos XML de configuración de ObjectGrid y, también, de ObjectGridDeployment que se basan en el archivo `persistence.xml` para configurar un sistema externo. Para ver ejemplos de estos archivos de configuración, consulte “Ejemplo: Archivos XML de ObjectGrid OpenJPA” en la página 345.

Resultados

Configuración EMBEDDED, EMBEDDED_PARTITION, o interna de dominio:

Cuando se inicia una aplicación, el plug-in detecta o inicia automáticamente un servicio de catálogo, inicia un servidor de contenedor y conecta los servidores de contenedor al servicio de catálogo. El plug-in se comunica con el contenedor ObjectGrid y sus iguales que se ejecutan en otros procesos de servidor de aplicaciones mediante la conexión de cliente.

Configuración REMOTE:

La política de despliegue se especifica aparte de la aplicación JPA. Un sistema de ObjectGrid externo tiene tanto procesos de servicio de catálogo como de servidor de contenedor. Debe iniciar un servicio de catálogo antes de iniciar servidores de contenedor. Consulte “Inicio de los servidores autónomos” en la página 395 y “Inicio de servidores de contenedor” en la página 398 para obtener más información.

Qué hacer a continuación

- Desarrolle una aplicación OpenJPA que utilice la configuración. Para obtener más información, consulte Ejemplo: Utilización del plug-in Hibernate para precargar datos en la memoria caché de ObjectGrid.
- En un entorno de producción, cree dominios de servicio de catálogo para los procesos creados automáticamente para la configuración EMBEDDED o EMBEDDED_PARTITION.
  - Entorno autónomo:

    Si no se ejecutan los servidores dentro de un proceso WebSphere Application Server, los hosts y los puertos del dominio de servicio de catálogo se especifican utilizando el archivo de propiedades denominado

objectGridServer.properties. Este archivo se debe almacenar en la classpath de la aplicación y tiene definida la propiedad **catalogServiceEndPoints**. El dominio de servicio de catálogo se inicia independientemente de los procesos de la aplicación y se debe iniciar antes de que se inicien los procesos de la aplicación.

el formato del archivo objectGridServer.properties es el siguiente:

```
PuntosFinalesServicioCatálogos=<nombrehost>:<puerto1>,<nombrehost2>:<puerto2>
```

– Entorno de WebSphere Application Server:

Si se ejecuta dentro de un proceso WebSphere Application Server, el plug-in de memoria caché JPA se conecta automáticamente el servicio de catálogo o al dominio de servicio de catálogo definido para la célula de WebSphere Application Server.

- Cuando se utiliza un valor de ObjectGridType EMBEDDED o EMBEDDED_PARTITION en un entorno Java SE, utilice el método System.exit(0) al final del programa para detener el servidor eXtreme Scale incorporado. De lo contrario, el programa puede dejar de responder.

Ejemplo: Archivos XML de ObjectGrid OpenJPA:

Los archivos XML OpenJPA ObjectGrid deben crearse según la configuración de la unidad de persistencia.

Archivo persistence.xml

A continuación, aparece un archivo persistence.xml que es un ejemplo que representa la configuración de una unidad de persistencia:

```
<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>
```

Archivo openjpa-objectGrid.xml

El archivo openjpa-objectGrid.xml se utiliza para personalizar la configuración de ObjectGrid tanto para el tipo EMBEDDED como para el tipo EMBEDDED_PARTITION. A continuación, aparece el archivo openjpa-objectGrid.xml que coincide con el archivo persistence.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
  <objectGrid name="Annuity">
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject"
                readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
    <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
                pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
    <backingMap name="ObjectGridQueryCache" readOnly="false" copyKey="false"
                lockStrategy="NONE" copyMode="NO_COPY" pluginCollectionRef="ObjectGridQueryCache"
                evictionTriggers="MEMORY_USAGE_THRESHOLD"  />
  </objectGrid>
</objectGrids>
<backingMapPluginCollections>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection
          id="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout">
    <bean id="ObjectTransformer"
          className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="ObjectGridQueryCache">
    <bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.HashIndex" >
        <property name="Name" type="java.lang.String"
                  value="QueryCacheKeyIndex" description="name of index"/>
        <property name="POJOKeyIndex" type="boolean" value="true" description="POJO Key Index "/>
    </bean>
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" >
```

```
     </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Importante:

1. Cada entidad se correlaciona con un BackingMap cuyo nombre es el nombre de clase de entidad totalmente calificado.

   De forma predeterminada, las entidades forman parte de la memoria caché de segundo nivel. En las clases Entity que es necesario excluir del almacenamiento en memoria caché, puede incluir la anotación `@DataCache(enabled=false)` en la clase de entidad que desea excluir de la memoria caché L2:

   ```
   import org.apache.openjpa.persistence.DataCache;
   @Entity
   @DataCache(enabled=false)
   public class OpenJPACacheTest { ... }
   ```

2. Si las clases de entidad están en una jerarquía de herencia, las clases hija se correlacionan con el BackingMap padre. La jerarquía de herencia comparte un solo BackingMap.
3. La correlación ObjectGridQueryCache es necesaria para dar soporte a QueryCache.
4. El objeto backingMapPluginCollection de cada correlación de entidad debe tener ObjectTransformer con la clase com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer.
5. El objeto backingMapPluginCollection de la correlación ObjectGridQueryCache debe tener el índice de clave denominado QueryCacheKeyIndex, como se muestra en el ejemplo.
6. El desalojador es opcional para cada correlación.

Archivo `openjpa-objectGridDeployment.xml`

El archivo `openjpa-objectGridDeployment.xml` se utiliza para personalizar la política de despliegue. A continuación, aparece el archivo `openjpa-objectGridDeployment.xml` que coincide con el archivo `persistence.xml`:

`openjpa-objectGridDeployment.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Annuity">
    <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1"
            minSyncReplicas="0" maxSyncReplicas="4" maxAsyncReplicas="0"
            replicaReadEnabled="true">
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistebleObject" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <map ref="ObjectGridQueryCache" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Nota: La correlación ObjectGridQueryCache es necesaria para dar soporte a QueryCache.

Configuración del plug-in Hibernate

Puede habilitar la memoria caché para utilizar el plug-in de memoria caché Hibernate especificando los archivos de propiedades.

Antes de empezar

- Debe determinar la topología de plug-in de memoria caché JPA que desea utilizar. Consulte “Plug-in de memoria caché de nivel 2 (L2) JPA” en la página 331 para obtener más información sobre las distintas configuraciones.
- Debe tener una aplicación que utiliza las API JPA. Si desea utilizar API WebSphere eXtreme Scale para acceder a datos con JPA, utilice el cargador JPA. Para obtener más información, consulte “Configuración de cargadores JPA” en la página 354.

Procedimiento

1. Si está utilizando WebSphere Application Server, coloque los archivos de archivado Java (JAR) en las ubicaciones correspondientes para su configuración.

   El plug-in de memoria caché Hibernate está empaquetado en el archivo `oghibernate-cache.jar` y se instala en el directorio *raíz_was*/`optionalLibraries/ObjectGrid`. Para utilizar el plug-in de memoria caché Hibernate, debe incluir el archivo `oghibernate-cache.jar` en la biblioteca Hibernate. Por ejemplo, si incluye la biblioteca Hibernate en la aplicación, debe incluir también el archivo `oghibernate-cache.jar`. Si define una biblioteca compartida para incluir la biblioteca Hibernate, debe añadir el archivo `oghibernate-cache.jar` en el directorio de la biblioteca compartida.

   eXtreme Scale no instala el archivo `cglib.jar` en el entorno de WebSphere Application Server. Si tiene aplicaciones o bibliotecas compartidas existentes como, por ejemplo, Hibernate, que dependen del archivo `cglib.jar`, localice el archivo `cglib.jar` e inclúyalo en la classpath. Por ejemplo, si la aplicación incluye todos los archivos JAR de biblioteca Hibernate, pero excluye el archivo `cglib.jar` disponible con Hibernate, debe incluir el archivo `cglib.jar` que procede de Hibernate en la aplicación.
2. Establezca las propiedades en el archivo `persistence.xml` para configurar el plug-in de memoria caché Hibernate

   A continuación se muestra la sintaxis para establecer las propiedades en el archivo `persistence.xml`:

   ```
   <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="<property>=<value>,..." />
   <property name="objectgrid.hibernate.regionNames" value="<regionName>,.." />
   ```

   - **`hibernate.cache.provider_class`**: El valor de la propiedad **`provider_class`** es la clase com.ibm.websphere.objectgrid.hibernate.cache. ObjectGridHibernateCacheProvider.
   - **`hibernate.cache.use_query_cache`**: para habilitar la memoria caché de consulta, establezca el valor en `true` en la propiedad **`use_query_cache`**.

     Nota: Sólo puede habilitar la memoria caché de consulta para topologías incorporadas e internas de dominio incorporadas.
   - **`objectgrid.configuration`**: utilice la propiedad objectgrid.configuration para especificar las propiedades de configuración de memoria caché de eXtreme Scale, incluido el atributo ObjectGridType que especifica cómo colocar los fragmentos en la cuadrícula de datos.

     Debe especificar un valor de propiedad ObjectGridName exclusivo para evitar posibles conflictos de denominación. Las otras propiedades de configuración de la memoria caché de eXtreme Scale son opcionales.

     Para habilitar el almacenamiento en memoria caché de grabación diferida, utilice los siguientes atributos de grabación diferida en la propiedad de configuración objectgrid.configuration. Cuando está habilitado el almacenamiento en memoria caché con escritura diferida, las actualizaciones

se almacenan temporalmente en un almacenamiento de datos con ámbito de JVM hasta que se cumplen las condiciones de writeBehindInterval o writeBehindMaxBatchSize, cuando los datos se vacían en la memoria caché.

```
writeBehind=true, writeBehindInterval=5000, writeBehindPoolSize=10, writeBehindMaxBatchSize=1000
```

Atención: A no ser que esté habilitado writeBehind, se omiten los demás valores de configuración de grabación diferida.

Para obtener más información sobre los valores que puede establecer en la propiedad **objectgrid.configuration**, consulte "Propiedades de configuración de la memoria caché JPA" en la página 338.

- **objectgrid.hibernate.regionNames**: la propiedad objectgrid.hibernate.regionNames es óptima y se debe especificar cuando los valores de regionNames se definen después de la inicialización de la memoria caché de eXtreme Scale. Considere el ejemplo de una clase de entidad que se ha correlacionado con un regionName con la clase de entidad no especificada en el archivo `persistence.xml` o no incluida en el archivo de correlaciones Hibernate. De forma adicional, tiene una anotación Entity. El regionName para esta clase de entidad se resuelve al cargar la clase cuando se inicializa la memoria caché de eXtreme Scale. Otro ejemplo es el método Query.setCacheRegion(String regionName) que se ejecuta después de la inicialización de la memoria caché de eXtreme Scale. En estas situaciones, incluya todos los regionNames dinámicos posibles determinados en la propiedad objectgrid.hibernate.regionNames de forma que la memoria caché de eXtreme Scale puede preparar las BackingMaps para todos los regionNames.

3. Opcional: Para personalizar adicionalmente la cuadrícula de datos utilizada por la memoria caché, puede proporcionar valores adicionales con archivos XML.

   Para la mayoría de los escenarios, definir las propiedades de memoria caché debería ser suficiente. Para personalizar de forma adicional el ObjectGrid utilizado por la memoria caché, puede proporcionar archivos XML de configuración de ObjectGrid Hibernate en el directorio `META-INF` de forma similar al archivo `persistence.xml`. Durante la inicialización, la memoria caché intenta localizar estos archivos XML y procesarlos si se encuentran.

   Existen tres tipos de archivos XML de configuración de ObjectGrid Hibernate:

   - `hibernate-objectGrid.xml` (configuración de ObjectGrid)

     Vía de acceso del archivo: `META-INF/hibernate-objectGrid.xml`

     De manera predeterminada, cada clase de entidad tiene un regionName asociado (el valor predeterminado es el nombre de la clase de entidad) que se correlaciona con una configuración de BackingMap de nombre regionName dentro de la configuración de ObjectGrid. Por ejemplo, la clase de entidad com.mycompany.Employee tiene un valor predeterminado de regionName asociado como com.mycompany.Employee BackingMap. La configuración predeterminada de BackingMap es readOnly="false", copyKey="false", lockStrategy="NONE" y `copyMode="NO_COPY"`. Puede personalizar algunos BackingMaps con una configuración elegida. La palabra clave reservada "ALL_ENTITY_MAPS" se puede utilizar para representar todas las correlaciones, excepto otras correlaciones personalizadas listas en el archivo `hibernate-objectGrid.xml`. Los BackingMaps que no aparecen listados en este archivo `hibernate-objectGrid.xml` utilizan la configuración predeterminada.

   - `hibernate-objectGridDeployment.xml` (política de despliegue)

     Vía de acceso del archivo: `META-INF/hibernate-objectGridDeployment.xml`

Este archivo se utiliza para personalizar la política de despliegue. Al personalizar la política de despliegue, si se proporciona `hibernate-objectGridDeployment.xml`, se descarta la política predeterminada de despliegue. Todos los valores del atributo de la política de despliegues procederán del archivo `hibernate-objectGridDeployment.xml` proporcionado.

- `hibernate-objectGrid-client-override.xml` (configuración de sustitución de ObjectGrid de cliente)

  Vía de acceso del archivo: `META-INF/hibernate-objectGrid-client-override.xml`

  Este archivo se utiliza para personalizar un ObjectGrid de cliente. De manera predeterminada, la memoria caché ObjectGrid aplica una configuración de alteración temporal de cliente predeterminada que inhabilita la memoria caché cercana. Si la aplicación requiere una memoria caché cercana, puede proporcionar este archivo y especificar `numberOfBuckets="xxx"`. La alteración temporal del cliente predeterminado inhabilita la memoria caché cercana estableciendo `numberOfBuckets="0"`. La memoria caché cercana se puede activar si se restablece el atributo numberOfBuckets en un valor mayor que cero con el archivo `hibernate-objectGrid-client-override.xml`. La forma en la que trabaja el archivo `hibernate-objectGrid-client-override.xml` es similar a `hibernate-objectGrid.xml`: Altera temporalmente o amplía la configuración predeterminada de alteración temporal de ObjectGrid de cliente.

En función de la topología configurada de eXtreme Scale, puede proporcionar cualquier de es tos tres archivos XML para personalizar dicha topología.

Para ambos tipos, EMBEDDED y EMBEDDED_PARTITION, puede proporcionar cualquiera de los tres archivos XML para personalizar el ObjectGrid, la política de despliegue y la configuración de alteración temporal de ObjectGrid de cliente.

Para un ObjectGrid REMOTE, la memoria caché no crea ningún ObjectGrid dinámico. La memoria caché sólo obtiene un ObjectGrid de cliente del servicio de catálogo. Sólo puede proporcionar un archivo `hibernate-objectGrid-client-override.xml` para personalizar la configuración de alteración temporal de ObjectGrid de cliente.

4. Opcional: (Solo configuraciones remotas) Configure un sistema eXtreme Scale externo si desea configurar una memoria caché con un tipo de ObjectGrid REMOTE.

   Debe configurar un sistema eXtreme Scale externo si desea configurar una memoria caché con un tipo de ObjectGrid REMOTE. Necesita ambos archivos XML de configuración de ObjectGrid y, también, de ObjectGridDeployment que se basan en el archivo `persistence.xml` para configurar un sistema externo. Para ver ejemplos de estos archivos de configuración, consulte “Ejemplo: Archivos XML de ObjectGrid Hibernate” en la página 351.

Resultados

Configuración EMBEDDED o EMBEDDED_PARTITION:

Cuando se inicia una aplicación, el plug-in detecta o inicia automáticamente un servicio de catálogo, inicia un servidor de contenedor y conecta los servidores de contenedor al servicio de catálogo. El plug-in se comunica con el contenedor ObjectGrid y sus iguales que se ejecutan en otros procesos de servidor de aplicaciones mediante la conexión de cliente.

Cada entidad JPA tiene una correlación de respaldo independiente asignada utilizando el nombre de clase de la entidad. Cada BackingMap tiene los atributos siguientes:

- `readOnly="false"`
- `copyKey="false"`
- `lockStrategy="NONE"`
- `copyMode="NO_COPY"`

Configuración REMOTE:

La política de despliegue se especifica aparte de la aplicación JPA. Un sistema de ObjectGrid externo tiene tanto procesos de servicio de catálogo como de servidor de contenedor. Debe iniciar un servicio de catálogo antes de iniciar servidores de contenedor. Consulte "Inicio de los servidores autónomos" en la página 395 y "Inicio de servidores de contenedor" en la página 398 para obtener más información.

Qué hacer a continuación

- Desarrolle una aplicación Hibernate que utilice la configuración. Para obtener más información, consulte Ejemplo: Utilización del plug-in Hibernate para precargar datos en la memoria caché de ObjectGrid.
- En un entorno de producción, cree dominios de servicio de catálogo para los procesos creados automáticamente para la configuración EMBEDDED o EMBEDDED_PARTITION.
  - Entorno autónomo:

    Si no se ejecutan los servidores dentro de un proceso WebSphere Application Server, los hosts y los puertos del dominio de servicio de catálogo se especifican utilizando el archivo de propiedades denominado `objectGridServer.properties`. Este archivo se debe almacenar en la classpath de la aplicación y tiene definida la propiedad **catalogServiceEndPoints**. El dominio de servicio de catálogo se inicia independientemente de los procesos de la aplicación y se debe iniciar antes de que se inicien los procesos de la aplicación.

    el formato del archivo `objectGridServer.properties` es el siguiente:

    ```
    PuntosFinalesServicioCatálogos=<nombrehost>:<puerto1>,<nombrehost2>:<puerto2>
    ```
  - Entorno de WebSphere Application Server:

    Si se ejecuta dentro de un proceso WebSphere Application Server, el plug-in de memoria caché JPA se conecta automáticamente el servicio de catálogo o al dominio de servicio de catálogo definido para la célula de WebSphere Application Server.
- Cuando se utiliza un valor de ObjectGridType `EMBEDDED` o `EMBEDDED_PARTITION` en un entorno Java SE, utilice el método System.exit(0) al final del programa para detener el servidor eXtreme Scale incorporado. De lo contrario, el programa puede dejar de responder.

Ejemplo: Archivos XML de ObjectGrid Hibernate:

Los archivos XML Hibernate ObjectGrid deben crearse según la configuración de una unidad de persistencia.

Archivo persistence.xml

A continuación, aparece un archivo persistence.xml de ejemplo que representa la configuración de una unidad de persistencia:

```
<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>
```

Archivo hibernate-objectGridDeployment.xml

Utilice el archivo hibernate-objectGridDeployment.xml para personalizar opcionalmente la política de despliegue. Si proporciona este archivo en el directorio META-INF/hibernate-objectGridDeployment.xml, la política de despliegue predeterminada se sustituye por la configuración en este archivo.

```
<?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>
```

Archivo hibernate-objectGrid.xml

Si no utiliza Hibernate con JPA (Java Persistence API), utilice el siguiente hibernate-objectGrid.xml de ejemplo para crear la configuración Hibernate:

```
<?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>
```

Nota: Las correlaciones org.hibernate.cache.UpdateTimestampsCache, org.hibernate.cache.StandardQueryCache y defaultCacheMap son necesarias.

Configuración de la integración de base de datos

Puede utilizar WebSphere eXtreme Scale para disminuir la carga en bases de datos. Puede utilizar una Java Persistence API (JPA) entre WebSphere eXtreme Scale y la base de datos para integrar los cambios como un cargador.

Antes de empezar

Para obtener un resumen de las diversas topologías que puede crear con una base de datos, consulte “Integración de base de datos: almacenamiento en memoria caché de grabación diferida, en línea y complementaria” en la página 17.

Configuración de cargadores JPA

Un cargador Java Persistence API (JPA) es una implementación de plug-in que utiliza JPA para interactuar con la base de datos.

Antes de empezar

- Debe tener una implementación de JPA como, por ejemplo, Hibernate u OpenJPA.
- La base de datos puede ser cualquier programa de fondo soportado por el proveedor JPA elegido.
- Decida si va a utilizar el plug-in JPALoader o el plug-in JPAEntityLoader. Utilice el plug-in JPALoader cuando almacene los datos utilizando la API ObjectMap. Utilice el plug-in JPAEntityLoader cuando almacene los datos mediante la API EntityManager.

  Nota: Si está utilizando las API JPA para acceder al origen de datos JPA, utilice el plug-in de memoria caché L2 JPA. El plug-in de memoria caché presenta la cuadrícula de datos entre la aplicación y el origen de datos JPA, al mismo tiempo que sigue utilizando una aplicación JPA. Para obtener más información, consulte "Plug-in de memoria caché de nivel 2 (L2) JPA" en la página 331.

Acerca de esta tarea

Para obtener más información sobre cómo funciona Java Persistence API (JPA) Loader, consulte Cargadores JPA.

Procedimiento

1. Configure los parámetros necesarios que requiere JPA para interactuar con una base de datos.

   Los siguientes parámetros son necesarios. Estos parámetros se configuran en el bean JPALoader o JPAEntityLoader, y en el bean JPATxCallback.

   - **`persistenceUnitName`**: especifica el nombre de la unidad de persistencia. Este parámetro es necesario para dos propósitos: para crear una fábrica de JPA EntityManagerFactory, y para localizar los metadatos de la entidad JPA en el archivo `persistence.xml`. Este atributo se establece en el bean JPATxCallback.
   - **`JPAPropertyFactory`**: especifica la fábrica para crear una correlación de propiedad de persistencia para alterar temporalmente las propiedades de persistencia predeterminadas. Este atributo se establece en el bean JPATxCallback.Para establecer este atributo, es necesaria la configuración del estilo Spring.
   - **`entityClassName`**: especifica el nombre de la clase de entidad necesaria para utilizar los métodos JPA como, por ejemplo, EntityManager.persist, EntityManager.find, etc. El plug-in JPALoader requiere este parámetro, pero el parámetro es opcional para JPAEntityLoader. Para el plug-in JPAEntityLoader, si no se configura un parámetro **`entityClassName`**, se utiliza la clase de entidad configurada en la correlación de entidad de ObjectGrid. Debe utilizar la misma clase para eXtreme Scale EntityManager y el proveedor JPA. Este atributo se establece en el bean JPALoader o JPAEntityLoader.
   - **`preloadPartition`**: especifica la partición en la que se inicia la precarga de la correlación. Si la partición de la carga previa es menor que cero, o mayor que el número total de particiones menos 1, no se inicia la precarga. El valor

predeterminado es -1, que significa que la precarga no se inicia de forma predeterminada. Este atributo se establece en el bean JPALoader o JPAEntityLoader.

Aparte de los cuatro parámetros JPA que se configuran en eXtreme Scale, se utilizan metadatos JPA para recuperar la clave de las entidades JPA. Los metadatos JPA se pueden configurar como una anotación, o como un archivo `orm.xml` especificado en el archivo `persistence.xml`. No forman parte de la configuración de eXtreme Scale.

2. Configure los archivos XML para la configuración JPA.

   Para configurar un JPALoader o JPAEntityLoader, consulte Plug-ins para la comunicación con bases de datos.

   Configure una devolución de llamada de transacción JPATxCallback junto con la configuración del cargador. El ejemplo siguiente es un archivo de descriptor XML de ObjectGrid (`objectgrid.xml`), que tiene configurado un JPAEntityLoader y JPATxCallback:

   configuración de un cargador incluida la devolución de llamada - ejemplo XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
      <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
        <bean id="TransactionCallback"
           className="com.ibm.websphere.objectgrid.jpa.JPATxCallback">
           <property
              name="persistenceUnitName"
              type="java.lang.String"
              value="employeeEMPU" />
       </bean>
       <backingMap name="Employee" pluginCollectionRef="Employee" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
        <bean id="Loader"
          className="com.ibm.websphere.objectgrid.jpa.JPAEntityLoader">
       <property
               name="entityClassName"
               type="java.lang.String"
               value="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
        </bean>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

   Si desea configurar un JPAPropertyFactory, debe utilizar una configuración de estilo Spring. A continuación se muestra un ejemplo de archivo de configuración XML, `JPAEM_spring.xml`, que configura un bean Spring que se utilizará para configuraciones eXtreme Scale.

   configuración de un cargador incluida la fábrica de propiedades JPA - ejemplo XML

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">

  <objectgrid:JPAEntityLoader id="jpaLoader"
  entityClassName="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
    <objectgrid:JPATxCallback id="jpaTxCallback" persistenceUnitName="employeeEMPU" />
</beans>
```

Sigue el archivo XML de configuración de `Objectgrid.xml`. Tenga en cuenta que el nombre de ObjectGrid es `JPAEM`, que coincide con el nombre de ObjectGrid en el archivo de configuración de Spring `JPAEM_spring.xml`.

Configuración del cargador JPAEM - ejemplo de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
      <bean id="TransactionCallback"
            className="{spring}jpaTxCallback"/>
        <backingMap name="Employee" pluginCollectionRef="Employee"
                    writeBehind="T4"/>
      </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
       <bean id="Loader" className="{spring}jpaLoader" />
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Se puede anotar una entidad con las anotaciones JPA y, también, las anotaciones del gestor de entidades eXtreme Scale. Cada anotación tiene un XML equivalente que puede utilizarse. Por lo tanto, eXtreme Scale se añade al espacio de nombres de Spring. También puede configurarlas mediante el uso del soporte de espacio de nombres Spring. Para obtener más información, consulte Visión general de la infraestructura Spring.

Configuración de un actualizador de datos basado en la hora de JPA

Puede configurar una actualización de base de datos basada en tiempo utilizando un XML para una configuración de eXtreme Scale local o distribuida. También puede configurar una configuración local a través de programa.

Acerca de esta tarea

Si desea más información sobre cómo funciona el actualizador de datos basado en la hora de Java Persistence API (JPA), consulte Actualizador de datos basado en la hora de JPA.

Procedimiento

Cree una configuración de timeBasedDBUpdate.

- **Con un archivo XML:**

  El siguiente ejemplo muestra un archivo `objectgrid.xml` que contiene una configuración de timeBasedDBUpdate:

  actualizador basado en la hora JPA - ejemplo de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
 xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="changeOG"
            entityMetadataXMLFile="userEMD.xml">
            <backingMap name="user" >
                <timeBasedDBUpdate timestampField="rowChgTs"
                      persistenceUnitName="userderby"
                      entityClass="com.test.UserClass"
                      mode="INVALIDATE_ONLY"
                   />
            </backingMap>
        </objectGrid>
    </objectGrids>
    <backingMapPluginCollections>
</objectGridConfig>
```

En este ejemplo, la correlación "user" se configura con una actualización de base de datos basada en tiempo. La modalidad de actualización de base de datos es INVALIDATE_ONLY, y el campo de indicación de hora es rowChgTs.

Cuando se inicia el ObjectGrid distribuido "changeOG" en el servidor de contenedor, se inicia automáticamente la hebra de actualización de base de datos basada en la hora en la partición 0.

- **A través de programa:**

  Si crea un ObjectGrid local, también puede crear un objeto TimeBasedDBUpdateConfig y establecerlo en la instancia de BackingMap:

  ```
  public void setTimeBasedDBUpdateConfig(TimeBasedDBUpdateConfig dbUpdateConfig);
  ```

  Si desea más información sobre cómo establecer un objeto en la instancia de BackingMap, consulte la información sobre la interfaz BackingMap en la documentación de la API.

  De forma alternativa, puede anotar el campo de indicación de fecha y hora en la clase de entidad utilizando la anotación com.ibm.websphere.objectgrid.jpa.dbupdate.annotation.Timestamp. Al configurar el valor en la clase, no tendrá que configurar el timestampField en la configuración del XML.

Qué hacer a continuación

Inicie el actualizador de datos basado en la hora de JPA. Si desea más información, consulte Inicio del actualizador basado en la hora de JPA.

Configuración de servicios de datos REST

Puede utilizar el servicio de datos REST de WebSphere eXtreme Scale con WebSphere Application Server versión 7.0, WebSphere Application Server Community Edition y Apache Tomcat.

Acerca de esta tarea

El ejemplo incluido tiene código fuente y binarios compilados para ejecutar una cuadrícula de datos particionada. Este ejemplo demuestra cómo crear una cuadrícula de datos simple y modelar los datos utilizando entidades y proporciona dos aplicaciones cliente de línea de mandatos que permiten añadir y consultar entidades utilizando Java o C#.

El cliente Java de ejemplo utiliza la API EntityManager Java para persistir y consultar datos en la cuadrícula de datos. Este cliente se puede ejecutar en Eclipse o utilizando un script de línea de mandatos. Tenga en cuenta que el cliente Java de ejemplo no ofrece una demostración del servicio de datos REST, pero permite actualizar datos en la cuadrícula para que un navegador de web u otros clientes puedan leer los datos.

El cliente C# Microsoft WCF Data Services de ejemplo se comunica con la cuadrícula de datos de eXtreme Scale por medio del servicio de datos REST utilizando la infraestructura .NET. El cliente de WCF Data Services se puede utilizar para actualizar y consultar la cuadrícula de datos.

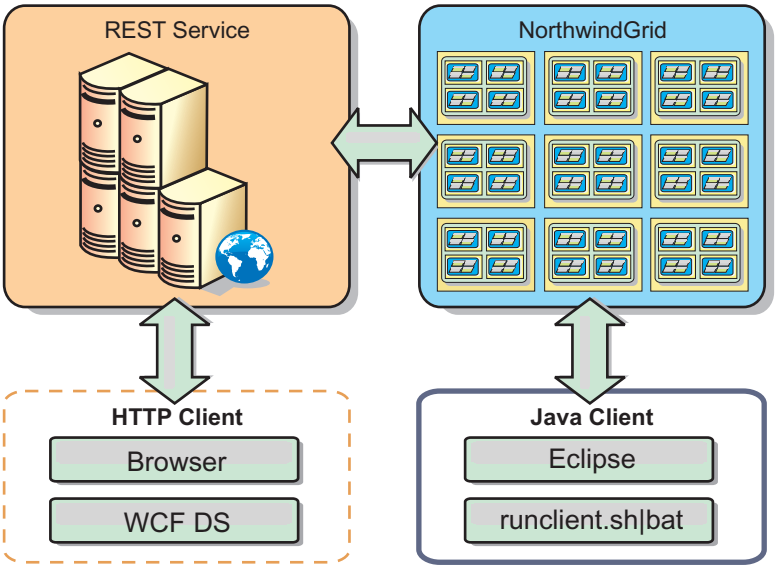


Figura 43. Iniciación a la topología de ejemplo. Los clientes HTTP que utilizan el servicio de datos REST y los clientes Java pueden acceder a la misma cuadrícula de datos.

Procedimiento

1. Configure e inicie la cuadrícula de datos de eXtreme Scale. Consulte "Habilitación del servicio de datos REST".
2. Configure e inicie el servicio de datos REST en un servidor web. Consulte "Configuración de servidores de aplicaciones para el servicio de datos REST" en la página 367.
3. Ejecute un cliente para que interactúe con el servicio de datos REST. Hay dos opciones disponibles:
   a. Ejecute el cliente Java de ejemplo para llenar la cuadrícula de datos utilizando la API EntityManager y consultar los datos de la cuadrícula utilizando un navegador web y el servicio de datos REST de eXtreme Scale. Consulte "Uso de un cliente Java con los servicios de datos REST" en la página 384.
   b. Ejecute el cliente C# de WCF Data Services. Consulte "Cliente Visual Studio 2008 WCF con servicio de datos REST" en la página 386.

Habilitación del servicio de datos REST

El servicio de datos REST representa metadatos de entidades de WebSphere eXtreme Scale para representar cada entidad como un EntitySet.

Inicio de una cuadrícula de datos de eXtreme Scale de ejemplo

En general, antes de iniciar el servicio de datos REST, inicie la cuadrícula de datos de eXtreme Scale. Los pasos siguientes iniciarán un solo proceso de servicio de catálogo de eXtreme Scale y dos procesos servidor de contenedor.

WebSphere eXtreme Scale se puede instalar utilizando tres métodos diferentes:

- Instalación de prueba
- Despliegue autónomo
- Despliegue integrado de WebSphere Application Server

Modelo de datos escalable de eXtreme Scale

El ejemplo Northwind de Microsoft utiliza la tabla Order Detail para establecer una asociación de muchos a muchos entre Orders y Products.

Las especificaciones de correlación de objeto a relacional (ORM) como ADO.NET Entity Framework y Java Persistence API (JPA) puede correlacionar las tablas y las relaciones utilizando entidades. No obstante, esta arquitectura no se escala. Todo debe encontrarse en la misma máquina, o en un clúster de máquinas caro, para que funcione bien.

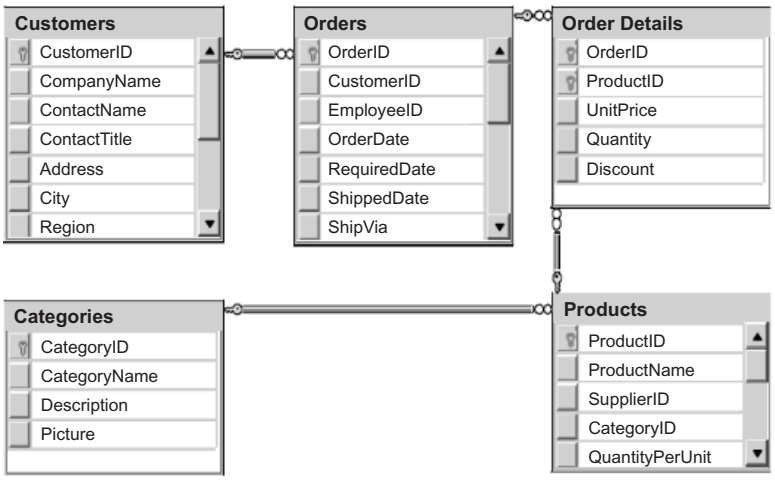


Figura 44. Diagrama del esquema de ejemplo Northwind de Microsoft SQL Server

A fin de crear una versión escalable del ejemplo, las entidades se deben modelar de modo que cada entidad o grupo de entidades relacionadas se pueda particionar basándose en una sola clave. Mediante la creación de particiones en una sola clave, las solicitudes se pueden distribuir entre varios servidores independientes. Para conseguir esta configuración, las entidades se han dividido en dos árboles: el árbol de clientes y pedidos (Customer and Order) y el árbol de productos y categorías (Product and Category). En este modelo, cada árbol se puede dividir independientemente y, por lo tanto, puede crecer a ritmos diferentes, con lo que aumenta la escalabilidad.

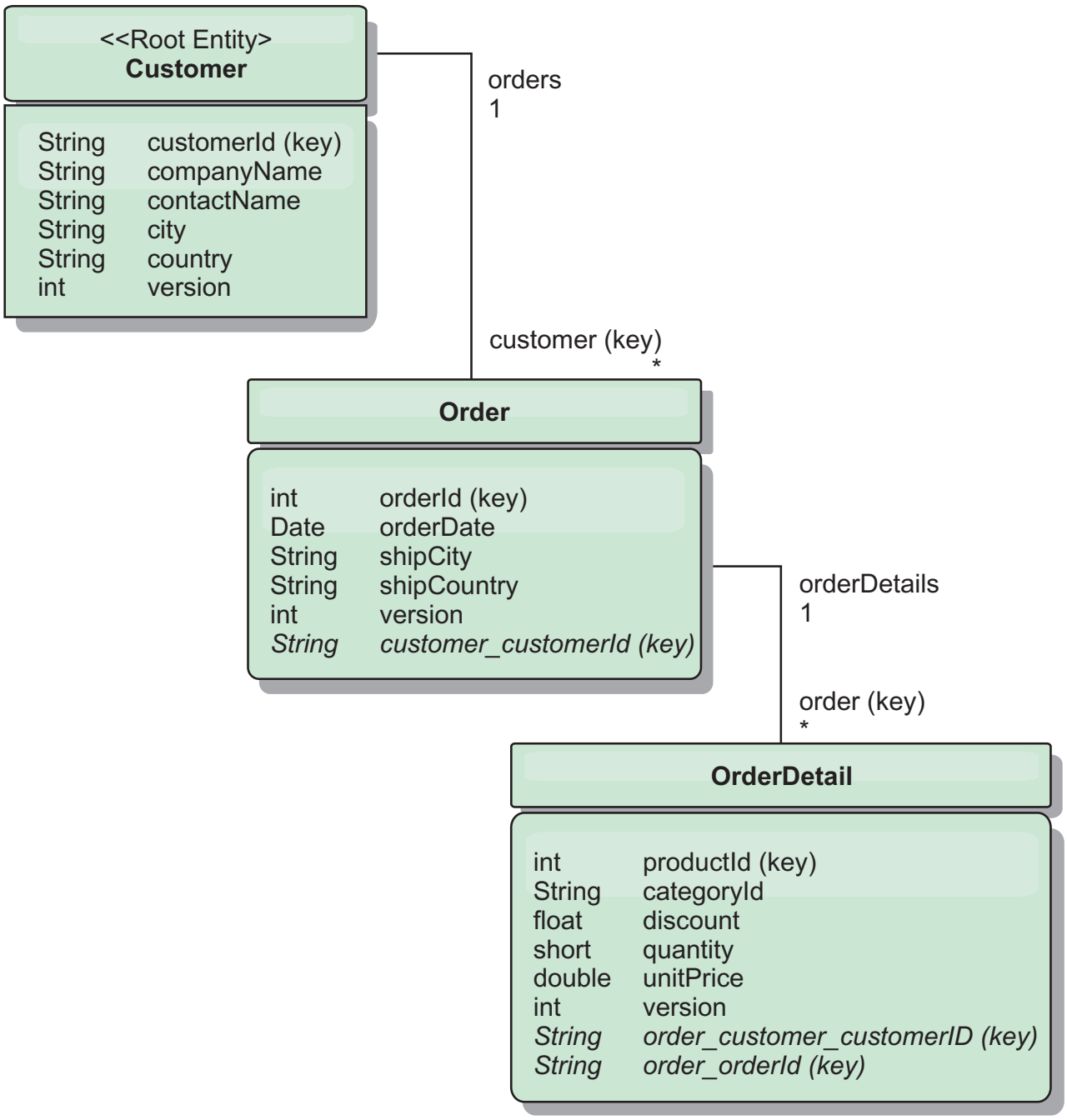


Figura 45. Diagrama del esquema de entidades Customer y Order

Por ejemplo, tanto Order como Product tienen enteros exclusivos distintos como claves. De hecho, las tablas Order y Product son realmente independientes entre sí. Por ejemplo, considere el efecto del tamaño de un catálogo, el número de productos que vende, con el número total de pedidos. Intuitivamente, podría parecer que tener muchos productos también implica tener muchos pedidos, pero esto no es necesariamente así. Si esto fuera verdad, podría aumentar fácilmente las ventas con el simple hecho de añadir más productos a su catálogo. Los pedidos y los productos tienen sus propias tablas independientes. Puede ampliar aún más este concepto de modo que los pedidos y los productos tengan sus propias cuadrículas de datos distintas. Con cuadrículas de datos independientes, puede controlar el número de particiones y servidores, además del tamaño de cada

cuadrícula por separado de modo que la aplicación se pueda escalar. Si dobla el tamaño de su catálogo, debe doblar la cuadrícula de datos de productos, pero la cuadrícula de pedidos puede permanecer inalterada. Y lo contrario es aplicable a una oleada de pedidos o a una oleada de pedidos prevista.

En el esquema, un cliente tiene cero o más pedidos y un pedido tiene elementos de línea (OrderDetail), cada uno con un producto específico. Un producto se identifica mediante el ID (la clave de producto) en cada OrderDetail. Una sola cuadrícula de datos almacena clientes, pedidos y detalles del pedido con Customer como entidad raíz de la cuadrícula de datos. Puede recuperar clientes por ID, pero deberá obtener los pedidos a partir del ID de cliente. Por lo tanto, el ID de cliente se añade al pedido como parte de su clave. Del mismo modo, el ID de cliente y el ID de pedido forman parte del ID de detalle del pedido (OrderDetail).

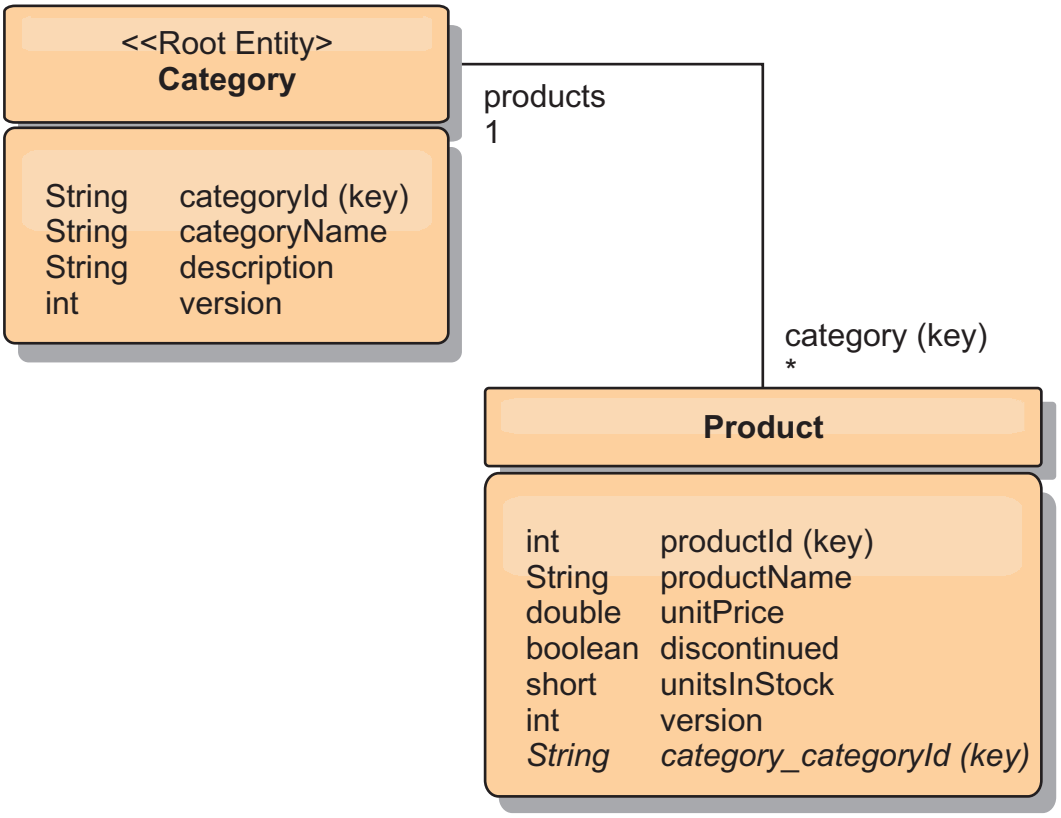


Figura 46. Diagrama del esquema de entidades Category y Product

En el esquema de categorías y productos, la categoría es la raíz del esquema. Con este esquema, los clientes pueden consultar productos por categoría. Consulte "Recuperación y actualización de datos con REST" para obtener detalles adicionales sobre las asociaciones de claves y su importancia.

Recuperación y actualización de datos con REST

El protocolo OData requiere que todas las entidades puedan ser dirigidas por su forma canónica. Esto significa que cada entidad debe incluir la clave de la entidad particionada, la entidad raíz y la raíz de esquema.

A continuación se ofrece un ejemplo de cómo utilizar la asociación desde una entidad raíz para dirigirse a un hijo en :

```
/Customer('ACME')/order(100)
```

En WCF Data Services, la entidad hijo debe ser directamente direccionable, lo que significa que la clave en la raíz de esquema debe formar parte de la clave del hijo: `/Order(customer_customerId='ACME', orderId=100)`. Esto se consigue creando una asociación con la entidad raíz donde la asociación de uno a uno o de muchos a uno con la entidad raíz también se identifica como una clave. Cuando las entidades se incluyen como parte de la clave, los atributos de la entidad padre se exponen como propiedades clave.

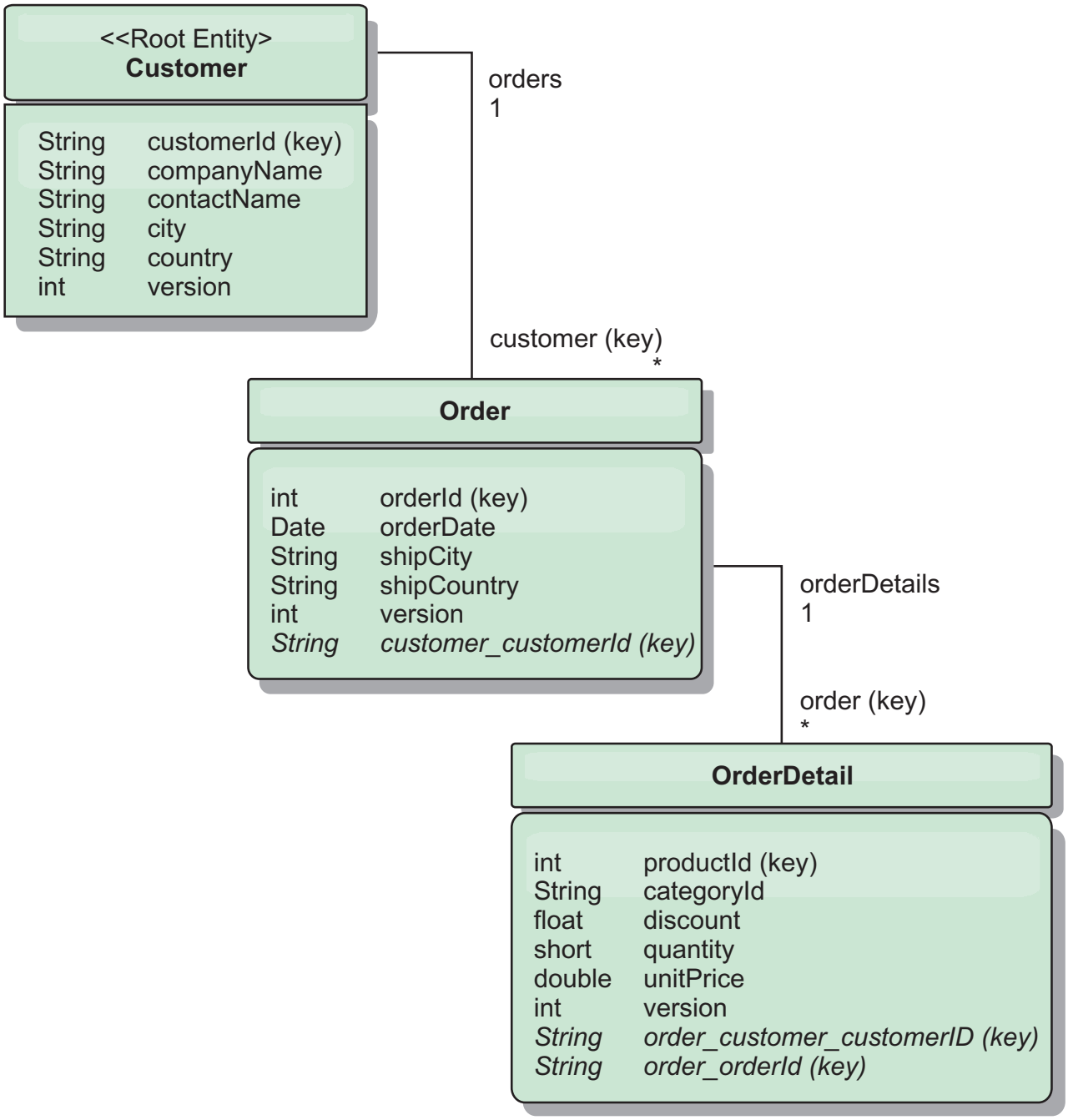


Figura 47. Diagrama del esquema de entidades Customer y Order

El diagrama de esquema de entidad Customer/Order ilustra cómo se particiona cada entidad utilizando Customer. La entidad Order incluye Customer como parte de su clave y, por lo tanto, se puede acceder a ella directamente. El servicio de

datos REST expone todas las asociaciones de clave como propiedades individuales: Order tiene customer_customerId y OrderDetail tiene order_customer_customerId y order_orderId.

Utilizando la API EntityManager, puede encontrar el pedido (Order) utilizando el cliente (Customer) y el ID de cliente:

```
transaction.begin();
// Buscar el pedido mediante el cliente. Sólo incluimos el ID
// en la clase Customer al crear la instancia de clave OrderId.
Order order = (Order) em.find(Order.class,
    new OrderId(100, new Customer('ACME')));
...
transaction.commit();
```

Al utilizar el servicio de datos REST, el pedido se puede recuperar con cualquiera de los URL:

- `/Order(orderId=100, customer_customerId='ACME')`
- `/Customer('ACME')/orders?$filter=orderId eq 100`

Para dirigirse a la clave de cliente se utiliza el nombre de atributo de la entidad Customer, un carácter de subrayado y el nombre de atributo del ID de cliente: customer_customerId.

Una entidad también puede incluir una entidad no raíz como componente de su clave si todos los ancestros de la entidad no raíz tienen asociaciones de clave con la raíz. En este ejemplo, OrderDetail tiene una asociación de clave con Order y Order tiene una asociación de clave con la entidad Customer raíz. Utilizando la API EntityManager:

```
transaction.begin();
// Construir una instancia de clave OrderDetailId. Ésta incluye
// Order y Customer con sólo el conjunto de claves.
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);
...
```

El servicio de datos REST permite dirigirse a OrderDetail directamente:

```
/OrderDetail(productId=500, order_customer_customerId='ACME', order_orderId =100)
```

La asociación de la entidad OrderDetail a la entidad Product se ha roto para permitir particionar el inventario de pedidos (Order) y productos (Product) independientemente. La entidad OrderDetail almacena la categoría y el ID de producto en lugar de una relación estricta. Al desacoplar los dos esquemas de entidad, sólo se accede a una partición a la vez.

El esquema de entidad Category y Product, ilustrado en el diagrama, muestra que la entidad raíz es Category y que cada entidad Product tiene una asociación con una entidad Category. La entidad Category se incluye en la identidad Product. El servicio de datos REST expone una propiedad clave: category_categoryId que permite dirigirse directamente a Product.

Puesto que Category es la entidad raíz, en un entorno particionado, se debe conocer la categoría para encontrar el producto. Utilizando la API EntityManager, la transacción se debe fijar a la entidad Category antes de buscar la entidad Product.

Utilizando la API EntityManager:

```
transaction.begin();
// Crear la entidad raíz Category con sólo la clave. Esto nos
// permite construir un ProductId sin necesidad de encontrar
// la entidad Category en primer lugar. La transacción queda
// fijada a la partición donde se almacena Category "COMP".
Category cat = new Category("COMP");
Product product = (Product) em.find(Product.class,
    new ProductId(500, cat));
...
```

El servicio de datos REST permite dirigirse a Product directamente:

```
/Product(productId=500, category_categoryId='COMP')
```

Inicio de una cuadrícula de datos autónoma para servicios de datos REST

Siga estos pasos para iniciar una cuadrícula de datos de ejemplo de servicio REST de WebSphere eXtreme Scale para un despliegue autónomo de eXtreme Scale.

Antes de empezar

Instale el producto completo o de prueba de WebSphere eXtreme Scale:

- Instale la versión autónoma del producto y aplique los arreglos subsiguientes, si los hay.
- Descargue y extraiga el producto de prueba WebSphere eXtreme Scale Versión 7.1 o posterior, que incluye el servicio de datos REST de WebSphere eXtreme Scale.

Acerca de esta tarea

Inicie la cuadrícula de datos de ejemplo de WebSphere eXtreme Scale.

Procedimiento

1. Inicie el proceso del servicio de catálogo. Abra una línea de mandatos o una ventana de terminal y defina la variable de entorno JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=inicio_java`
   - Windows `set JAVA_HOME=inicio_java`
2. `cd restservice_home/gettingstarted`
3. Inicie el proceso del servicio de catálogo. Para iniciar el servicio *sin* la seguridad de eXtreme Scale, utilice los mandatos siguientes.
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Para iniciar el servicio con la seguridad de eXtreme Scale, utilice los mandatos siguientes.
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
4. Inicie dos procesos de servidor de contenedores. Abra otra línea de mandatos o una ventana de terminal y defina la variable de entorno JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=inicio_java`
   - Windows `set JAVA_HOME=inicio_java`
5. `cd restservice_home/gettingstarted`

6. Inicie un proceso de servidor de contenedor:

   Para iniciar el servidor sin la seguridad de eXtreme Scale, utilice los mandatos siguientes:

   - Linux UNIX `./runcontainer.sh container0`
   - Windows `runcontainer.bat container0`

   Para iniciar el servidor con la seguridad de eXtreme Scale, utilice los mandatos siguientes.

   - Linux UNIX `./runcontainer_secure.sh container0`
   - Windows `runcontainer_secure.bat container0`
7. Abra otra línea de mandatos o una ventana de terminal y defina la variable de entorno JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=`*`inicio_java`*
   - Windows `set JAVA_HOME=`*`inicio_java`*
8. `cd restservice_home/gettingstarted`
9. Inicie un segundo proceso de servidor de contenedor.

   Para iniciar el servidor sin la seguridad de eXtreme Scale, utilice los mandatos siguientes.

   - Linux UNIX `./runcontainer.sh container1`
   - Windows `runcontainer.bat container1`

   Para iniciar el servidor con la seguridad de eXtreme Scale, utilice los mandatos siguientes.

   - Linux UNIX `./runcontainer_secure.sh container1`
   - Windows `runcontainer_secure.bat container1`

Resultados

Espere hasta que los contenedores de eXtreme Scale estén preparados antes de continuar con los siguientes pasos. Los servidores de contenedor están preparados cuando aparece el mensaje siguiente en la ventana terminal:

`CWOBJ1001I: ObjectGrid Server `*`nombre_contenedor`*` está listo para procesar peticiones.`

Donde *nombre_contenedor* es el nombre del contenedor que se ha iniciado.

Inicio de una cuadrícula de datos para servicios de datos REST en WebSphere Application Server

Siga estos pasos para iniciar una cuadrícula de datos de ejemplo de servicio REST de WebSphere eXtreme Scale autónomo para un despliegue de WebSphere eXtreme Scale integrado con WebSphere Application Server. Aunque WebSphere eXtreme Scale está integrado con WebSphere Application Server, estos pasos inician un contenedor y un proceso de servicio de catálogo de WebSphere eXtreme Scale.

Antes de empezar

Instale el producto en un directorio de instalación de WebSphere Application Server Versión 7.0.0.5 o posterior con la seguridad habilitada. Aumente como mínimo un perfil de servidor de aplicación.

Acerca de esta tarea

Inicie la cuadrícula de datos de ejemplo de WebSphere eXtreme Scale.

Procedimiento

1. Inicie el proceso del servicio de catálogo. Abra una línea de mandatos o una ventana de terminal y defina la variable de entorno JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=inicio_java`
   - Windows `set JAVA_HOME=inicio_java`

   `cd restservice_home/gettingstarted`
2. Inicie el proceso del servicio de catálogo.

   Para iniciar el servidor sin la seguridad de eXtreme Scale, utilice los mandatos siguientes.
   - Linux UNIX `./runcat.sh`
   - Windows `runcat.bat`

   Para iniciar el servidor con la seguridad de eXtreme Scale, utilice los mandatos siguientes.
   - Linux UNIX `./runcat_secure.sh`
   - Windows `runcat_secure.bat`
3. Inicie dos procesos de servidor de contenedores. Abra otra línea de mandatos o una ventana de terminal y defina la variable de entorno JAVA_HOME:
   - Linux UNIX `export JAVA_HOME=inicio_java`
   - Windows `set JAVA_HOME=inicio_java`
4. Inicie un proceso de servidor de contenedor.

   Para iniciar el servidor sin la seguridad de eXtreme Scale, utilice los mandatos siguientes.
   a. Abra una ventana de línea de mandatos.
   b. `cd restservice_home/gettingstarted`
   c. Para iniciar el servidor *sin* la seguridad de eXtreme Scale, utilice los mandatos siguientes.
      - Linux UNIX `./runcontainer.sh container0`
      - Windows `runcontainer.bat container0`
   d. Para iniciar el servidor con la seguridad de eXtreme Scale, utilice los mandatos siguientes.
      - Linux UNIX `./runcontainer_secure.sh container0`
      - Windows `runcontainer_secure.bat container0`
5. Inicie un segundo proceso de servidor de contenedor.
   a. Abra una ventana de línea de mandatos.
   b. `cd restservice_home/gettingstarted`
   c. Para iniciar el servidor *sin* la seguridad de eXtreme Scale, utilice los mandatos siguientes.
      - Linux UNIX `./runcontainer.sh container1`
      - Windows `runcontainer.bat container1`

d. Para iniciar el servidor *con* la seguridad de eXtreme Scale, utilice los mandatos siguientes.

- Linux UNIX `./runcontainer_secure.sh container1`
- Windows `runcontainer_secure.bat container1`

Resultados

Espere hasta que los servidores de contenedor estén preparados antes de continuar con los siguientes pasos. Los servidores de contenedor están preparados cuando aparece el mensaje siguiente:

`CWOBJ1001I: ObjectGrid Server` *`nombre_contenedor`* `está listo para procesar peticiones.`

Donde *nombre_contenedor* es el nombre del contenedor que se inició en el paso anterior.

Configuración de servidores de aplicaciones para el servicio de datos REST

Puede configurar varios servidores de aplicaciones para utilizar el servicio de datos REST.

Despliegue del servicio de datos REST en WebSphere Application Server

Este tema describe cómo configurar el servicio de datos REST de WebSphere eXtreme Scale en WebSphere Application Server o WebSphere Application Server Network Deployment Versión 6.1.0.25 o posterior. Estas instrucciones se aplican también a despliegues donde WebSphere eXtreme Scale se integre con el despliegue de WebSphere Application Server.

Antes de empezar

Debe tener uno de los entornos siguientes en el sistema para configurar y desplegar el servicio de datos REST para WebSphere eXtreme Scale.

- WebSphere Application Server con el cliente autónomo de WebSphere eXtreme Scale:
  - El producto de prueba WebSphere eXtreme Scale Versión 7.1 con el servicio de datos REST se descarga y extrae o se instala el producto WebSphere eXtreme Scale Versión 7.1.0.0 con el arreglo acumulativo 2 en un directorio autónomo.
  - WebSphere Application Server Versión 6.1.0.25 o 7.0.0.5 o posterior está instalado y en ejecución.
- WebSphere Application Server integrado con WebSphere eXtreme Scale:

  WebSphere eXtreme Scale Versión 7.1.0.0 con arreglo acumulativo 2 o posterior se instala sobre WebSphere Application Server Versión 6.1.0.25 o 7.0 o posterior.

  Consejo: El servicio de datos REST de WebSphere eXtreme Scale sólo requiere que la opción de cliente de WebSphere eXtreme Scale esté instalada. No es necesario aumentar el perfil.

  Lea sobre cómo habilitar la seguridad Java 2 en el Information Center de WebSphere Application Server.

Procedimiento

1. Configure e inicie una cuadrícula de datos.
   a. Para obtener información sobre cómo configurar una cuadrícula de datos para su uso con el servicio de datos REST, consulte "Inicio de una cuadrícula de datos para servicios de datos REST en WebSphere Application Server" en la página 365.
   b. Compruebe que un cliente se pueda conectar y acceder a las entidades de la cuadrícula de datos. Si desea ver un ejemplo, consulte "Guía de aprendizaje: Cómo empezar con WebSphere eXtreme Scale" en la página 1.
2. Cree el archivo JAR o el directorio de configuración del servicio REST de eXtreme Scale. Consulte la información sobre cómo empaquetar y desplegar el servicio REST en "Instalación del servicio de datos REST" en la página 200.
3. Añada el archivo JAR o el directorio de configuración del servicio de datos REST a la vía de acceso de clases del servidor de aplicaciones:
   a. Abra la consola administrativa de WebSphere Application Server.
   b. Vaya a **Entorno** > **Bibliotecas compartidas**
   c. Pulse **Nuevo**
   d. Añada las entradas siguientes a los campos apropiados:
      - Nombre: `extremescale_rest _configuration`
      - Classpath: <jar o directorio de configuración del servicio REST>
   e. Pulse **Aceptar**
   f. Guarde los cambios en la configuración maestra
4. Añada el archivo JAR del tiempo de ejecución del cliente de WebSphere eXtreme Scale, `wsogclient.jar`, y el archivo JAR o el directorio de la configuración del servicio de datos REST a la classpath del servidor de aplicaciones. Este paso no es necesario si WebSphere eXtreme Scale está integrado con la instalación de WebSphere Application Server.
   a. Abra la consola administrativa de WebSphere Application Server.
   b. Vaya a **Entorno** > **Bibliotecas compartidas**.
   c. Pulse **Nueva**.
   d. Añada las entradas siguientes a los campos:
      - Nombre: `extremescale_client_v71`
      - Classpath: `inicio_wxs/lib/wsogclient.jar`

      Recuerde: Añada cada vía de acceso a una línea distinta.
   e. Pulse **Aceptar**.
   f. Guarde los cambios en la configuración maestra.
5. Instale el archivo EAR del servicio de datos REST, `wxsrestservice.ear`, en WebSphere Application Server con la consola administrativa:
   a. Abra la consola administrativa de WebSphere Application Server.
   b. Pulse **Aplicaciones** > **Nueva aplicación**.
   c. Vaya al archivo `/lib/wxsrestservice.ear` en el sistema de archivos, selecciónelo y pulse **Siguiente**.
      - Si utiliza WebSphere Application Server Versión 7.0, pulse Siguiente.
      - Si utiliza WebSphere Application Server Versión 6.1, especifique un valor de raíz de contexto con el nombre: `/wxsrestservice` y continúe al paso siguiente.
   d. Elija la opción de instalación detallada y pulse **Siguiente**.

e. En la pantalla de avisos de seguridad de la aplicación, pulse **Continuar**.
f. Elija las opciones de instalación predeterminadas y pulse **Siguiente**.
g. Elija un servidor al que correlacionar la aplicación y pulse **Siguiente**.
h. En la página de recarga de JSP, utilice los valores predeterminados y pulse **Siguiente**.
i. En la página de bibliotecas compartidas, correlacione el módulo `wxsrestservice.war` con las bibliotecas compartidas que ha definido:
   - `extremescale_rest_configuration`
   - `extremescale_client_v71`

   Consejo: Esta biblioteca compartida solo es necesaria si WebSphere eXtreme Scale no está integrado con WebSphere Application Server.
j. En la página de correlación de la relación de bibliotecas compartidas, utilice los valores predeterminados y pulse **Siguiente**.
k. En la página de correlación de sistemas principales virtuales, utilice los valores predeterminados y pulse **Siguiente**.
l. En la página de correlación de raíces de contexto, defina wxsrestservice como raíz de contexto
m. En la pantalla Resumen, pulse **Finalizar** para completar la instalación.
n. Guarde los cambios en la configuración maestra.

6. Inicie la aplicación de servicio de datos REST `wxsrestservice`:
   a. Vaya a la aplicación en la consola administrativa.
      - WebSphere Application Server Versión 7.0: en la consola administrativa, pulse **Aplicaciones** > **Tipos de aplicación** > **Aplicaciones WebSphere**.
      - WebSphere Application Server Versión 6.1: en la consola administrativa, pulse **Aplicaciones** > **Aplicaciones empresariales**.
   b. Marque el recuadro de selección junto a la aplicación `wxsrestservice` y pulse **Iniciar**.
   c. Revise el archivo `SystemOut.log` para el perfil del servidor de aplicaciones. Cuando el servicio de datos REST se ha iniciado satisfactoriamente, se visualiza el mensaje siguiente en el archivo `SystemOut.log` para el perfil del servidor:

      ```
      CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.
      ```
7. Verifique que el servicio de datos REST funcione: El número de puerto se puede encontrar en el archivo `SystemOut.log` en el directorio de registros del perfil del servidor de aplicaciones mirando el primer puerto visualizado para el identificador de mensaje: SRVE0250I. El puerto predeterminado es 9080.

   Por ejemplo:`http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/` Resultado: Se visualiza el documento del servicio AtomPub.

   Por ejemplo: `http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata`. Aparecerá el documento de extensiones de datos del modelo de entidad (EDMX).
8. Para detener los procesos de la cuadrícula de datos, utilice `CTRL+C` en la ventana de mandatos correspondiente.

Inicio del servicio de datos REST con WebSphere eXtreme Scale integrado en WebSphere Application Server 7.0:

Este tema describe cómo configurar e iniciar el servicio de datos REST de eXtreme Scale utilizando WebSphere Application Server version 7.0 integrado y aumentado con WebSphere eXtreme Scale.

Antes de empezar

Compruebe que la cuadrícula de datos autónoma de eXtreme Scale se haya iniciado. Consulte "Habilitación del servicio de datos REST" en la página 358 para obtener información detallada sobre cómo iniciar la cuadrícula de datos.

Acerca de esta tarea

Para empezar a utilizar el servicio de datos REST de WebSphere eXtreme Scale utilizando WebSphere Application Server, siga los pasos que se indican a continuación:

Procedimiento

1. Añada el JAR de configuración del ejemplo de servicio de datos REST de WebSphere eXtreme Scale a la vía de acceso de clases:
   a. Abra la consola de administración de WebSphere
   b. Vaya a Entorno -> Bibliotecas compartidas
   c. Pulse Nuevo
   d. Añada las entradas siguientes a los campos apropiados:
      1) Nombre: extremescale_gettingstarted_config
      2) Classpath
         - `inicio_restservice/gettingstarted/restclient/bin`
         - `inicio_restservice/gettingstarted/common/bin`

         Recuerde: Cada vía de acceso debe aparecer en una línea distinta.
   e. Pulse **Aceptar**
   f. Guarde los cambios en la configuración maestra
2. Instale el archivo EAR del servicio de datos REST, wxsrestservice.ear, en WebSphere Application Server utilizando la consola de administración de WebSphere:
   a. Abra la consola de administración de WebSphere
   b. Vaya a Aplicaciones -> Nueva aplicación
   c. Vaya al archivo `inicio_restservice/lib/wxsrestservice.ear` en el sistema de archivos. Seleccione el archivo y pulse **Siguiente**.
   d. Elija las opciones de instalación detalladas y pulse **Siguiente**.
   e. En la pantalla de avisos de seguridad de la aplicación, pulse **Continuar**.
   f. Elija las opciones de instalación predeterminadas y pulse **Siguiente**.
   g. Seleccione un servidor con el que correlacionar el módulo wxsrestservice.war y pulse **Siguiente**.
   h. En la página de recarga de JSP, utilice los valores predeterminados y pulse **Siguiente**.
   i. En la página de bibliotecas compartidas, correlacione el módulo "wxsrestservice.war" con las bibliotecas compartidas siguientes, definidas durante el paso 1: `extremescale_ gettingstarted _config`
   j. En la página de correlación de la relación de bibliotecas compartidas, utilice los valores predeterminados y pulse **Siguiente**.

k. En la página de correlación de sistemas principales virtuales, utilice los valores predeterminados y pulse **Siguiente**.

l. En la página de correlación de raíces de contexto, defina wxsrestservice como raíz de contexto.

m. En la pantalla Resumen, pulse **Finalizar** para completar la instalación.

n. Guarde los cambios en la configuración maestra.

3. Si la cuadrícula de datos de eXtreme Scale se ha iniciado con la seguridad de eXtreme Scale habilitada, establezca la propiedad siguiente en el archivo `inicio_restservice/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=inicio_restservice/gettingstarted/security/security.ogclient.properties
```

4. Inicie el servidor de aplicaciones y la aplicación de servicio de datos REST de eXtreme Scale "wxsrestservice".

   Después de que la aplicación se haya iniciado, consulte el archivo SystemOut.log correspondiente al servidor de aplicaciones y verifique que aparezca el mensaje siguiente: `CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.`

5. Verifique que el servicio de datos REST funcione

   a. Abra un navegador y acceda a:

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid

      Aparecerá el documento de servicio correspondiente a NorthwindGrid.

   b. Vaya a:

      http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata

      Aparecerá el documento de extensiones de datos del modelo de entidad (EDMX).

6. Para detener los procesos de cuadrícula de datos, utilice CTRL+C en la ventana de mandatos respectiva para detener los procesos.

Despliegue del servicio de datos REST en WebSphere Application Server Community Edition

Puede configurar los servicios de datos REST de eXtreme Scale en WebSphere Application Server Community Edition Versión 2.1.1.3 o posterior.

Antes de empezar

- Debe haber un JRE o JDK IBM (recomendado) o Sun, Versión 5 o posterior instalado y la variable de entorno JAVA_HOME debe estar definida.
- Descargue e instale WebSphere Application Server Community Edition Versión 2.1.1.3 o posterior en el directorio `raíz_wasce`, por ejemplo, el directorio `/opt/IBM/wasce`. Lea las instrucciones de instalación para obtener información sobre laversión 2.1.1 o sobre otras versiones.

Procedimiento

1. Configure e inicie una cuadrícula de datos.

   a. Para obtener más información sobre cómo configurar una cuadrícula de datos de eXtreme Scale para su uso con el servicio de datos REST, lea sobre "Inicio de una cuadrícula de datos autónoma para servicios de datos REST" en la página 364.

b. Compruebe que un cliente de eXtreme Scale se pueda conectar y acceder a las entidades de la cuadrícula de datos. Si desea ver un ejemplo, consulte "Guía de aprendizaje: Cómo empezar con WebSphere eXtreme Scale" en la página 1.

2. Cree el archivo JAR o el directorio de configuración del servicio REST de eXtreme Scale. Consulte la información sobre empaquetamiento y despliegue en el tema "Instalación del servicio de datos REST" en la página 200 para obtener los detalles.
3. Inicie el servidor WebSphere Application Server Community Edition:
   a. Para iniciar el servidor sin la seguridad Java SE habilitada, ejecute el mandato siguiente:

      UNIX Linux `raíz_wasce/bin/startup.sh`

      Windows `raíz_wasce/bin/startup.bat`

   b. Para iniciar el servidor con la seguridad Java SE habilitada, siga los pasos que se indican a continuación: UNIX Linux
      1) Abra una ventana de línea de mandatos o de terminal y ejecute el mandato de copia siguiente (o copie el contenido del archivo de política especificado en la política existente): `cp inicio_restservice/gettingstarted/wasce/geronimo.policy raíz_wasce/bin`
      2) Edite el archivo `raíz_wasce/bin/setenv.sh`
      3) Después de la línea que contiene `"WASCE_JAVA_HOME="`, añada lo siguiente:
      ```
      export JAVA_OPTS="-Djava.security.manager
      -Djava.security.policy=geronimo.policy"
      ```

      Windows
      1) Abra una ventana de línea de mandatos y ejecute el siguiente mandato copy o bien copie el contenido del archivo de política especificado en la política existente:
      ```
      copy inicio_restservice\gettingstarted\wasce\geronimo.policy\bin
      ```
      2) Edite el archivo `raíz_wasce\bin\setenv.bat`
      3) Después de la línea que contiene "set WASCE_JAVA_HOME=", añada lo siguiente:
      ```
      set JAVA_OPTS="-Djava.security.manager
      -Djava.security.policy=geronimo.policy"
      ```
4. Añada el JAR de tiempo de ejecución del cliente ObjectGrid al depósito de WebSphere Application Server Community Edition:
   a. Abra la consola de administración de WebSphere Application Server Community Edition e inicie sesión. El URL predeterminado es: `http://localhost:8080/console`, el ID de usuario predeterminado es `system` y la contraseña es `manager`.
   b. Pulse el enlace **Depósito** del lado izquierdo de la ventana de la consola, en la carpeta **Servicios**.
   c. En la sección **Añadir archivo al depósito**, introduzca los datos siguientes en los recuadros de texto de entrada:

Tabla 24. Añadir archivo al depósito

Recuadro de texto	Valor
Archivo	`wxs_home/lib/ogclient.jar`
Grupo	com.ibm.websphere.xs
Artefacto	ogclient

Tabla 24. Añadir archivo al depósito (continuación)

Recuadro de texto	Valor
Versión	7.1
Tipo	JAR

d. Pulse el botón Instalar

Consulte la nota técnica siguiente para obtener información detallada sobre los distintos métodos que se pueden utilizar para configurar las dependencias de clase y biblioteca: Specifying external dependencies to applications running on WebSphere Application Server Community Edition (Especificación de dependencias externas con aplicaciones que se ejecutan en WebSphere Application Server Community Edition).

5. Despliegue el módulo del servicio de datos REST, el archivo `wxsrestservice.war`, en el servidor WebSphere Application Server Community Edition.
   a. Copie y edite el archivo XML de plan de despliegue de ejemplo: `inicio_restservice/gettingstarted/wasce/geronimo-web.xml` para incluir las dependencias de vía de acceso del JAR o directorio de configuración del servicio de datos REST. Consulte la sección para ver un ejemplo de cómo definir la vía de acceso de clases para incluir el archivo `wxsRestService.properties` y otros archivos de configuración y clases de metadatos.
   b. Abra la consola de administración de WebSphere Application Server Community Edition e inicie sesión.

      Consejo: El URL predeterminado es: `http://localhost:8080/console`. El ID de usuario predeterminado es `system` y la contraseña es `manager`.
   c. Pulse el enlace **Desplegar nuevas** del lado izquierdo de la ventana de la consola.
   d. En la página **Instalar aplicaciones nuevas**, escriba los valores siguientes en los recuadros de texto:

Tabla 25. Instalar aplicaciones nuevas

Recuadro de texto	Valor
Archivo	`inicio_restservice/lib/wxsrestservice.war`
Plan	`inicio_restservice/gettingstarted/wasce/geronimo-web.xml`

      Consejo: Utilice la vía de acceso al archivo `geronimo-web.xml` que ha copiado y editado en el paso 3.
   e. Pulse el botón Instalar. La página de la consola indica entonces que la aplicación se ha instalado e iniciado satisfactoriamente.
   f. Examine la consola o el registro de salida del sistema WebSphere Application Server Community Edition para comprobar que el servicio de datos REST se haya iniciado satisfactoriamente. Debe aparecer el mensaje siguiente:

      ```
      CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere
      eXtreme Scale.
      ```
6. Inicie el servidor WebSphere Application Server Community Edition ejecutando el mandato siguiente:

- UNIX Linux `raíz_wasce/bin/startup.sh`
- Windows `raíz_wasce/bin/startup.bat`

7. Instale el servicio de datos REST de eXtreme Scale y el ejemplo proporcionado en el servidor WebSphere Application Server Community Edition:
   a. Añada el JAR de tiempo de ejecución del cliente ObjectGrid al depósito de WebSphere Application Server Community Edition:
      1) Abra la consola de administración de WebSphere Application Server Community Edition e inicie sesión. El URL predeterminado es: `http://localhost:8080/console`. El ID de usuario predeterminado es `system` y la contraseña es `manager`.
      2) Pulse el enlace **Depósito** del lado izquierdo de la ventana de la consola, en la carpeta Servicios.
      3) En la sección **Añadir archivo al depósito**, introduzca los datos siguientes en los recuadros de texto de entrada:

Tabla 26. Añadir archivo al depósito

Recuadro de texto	Valor
Archivo	wxs_home/lib/ogclient.jar
Grupo	com.ibm.websphere.xs
Artefacto	ogclient
Versión	7.1
Tipo	JAR

      4) Pulse el botón Instalar.

         Consejo: Consulte la nota técnica siguiente para obtener información detallada sobre los distintos métodos que se pueden utilizar para configurar las dependencias de clase y biblioteca: Specifying external dependencies to applications running on WebSphere Application Server Community Edition (Especificación de dependencias externas con aplicaciones que se ejecutan en WebSphere Application Server Community Edition).
   b. Despliegue el módulo del servicio de datos REST: `wxsrestservice.war` en el servidor WebSphere Application Server Community Edition.
      1) Edite el archivo XML de despliegue de ejemplo `restservice_home/gettingstarted/wasce/geronimo-web.xml` para incluir las dependencias de vía de acceso a los directorios de vía de acceso de clases del ejemplo de iniciación:
         - Cambie los "classesDirs" para los dos GBeans del cliente de iniciación:

         La vía de acceso "classesDirs" para el GBean GettingStarted_Client_SharedLib se debe definir como: `inicio_restservice/gettingstarted/restclient/bin`

         La vía de acceso "classesDirs" para el GBean GettingStarted_Common_SharedLib se debe definir como: `inicio_restservice/gettingstarted/common/bin`
      2) Abra la consola de administración de WebSphere Application Server Community Edition e inicie sesión.
      3) Pulse el enlace **Desplegar nuevas** del lado izquierdo de la ventana de la consola.

4) En la página **Instalar aplicaciones nuevas**, escriba los valores siguientes en los recuadros de texto:

Tabla 27. Instalar aplicaciones nuevas

Recuadro de texto	Valor
Archivo	*inicio_restservice*/lib/wxsrestservice.war
Plan	*inicio_restservice*/gettingstarted/wasce/geronimo-web.xml

5) Pulse el botón **Instalar**.

La página de la consola indica entonces que la aplicación se ha instalado e iniciado satisfactoriamente.

6) Examine el registro de salida del sistema WebSphere Application Server Community Edition para verificar que el servicio de datos REST se ha iniciado satisfactoriamente comprobando si el mensaje siguiente está presente:

```
CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.
```

8. Verifique que el servicio de datos REST funcione

Abra un navegador web y acceda al URL siguiente: http://<*host*>:<*puerto*>/<*raíz de contexto*>/restservice/<nombre de cuadrícula>

El puerto predeterminado para WebSphere Application Server Community Edition es 8080 y se define utilizando la propiedad "HTTPPort" en el archivo /var/config/config-substitutions.properties.

Por ejemplo: http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/

Resultados

Aparecerá el documento de servicio de AtomPub.

Inicio del servicio de datos REST en WebSphere Application Server Community Edition:

Este tema describe cómo configurar e iniciar el servicio de datos REST de eXtreme Scale utilizando WebSphere Application Server Community Edition.

Antes de empezar

Verifique que la cuadrícula de datos de ejemplo se haya iniciado. Consulte el apartado “Habilitación del servicio de datos REST” en la página 358 para obtener información detallada sobre cómo iniciar la cuadrícula.

Procedimiento

1. Descargue e instale WebSphere Application Server Community Edition Versión 2.1.1.3 o posterior en raíz_wasce, como: /opt/IBM/wasce
2. Inicie el servidor WebSphere Application Server Community Edition ejecutando el mandato siguiente:
   - Linux UNIX raíz_wasce/bin/startup.sh
   - Windows raiz_wasce/bin/startup.bat

3. Si se ha iniciado la cuadrícula eXtreme Scale con la seguridad de eXtreme Scale habilitada, establezca las propiedades siguientes en el archivo `restservice_home/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=inicio_restservice/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Instale el servicio de datos REST de eXtreme Scale y el ejemplo proporcionado en el servidor WebSphere Application Server Community Edition:
   a. Añada el JAR de tiempo de ejecución del cliente ObjectGrid al depósito de WebSphere Application Server Community Edition:
      1) Abra la consola de administración de WebSphere Application Server Community Edition e inicie sesión.

         Consejo: El URL predeterminado es: `http://localhost:8080/console`. El ID de usuario predeterminado es `system` y la contraseña es `manager`.
      2) Pulse el **Depósito**, en la carpeta Servicios.
      3) En la sección **Añadir archivo al depósito**, introduzca los datos siguientes en los recuadros de texto de entrada:

Tabla 28. Archivo al depósito

Recuadro de texto	Valor
Archivo	wxs_home/lib/ogclient.jar
Grupo	com.ibm.websphere.xs
Artefacto	ogclient
Versión	7.0
Tipo	jar

      4) Pulse el botón Instalar.

         Consejo: Consulte la nota técnica siguiente para obtener información detallada sobre los distintos métodos de dependencias configuración de clase y biblioteca: Specifying external dependencies to applications running on WebSphere Application Server Community Edition (Especificación de dependencias externas con aplicaciones que se ejecutan en WebSphere Application Server Community Edition).
   b. Despliegue el módulo del servicio de datos REST, que es el archivo `wxsrestservice.war`, en el servidor WebSphere Application Server Community Edition.
      1) Edite el archivo XML de despliegue de ejemplo `restservice_home/gettingstarted/wasce/geronimo-web.xml` para incluir las dependencias de vía de acceso a los directorios de vía de acceso de clases del ejemplo de iniciación:

         Cambie las vías de acceso classesDirs para los dos GBeans del cliente de iniciación:

         - La vía de acceso "classesDirs" para el GBean GettingStarted_Client_SharedLib se debe definir como: `inicio_restservice/gettingstarted/restclient/bin`
         - La vía de acceso "classesDirs" para el GBean GettingStarted_Common_SharedLib se debe definir como: `inicio_restservice/gettingstarted/common/bin`

2) Abra la consola administrativa de WebSphere Application Server Community Edition e inicie sesión.

   Consejo: El URL predeterminado es: `http://localhost:8080/console`. El ID de usuario predeterminado es `system` y la contraseña es `manager`.

3) Pulse **Desplegar nuevas**.
4) En la página **Instalar aplicaciones nuevas**, escriba los valores siguientes en los recuadros de texto:

Tabla 29. Valores de instalación

Recuadro de texto	Valor
Archivo	`inicio_restservice/lib/wxsrestservice.war`
Plan	`inicio_restservice/gettingstarted/wasce/geronimo-web.xml`

5) Pulse el botón Instalar.

   La página de la consola debe indicar que la aplicación se ha instalado e iniciado satisfactoriamente.

6) Examine el registro de salida del sistema WebSphere Application Server Community Edition o la consola para verificar que el servicio de datos REST se ha iniciado satisfactoriamente comprobando si el mensaje siguiente está presente:

   ```
   CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.
   ```

5. Verifique que el servicio de datos REST funcione
   a. Abra el enlace siguiente en una ventana de navegador: `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`. Aparecerá el documento de servicio correspondiente a la cuadrícula NorthwindGrid.
   b. Abra el enlace siguiente en una ventana de navegador: `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata`. Aparecerá el documento de extensiones de datos del modelo de entidad (EDMX).
6. Para detener los procesos de cuadrícula, utilice la combinación de teclas `CTRL+C` en la ventana de mandato respectiva.
7. Para detener WebSphere Application Server Community Edition, utilice el mandato siguiente:
   - UNIX Linux `raíz_wasce/bin/shutdown.sh`
   - Windows `raíz_wasce\bin\shutdown.bat`

   Consejo: El ID de usuario predeterminado es `system` y la contraseña es `manager`.Si utiliza un puerto personalizado, utilice la opción `-port`.

Despliegue del servicio de datos REST en Apache Tomcat

Este tema describe cómo configurar el servicio de datos REST de WebSphere eXtreme Scale en Apache Tomcat Versión 5.5 o posterior.

Acerca de esta tarea

- Debe haber un JRE o JDK IBM o Sun, Versión 5 o posterior instalado y una variable de entorno JAVA_HOME especificada.

- Apache Tomcat Versión 5.5 o posterior debe estar instalado. Consulte Apache Tomcat para obtener información detallada sobre cómo instalar Tomcat.
- Una instalación autónoma de WebSphere eXtreme Scale.

Procedimiento

1. Si utiliza un Sun JRE o JDK, instale IBM ORB en Tomcat:
   a. Tomcat versión 5.5:

      Copie todos los archivos JAR de:

      el directorio *inicio_wxs*/lib/endorsed

      a:

      el directorio *raíz_tomcat*/common/endorsed
   b. Tomcat versión 6.0:

      Cree un directorio "endorsed":

      UNIX Linux `mkdir raíz_tomcat/endorsed`

      Windows `md raíz_tomcat/endorsed`

      Copie todos los archivos JAR de:

      `inicio_wxs/lib/endorsed`

      a:

      `raíz_tomcat/common/endorsed`
2. Configure e inicie una cuadrícula de datos.
   a. Para obtener información sobre cómo configurar una cuadrícula de datos para su uso con el servicio de datos REST, consulte Capítulo 6, “Configuración”, en la página 223.
   b. Verifique que un cliente de eXtreme Scale se pueda conectar a entidades de la cuadrícula y acceder a ellas. Si desea ver un ejemplo, consulte “Configuración de servicios de datos REST” en la página 357.
3. Cree el archivo JAR o el directorio de configuración del servicio REST de eXtreme Scale. Consulte la información sobre empaquetamiento y despliegue en “Instalación del servicio de datos REST” en la página 200 para obtener los detalles.
4. Despliegue el módulo del servicio de datos REST: wxsrestservice.war al servidor Tomcat.

   Copie el archivo `wxsrestservice.war` de:

   inicio_servicioRest/lib

   a:

   raíz_tomcat/webapps
5. Añada el JAR de tiempo de ejecución del cliente ObjectGrid y el JAR de la aplicación a la vía de acceso de clases compartida de Tomcat:
   a. Edite el archivo *raíz_tomcat*/conf/catalina.properties
   b. Añada los nombres de vía de acceso siguientes al final de la propiedad shared.loader, separándolos mediante comas:
      - *inicio_wxs*/lib/ogclient.jar
      - *inicio_servicioRest*/gettingstarted/restclient/bin
      - *inicio_servicioRest*/gettingstarted/common/bin
6. Si utiliza la seguridad Java 2 y añade permisos de seguridad al archivo de política tomcat:
   - Si utiliza Tomcat versión 5.5:

Fusione el contenido del archivo de política catalina 5.5 de ejemplo que se encuentra en

inicio_servicioRest/gettingstarted/tomcat/catalina-5_5.policy con el archivo *raíz_tomcat*/conf/catalina.policy.

- Si utiliza Tomcat versión 6.0:

  Fusione el contenido del archivo de política catalina 6.0 de ejemplo que se encuentra en

  inicio_servicioRest/gettingstarted/tomcat/catalina-6_0.policy con el archivo *raíz_tomcat*/conf/catalina.policy.

7. Inicie el servidor Tomcat:
   - **Si utiliza Tomcat 5.5 en UNIX o Windows o la distribución en ZIP de Tomcat 6.0:**
     a. cd *raíz_tomcat*/bin
     b. Inicie el servidor:
        - Sin la seguridad Java 2 habilitada:

          UNIX Linux ./catalina.sh run

          Windows catalina.bat run
        - Con la seguridad Java 2 habilitada:

          UNIX Linux ./catalina.sh run -security

          Windows catalina.bat run -security

     c. Los registros de Apache Tomcat se muestran en la consola. Cuando el servicio de datos REST se haya iniciado con éxito, aparecerá el mensaje siguiente en la consola administrativa:

        CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.
   - **Si utiliza Tomcat 6.0 en Windows utilizando la distribución del instalador de Windows:**
     a. cd /bin
     b. Inicie la herramienta de configuración de Apache Tomcat 6:

        tomcat6w.exe
     c. Para habilitar la seguridad Java 2 (opcional):

        Añada las entradas siguientes a las Opciones de Java en el separador Java en la ventana de propiedades de Apache Tomcat 6:

        -Djava.security.manager

        -Djava.security.policy=\conf\catalina.policy
     d. Pulse el botón Inicio de la ventana de propiedades de Apache Tomcat 6 para iniciar al servidor Tomcat.
     e. Examine los registros siguientes para verificar que el servidor Tomcat se haya iniciado con éxito:
        - *raíz_tomcat*/bin/catalina.log

          Muestra el estado del motor del servidor Tomcat
        - *raíz_tomcat*/bin/stdout.log

          Muestra el registro de salida del sistema.
     f. Cuando el servicio de datos REST se haya iniciado con éxito, aparecerá el mensaje siguiente en el registro de salida del sistema:

        CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.

8. Verifique que el servicio de datos REST funcione.
   Abra un navegador web y acceda al URL siguiente:
   `http://host:puerto/raíz_contexto/restservice/nombre_cuadrícula`
   El puerto predeterminado para Tomcat es 8080 y se configura en el archivo *raíz_tomcat*`/conf/server.xml` en el elemento <Connector>.
   Por ejemplo:
   `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/`

Resultados

Aparecerá el documento de servicio de AtomPub.

Inicio de los servicios de datos REST en Apache Tomcat:

Este tema describe cómo configurar e iniciar el servicio de datos REST de eXtreme Scale utilizando Apache Tomcat versión 5.5 o posterior.

Antes de empezar

Compruebe que se haya iniciado la cuadrícula de datos de ejemplo de eXtreme Scale. Consulte "Habilitación del servicio de datos REST" en la página 358 para obtener información detallada sobre cómo iniciar la cuadrícula de datos.

Procedimiento

1. Descargue e instale Apache Tomcat Versión 5.5 o posterior en raíz_tomcat, por ejemplo: `/opt/tomcat`
2. Instale el servicio de datos REST de eXtreme Scale y el ejemplo proporcionado en el servidor Tomcat tal como se indica a continuación:
   a. Si utiliza un Sun JRE o JDK, debe instalar IBM ORB en Tomcat:
      - Para Tomcat versión 5.5
        Copie todos los archivos JAR de:
        `inicio_wxs/lib/endorsed`
        a
        `raíz_tomcat/common/endorsed`
      - Para Tomcat versión 6.0
        1) Cree un directorio "endorsed"
           – UNIX Linux `mkdir raíz_tomcat/endorsed`
           – Windows `md raíz_tomcat/endorsed`
        2) Copie todos los archivos JAR de:
           `inicio_wxs/lib/endorsed`
           a
           `raíz_tomcat/endorsed`
   b. Despliegue el módulo del servicio de datos REST: wxsrestservice.war al servidor Tomcat.
      Copie el archivo wxsrestservice.war de:
      `inicio_restservice/lib`
      a:
      `raíz_tomcat/webapps`

c. Añada el JAR de tiempo de ejecución del cliente ObjectGrid y el JAR de la aplicación a la vía de acceso de clases compartida de Tomcat:
   1) Edite el archivo `raíz_tomcat/conf/catalina.properties`
   2) Añada los nombres de vía de acceso siguientes al final de la propiedad shared.loader en forma de lista delimitada por comas:
      - wxs_home/lib/ogclient.jar
      - inicio_restservice/gettingstarted/restclient/bin
      - inicio_restservice/gettingstarted/common/bin

      Importante: El separador de la vía de acceso debe ser una **barra inclinada**.

3. Si se ha iniciado la cuadrícula de datos de eXtreme Scale con la seguridad de eXtreme Scale habilitada, establezca las propiedades siguientes en el archivo `inicio_restservice/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=inicio_restservice/gettingstarted/security/security.ogclient.properties
loginType=none
```

4. Inicie el servidor Tomcat con el servicio de datos REST:
   - Si utiliza Tomcat 5.5 en UNIX o Windows, o bien Tomcat 6.0 en UNIX:
     a. `cd raíz_tomcat/bin`
     b. Inicie el servidor:
        – UNIX Linux `./catalina.sh run`
        – Windows `catalina.bat run`
     c. La consola mostrará entonces los registros de Apache Tomcat. Cuando el servicio de datos REST se haya iniciado con éxito, aparecerá el mensaje siguiente en la consola de administración:

        `CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.`
   - Si utiliza Tomcat 6.0 en Windows:
     a. `cd raíz_tomcat/bin`
     b. Inicie la herramienta de configuración de Apache Tomcat 6 con el mandato siguiente: `tomcat6w.exe`
     c. Pulse el botón Inicio de la ventana de propiedades de Apache Tomcat 6 para iniciar al servidor Tomcat.
     d. Examine los registros siguientes para verificar que el servidor Tomcat se haya iniciado con éxito:
        – `raíz_tomcat/bin/catalina.log`

          Muestra el estado del motor del servidor Tomcat
        – `raíz_tomcat/bin/stdout.log`

          Muestra el registro de salida del sistema.
     e. Cuando el servicio de datos REST se haya iniciado con éxito, aparecerá el mensaje siguiente en el registro de salida del sistema: `CWOBJ4000I: Se ha iniciado el servicio de datos REST de WebSphere eXtreme Scale.`
5. Verifique que el servicio de datos REST funcione
   a. Abra un navegador y acceda a:

      http://localhost:8080/wxsrestservice/restservice/NorthwindGrid

      Aparecerá el documento de servicio correspondiente a NorthwindGrid.
   b. Vaya a:

http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata

Aparecerá el documento de extensiones de datos del modelo de entidad (EDMX).

6. Para detener los procesos de la cuadrícula de datos, utilice CTRL+C en la ventana de mandatos correspondiente.
7. Para detener Tomcat, utilice CTRL +C en la ventana en la cual lo haya iniciado.

Configuración de navegadores web para acceder a feeds ATOM de servicio de datos REST

El servicio de datos REST de eXtreme Scale crea canales de información ATOM de forma predeterminada al utilizar un navegador web. Es posible que el formato del canal de información ATOM no sea compatible con navegadores antiguos o que se interprete de forma que los datos no se puedan ver como XML. Puede configurar Internet Explorer Versión 8 y Firefox Versión 3 para visualizar los feeds ATOM y XML en el navegador.

Acerca de esta tarea

El servicio de datos REST de eXtreme Scale crea canales de información ATOM de forma predeterminada al utilizar un navegador web. Es posible que el formato del canal de información ATOM no sea compatible con navegadores antiguos o que se interprete de forma que los datos no se puedan ver como XML. En el caso de los navegadores antiguos, se le solicitará que guarde los archivos en el disco. Cuando se hayan descargado los archivos, utilice su lector XML favorito para examinar los archivos. Al XML generado no se la da formato para visualizarlo, de modo que todo se imprimirá en una sola línea. La mayor parte de los programas que leen XML, como Eclipse, permiten cambiar el formato XML por un formato legible.

En el caso de los navegadores modernos, Microsoft Internet Explorer Versión 8 y Firefox Versión 3, los archivos XML ATOM se pueden mostrar de forma nativa en el navegador. Los temas siguientes proporcionan información detallada sobre cómo configurar Internet Explorer Versión 8 y Firefox Versión 3 para mostrar los canales de información ATOM y XML en el navegador.

Procedimiento

Configurar Internet Explorer Version 8

- Para habilitar Internet Explorer para leer los feeds ATOM que genera el servicio de datos REST, utilice los pasos siguientes:
  1. Pulse **Herramientas** > **Opciones de Internet**
  2. Seleccione el separador **Contenido**
  3. Pulse el botón **Configuración** de la sección **Fuentes y Web Slices**
  4. Quite la marca del recuadro: "Activar la vista de lectura de fuentes"
  5. Pulse **Aceptar** para volver al navegador.
  6. Reinicie Internet Explorer.

Configurar Firefox Versión 3

- Firefox no muestra automáticamente las páginas con el tipo de contenido application/atom+xml. La primera vez que aparezca una página de este tipo, Firefox le solicitará que guarde el archivo. Para visualizar la página, abra el propio archivo con Firefox tal como se indica a continuación:

1. En el recuadro de diálogo de selección de aplicaciones, seleccione el botón de selección "Abrir con" y pulse el botón **Examinar**.
2. Vaya al directorio de instalación de Firefox. Por ejemplo: `C:\Archivos de programa\Mozilla Firefox`
3. Seleccione `firefox.exe` y pulse el botón **Aceptar**.
4. Marque el recuadro de selección “Utilizar siempre el programa seleccionado para abrir este tipo de archivos”.
5. Pulse el botón **Aceptar**.
6. A continuación, Firefox mostrará la página XML ATOM en una ventana o pestaña de navegador nueva

- Firefox representa automáticamente los canales de información ATOM en formato legible. No obstante, los canales de información que el servicio de datos REST crea incluyen XML. Firefox no puede mostrar el XML a menos que se inhabilite el representador de canales de información. A diferencia de Internet Explorer, en Firefox, el plug-in de representación de canales de información ATOM se debe editar explícitamente. Para configurar Firefox para leer canales de información ATOM como archivos XML, siga estos pasos:
  1. Abra el archivo siguiente en un editor de texto: `<raízInstalFirefox>\components\FeedConverter.js`. En la vía de acceso, `<raízInstalFirefox>` es el directorio raíz donde está instalado Firefox.

     Para los sistemas operativos Windows, el directorio predeterminado es: `C:\Archivos de programas\Mozilla Firefox`.
  2. Busque el fragmento de código igual al siguiente:

     ```
     // 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. Marque como comentario las dos líneas que empiezan por `if` y `result` colocando `//` (dos barras inclinadas) delante de ellas.
  4. Añada la siguiente sentencia al fragmento de código: `if(0) {`.
  5. El texto resultante debe ser como el que se indica a continuación:

     ```
     // 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. Guarde el archivo.
  7. Reinicie Firefox
  8. Ahora Firefox puede mostrar automáticamente todos los canales de información en el navegador.
- Pruebe su configuración intentando algunos URL.

Ejemplo

Esta sección describe algunos URL de ejemplo que se pueden utilizar para visualizar los datos añadidos por el ejemplo de iniciación proporcionado con el servicio de datos REST. Antes de utilizar los URL siguientes, añada el conjunto de datos predeterminado a la cuadrícula de datos de ejemplo de eXtreme Scale utilizando el cliente Java de ejemplo o el cliente de WCF Data Services de Visual Studio.

En los ejemplos siguientes se presupone que el puerto es 8080, aunque puede variar. Consulte la sección correspondiente para obtener información detallada sobre cómo configurar los servicios de datos REST en distintos servidores de aplicaciones.

- Ver un solo cliente con el ID "ACME":

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')
  ```

- Ver todos los pedidos del cliente "ACME":

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')/orders
  ```

- Ver el cliente "ACME" y los pedidos:

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')?$expand=orders
  ```

- Ver el pedido 1000 del cliente "ACME":

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
  Order(orderId=1000,customer_customerId='ACME')
  ```

- Ver el pedido 1000 del cliente "ACME" y su cliente asociado:

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
  Order(orderId=1000,customer_customerId='ACME')?$expand=customer
  ```

- Ver el pedido 1000 del cliente "ACME" y su cliente y detalles de pedido asociados:

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
  Order(orderId=1000,customer_customerId='ACME')?$expand=customer,orderDetails
  ```

- Ver todos los pedidos del cliente "ACME" del mes de octubre de 2009 (GMT):

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
  Customer(customerId='ACME')/orders?$filter=orderDate
  ge datetime'2009-10-01T00:00:00'
  and orderDate lt datetime'2009-11-01T00:00:00'
  ```

- Ver los 3 primeros pedidos y detalles de pedido del cliente "ACME" del mes de octubre de 2009 (GMT):

  ```
  http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/
  Customer(customerId='ACME')/orders?$filter=orderDate
  ge datetime'2009-10-01T00:00:00'
  and orderDate lt datetime'2009-11-01T00:00:00'
  &$orderby=orderDate&$top=3&$expand=orderDetails
  ```

Uso de un cliente Java con los servicios de datos REST

La aplicación cliente Java utiliza la API EntityManager de eXtreme Scale para insertar datos en la cuadrícula.

Acerca de esta tarea

Las secciones anteriores describen cómo crear una cuadrícula de datos de eXtreme Scale y configurar e iniciar el servicio de datos REST de eXtreme Scale. La aplicación cliente Java utiliza la API EntityManager de eXtreme Scale para insertar datos en la cuadrícula.No demuestra cómo utilizar las interfaces de REST. La finalidad de este cliente es demostrar cómo se utiliza la API EntityManager para interactuar con la cuadrícula de datos de eXtreme Scale y permitir la modificación de datos en la cuadrícula. Para ver datos de la cuadrícula utilizando el servicio de datos REST, utilice un navegador web o utilice la aplicación cliente Visual Studio 2008.

Procedimiento

Para añadir rápidamente contenido a la cuadrícula de datos de eXtreme Scale, ejecute el mandato siguiente:

1. Abra una línea de mandatos o una ventana de terminal y defina la variable de entorno JAVA_HOME:

- Linux UNIX `export JAVA_HOME=inicio_java`
- Windows `set JAVA_HOME=inicio_java`

2. `cd restservice_home/gettingstarted`
3. Inserte datos en la cuadrícula. Los datos que se inserten se recuperarán posteriormente utilizando un navegador web y el servicio de datos REST.

   Si la cuadrícula de datos se ha iniciado *sin* seguridad de eXtreme Scale, utilice los mandatos siguientes.

   - UNIX Linux `./runclient.sh load default`
   - Windows `runclient.bat load default`

   Si la cuadrícula de datos se ha iniciado *con* seguridad de eXtreme Scale, utilice los mandatos siguientes.

   - UNIX Linux `./runclient_secure.sh load default`
   - Windows `runclient_secure.bat load default`

   Para un cliente Java client, utilice la sintaxis del mandato siguiente:

   - UNIX Linux `runclient.sh mandato`
   - Windows `runclient.bat mandato`

   Están disponibles los siguientes mandatos:

   - `load default`

     Carga un conjunto predefinido de entidades Customer, Category y Product en la cuadrícula de datos y crea un conjunto aleatorio de pedidos para cada cliente.
   - `load category categoryId categoryName firstProductId num_products`

     Crea una categoría de producto y un número fijo de entidades de producto en la cuadrícula de datos. El parámetro firstProductId identifica el número de ID del primer producto y a cada producto siguiente se le asigna el ID siguiente hasta que se crea el número de productos especificado.
   - `load customer companyCode contactNamecompanyName numOrders firstOrderIdshipCity maxItems discountPct`

     Carga un nuevo cliente en la cuadrícula de datos y crea un conjunto fijo de entidades Order para cualquier producto aleatorio cargado actualmente en la cuadrícula. El número de entidades Order se determina definiendo el parámetro <numOrders>. Cada Order tendrá un número aleatorio de entidades OrderDetail hasta alcanzar el valor de <maxItems>
   - `display customer companyCode`

     Muestra una entidad Customer y las entidades Order y OrderDetail asociadas.
   - `display category categoryId`

     Muestra una entidad Category de producto y las entidades Product asociadas.

Resultados

- `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`

Ejecución y creación de la cuadrícula de datos de ejemplo y del cliente Java con Eclipse

El ejemplo de iniciación del servicio de datos REST se puede actualizar y ampliar mediante Eclipse. Para obtener información detallada sobre cómo configurar el entorno de Eclipse, consulte el documento de texto: `inicio_restservice/gettingstarted/ECLIPSE_README.txt`.

Después de que el proyecto WXSRestGettingStarted se haya importado a Eclipse y se haya creado satisfactoriamente, el ejemplo volverá a compilar automáticamente los archivos de script utilizados para iniciar el servidor de contenedores y el cliente obtendrá automáticamente los archivos de clases y los archivos XML. El servicio de datos REST también detectará automáticamente cualquier cambio, ya que el servidor web está configurado para leer los directorios de compilación de Eclipse automáticamente.

Importante: Al cambiar los archivos fuente o de configuración, tanto el servidor de contenedores de eXtreme Scale como la aplicación del servicio de datos REST se deben reiniciar. El servidor de contenedores de eXtreme Scale se debe iniciar antes de la aplicación web del servicio de datos REST.

Cliente Visual Studio 2008 WCF con servicio de datos REST

El ejemplo de iniciación del servicio de datos REST de eXtreme Scale incluye un cliente WCF Data Services que puede interactuar con el servicio de datos REST de eXtreme Scale. El ejemplo está escrito como una aplicación de línea de mandatos en C#.

Requisitos de software

El cliente de ejemplo C# de WCF Data Services requiere lo siguiente:

- Sistema operativo
  - Microsoft Windows XP
  - Microsoft Windows Server 2003
  - Microsoft Windows Server 2008
  - Microsoft Windows Vista
- Microsoft Visual Studio 2008 con Service Pack 1

  Consejo: Consulte el enlace anterior para conocer los requisitos adicionales de hardware y software.
- Microsoft .NET Framework 3.5 Service Pack 1
- Microsoft Support: An update for the .NET Framework 3.5 Service Pack 1 is available (Soporte de Microsoft: hay una actualización de NET Framework 3.5 Service Pack 1 disponible)

Creación y ejecución del cliente de iniciación

El cliente de ejemplo de WCF Data Services incluye un proyecto y una solución de Visual Studio 2008 y el código fuente para ejecutar el ejemplo. El ejemplo se debe cargar en Visual Studio 2008 y se debe compilar en un programa Windows ejecutable para poder ejecutarlo. Para crear y ejecutar el ejemplo, consulte el

documento de texto siguiente: `inicio_restservice/gettingstarted/VS2008_README.txt`.

Sintaxis del mandato del cliente C# de WCF Data Services

Windows `WXSRESTGettingStarted.exe <URL de servicio> <mandato>`

El <URL de servicio> es el URL del servicio de datos REST de eXtreme Scale configurado en la sección correspondiente.

Están disponibles los siguientes mandatos:

- `load default`

  Carga un conjunto predefinido de entidades Customer, Category y Product en la cuadrícula de datos y crea un conjunto aleatorio de pedidos para cada cliente.
- `load category <categoryId> <categoryName> <firstProductId> <numProducts>`

  Crea una categoría de producto y un número fijo de entidades de producto en la cuadrícula de datos. El parámetro firstProductId identifica el número de ID del primer producto y a cada producto siguiente se le asigna el ID siguiente hasta que se crea el número de productos especificado.
- `load customer <companyCode> <contactName> <companyName> <numOrders> <firstOrderId> <shipCity> <maxItems> <discountPct>`

  Carga un nuevo cliente en la cuadrícula de datos y crea un conjunto fijo de entidades Order para cualquier producto aleatorio cargado en la cuadrícula de datos. El número de entidades Order se determina definiendo el parámetro <numOrders>. Cada Order tendrá un número aleatorio de entidades OrderDetail hasta alcanzar el valor de <maxItems>
- `display customer <companyCode>`

  Muestra una entidad Customer y las entidades Order y OrderDetail asociadas.
- `display category <categoryId>`

  Muestra una entidad Category de producto y las entidades Product asociadas.
- `unload`

  Elimina todas las entidades cargadas utilizando el mandato "default load".

Los ejemplos siguientes ilustran distintos mandatos.

- `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`

Configuración de servidores para OSGi

WebSphere eXtreme Scale incluye un paquete OSGi de servidor, que permite iniciar y configurar servidores y contenedores en una infraestructura OSGi. Los temas de configuración describen cómo utilizar el paquete de servidor de eXtreme Scale, el servicio OSGi Blueprint y la configuración de eXtreme Scale para ejecutar servidores eXtreme Scale en una infraestructura OSGi de Eclipse Equinox.

Acerca de esta tarea

Son necesarias las tareas siguientes para iniciar un servidor eXtreme Scale en Eclipse Equinox:

Procedimiento

1. Crear un paquete OSGi que almacenará los plug-ins de eXtreme Scale, exponiéndolos como servicios y actualizar el archivo XML de descriptor de ObjectGrid para hacer referencia a los servicios.
2. Configurar OSGi para iniciar un servidor de contenedor de eXtreme Scale.
3. Instalar e iniciar el paquete de servidor de eXtreme Scale en la infraestructura OSGi.
4. Instalar e iniciar el paquete OSGi que contiene los plug-ins de eXtreme Scale.

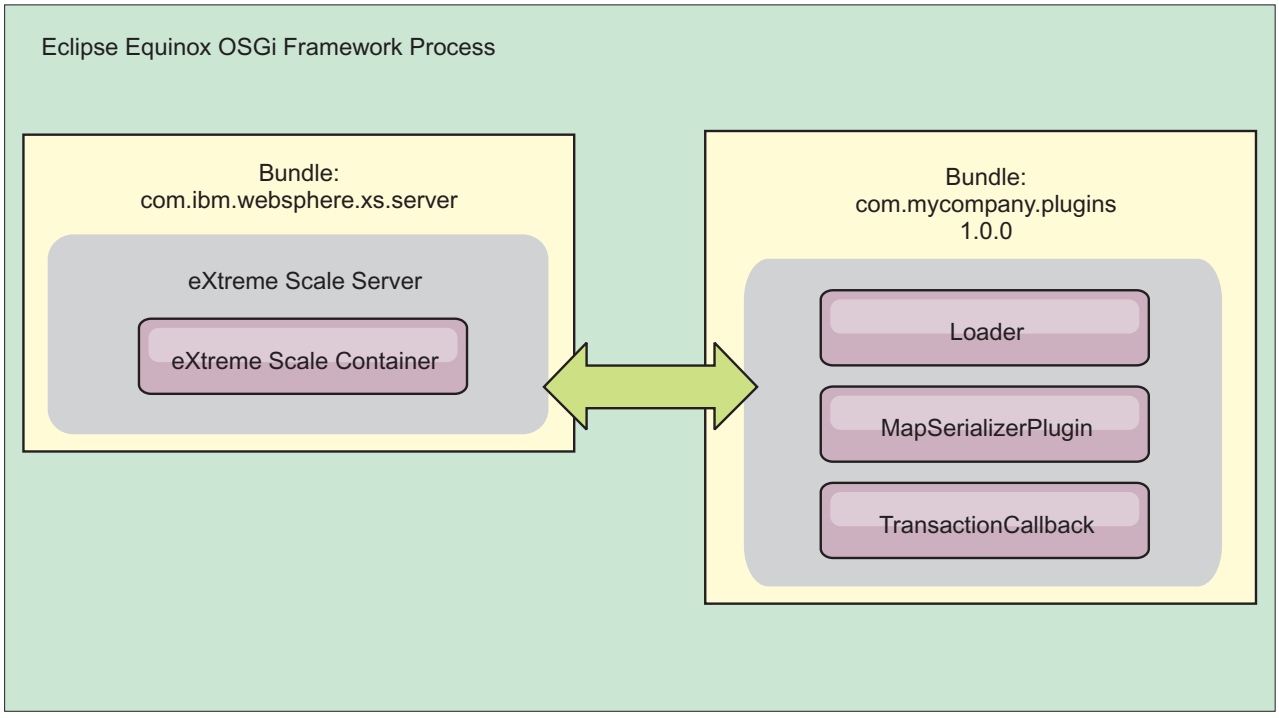


Figura 48. Proceso de Eclipse Equinox para instalar e iniciar paquetes OSGi con plug-ins de eXtreme Scale

Configuración de plug-ins de eXtreme Scale con OSGi Blueprint

Todos los plug-ins de eXtreme Scale ObjectGrid y BackingMap se pueden definir como servicios y beans OSGi utilizando el servicio OSGi Blueprint disponible con Eclipse Gemini o Apache Aries.

Antes de empezar

Antes de configurar los plug-ins como servicios OSGi, primero debe empaquetar los plug-ins en un paquete OSGi y conocer los principios fundamentales de los plug-ins necesarios. El paquete debe importar los paquetes de servidor o cliente de WebSphere eXtreme Scale y otros paquetes dependientes necesarios para los plug-ins o crear una dependencia de paquete en los paquetes de servidor o cliente de eXtreme Scale. Este tema describe cómo configurar el XML de Blueprint para crear beans de plug-ins y exponerlos como servicios OSGi para que eXtreme Scale los utilice.

Acerca de esta tarea

Los beans y servicios están definidos en un archivo XML Blueprint y el contenedor Blueprint descubre, crea y conecta los beans entre ellos y los expone como servicios. El proceso deja los beans disponibles para otros paquetes OSGi, incluidos los paquetes de servidor y cliente de eXtreme Scale.

Al crear servicios de plug-in personalizados para utilizarlos con eXtreme Scale, el paquete que va a alojar los plug-ins, debe estar configurado para utilizar Blueprint. Además, se debe crear y almacenar un archivo XML Blueprint dentro del paquete. Para obtener una visión general de la especificación, lea la información sobre la creación de aplicaciones OSGi con la especificación de contenedor Blueprint.

Procedimiento

1. Cree un archivo XML Blueprint. Puede utilizar el nombre que desee para el archivo. No obstante, debe incluir el espacio de nombre blueprint:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
   ...
   </blueprint>
   ```

2. Cree definiciones de bean en el archivo XML Blueprint para cada plug-in de eXtreme Scale.

   Los beans se definen utilizando el elemento `<bean>`, se pueden conectar a otras referencias de bean y pueden incluir parámetros de inicialización.

   Importante: Al definir un bean, debe utilizar el ámbito correcto. Blueprint soporta los ámbitos de singleton y prototipo. eXtreme Scale también soporta un ámbito de fragmento personalizado.

   Defina la mayoría de los plug-ins de eXtreme Scale como beans de ámbito de fragmento o prototipo, ya que todos los beans deben ser exclusivos para cada fragmento ObjectGrid o instancia de BackingMap con los que estén asociados. Los beans de ámbito de fragmento pueden ser útiles cuando se utilizan los beans en otros contextos para permitir recuperar la instancia correcta.

   Para definir un bean de ámbito de prototipo, utilice el atributo `scope="prototype"` en el bean:

   ```
   <bean id="myPluginBean" class="com.mycompany.MyBean" scope="prototype">
   ...
   </bean>
   ```

   Para definir un bean de ámbito de fragmento, debe añadir el espacio de nombres `objectgrid` al esquema XML y utilizar el atributo `scope="objectgrid:shard"` en el bean:

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
   ```

```
            xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"

    xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                http://www.ibm.com/schema/objectgrid/objectgrid.xsd">

    <bean id="myPluginBean" class="com.mycompany.MyBean"
   scope="objectgrid:shard">
    ...
    </bean>

...
```

3. Cree definiciones de bean PluginServiceFactory para cada bean de plug-in. Todos los beans de eXtreme Scale deben tener un bean PluginServiceFactory definido para que se pueda aplicar el ámbito de bean correcto. eXtreme Scale incluye un BlueprintServiceFactory que se puede utilizar. Incluye dos propiedades que se deben establecer. Debe establecer la propiedad blueprintContainer en la referencia `blueprintContainer` y la propiedad beanId se debe establecer en el nombre de identificador de bean. Cuando eXtreme Scale busca el servicio para instanciar los beans adecuados, el servidor busca la instancia de componente de bean utilizando el contenedor Blueprint.

   ```
   bean id="myPluginBeanFactory"
       class="com.ibm.websphere.objectgrid.plugins.osgi.BluePrintServiceFactory">
         <property name="blueprintContainer" ref="blueprintContainer"/>
   <property name="beanId" value="myPluginBean" />
   </bean>
   ```

4. Crear un administrador de servicios para cada bean PluginServiceFactory. Cada administrador de servicios expone el bean PluginServiceFactory, utilizando el elemento `<service>`. El elemento de servicio identifica el nombre a exponer en OSGi, la referencia al bean PluginServiceFactory, la interfaz a exponer y la clasificación del servicio. eXtreme Scale utiliza la clasificación de administrador de servicios para realizar actualizaciones de servicio cuando la cuadrícula de eXtreme Scale está activa. Si no se especifica la clasificación, la infraestructura OSGi supone una clasificación de 0. Lea la información sobre la actualización de clasificaciones de servicio para obtener más información.

   Blueprint incluye varias opciones para configurar administradores de servicios. Para definir un administrador de servicios simple para un bean PluginServiceFactory, cree un elemento `<service>` para cada bean PluginServiceFactory:

   ```
   <service ref="myPluginBeanFactory"
       interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
       ranking="1">
   </service>
   ```

5. Almacene el archivo XML Blueprint en el paquete de plug-ins. El archivo XML Blueprint debe almacenarse en el directorio `OSGI-INF/blueprint` para que se descubra el contenedor Blueprint.

   Para almacenar el archivo XML Blueprint en un directorio diferente, debe especificar la siguiente cabecera de manifiesto Bundle-Blueprint:

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

Resultados

Los plug-ins de eXtreme Scale están ahora configurados para exponerse en un contenedor OSGi Blueprint. Además, el archivo XML de descriptor ObjectGrid está configurado para hacer referencia a los plug-ins utilizando el servicio OSGi Blueprint.

Configuración de servidores con OSGi Blueprint

Puede configurar servidores de contenedor de WebSphere eXtreme Scale utilizando un archivo XML de OSGi Blueprint, lo que permite simplificar el empaquetado y el desarrollo de paquetes de servidor autocontenidos.

Antes de empezar

En este tema se supone que se han completado las tareas siguientes:

- Se ha instalado e iniciado la infraestructura OSGi de Eclipse Equinox con el contenedor Eclipse Gemini o Apache Aries Blueprint.
- Se ha instalado e iniciado el paquete de servidor de eXtreme Scale.
- Se ha creado el paquete de plug-ins dinámicos de eXtreme Scale.
- Se han creado el archivo XML de política de despliegue y el archivo XML de descriptor de ObjectGrid de eXtreme Scale.

Acerca de esta tarea

Esta tarea describe cómo configurar un servidor de eXtreme Scale con un contenedor utilizando un archivo XML Blueprint. El resultado del procedimiento es paquete de contenedor. Cuando se inicie el paquete de contenedor, el paquete de el servidor eXtreme Scale realizará un seguimiento del paquete, analizará el XML de servidor e iniciará un servidor y contenedor.

Un paquete de contenedor se puede combinar de manera opcional con la aplicación y los plug-ins de eXtreme Scale cuando no son necesarias actualizaciones de plug-in dinámicas o los plug-ins no soportan la actualización dinámica.

Procedimiento

1. Cree un archivo XML Blueprint con el espacio de nombres `objectgrid` incluido. Puede utilizar el nombre que desee para el archivo. No obstante, debe incluir el espacio de nombres blueprint:

```
<?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. Añada la definición XML para el servidor de eXtreme Scale con las propiedades de servidor adecuadas. Consulte el archivo XML de descriptor Spring para obtener detalles sobre todas las propiedades de configuración disponibles. Consulte el ejemplo siguiente de la definición XML:

```
objectgrid:server
    id="xsServer"
tracespec="ObjectGridOSGi=all=enabled"
   tracefile="logs/osgi/wxsserver/trace.log"
    jmxport="1199"
   listenerPort="2909">
    <objectgrid:catalog host="catserver1.mycompany.com" port="2809" />
    <objectgrid:catalog host="catserver2.mycompany.com" port="2809" />
</objectgrid:server>
```

3. Añadir la definición XML para el contenedor de eXtreme Scale con la referencia a la definición de servidor y a los archivos XML de descriptor ObjectGrid y XML de despliegue ObjectGrid incorporados en el paquete; por ejemplo:

```
<objectgrid:container id="container"
    objectgridxml="/META-INF/objectGrid.xml"
    deploymentxml="/META-INF/objectGridDeployment.xml"
    server="xsServer" />
```

4. Almacene el archivo XML Blueprint en el paquete de contenedor. El XML Blueprint se debe almacenar en el directorio `OSGI-INF/blueprint` para que se encuentre el contenedor Blueprint.

   Para almacenar el archivo XML Blueprint en un directorio diferente, debe especificar la cabecera de manifiesto Bundle-Blueprint; por ejemplo:

   ```
   Bundle-Blueprint: OSGI-INF/blueprint.xml
   ```

5. Empaquete los archivos en un solo archivo JAR de paquete. Consulte el ejemplo siguiente de una jerarquía de directorios de paquete:

   ```
   MyBundle.jar
       /META-INF/manifest.mf
       /META-INF/objectGrid.xml
       /META-INF/objectGridDeployment.xml
       /OSGI-INF/blueprint/blueprint.xml
   ```

Resultados

Ya se ha creado un paquete de contenedor de eXtreme Scale y ahora se puede instalar en Eclipse Equinox. Cuando se inicia el paquete de contenedor, el entorno de ejecución de servidor de eXtreme Scale del paquete de servidor de eXtreme Scale inicia automáticamente el servidor de eXtreme Scale de singleton utilizando los parámetros definidos en el paquete e inicia un servidor de contenedor. El paquete se puede detener e iniciar, lo que hace que el contenedor se detenga y se inicie. El servidor es un singleton y no se detiene cuando el paquete se inicia por primera vez.

Configuración de servidores con la administración de configuración de OSGi

Puede utilizar el servicio de administración de configuración de OSGi (config admin) para configurar los servidores de contenedor de WebSphere eXtreme Scale.

Acerca de esta tarea

Para configurar un servidor, el identificador permanente (PID) de ManagedService, com.ibm.websphere.xs.server, se establece para hacer referencia al archivo de propiedades de servidor ObjectGrid en el sistema de archivos. Para configurar un contenedor, el PID de ManagedServiceFactory, com.ibm.websphere.xs.container, se establece para hacer referencia al archivo XML de despliegue de ObjectGrid y el archivo XML de política de despliegue de ObjectGrid en el sistema de archivos.

Cuando los dos PID se establecen en el servicio de administración de configuración, el servicio de servidor de eXtreme Scale inicializa automáticamente el servidor e inicia el contenedor con los archivos de configuración especificados. Los PID de administración de configuración se conservan en el directorio de configuración de OSGi. Si la configuración no se borra, los valores se mantienen entre los reinicios de infraestructura.

Existen varios programas de utilidad de otras empresas para establecer propiedades de administración de configuración. Los siguientes programas de utilidad son ejemplos de herramientas que el producto soporta:

- El Cliente de línea de mandatos de administración Luminis OSGi Configuration Admin permite la configuración de línea de mandatos.

- Apache Felix File Install permite especificar valores de PID de administración de configuración en archivos de propiedades estándares.

Para configurar servidores de contenedor de eXtreme Scale con el cliente de línea de mandatos de administración de configuración de OSGi para Luminis, realice los pasos siguientes

Procedimiento

1. Crear un PID de servicio gestionado para el archivo de propiedades de servidor ObjectGrid en la consola de OSGi, ejecutando los mandatos siguientes:

   ```
   osgi> cm create com.ibm.websphere.xs.server
   osgi> cm put com.ibm.websphere.xs.server objectgrid.server.props /mypath/server.properties
   ```

2. Cree un PID de identificador de persistencia de fábrica de servicio gestionado para el contenedor ObjectGrid en la consola de OSGi ejecutando los mandatos siguientes.

   Atención: Utilice el PID que se crea con el mandato de administración de configuración **`createf`**. El PID que se utiliza en el siguiente fragmento de código es sólo un ejemplo.

   ```
   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
   ```

Resultados

Ahora los servidores de contenedor de eXtreme Scale están configurados para iniciarse en una infraestructura OSGi de Eclipse Equinox.

Qué hacer a continuación

Los servidores de contenedor también se pueden crear mediante programa utilizando la API ServerFactory y activadores de paquetes OSGi. Para obtener detalles sobre la utilización de la API ServerFactory, consulte la documentación de la API.

Capítulo 7. Administración

La administración y la utilización del entorno del producto incluye iniciar y detener servidores, gestionar la disponibilidad de la cuadrícula de datos y realizar la recuperación de los escenarios de anomalía del centro de datos. Después de configurar los servidores de catálogo y los servidores de contenedor, puede iniciar y detener los servidores mediante diversos métodos. El método que utiliza para iniciar y detener servidores depende de si utiliza una topología incorporada, una topología autónoma o una topología que se ejecuta en WebSphere Application Server.

Inicio y detención de los servidores autónomos

Puede iniciar y detener los servidores de contenedor y de catálogo autónomos con los scripts **startOgServer** y **stopOgServer** o la API de servidor incorporado.

Antes de empezar

Si está iniciando o deteniendo los servidores en un entorno autónomo que esté utilizando un proveedor de seguridad de cliente externo, debe establecer la variable de entorno *CLIENT_AUTH_LIB* antes de ejecutar los scripts **startOgServer** o **stopOgServer**. Para obtener más información sobre establecer esta variable de entorno, consulte "Inicio de servidores seguros en un entorno autónomo" en la página 530.

Inicio de los servidores autónomos

Cuando se ejecuta una configuración autónoma, el entorno está formado por servidores de catálogo, servidores de contenedor y procesos de cliente. Los servidores WebSphere eXtreme Scale también pueden incorporarse en las aplicaciones Java existentes con la API de servidor incorporado. Debe configurar e iniciar manualmente estos procesos.

Antes de empezar

Puede iniciar servidores WebSphere eXtreme Scale en un entorno que no tiene WebSphere Application Server instalado. Si utiliza WebSphere Application Server, consulte "Configuración de WebSphere eXtreme Scale con WebSphere Application Server" en la página 256.

Inicio de un servicio de catálogo autónomo

Debe iniciar el servicio de catálogo manualmente cuando utilice un entorno distribuido de WebSphere eXtreme Scale que no se ejecute en WebSphere Application Server.

Antes de empezar

- Si utiliza WebSphere Application Server, el servicio de catálogo se inicia automáticamente en los procesos existentes. Consulte Inicio del servicio de catálogo en WebSphere Application Server si desea más información.

Acerca de esta tarea

Inicie el servicio de catálogo con el script **startOgServer**. Cuando llame al mandato start, utilice el script **startOgServer.sh** en las plataformas Unix o **startOgServer.bat** en Windows.

El servicio de catálogo se puede ejecutar en un único proceso o puede incluir varios servidores de catálogo para formar un dominio de servidor de catálogo. Se requiere un dominio de servicio de catálogo en un entorno de producción para la alta disponibilidad. Para obtener más información sobre los dominios de servicio de catálogo, consulte la información sobre los dominios de servicio de catálogo en la *Visión general del producto*. También puede especificar parámetros adicionales en el script para enlazar el intermediario de solicitud de objetos (ORB) con un host y puerto específicos, especificar el dominio o habilitar la seguridad.

Procedimiento

- **Inicie un único proceso de servidor de catálogo.**

  Para iniciar un servidor de catálogo único, escriba los siguientes mandatos desde la línea de mandatos:

  1. Vaya al directorio `bin`.

     ```
     cd objectgridRoot/bin
     ```

  2. Ejecute el mandato **startOgServer**.

     ```
     startOgServer.bat|sh catalogServer
     ```

  Para obtener una lista de todos los parámetros de línea de mandatos disponibles, consulte "Script **startOgServer**" en la página 401. No utilice una sola Máquina virtual Java (JVM) para ejecutar el servicio de catálogo en un entorno de producción. Si el servicio de catálogo falla, ningún cliente nuevo podrá direccionarse al eXtreme Scale desplegado, y no se podrá añadir ninguna instancia nueva de ObjectGrid al dominio. Por estos motivos, deberá iniciar un conjunto de Mäquinas virtuales Java para ejecutar un dominio de servicio de catálogo.

- **Inicie un dominio de servicio de catálogo que conste de varios puntos finales.**

  Para iniciar un conjunto de servidores para que ejecuten un servicio de catálogo, debe utilizar la opción **-catalogServiceEndPoints** en el scriptstartOgServer. Este argumento acepta una lista de puntos finales de servicio de catálogo con el formato de *nombreServidor*:*nombreHost*:*puertoCliente*:*puertoIgual*.En el siguiente ejemplo se muestra cómo iniciar la primera de tres Mäquinas virtuales Java para alojar un servicio de catálogo:

  1. Desplácese al directorio bin.

     ```
     cd raíz_intal_wxs/bin
     ```

  2. Ejecute el mandato **startOgServer**.

     ```
     startOgServer.bat|sh cs1 -catalogServiceEndPoints
     cs1:MyServer1.company.com:6601:6602,
     cs2:MyServer2.company.com:6601:6602,
     cs3:MyServer3.company.com:6601:6602
     ```

     En este ejemplo, se inicia el servidor `cs1` en el host `MyServer1.company.com`. Este nombre de servidor es el primer argumento que se pasado al script. Durante la inicialización del usuario `cs1`, los parámetros catalogServiceEndpoints se examinan para determinar qué puertos se asignan para este proceso. La lista también se utiliza para permitir al servidor `cs1` aceptar conexiones de otros servidores: `cs2` y `cs3`.

3. Para iniciar los servidores de catálogo restantes de la lista, pase los siguientes argumentos al script startOgServer. Inicio del servidor `cs2` en el host `MyServer2.company.com`.

   ```
   startOgServer.bat|sh cs2 -catalogServiceEndPoints
   cs1:MyServer1.company.com:6601:6602,
   cs2:MyServer2.company.com:6601:6602,
   cs3:MyServer3.company.com:6601:6602
   ```

   Inicio de `cs3` en `MyServer3.company.com`:

   ```
   startOgServer.bat|sh cs3 -catalogServiceEndPoints
   cs1:MyServer1.company.com:6601:6602,
   cs2:MyServer2.company.com:6601:6602,
   cs3:MyServer3.company.com:6601:6602
   ```

   Importante: Inicie como mínimo dos servidores de catálogo en paralelo.

   Debe iniciar los servidores de catálogo que se encuentran en una cuadrícula de datos en paralelo, ya que cada servidor se pausa para esperar a que los otros servidores de catálogo se unan al grupo principal. Un servidor de catálogo configurado para una cuadrícula de datos no se inicia hasta que identifica otros miembros del grupo. Con el tiempo, el servidor de catálogo excede el tiempo de espera si no hay otros servidores disponibles.

- **Enlace el ORB con un host y un puerto específicos.**

  Aparte de los puertos definidos en el argumento**`catalogServiceEndpoints`**, cada servicio de catálogo también utiliza un intermediario para solicitudes de objetos (ORB) para aceptar conexiones de clientes y contenedores. De forma predeterminada, el ORB está a la escucha en el puerto `2809` del host local. Si desea enlazar el ORB con un host y puerto específicos en una JVM de servicio de catálogo, utilice los argumentos **`-listenerHost`** y **`-listenerPort`**. El ejemplo siguiente muestra cómo iniciar un único servidor de catálogo de JVM con su ORB enlazado al puerto `7000` en `MyServer1.company.com`:

  ```
  startOgServer.sh catalogServer -listenerHost MyServer1.company.com
  -listenerPort 7000
  ```

  Cada contenedor y cliente de eXtreme Scale deben proporcionarse con los datos de punto final de ORB de servicio de catálogo. Los clientes sólo necesitan un subconjunto de estos datos, aunque como mínimo debe utilizar dos puntos finales para la alta disponibilidad.

- Opcional: **Dé un nombre al dominio de servicio de catálogo**

  No se necesario un nombre de dominio de servicio de catálogo al iniciar un servicio de catálogo. Sin embargo, si utiliza réplica multimaestro o si utiliza varios dominios de servicio de catálogo en el mismo conjunto de procesos, debe configurar un único nombre de dominio de servicio de catálogo. El nombre de dominio predeterminado es `DefaultDomain`. Para proporciona un nombre al dominio, utilice la opción **`-domain`**. En el siguiente ejemplo se muestra cómo iniciar una JVM de servicio de catálogo único con el nombre de dominio `myDomain`.

  ```
  startOgServer.sh catalogServer -domain myDomain
  ```

  Para obtener más información sobre cómo configurar réplica multimaestro, consulte “Configuración de topologías de varios centros de datos” en la página 281.

- **Inicie un servicio de catálogo seguro.** Si desea más información, consulte “Inicio de servidores seguros en un entorno autónomo” en la página 530.
- **Inicie el servicio de catálogo mediante programación.**

Cualquier valor de JVM marcado por el método CatalogServerProperties.setCatalogServer puede alojar el servicio de catálogo para eXtreme Scale. Este método indica al tiempo de ejecución del servidor eXtreme Scale que cree una instancia del servicio de catálogo cuando se inicie el servidor. El código siguiente muestra cómo crear la instancia del servidor de catálogo de eXtreme Scale:

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

//El método getInstance() iniciará el servicio de catálogo.
Server server = ServerFactory.getInstance();
```

Para obtener más información sobre cómo iniciar los servidores mediante programación, consulte "Utilización de la API de servidor incorporado para iniciar y detener servidores" en la página 410.

Inicio de servidores de contenedor

Puede iniciar los servidores de contenedor desde la línea de mandatos utilizando una topología de despliegue o utilizando el archivo `server.properties`.

Acerca de esta tarea

Para iniciar un proceso de contenedor, necesita un archivo XML de ObjectGrid. El archivo XML de ObjectGrid especifica qué servidores eXtreme Scale aloja el contenedor. Asegúrese de que el contenedor esté equipado para alojar cada ObjectGrid en el XML que le pase. Todas las clases que estas ObjectGrids requieren deben estar en la vía de acceso de clases para el contenedor. Si desea más información sobre el archivo XML de ObjectGrid, consulte Archivo `objectGrid.xsd` .

Procedimiento

- **Inicie el proceso de contenedor desde la línea de mandatos.**
  1. En la línea de mandatos, vaya al directorio bin:

     ```
     cd raíz_intal_wxs/bin
     ```

  2. Ejecute el siguiente mandato:

     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

     Importante: En el contenedor, la opción **-catalogServiceEndPoints** se utiliza para hacer referencia al host y al puerto del intermediario de solicitud de objetos (ORB) en el servicio de catálogo. El servicio de catálogo utiliza las opciones **-listenerHost** y **-listenerPort** para especificar el host y el puerto para el enlace ORB o acepta el enlace predeterminado. Cuando inicie un contenedor, utilice la opción **-catalogServiceEndPoints** para hacer referencia a los valores pasados en las opciones **-listenerHost** y **-listenerPort** en el servicio de catálogo. Si las opciones **-listenerHost** y **-listenerPort** no se utilizan cuando se inicia el servicio de catálogo, el ORB enlaza con el puerto 2809 del host local para el servicio de catálogo. No utilice la opción **-catalogServiceEndPoints** para hacer referencia a los hosts y puertos que se pasaron en la opción **-catalogServiceEndPoints** en el servicio de catálogo. En el servicio de catálogo, la opción **-catalogServiceEndPoints** se utiliza para especificar los puertos necesarios para la configuración de servidor estático.
     Este proceso se identifica por c0, el primer argumento pasado al script. Utilice `companyGrid.xml` para iniciar el contenedor. Si el ORB del servidor de catálogo se ejecuta en un host distinto que el del contenedor o utiliza un puerto no predeterminado, debe utilizar el argumento **-catalogServiceEndPoints** para

conectarse al ORB. Para este ejemplo, suponga que un servicio de catálogo simple se ejecuta en el puerto 2809 en MyServer1.company.com

- **Inicie el contenedor utilizando una política de despliegue.**

  Aunque no es necesario, se recomienda una política de despliegue durante el inicio del contenedor. La política de despliegue se utiliza para configurar el particionamiento y la réplica para eXtreme Scale. La política de despliegue también se puede utilizar para influir en el comportamiento de la colocación. Como el ejemplo anterior no ha proporcionado un archivo de política de despliegue, el ejemplo recibe todos los valores predeterminados con relación a la réplica, al réplica y a la colocación. Por ello, las correlaciones de CompanyGrid están en un mapSet. El mapSet no está particionado ni replicado. Si desea más información sobre los archivos de política de despliegue, consulte Archivo XML de descriptor de política de despliegue . En el siguiente ejemplo se utiliza el archivo companyGridDpReplication.xml para iniciar una JVM de contenedor, la JVM c0:

  1. En la línea de mandatos, vaya al directorio bin:

     ```
     cd raíz_intal_wxs/bin
     ```

  2. Ejecute el siguiente mandato:

     ```
     startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Nota: Si tiene clases Java almacenadas en un directorio específico, en lugar de alterar el script StartOgServer, puede lanzar el servidor con argumentos del modo siguiente: -jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar . En el archivo companyGridDpReplication.xml, un único conjunto de correlaciones contiene todas las correlaciones. Este mapSet está dividido en 10 particiones. Cada partición tiene una réplica síncrona y ninguna réplica asíncrona. Cualquier contenedor que utilice la política de despliegue de companyGridDpReplication.xml emparejado con el archivo XML de ObjectGrid companyGrid.xml también puede alojar fragmentos de CompanyGrid. Inicie otra JVM de contenedor, la JVM c1:

  1. En la línea de mandatos, vaya al directorio bin:

     ```
     cd raíz_intal_wxs/bin
     ```

  2. Ejecute el siguiente mandato:

     ```
     startOgServer.sh c1 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplication.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     ```

  Cada política de despliegue contiene uno o más elementos objectgridDeployment. Cuando se inicia un contenedor, éste publica su política de despliegue en el servicio de catálogo. El servicio de catálogo examina cada elemento objectgridDeployment. Si el atributo objectgridName coincide con el atributo objectgridName del elemento objectgridDeployment recibido anteriormente, se ignora el último elemento objectgridDeployment. El primer elemento objectgridDeployment recibido para un atributo objectgridName específico se utiliza como elemento maestro. Por ejemplo, suponga que la JVM c2 utiliza una política de despliegue que divide el mapSet en un número de particiones distinto:

  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>
```

Ahora, puede iniciar una tercera JVM, la JVM c2:

1. En la línea de mandatos, vaya al directorio bin:

   ```
   cd raíz_intal_wxs/bin
   ```

2. Ejecute el siguiente mandato:

   ```
   startOgServer.sh c2 -objectGridFile ../xml/companyGrid.xml
   -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
   -catalogServiceEndPoints MyServer1.company.com:2809
   ```

El contenedor de la JVM c2 se inicia con una política de despliegue que especifica 5 particiones para mapSet1. Sin embargo, el servicio de catálogo ya mantiene la copia maestra de objectgridDeployment en CompanyGrid. Cuando se inició la JVM c0, se especificó que existen 10 particiones para este mapSet. Puesto que era el primera contenedor para iniciarse y publicar su política de despliegue, su política de despliegue pasa a ser la maestra. Por lo tanto, se ignora cualquier valor de atributo de objectgridDeployment que sea igual a `CompanyGrid` en una política de despliegue posterior.

- **Inicie un contenedor utilizando un archivo de propiedades de servidor.**

  Puede utilizar un archivo de propiedades de servidor para configurar el rastreo y la seguridad en un contenedor. Ejecute los siguientes mandatos para iniciar el contenedor c3 con un archivo de propiedades de servidor:

  1. En la línea de mandatos, vaya al directorio bin:

     ```
     cd raíz_intal_wxs/bin
     ```

  2. Ejecute el siguiente mandato:

     ```
     startOgServer.sh c3 -objectGridFile ../xml/companyGrid.xml
     -deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
     -catalogServiceEndPoints MyServer1.company.com:2809
     -serverProps ../serverProps/server.properties
     ```

  A continuación, aparece un ejemplo de archivo `server.properties`:

  ```
  server.properties
  workingDirectory=
  traceSpec=*=all=disabled
  systemStreamToFileEnabled=true
  enableMBeans=true
  memoryThresholdPercentage=50
  ```

  Éste es el archivo de propiedades de servidor básico que no tiene la seguridad habilitada. Si desea más información sobre el archivo `server.properties`, consulte Archivo de propiedades de servidor .

- **Inicie un servidor de contenedor mediante programación.**

  Para obtener más información sobre cómo iniciar los servidores de contenedor mediante programación, consulte “Utilización de la API de servidor incorporado para iniciar y detener servidores” en la página 410.

Script startOgServer

El script **startOgServer** inicia los servidores de contenedor y catálogo. Puede utilizar diversos parámetros al iniciar los servidores para habilitar el rastreo, especificar números de puerto, etc.

Finalidad

Puede utilizar el script **startOgServer** para iniciar servidores.

Ubicación

El script **startOgServer** se encuentra en el directorio `bin` del directorio raíz, por ejemplo:

```
cd raíz_intal_wxs/bin
```

Nota: Si tiene clases Java almacenadas en un directorio específico, en lugar de alterar el script startOgServer, puede lanzar el servidor con argumentos del modo siguiente: `-jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar`

.

Uso para servidores de catálogo

Para iniciar un servidor de catálogo:

Windows

```
startOgServer.bat <servidor> [options]
```

UNIX

```
startOgServer.sh <servidor>[options]
```

Para iniciar un servidor de catálogo configurado predeterminado, emplee los siguientes mandatos:

Windows

```
startOgServer.bat catalogServer
```

UNIX

```
startOgServer.sh catalogServer
```

Opciones para iniciar servidores de catálogo

Todos los parámetros siguientes son opcionales.

Parámetros para iniciar un servidor de catálogo:

-catalogServiceEndPoints
<nombreServidor:nombreHost:puertoCliente:puertoIgual>
En el contenedor, hace referencia al host y puerto ORB (Object Request Broker - Intermediario de solicitud de objetos) en el servicio de catálogo. Cada atributo se define de la manera siguiente:

serverName
Especifica un nombre para identificar el proceso que está iniciando.

hostName
Especifica el nombre de host para el sistema donde se inicia el servidor.

clientPort
Especifica el puerto que se utiliza para la comunicación de servicio de catálogo de igual.

peerPort
Este valor es el mismo que el de haManagerPort. Especifica el puerto que se utiliza para la comunicación de servicio de catálogo de igual.

El ejemplo siguiente inicia el servidor de catálogo cs1, que está en el dominio de servicio de catálogo que los servidores cs2 y cs3:

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

-clusterSecurityFile <archivo xml de seguridad de clúster>
Especifica el archivo `objectGridSecurity.xml` en el disco duro, que describe las propiedades de seguridad comunes para todos los servidores (incluidos los servidores de catálogo y los servidores de contenedor). Un ejemplo de propiedad es la configuración de autenticador que representa el mecanismo de autenticación y el registro de usuarios.

Ejemplo:`/opt/xs/ogsecurity.xml`

-clusterSecurityUrl <URL de xml de seguridad de clúster>
Especifica el archivo `objectGridSecurity.xml` como URL al archivo en el disco duro o en la red, que describe las propiedades de seguridad comunes para todos los servidores (incluidos los servidores de catálogo y los servidores de contenedor). Un ejemplo de propiedad es la configuración de autenticador que representa el mecanismo de autenticación y el registro de usuarios.

Ejemplo:`file:///opt/xs/ogsecurity.xml`

-domain <nombre de dominio>
Especifica el nombre del dominio de servicio de catálogo para este servidor de catálogo. El dominio de servicio de catálogo crea un grupo de servidores de catálogo con alta disponibilidad. Cada servidor de catálogo para un único dominio debe especificar el mismo valor para el parámetro **-domain**.

-JMXConnectorPort <puerto>
Define el puerto SSL (Secure Sockets Layer) al que se enlaza el servicio JMX (Java Management Extensions).

-haManagerPort <puerto>
Sinónimo con puerto de igual. Especifica el número de puerto que utiliza el High Availability Manager. Si esta propiedad no está establecida, el servicio de catálogos genera un puerto disponible de forma automática. Esta propiedad se aplica tanto al servidor de contenedor, como al servicio catálogos. (Sólo necesario para entornos de WebSphere Application Server.)

-JMXServicePort <puerto>
Especifica el número de puerto en el que el servidor MBean escucha las comunicaciones con Java Management Extensions (JMX). Debe utilizar un número de puerto distinto para cada JVM de la configuración. Si desea utilizar JMX/RMI, especifique explícitamente **JMXServicePort** y el número de puerto, incluso si desea utilizar el valor de puerto predeterminado. Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.

Valor predeterminado: 1099

-jvmArgs <argumento de JVM>
Especifica un conjunto de argumentos de JVM. Cada opción después de la opción **-jvmArgs** se utiliza para iniciar la máquina virtual Java (JVM) del servidor. Cuando se utilice el parámetro **-jvmArgs**, asegúrese de que sea el último argumento del script opcional especificado.

Ejemplo: -jvmArgs -Xms256M -Xmx1G

-listenerHost <nombre de host>

Especifique el nombre de host al que se enlaza el Intermediario para solicitudes de objetos (ORB) para la comunicación con el protocolo Inter-ORB de Internet (IIOP). El valor debe ser un nombre de dominio totalmente calificado o una dirección IP. Si la configuración incluye varias tarjetas de red, establezca el puerto y el host de escucha para permitir que el Intermediario para solicitudes de objetos de la JVM conozca la dirección IP a la que enlazarse. Si no especifica qué dirección IP se debe utilizar, se pueden producir síntomas como tiempos de espera de conexión excedidos, anomalías de API inusuales y clientes que parece que se cuelgan.**Valor predeterminado:** localhost

-listenerPort <puerto>
Especifica el número de puerto al que se enlaza el intermediario para solicitudes de objetos (ORB). Este valor configura los contenedores y clientes para que se comuniquen con el servicio de catálogo mediante el ORB. En WebSphere Application Server, el listenerPort se hereda de la configuración de puerto BOOTSTRAP_ADDRESS.Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.**Valor predeterminado:** 2809

-quorum true|false
Habilita el quórum en el servidor de catálogo. Si desea más información, consulte Quórums del servidor de catálogos .

-script <archivo de script>
Especifica la ubicación de un script personalizado para mandatos que especifique para iniciar servidores de catálogo o contenedores y a continuación parametrizarlos o editarlos, según se requiera.

-serverProps <archivo de propiedades de servidor>
Especifica el archivo de propiedades que contiene las propiedades de seguridad específicas del servidor. El nombre de archivo especificado para esta propiedad está en formato de vía de acceso de archivo sencillo, como c:/tmp/og/catalogserver.props.

-traceSpec <especificación de rastreo>
Especifica una serie que especifica el ámbito del rastreo que está habilitado cuando se inicia el servidor.

Ejemplo:

- ObjectGrid=all=enabled
- ObjectGrid*=all=enabled

-traceFile <archivo de rastreo>
Especifica la vía de acceso a un archivo en el que guardar información de rastreo.

Ejemplo: ../logs/c4Trace.log

-timeout <segundos>
Especifica un número de segundos antes de que el inicio del servidor exceda el tiempo de espera.

Uso para servidores de contenedor Windows

```
startOgServer.bat <servidor> -objectgridFile <archivo xml>
-deploymentPolicyFile <archivo xml> [options]
```

Windows

```
startOgServer.bat <servidor> -objectgridUrl <URL de xml>
-deploymentPolicyUrl <URL de xml> [options]
```

UNIX

```
startOgServer.sh <servidor> -objectgridFile <archivo xml>
-deploymentPolicyFile <archivo xml> [options]
```

UNIX

```
 startOgServer.sh <servidor> -objectgridUrl <URL de xml>
-deploymentPolicyUrl <URL de xml> [options]
```

Opciones para servidores de contenedor

-catalogServiceEndPoints<nombreHost:puerto,nombreHost:puerto>
Especifica el host y el puerto del intermediario de solicitud de objetos (ORB) en el servicio de catálogo.

Valor predeterminado: localhost:2809

-deploymentPolicyFile <archivo xml de política de despliegue>
Especifica la vía de acceso del archivo de política de despliegue en el disco duro. La política de despliegue se utiliza para configurar el particionamiento y la réplica. La política de despliegue también se puede utilizar para influir en el comportamiento de la colocación.

Ejemplo: `../xml/SimpleDP.xml`

-deploymentPolicyUrl <url de política de despliegue>
Especifica el URL del archivo de política de despliegue en el disco duro o en la red. La política de despliegue se utiliza para configurar el particionamiento y la réplica. La política de despliegue también se puede utilizar para influir en el comportamiento de la colocación.

Ejemplo: `file://xml/SimpleDP.xml`

-JMXConnectorPort <puerto>
Define el puerto SSL (Capa de sockets seguros) al que se enlaza el servicio JMX (Java Management Extensions).

-JMXServicePort <puerto>

Especifica el número de puerto en el que el servidor MBean escucha las comunicaciones con Java Management Extensions (JMX). Debe utilizar un número de puerto distinto para cada JVM de la configuración. Si desea utilizar JMX/RMI, especifique explícitamente **JMXServicePort** y el número de puerto, incluso si desea utilizar el valor de puerto predeterminado. Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.**Valor predeterminado:** `1099`

-jvmArgs <argumento de JVM>
Especifica un conjunto de argumentos de JVM. Cada opción después de la

opción **-jvmArgs** se utiliza para iniciar la máquina virtual Java (JVM) del servidor. Cuando se utilice el parámetro **-jvmArgs**, asegúrese de que sea el último argumento del script opcional especificado.

Ejemplo: -jvmArgs -Xms256M -Xmx1G

-listenerHost <nombre de host>

Especifique el nombre de host al que se enlaza el Intermediario para solicitudes de objetos (ORB) para la comunicación con el protocolo Inter-ORB de Internet (IIOP). El valor debe ser un nombre de dominio totalmente calificado o una dirección IP. Si la configuración incluye varias tarjetas de red, establezca el puerto y el host de escucha para permitir que el Intermediario para solicitudes de objetos de la JVM conozca la dirección IP a la que enlazarse. Si no especifica qué dirección IP se debe utilizar, se pueden producir síntomas como tiempos de espera de conexión excedidos, anomalías de API inusuales y clientes que parece que se cuelgan.**Valor predeterminado:** localhost

-listenerPort <puerto>

Especifica el número de puerto al que se enlaza el intermediario para solicitudes de objetos (ORB). Este valor configura los contenedores y clientes para que se comuniquen con el servicio de catálogo mediante el ORB. En WebSphere Application Server, el listenerPort se hereda de la configuración de puerto BOOTSTRAP_ADDRESS.Esta propiedad se aplica al servidor de contenedor y al servicio de catálogo.**Valor predeterminado:** 2809

-objectgridFile <archivo xml de descriptor ObjectGrid>

Especifica la vía de acceso al archivo descriptor de ObjectGrid.El archivo XML de ObjectGrid especifica qué servidores eXtreme Scale aloja el contenedor.

-objectgridUrl <url de descriptor ObjectGrid>

Especifica un URL para el archivo de descriptor de ObjectGrid. El archivo XML de ObjectGrid especifica qué servidores eXtreme Scale aloja el contenedor.

-script <archivo de script>

Especifica la ubicación de un script personalizado para mandatos que especifique para iniciar servidores de catálogo o contenedores y a continuación parametrizarlos o editarlos, según se requiera.

-serverProps <archivo de propiedades de servidor>

Especifica la vía de acceso al archivo de propiedades del servidor.

Ejemplo:../security/server.props

-timeout <segundos>

Especifica un número de segundos antes de que el inicio del servidor exceda el tiempo de espera.

-traceFile <archivo de rastreo>

Especifica la vía de acceso a un archivo en el que guardar información de rastreo.

Ejemplo: ../logs/c4Trace.log

-traceSpec <especificación de rastreo>

Especifica una serie que especifica el ámbito del rastreo que está habilitado cuando se inicia el servidor.

Ejemplo:

- ObjectGrid=all=enabled
- ObjectGrid*=all=enabled

-zone <nombre de zona>
Especifica la zona que se va a utilizar para todos los contenedores dentro del servidor. Consulte "Direccionamiento a zonas según preferencias" en la página 243 la información sobre las zonas en la *Visión general del producto* para obtener más información sobre cómo configurar las zonas.

Detención de servidores autónomos

Puede utilizar el script `stopOgServer` para detener los procesos de servidor eXtreme Scale.

Acerca de esta tarea

Ejecute el script **stopOgServer** yendo al directorio `bin`:

```
cd raíz_intal_wxs/bin
```

Procedimiento

- **Detenga un único servidor de contenedor.**

  Ejecute el script **stopOgServer** para detener el servidor de contenedor. Utilice este mandato solo al detener un único servidor de contenedor. Si ejecuta el mandato stop de servidor de catálogo único en varios servidores de contenedor sucesivamente, es posible que vea problemas de rendimiento y rotación para la colocación de fragmentos.

  ```
  stopOgServer servidorContenedor -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Atención: La opción **-catalogServiceEndPoints** debe coincidir con el valor de la opción **-catalogServiceEndPoints** que se ha utilizado para iniciar el contenedor. Si no se ha utilizado **-catalogServiceEndPoints** para iniciar el contenedor, los valores predeterminados probablemente sean localhost o el nombre de host y 2809 para el puerto ORB para conectarse al servicio de catálogo. De lo contrario, utilice los valores proporcionados a **-listenerHost** y **-listenerPort** en el servicio de catálogo. Si las opciones **-listenerHost** y**-listenerPort** no se utilizan al iniciar el servicio de catálogo, ORB se enlaza al puerto 2809 en el host local para el servicio de catálogo.

- **Detenga varios servidores de contenedor.**

  Para evitar problemas de rotación y rendimiento para la colocación de fragmentos cuando desee detener varios servidores de contenedor simultáneamente, utilice el siguiente formato de mandato. Separe una lista de servidores de contenedor con comas:

  ```
  stopOgServer servidorContenedor0,servidorContenedor1,servidorContenedor2
  -catalogServiceEndPoints MyServer1.company.com:2809
  ```

  Si desea detener todos los contenedores en una zona o host específico, puede utilizar el parámetro **-teardown**. Si desea más información, consulte "Detención correcta de servidores con el programa de utilidad **xscmd**" en la página 409.

- **Detenga los servidores de catálogo.**

  Ejecute el script **stopOgServer** para detener el servidor de catálogo.

  ```
  stopOgServer.sh catalogServer -catalogServiceEndPoints MyServer1.company.com:2809
  ```

Atención: Al detener un servicio de catálogo, utilice la opción **-catalogServiceEndPoints** para hacer referencia al host y puerto de intermediario de solicitud de objetos (ORB) en el servicio de catálogo. El servicio de catálogo utiliza las opciones **-listenerHost** y **-listenerPort** para especificar el host y el puerto para el enlace ORB o acepta el enlace predeterminado. Si las opciones **-listenerHost** y**-listenerPort** no se utilizan al iniciar el servicio de catálogo, ORB se enlaza al puerto 2809 en el host local para el servicio de catálogo. La opción **-catalogServiceEndPoints** es distinta al detener un servicio de catálogo que al iniciar el servicio de catálogo.

El inicio de un servicio de catálogo requiere puertos de acceso de igual y puertos de acceso de cliente, si no se han utilizado los puertos predeterminados. La detención de un servicio de catálogo requiere solo el puerto ORB.

- **Detenga el servidor de la consola web.** Para detener el servidor de la consola web, ejecute el script **stopConsoleServer.bat|sh**. Este script se encuentra en el directorio *raíz_intal_wxs*/ObjectGrid/bin de la instalación. Para obtener más información, consulte "Inicio e inicio de sesión en la consola web" en la página 443.
- **Habilite el rastreo para el proceso de detención del servidor.**

  Si el contenedor no puede detenerse, puede habilitar el rastreo para ayudar a realizar la depuración del problema. Para habilitar el rastreo durante la detención de un servidor, añada los parámetros **-traceSpec** y **-traceFile** a los mandatos de detención. El parámetro **-traceSpec** especifica el tipo de rastreo que se va a habilitar y el parámetro **-traceFile** especifica la vía de acceso y el nombre del archivo que se va a crear y utilizar para los datos de rastreo.

  1. En la línea de mandatos, vaya al directorio bin.

     ```
     cd raíz_intal_wxs/bin
     ```

  2. Ejecute el script **stopOgServer** con el rastreo habilitado.

     ```
     stopOgServer.sh c4 -catalogServiceEndPoints MyServer1.company.com:2809
      -traceFile ../logs/c4Trace.log -traceSpec ObjectGrid=all=enabled
     ```

     Una vez que se ha obtenido el rastreo, busque los errores relacionados con los conflictos del puerto, clases que faltan, archivos XML incorrectos o que faltan, o cualquier seguimiento de la pila. Las especificaciones de rastreo de inicio sugeridas son:

     - `ObjectGrid=all=enabled`
     - `ObjectGrid*=all=enabled`

     Para todas las opciones de especificación de rastreo, consulte "Opciones de rastreo" en la página 537.
- **Detenga los servidores incorporados mediante programación.**

  Para obtener más información sobre cómo detener los servidores incorporados mediante programación, consulte "Utilización de la API de servidor incorporado para iniciar y detener servidores" en la página 410.

Script stopOgServer

El script **stopOgServer** detiene los servidores de catálogo y contenedor.

Finalidad

Utilice el script **stopOgServer** para detener un servidor. Debe proporcionar el nombre del servidor y sus puntos finales de servicio de catálogo.

Ubicación

El script **stopOgServer** se encuentra en el directorio `bin` del directorio `root`, por ejemplo:

```
cd raíz_intal_wxs/bin
```

Uso

Para detener un servidor de catálogo o contenedor: Windows

```
stopOgServer.bat <nombre_servidor> -catalogServiceEndPoints
<host_sc:puerto_escucha_sc,host_sc:puerto_escucha_sc> [opciones]
```

UNIX

```
stopOgServer.sh <nombre_servidor> -catalogServiceEndPoints
<host_sc:puerto_escucha_sc,host_sc:puerto_escucha_sc> [opciones]
```

Opciones

-catalogServiceEndPoints <host_cs:puerto_escucha_cs, host_cs:puerto-escucha_cs...>

Especifica el host y el número de puerto del intermediario de solicitud de objetos (ORB).

Para servidores de contenedor: la lista de puntos finales de servicio de catálogo debe ser la misma que la lista utilizada para iniciar el servidor de contenedor. Si no ha especificado esta opción al iniciar el servidor de contenedor, utilice el valor predeterminado de `localhost:2809`.

Para servidores de catálogo: si está deteniendo el servicio de catálogo, utilice los valores indicados para las opciones **-listenerHost** y **-listenerPort** cuando inició el servicio de catálogo. Si no ha especificado estas opciones al iniciar el servidor de catálogo, utilice el valor predeterminado de `localhost:2809`. El valor de **-catalogServiceEndPoints** que utiliza al detener el servicio de catálogo es distinto del que utiliza al iniciar el servicio de catálogo.

-clientSecurityFile <archivo de propiedades de seguridad>

Especifica la vía de acceso del archivo de propiedades de cliente que define las propiedades de seguridad para el cliente. Consulte Archivo de propiedades de cliente para obtener más información sobre los valores de seguridad de este archivo.

-traceSpec <especificación de rastreo>

Especifica una serie que especifica el ámbito del rastreo que está habilitado cuando se inicia el servidor.

Ejemplo:

- `ObjectGrid=all=enabled`
- `ObjectGrid*=all=enabled`

-traceFile <archivo de rastreo>

Especifica la vía de acceso a un archivo en el que guardar información de rastreo.

Ejemplo: `../logs/c4Trace.log`

-jvmArgs <argumento de JVM>

Especifica un conjunto de argumentos de JVM. Cada opción después de la opción **-jvmArgs** se utiliza para iniciar la máquina virtual Java (JVM) del

servidor. Cuando se utilice el parámetro **-jvmArgs**, asegúrese de que sea el último argumento del script opcional especificado.

Ejemplo: -jvmArgs -Xms256M -Xmx1G

Detención correcta de servidores con el programa de utilidad xscmd

Puede utilizar el programa de utilidad **xscmd** con el mandato **-c teardown** para detener una lista o un grupo de servidores de catálogo y contenedor. Este mandato simplifica el cierre de toda la cuadrícula de datos, o de partes de ella, y evita acciones de colocación y recuperación innecesarias del servicio de catálogo que se producen normalmente cuando se detienen o matan los procesos.

Procedimiento

- Detenga una lista específica de servidores.

  Proporcione una lista de servidores después del parámetro **-teardown**:

  ```
  xscmd –c teardown
  ```

- Detenga todos los servidores de una zona específica.

  Utilice el parámetro **-z** y proporcione el nombre de la zona. El servidor de catálogo determina los servidores que se ejecutan en la zona, y el programa de utilidad **xscmd** le presenta una lista de los servidores de la zona seleccionada antes de concluir los servidores:

  ```
  xscmd –c teardown –z nombre_zona
  ```

- Detenga todos los servidores de un host específico.

  Utilice el parámetro **-hf** y proporcione el nombre del host. Por ejemplo, para concluir todos los servidores de myhost.mycompany.com, entre -hf myhost.mycompany.com. El servidor de catálogo determina los servidores que se ejecutan en el host, y el programa de utilidad **xscmd** le presenta una lista de los servidores del host seleccionado antes de concluir los servidores:

  ```
  xscmd –teardown –hf <nombre_host>
  ```

Inicio y detención de servidores en un entorno de WebSphere Application Server

Los servidores de catálogo y contenedor se pueden iniciar automáticamente en un entorno de WebSphere Application Server o WebSphere Application Server Network Deployment.

Antes de empezar

Configure los servidores de catálogo y los servidores de contenedor para ejecutarse en WebSphere Application Server:

- “Configuración del servicio de catálogo en WebSphere Application Server” en la página 256
- “Configuración de servidores de contenedor en WebSphere Application Server” en la página 274

Acerca de esta tarea

El ciclo de vida de los servidores de catálogo y contenedor en WebSphere Application Server está vinculado con los procesos en los que se ejecutan estos servidores.

Procedimiento

- **Inicio de servicios de catálogo en WebSphere Application Server**:

  El ciclo de vida al que está vinculado un servidor de catálogo al proceso de WebSphere Application Server Después de configurar el dominio de servicio de catálogo en WebSphere Application Server, reinicie cada uno de los servidores que ha definido como parte del dominio de servicio de catálogo. El servicio de catálogo se inicia automáticamente en los servidores que ha asociado al dominio de servicio de catálogo. El servicio de catálogo también se puede iniciar automáticamente en los escenarios siguientes, dependiendo de la edición de WebSphere Application Server:

  – **WebSphere Application Server base**: puede configurar la aplicación para iniciar automáticamente un servidor de contenedor y un servicio de catálogo. Esta característica simplifica la prueba de unidades en entornos de despliegue, como por ejemplo Rational Application Developer porque no es necesario iniciar de forma explícita un servicio de catálogo. Consulte "Configuración de aplicaciones WebSphere Application Server para el inicio automático de servidores de contenedor" en la página 274 para obtener más información.

  – **WebSphere Application Server Network Deployment**: el servicio de catálogo se inicia automáticamente en el proceso del gestor de despliegue si el nodo del gestor de despliegue tiene instalado WebSphere eXtreme Scale y se aumenta el perfil del gestor de despliegue. Si desea más información, consulte "Configuración del servicio de catálogo en WebSphere Application Server" en la página 256.

- **Inicio de servidores de contenedor en WebSphere Application Server**:

  El ciclo de vida de un servidor de contenedor está vinculado a la aplicación de WebSphere Application Server. Cuando se inicia la aplicación configurada, los servidores de contenedor también se inician.

- **Detención de toda una cuadrícula de datos de servidores**:

  Puede detener servidores de catálogo y contenedor deteniendo las aplicaciones y los servidores de aplicaciones asociados. Sin embargo, puede detener también toda una cuadrícula de datos con el programa de utilidad **xscmd** o MBeans:

  – **En el programa de utilidad xscmd:**

    Consulte "Detención correcta de servidores con el programa de utilidad **xscmd**" en la página 409 para obtener más información sobre la detención de toda una cuadrícula de datos.

  – **Con Mbeans:**

    Utilice la operación tearDownServers con el MBean PlacementServiceMBean.

Utilización de la API de servidor incorporado para iniciar y detener servidores

Con WebSphere eXtreme Scale, puede utilizar una API programática para gestionar el ciclo de vida de servidores y contenedores incorporados. Puede configurar a través de programas el servidor con cualquiera de las opciones que también puede configurar con las opciones de la línea de mandatos o las propiedades de servidor basadas en archivo. Puede configurar el servidor incorporado para que sea un servidor de contenedor, un servicio de catálogo, o ambos.

Antes de empezar

Debe tener un método para ejecutar el código desde una Máquina virtual Java ya existente. Las clases de eXtreme Scale deben estar disponibles a través del árbol del cargador de clases.

Acerca de esta tarea

Puede ejecutar muchas tareas de administración con la API Administration. Un uso común de la API es su uso como servidor interno para almacenar el estado de la aplicación web. El servidor web puede iniciar un servidor WebSphere eXtreme Scale incorporado, realizar informes del servidor de contenedor en el servicio de catálogo y después añadirlo como un miembro de una cuadrícula distribuida mayor. Este uso puede proporcionar escalabilidad y alta disponibilidad en un almacén de datos que de lo contrario es volátil.

Puede controlar mediante programación el ciclo de vida completo de un servidor eXtreme Scale incorporado. Los ejemplos son lo más genéricos posibles y sólo muestran códigos de ejemplo de código directo para los pasos descritos.

Procedimiento

1. Obtenga el objeto ServerProperties desde la clase ServerFactory y configure las opciones necesarias.

   Cada servidor eXtreme Scale tiene un conjunto de propiedades configurables. Cuando un servidor se inicia desde la línea de mandatos, estas propiedades toman los valores predeterminado, pero puede alterar temporalmente varias propiedades proporcionando un origen o archivo externo. En el ámbito incorporado, puede establecer directamente las propiedades con un objeto ServerProperties. Debe establecer estas propiedades antes de obtener una instancia de servidor desde la clase ServerFactory. El siguiente fragmento de código obtiene un objeto ServerProperties, establece el campo CatalogServiceBootStrap e inicializa varios valores de servidor opcionales. Consulte la documentación de la API para ver una lista de los valores configurables.

   ```
   ServerProperties props = ServerFactory.getServerProperties();
   props.setCatalogServiceBootstrap("host:port"); // necesario para conectarse a un servicio
                                                  // de catálogo específico
   props.setServerName("ServerOne"); // name server
   props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // Establece la espec. de rastreo
   ```

2. Si desea que el servidor sera un servicio de catálogo, obtenga el objeto CatalogServerProperties.

   Todos los servidores incorporados pueden ser un servicio de catálogo, un servidor de contenedor, o ambos, un servidor de contenedor y un servicio de catálogo. El siguiente ejemplo obtiene el objeto CatalogServerProperties, habilita la opción del servicio de catálogo y configura distintos valores de servicio de catálogo.

   ```
   CatalogServerProperties catalogProps = ServerFactory.getCatalogProperties();
   catalogProps.setCatalogServer(true); // false de forma predeterminada, es necesario establecerlo
                                        // como un servicio de catálogo
   catalogProps.setQuorum(true); // habilitar / inhabilitar el quórum
   ```

3. Obtenga una instancia de Server desde la clase ServerFactory. La instancia de Server es un singleton con un ámbito de proceso que es responsable de gestionar la pertenencia de la cuadrícula. Después de que se haya creado una instancia, este proceso se conecta y está muy disponible con los otros servidores de la cuadrícula. El siguiente ejemplo muestra cómo crear la instancia de Server:

   ```
   Server server = ServerFactory.getInstance();
   ```

Mediante la revisión del ejemplo anterior, la clase ServerFactory proporciona un método estático que devuelve una instancia de Server. La clase ServerFactory tiene como objetivo ser la única interfaz para obtener una instancia de Server. Por lo tanto, la clase garantiza que la instancia es un singleton, o una instancia para cada JVM o cargador de clases aislado. El método getInstance inicializa la instancia de Server. Debe configurar todas las propiedades de servidor antes de inicializar la instancia. La clase Server es responsable de crear las nuevas instancias de Container. Puede utilizar ambas clases, ServerFactory y Server, para gestionar el ciclo de vida de la instancia de Server incorporada.

4. Inicie una instancia de Container utilizando la instancia de Server.

   Antes de que los fragmentos se puedan colocar en un servidor incorporado, debe crear un contenedor en el servidor. La interfaz Server tiene un método createContainer que adopta un argumento DeploymentPolicy. El siguiente ejemplo utiliza la instancia del servidor que ha obtenido para crear un contenedor utilizando un archivo DeploymentPolicy creado. Tenga en cuenta que los contenedores requieren un cargador de clases que tiene los binarios de aplicaciones disponibles para la serialización. Puede hacer que estos binarios estén disponibles llamando al método createContainer con el cargador de clases del contexto Thread establecido en el cargador de clases que desee utilizar.

   ```
   Política de despliegue = DeploymentPolicyFactory.createDeploymentPolicy(new
   URL("file://urltodeployment.xml"),
    new URL("file://urltoobjectgrid.xml"));
   Container container = server.createContainer(policy);
   ```

5. Eliminar y borrar un contenedor.

   Puede eliminar y borrar un servidor de contenedor utilizando el método teardown en ejecución en la instancia de Container obtenida. Ejecutar el método teardown en un contenedor correctamente borra el contenedor y elimina el contenedor del servidor incorporado.

   El proceso de limpieza del contenedor incluye el movimiento y la destrucción de todos los fragmentos que se han colocado dentro de dicho contenedor. Cada servidor puede contener muchos contenedores y fragmentos. La limpieza de un contenedor no afecta al ciclo de vida de la instancia padre de Server. El siguiente ejemplo demuestra cómo ejecutar el método teardown en un servidor. El método teardown se ha hecho disponible a través de la interfaz ContainerMBean. Mediante el uso de la interfaz ContainerMBean, si deja de tener acceso mediante programa a este contenedor, puede seguir eliminando y limpiando el contenedor con su MBean. También existe un método terminate en la interfaz Container, no utilice este método, a menos que sea absolutamente necesario. Este método es más potente y no coordina el movimiento y la limpieza de fragmentos apropiados.

   ```
   container.teardown();
   ```

6. Detenga el servidor incorporado.

   Cuando detenga un servidor incorporado, también puede detener los contenedores y los fragmentos que se ejecutan en el servidor. Cuando detenga un servidor incorporado, debe limpiar todas las conexiones abiertas o mover o destruir todos los fragmentos. El siguiente ejemplo demuestra cómo detener un servidor y cómo utilizar el método waitFor en la interfaz Server para asegurarse de que la instancia de Server se termina por completo. De forma similar al ejemplo del contenedor, el método stopServer pasa a estar disponible a través de la interfaz ServerMBean. Con esta interfaz, puede detener un servidor con el bean gestionado (MBean) correspondiente.

   ```
   ServerFactory.stopServer();  // Utiliza la fábrica para matar el proceso singleton de Server
    // o
   server.stopServer(); // Utiliza directamente la instancia de Server
   server.waitFor(); // Se devuelve este valor cuando el servidor ha completado correctamente
   sus procedimientos de conclusión.
   ```

Ejemplo de código completo:

```
 import java.net.MalformedURLException;
import java.net.URL;

import com.ibm.websphere.objectgrid.ObjectGridException;
import com.ibm.websphere.objectgrid.deployment.DeploymentPolicy;
import com.ibm.websphere.objectgrid.deployment.DeploymentPolicyFactory;
import com.ibm.websphere.objectgrid.server.Container;
import com.ibm.websphere.objectgrid.server.Server;
import com.ibm.websphere.objectgrid.server.ServerFactory;
import com.ibm.websphere.objectgrid.server.ServerProperties;

public class ServerFactoryTest {

    public static void main(String[] args) {

        try {

            ServerProperties props = ServerFactory.getServerProperties();
            props.setCatalogServiceBootstrap("catalogservice-hostname:catalogservice-port");
            props.setServerName("ServerOne"); // name server
            props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // TraceSpec

            /*
             * En la mayoría de los casos, el servidor actuará sólo como un servidor de contenedor
             * y se conectará a un servicio de catálogo externo. Este es un método más disponible
             * de realizar acciones. La siguiente excepción de código comentada
             * permitirá que este Server sea un servicio de catálogo.
             *
             *
             * CatalogServerProperties catalogProps =
             * ServerFactory.getCatalogProperties();
             * catalogProps.setCatalogServer(true); // habilitar el servicio de catálogo
             * catalogProps.setQuorum(true); // habilitar quórum
             */

            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);

            /*
             * Ahora el fragmento se colocará en este contenedor, si se cumplen los
             * requisitos de despliegue.
             * Esto engloba la creación del contenedor y del servidor incorporados.
             *
             * Las líneas siguientes simplemente demostrarán la llamada a métodos de limpieza
             */

            container.teardown();
            server.stopServer();
            int success = server.waitFor();

        } catch (ObjectGridException e) {
            // El contenedor no se ha podido inicializar
        } catch (MalformedURLException e2) {
            // url no válido para los archivos xml
        }

    }

}
```

API de servidor incorporado

WebSphere eXtreme Scale incluye interfaces de programación de aplicaciones (API) e interfaces de programación del sistema para incorporar servidores y clientes de eXtreme Scale a sus aplicaciones Java existentes. El tema siguiente describe las API de servidor incorporado disponibles.

Creación de instancias del servidor eXtreme Scale

Puede utilizar diversas propiedades para configurar la instancia del servidor eXtreme Scale, que puede recuperar del método ServerFactory.getServerProperties. El objeto ServerProperties es un singleton, de modo que cada llamada al método getServerProperties recupera la misma instancia.

Puede crear un servidor nuevo con el código siguiente.

```
Server server = ServerFactory.getInstance();
```

Todas las propiedades definidas antes de la primera invocación de getInstance se utilizan para inicializar el servidor.

Establecimiento de las propiedades de servidor

Puede establecer las propiedades de servidor hasta que se llame a ServerFactory.getInstance por primera vez. La primera llamada del método getInstance crea una instancia del servidor eXtreme Scale y lee todas las propiedades configuradas. Establecer las propiedades después de la creación no tiene ningún efecto. El ejemplo siguiente muestra cómo definir propiedades antes de crear una instancia de Server.

```
// Obtener las propiedades de servidor asociadas con este proceso.
ServerProperties serverProperties = ServerFactory.getServerProperties();

// Establecer el nombre del servidor para este proceso.
serverProperties.setServerName("EmbeddedServerA");

// Establecer el nombre de la zona donde está contenido este proceso.
serverProperties.setZoneName("EmbeddedZone1");

// Establecer la información de punto final necesaria para crear una rutina de
carga para el servicio de catálogo.
serverProperties.setCatalogServiceBootstrap("localhost:2809");

// Establecer el nombre de host de escucha ORB que se va a utilizar en los enlaces.
serverProperties.setListenerHost("host.local.domain");

// Establecer el puerto de escucha ORB que se va a utilizar en los enlaces.
serverProperties.setListenerPort(9010);

// Desactivar todos los MBeans de este proceso.
serverProperties.setMBeansEnabled(false);

Server server = ServerFactory.getInstance();
```

Incorporación del servicio de catálogo

Cualquier valor de JVM señalado por el método CatalogServerProperties.setCatalogServer puede albergar el servicio de catálogo de eXtreme Scale. Este método indica al tiempo de ejecución del servidor eXtreme Scale que cree una instancia del servicio de catálogo cuando se inicie el servidor. El código siguiente muestra cómo crear la instancia del servidor de catálogo de eXtreme Scale:

```
CatalogServerProperties catalogServerProperties =
 ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
```

Incorporación del contenedor de eXtreme Scale

Emita el método Server.createContainer de cualquier JVM para albergar varios contenedores de eXtreme Scale. El código siguiente muestra cómo crear la instancia de un contenedor de eXtreme Scale:

```
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);
```

Proceso de servidor autónomo

Puede iniciar todos los servicios de forma conjunta, que es práctico para el desarrollo y, también, práctico para la producción. Al iniciar los servicios de forma conjunta, un único proceso realiza todo lo siguiente: inicia el servicio de catálogo, inicia un conjunto de contenedores y ejecutar la lógica de conexión de cliente. Iniciar los servidores de esta forma clasifica los problemas de programación antes del despliegue en un entorno distribuido. El código siguiente muestra como crear la instancia de un servidor eXtreme Scale autónomo:

```
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);
```

Incorporación de eXtreme Scale in WebSphere Application Server

La configuración de eXtreme Scale se realiza automáticamente al instalar eXtreme Scale en un entorno de WebSphere Application Server. No es necesario establecer ninguna propiedad antes de acceder al servidor para crear un contenedor. El código siguiente muestra cómo crear instancias de un servidor eXtreme Scale en WebSphere Application Server:

```
Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL);
Container container = server.createContainer(policy);
```

Para ver un ejemplo paso a paso de cómo iniciar un servicio de catálogo y contenedor incorporado mediante programación, consulte el apartado “Utilización de la API de servidor incorporado para iniciar y detener servidores” en la página 410.

Administración con el programa de utilidad `xscmd`

Con **`xscmd`** puede completar tareas administrativas en el entorno como, por ejemplo: establecer enlaces de réplica multimaestro, sustituir quórum y detener grupos de servidores con el mandato teardown.

Antes de empezar

- Los servidores de catálogo y servidores de contenedor deben estar iniciados. Si los servidores de catálogo están en un dominio de servicio de catálogo, al menos dos servidores de catálogo deben estar iniciados.
- Verifique que la variable de entorno *JAVA_HOME* esté establecida para utilizar el entorno de ejecución que se ha instalado con el producto. Si está utilizando la versión de prueba del producto, debe establecer la variable de entorno *JAVA_HOME*.

Acerca de esta tarea

El programa de utilidad **xscmd** sustituye al programa de utilidad **xsadmin** de ejemplo como una herramienta de administración y supervisión completamente soportada. Puede completar operaciones similares con la herramienta **xsadmin**, pero esta herramienta no se soporta. El ejemplo **xsadmin** proporciona un método para analizar y descubrir datos de despliegue actuales y se puede utilizar como base para escribir programas de utilidad personalizados. Si anteriormente estaba utilizando la herramienta **xsadmin** para la supervisión y administración, considere actualizar los scripts para utilizar el programa de utilidad **xscmd**. Para obtener más información sobre la correlación de mandatos **xsadmin** con los nuevos mandatos **xscmd**, consulte "Migración de la herramienta **xsadmin** a la herramienta **xscmd**" en la página 216.

Procedimiento

1. Abra una ventana de línea de mandatos.En la línea de mandatos, establezca las variables de entorno correspondientes.
   a. Establezca la variable de entorno *CLIENT_AUTH_LIB*:
      - Windows `set CLIENT_AUTH_LIB=<vía_acceso_a_JAR_o_clases_seguridad>`
      - UNIX `set CLIENT_AUTH_LIB=<vía_acceso_a_JAR_o_clases_seguridad>`
        `export CLIENT_AUTH_LIB`
2. Vaya al directorio *inicio_wxs*`/bin`.
   `cd inicio_wxs/bin`
3. Visualice la ayuda para las diversas opciones de **xscmd**.
   - Para visualizar la ayuda general, ejecute el mandato siguiente:
     - UNIX `./xscmd.sh -h`
     - Windows `xscmd.bat -h`
   - Para visualizar una lista de todos los mandatos, ejecute el mandato siguiente:
     - UNIX `./xscmd.sh -lc`
     - Windows `xscmd.bat -lc`
   - Para visualizar la ayuda correspondiente a un mandato específico, ejecute el mandato siguiente:
     - UNIX `./xscmd.sh -h nombre_mandato`
     - Windows `xscmd.bat -h nombre_mandato`
   - Para visualizar una lista de los grupos de mandatos, ejecute el mandato siguiente:
     - UNIX `./xscmd.sh -lcg`
     - Windows `xscmd.bat -lcg`
   - Para visualizar una lista de los mandatos de un grupo de mandatos, ejecute el mandato siguiente:
     - UNIX `./xscmd.sh -lc nombre_grupo_mandatos`
     - Windows `xscmd.bat -lc nombre_grupo_mandatos`
4. Ejecute mandatos que se conectan a servidores de catálogo específicos. De forma predeterminada, **xscmd** se conecta al servidor de catálogo en el host local, utilizando el nombre de host y el puerto de `localhost:2809`. También puede proporcionar una lista de nombres de host y puertos en el mandato para poder conectarse a los servidores de catálogo de otros hosts. En la lista, uno de los

programas de utilidad **xscmd** se conecta a un host aleatorio. La lista de hosts que proporcione debe estar dentro del mismo dominio de servicio de catálogo.

- Proporcione una lista de servidores de catálogo autónomos a conectar:
  - UNIX `./xscmd.sh -c <nombre_mandato> -cep nombrehost:puerto(,nombrehost:puerto)`
  - Windows `xscmd.bat -c <nombre_mandato> -cep nombrehost:puerto(,nombrehost:puerto)`

  En los mandatos anteriores, *nombre_mandato* es el nombre del mandato que está ejecutando. El valor de *nombrehost:puerto* es el nombre de host de servidor de catálogo y el puerto de escucha. El valor de puerto de escucha en un servidor de catálogo autónomo se especifica cuando se ejecuta el mandato **startOgServer**.
- Proporcione una lista de servidores de catálogo de WebSphere Application Server que se deben conectar. No se puede conectar a servidores de catálogo que se ejecutan en WebSphere Application Server con el valor de hostlocal predeterminado:
  - UNIX `./xscmd.sh -c <nombre_mandato> -cep nombrehost_was:puerto(,nombrehost:puerto)`
  - Windows `xscmd.bat -c <nombre_mandato> -cep nombrehost_was:puerto(,nombrehost:puerto)`

  En los mandatos anteriores, *nombre_mandato* es el nombre del mandato que está ejecutando. El valor de *nombrehost_was* es el nombre de host del servidor de catálogo en la célula de WebSphere Application Server. El valor *puerto* es el puerto de escucha. El valor de puerto de escucha en WebSphere Application Server lo hereda la configuración de puerto BOOTSTRAP_ADDRESS. El valor predeterminado es `9809` si el servidor de catálogo se ejecuta en el gestor de despliegue. Si está ejecutando el servidor de catálogo en un servidor de aplicaciones, compruebe la configuración de puerto BOOTSTRAP_ADDRESS del servidor de aplicaciones para determinar el número de puerto.

Inicio de servidores eXtreme Scale utilizando la infraestructura OSGi de Eclipse Equinox

Los servidores de contenedor de WebSphere eXtreme Scale se pueden iniciar en una infraestructura OSGi de Eclipse Equinox utilizando varios métodos.

Antes de empezar

Para poder iniciar un contenedor eXtreme Scale, debe haber completado las siguientes tareas:

1. El paquete de servidor de WebSphere eXtreme Scale debe estar instalado en Eclipse Equinox.
2. La aplicación debe estar empaquetado como un paquete OSGi.
3. Los plug-ins de WebSphere eXtreme Scale (si existen) deben estar empaquetados como un paquete OSGi. Pueden estar empaquetados en el mismo paquete que la aplicación o como paquetes independientes.

Acerca de esta tarea

Esta tarea describe cómo iniciar un servidor de contenedor eXtreme Scale en una infraestructura OSGi de Eclipse Equinox. Puede utilizar cualquiera de los métodos siguientes para iniciar los servidores de contenedor utilizando la implementación de Eclipse Equinox:

- Servicio OSGi Blueprint

  Puede incluir toda la configuración y los metadatos en un paquete OSGi. Consulte la imagen siguiente para comprender el proceso de Eclipse Equinox para este método:

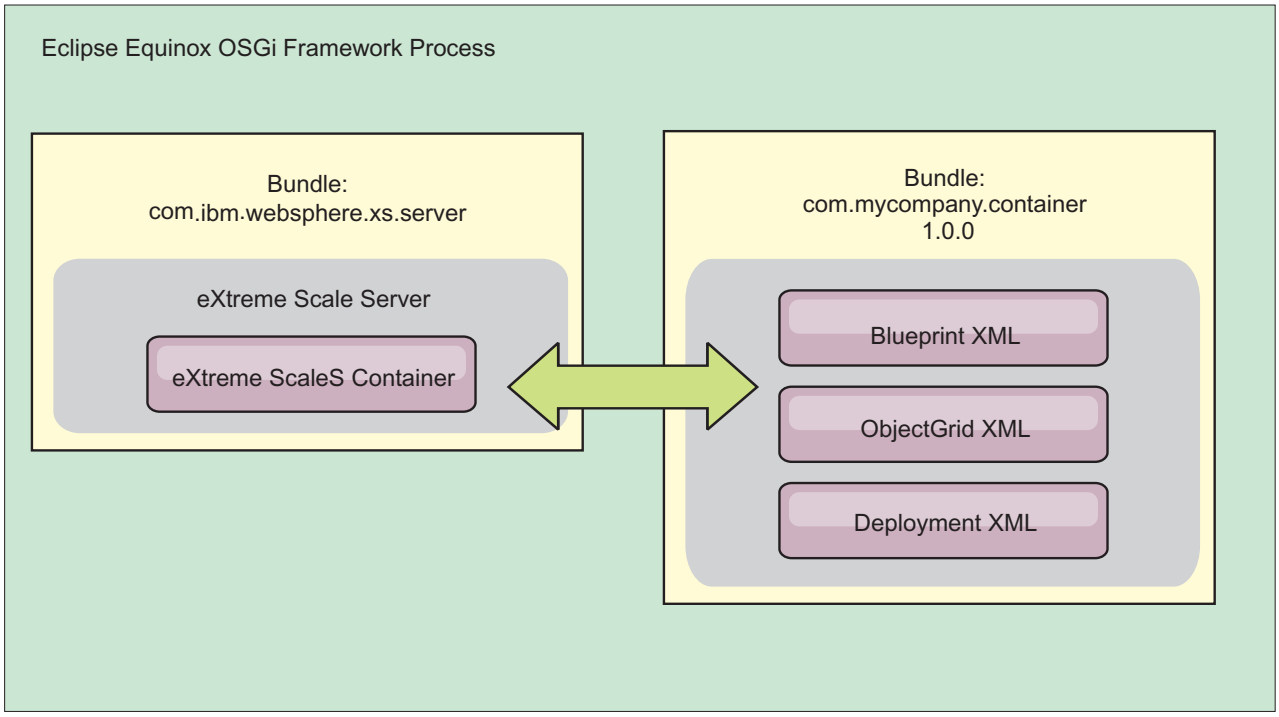


Figura 49. Proceso de Eclipse Equinox para incluir toda la configuración y los metadatos en un paquete OSGi

- Servicio de administración de configuración OSGi

  Puede especificar la configuración y los metadatos fuera de un paquete OSGi. Consulte la imagen siguiente para comprender el proceso de Eclipse Equinox para este método:

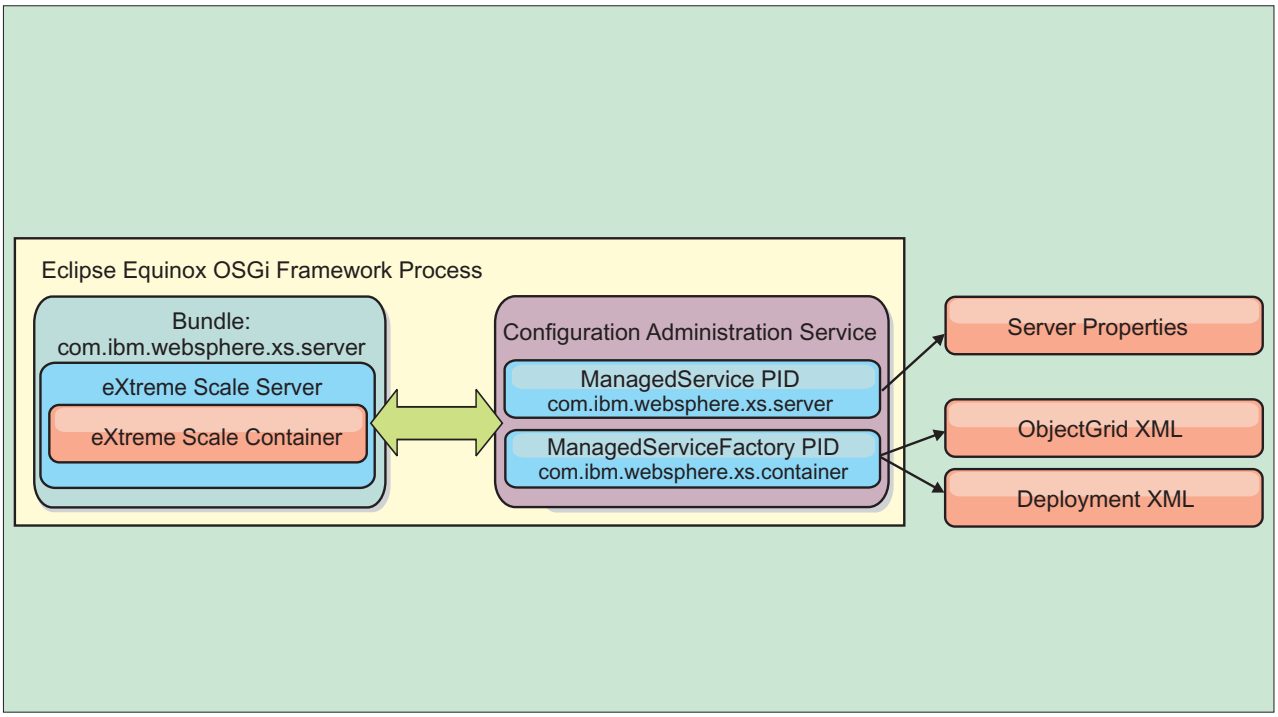


Figura 50. Proceso de Eclipse Equinox para especificar la configuración y los metadatos fuera de un paquete OSGi

- A través de programas
  Soporta soluciones de configuración personalizadas.

En cada caso, se configura un singleton de servidor eXtreme Scale y se configuran uno o varios contenedores.

El paquete de servidor eXtreme Scale, `objectgrid.jar`, incluye todas las bibliotecas necesarias para iniciar y ejecutar un contenedor de cuadrícula de eXtreme Scale en una infraestructura OSGi. El entorno de ejecución de servidor se comunica con los objetos de datos y los plug-ins proporcionados por el usuario utilizando el administrador de servicios OSGi.

Importante: Después de que un paquete de servidor eXtreme Scale se haya iniciado y el servidor eXtreme Scale se haya inicializado, no se puede reiniciar. Se debe reiniciar el proceso de Eclipse Equinox para reiniciar un servidor eXtreme Scale.

Puede utilizar el soporte de eXtreme Scale para el espacio de nombres Spring para configurar los servidores de contenedor de eXtreme Scale en un archivo XML Blueprint. Cuando se añaden los elementos XML de servidor y contenedor al archivo XML Blueprint, el manejador de espacio de nombres de eXtreme Scale inicia automáticamente un servidor de contenedor utilizando los parámetros definidos en el archivo XML Blueprint cuando se inicia el paquete. El manejador detiene el contenedor cuando se detiene el paquete.

Para configurar servidores de contenedor de eXtreme Scale con XML Blueprint, realice los pasos siguientes:

Procedimiento

- Inicie un servidor de contenedor de eXtreme Scale utilizando OSGi Blueprint.
  1. Cree un paquete de contenedor.
  2. Instale el paquete de contenedor en la infraestructura OSGi de Eclipse Equinox. Consulte "Instalación e inicio de plug-ins habilitados para OSGi".
  3. Inicie el paquete de contenedor.
- Inicie un servidor de contenedor de eXtreme Scale utilizando la administración de configuración de OSGi.
  1. Configure el servidor y el contenedor utilizando la administración de configuración.
  2. Cuando el paquete de servidor de eXtreme Scale se ha iniciado o los identificadores persistentes se crean con la administración de configuración, el servidor y el contenedor se inician automáticamente.
- Inicie un servidor de contenedor de eXtreme Scale utilizando la API ServerFactory. Consulte la documentación de API de servidor.
  1. Cree una clase de activador de paquete OSGi y utilice la API ServerFactory de eXtreme Scale para iniciar un servidor.

Instalación e inicio de plug-ins habilitados para OSGi

En esta tarea, instalará el paquete de plug-in dinámico en la infraestructura OSGi. A continuación, iniciará el plug-in.

Antes de empezar

En este tema se supone que se han completado las tareas siguientes:

- Se ha instalado el paquete de servidor o cliente de eXtreme Scale en la infraestructura OSGi de Eclipse Equinox. Consulte "Instalación de paquetes de eXtreme Scale" en la página 204.
- Se han implementado uno o varios plug-ins dinámicos de BackingMap u ObjectGrid. Consulte Creación de plug-ins dinámicos de eXtreme Scale.
- Los plug-ins dinámicos se han empaquetado como servicios OSGi en paquetes OSGi.

Acerca de esta tarea

Esta tarea describe cómo instalar el paquete utilizando la consola Eclipse Equinox. El paquete se puede instalar utilizando varios métodos diferentes, incluida la modificación del archivo de configuración `config.ini`. Los productos que incorporan Eclipse Equinox incluyen métodos alternativos para gestionar paquetes. Para obtener más información sobre cómo añadir paquetes en el archivo `config.ini` de Eclipse Equinox, consulte los opciones de ejecución de Eclipse.

OSGi permite que se inicien paquetes que tienen servicios duplicados. WebSphere eXtreme Scale utiliza la clasificación de servicios más reciente. Al iniciar varias infraestructuras OSGi en una cuadrícula de datos de eXtreme Scale, debe asegurarse de que se inician las clasificaciones de servicio correctas en cada servidor. Si no es así, la cuadrícula se inicia con una mezcla de versiones diferentes.

Para ver qué versiones están siendo utilizadas por la cuadrícula de datos, utilice el programa de utilidad xscmd para comprobar las clasificaciones actuales y

disponibles. Para obtener más información sobre las clasificaciones de servicio disponibles, consulte "Actualización de servicios OSGi para plug-ins de eXtreme Scale con **xscmd**" en la página 425.

Procedimiento

Instalar el paquete de plug-in en la infraestructura OSGi de Eclipse Equinox utilizando la consola OSGi.

1. Inicie la infraestructura de Eclipse Equinox con la consola habilitada; por ejemplo:

   ```
   <inicio_java>/bin/java -jar <raíz_equinox>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```

2. Instale el paquete de plug-in en la consola de Equinox.

   ```
   osgi> install file:///<vía_acceso_archivo>
   ```

   Equinox visualiza el ID de paquete para el paquete recién instalado:

   ```
   Bundle id is 17
   ```

3. Entre la línea siguiente para iniciar el paquete en la consola de Equinox, donde `<id>` es el ID de paquete asignado al instalar el paquete:

   ```
   osgi>  install <id>
   ```

4. Recupere el estado de servicio en la consola de Equinox para verificar que el paquete se ha iniciado:

   ```
   osgi> ss
   ```

   Cuando el paquete se ha iniciado satisfactoriamente, visualiza el estado ACTIVO; por ejemplo:

   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Instalar el paquete de plug-in en la infraestructura OSGi de Eclipse Equonix utilizando el archivo config.ini.

5. Copie el paquete de plug-in en el directorio de plug-ins de Eclipse Equinox; por ejemplo:

   ```
   <raíz_equinox>/plugins
   ```

6. Edite el archivo de configuración `config.ini` de Eclipse Equinox y añada el paquete a la propiedad osgi.bundles; por ejemplo:

   ```
   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
   ```

   Importante: Verifique que haya una línea en blanco después del último nombre de paquete. Cada paquete está separado por una coma.

7. Inicie la infraestructura de Eclipse Equinox con la consola habilitada; por ejemplo:

   ```
   <inicio_java>/bin/java -jar <raíz_equinox>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
   ```

8. Recupere el estado de servicio en la consola de Equinox para verificar que el paquete se ha iniciado; por ejemplo:

   ```
   osgi> ss
   ```

   Cuando el paquete se ha iniciado satisfactoriamente, visualiza el estado ACTIVO; por ejemplo:

   ```
   17      ACTIVE      com.mycompany.plugin.bundle_VRM
   ```

Resultados

El paquete de plug-in ya está instalado e iniciado. Ahora ya se puede iniciar el contenedor o cliente de eXtreme Scale. Para obtener más información sobre el desarrollo de plug-ins de eXtreme Scale, consulte el tema Plug-ins y API del sistema.

Administración de servicios habilitado para OSGi utilizando el programa de utilidad `xscmd`

Puede utilizar el programa de utilidad **xscmd** para completar las tareas de administrador, por ejemplo ver los servicios y sus clasificaciones que están siendo utilizados por cada contenedor y actualizar el entorno de ejecución para utilizar las nuevas versiones de los paquetes.

Acerca de esta tarea

Con la infraestructura OSGi de Eclipse Equinox, puede instalar varias versiones del mismo paquete y puede actualizar esos paquetes durante la ejecución. WebSphere eXtreme Scale es un entorno distribuido que ejecuta los servidores de contenedor en muchas instancias de infraestructura OSGi.

Los administradores son responsables de copiar, instalar e iniciar manualmente paquetes en la infraestructura OSGi. eXtreme Scale incluye un ServiceTrackerCustomizer OSGi para realizar un seguimiento de los servicios que se han identificado como plug-ins de eXtreme Scale en el archivo XML de descriptor de ObjectGrid. Utilice el programa de utilidad **xscmd** para validar qué versión del plug-in se utiliza, qué versiones están disponibles para utilizarse y para realizar actualizaciones de paquete.

eXtreme Scale utiliza el número de clasificación de servicio para identificar la versión de cada servicio. Cuando se cargan dos o más servicios con la misma referencia, eXtreme Scale utiliza automáticamente el servicio con la clasificación más alta.

Procedimiento

- Ejecute el mandato **osgiCurrent** y verifique que cada servidor de eXtreme Scale utiliza la clasificación de servicio de plug-in correcta.

  Dado que eXtreme Scale elige automáticamente la referencia de servicio con la clasificación más alta, es posible que la cuadrícula de datos empiece con varias clasificaciones de un servicio de plug-in.

  Si el mandato detecta una discrepancia de clasificaciones o si no puede encontrar un servicio, se establece un nivel de error distinto de cero. Si el mandato se ha completado satisfactoriamente, el nivel de error se establece en 0.

  El siguiente ejemplo muestra la salida del mandato **osgiCurrent** cuando dos plug-ins se instalan en la misma cuadrícula en cuatro servidores. El plug-in loaderPlugin utiliza la clasificación 1 y txCallbackPlugin utiliza la clasificación 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
```

El siguiente ejemplo muestra la salida del mandato **osgiCurrent** cuando server2 se ha iniciado con una clasificación más reciente de loaderPlugin:

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
----------------- --------------- --------------- ----------- -----------
loaderPlugin      1               MyGrid          MapSetA     server1
loaderPlugin      2               MyGrid          MapSetA     server2
loaderPlugin      1               MyGrid          MapSetA     server3
loaderPlugin      1               MyGrid          MapSetA     server4
txCallbackPlugin  2               MyGrid          MapSetA     server1
txCallbackPlugin  2               MyGrid          MapSetA     server2
txCallbackPlugin  2               MyGrid          MapSetA     server3
txCallbackPlugin  2               MyGrid          MapSetA     server4
```

- Ejecute el mandato **osgiAll** para verificar que los servicios de plug-in se han iniciado correctamente en cada servidor de contenedor de eXtreme Scale.

  Al iniciar paquetes que contienen servicios a los que una configuración ObjectGrid hace referencia, el entorno de ejecución de eXtreme Scale realiza automáticamente un seguimiento del plug-in, pero no lo utiliza inmediatamente. El mandato **osgiAll** muestra qué plug-ins están disponibles para cada servidor.

  Cuando se ejecuta sin parámetros, se muestran todos los servicios para todas las cuadrículas y servidores. Se pueden especificar filtros adicionales, incluido el filtro **-serviceName** <nombre_servicio>, para limitar la salida a un solo servicio o a un subconjunto de la cuadrícula de datos.

  El ejemplo siguiente muestra la salida del mandato **osgiAll** cuando se inician dos plug-ins en dos servidores. loaderPlugin tiene las dos clasificaciones 1 y 2 iniciadas y txCallbackPlugin tiene la clasificación 1 iniciada. El mensaje de resumen al final de la salida confirma que ambos servidores ven las mismas clasificaciones de servicio:

```
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.
```

  El ejemplo siguiente muestra la salida del mandato **osgiAll** cuando el paquete que incluye loaderPlugin con la clasificación 1 se detiene en server1. El mensaje de resumen en la parte inferior de la salida confirma que en server1 falta ahora loaderPlugin con la clasificación 1:

```
Server: server1
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        2
   txCallbackPlugin    1

Server: server2
   OSGi Service Name   Available Rankings
   -----------------   ------------------
   loaderPlugin        1, 2
   txCallbackPlugin    1
```

```
Summary - The following servers are missing service rankings:
   Server  OSGi Service Name Missing Rankings
   ------  ----------------- ----------------
   server1 loaderPlugin      1
```

El siguiente ejemplo muestra la salida si el nombre de servicio se especifica con el argumento **-sn**, pero el servicio no existe:

```
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.
```

- Ejecute el mandato **osgiCheck** para comprobar conjuntos de servicios de plug-in y clasificaciones para ver si están disponibles.

  El mandato **osgiCheck** acepta uno o más conjuntos de clasificaciones de servicio en el formato: `-serviceRankings <nombre de servicio>;<clasificación>[,<nombreServicio>;<clasificación>]`

  Cuando las clasificaciones están todas disponibles, el método vuelve con un nivel de error de 0. Si una o más clasificaciones no están disponibles, se establece un nivel de error distinto de cero y una tabla de todos los servidores que no incluyen las clasificaciones de servicio especificadas. Se pueden utilizar filtros adicionales para limitar la comprobación de servicio a un subconjunto de los servidores disponibles en el dominio de eXtreme Scale.

  Por ejemplo, si la clasificación o el servicio especificados están ausentes, se visualiza el siguiente mensaje:

  ```
  Server  OSGi Service Unavailable Rankings
  ------  ------------ --------------------
  server1 loaderPlugin 3
  server2 loaderPlugin 3
  ```

- Ejecute el mandato **osgiUpdate** para actualizar la clasificación de uno o más plug-ins para todos los servidores de una sola ObjectGrid y MapSet en una sola operación.

  El mandato acepta uno o más conjuntos de clasificaciones de servicio con el formato: `-serviceRankings <nombre de servicio>;<clasificación>[,<nombreServicio>;<clasificación>] -g <nombre de cuadrícula> -ms <nombre de conjunto de correlaciones>`

  Con este mandato, puede completar las siguientes operaciones:

  – Verifique que los servicios especificados están disponibles para actualizarse en cada uno de los servidores.

  – Cambie el estado de la cuadrícula a fuera de línea utilizando la interfaz StateManager. Si desea más información, consulte “Gestión de la disponibilidad del ObjectGrid” en la página 430. Este proceso inmoviliza la cuadrícula, espera a que se hayan completado las transacciones en ejecución e impide que se inicien transacciones nuevas. Este proceso también señala a los plug-ins ObjectGridLifecycleListener y BackingMapLifecycleListener que interrumpan cualquier actividad transaccional. Consulte Plug-ins para proporcionar escuchas de sucesos para obtener información sobre los plug-ins de escucha de sucesos.

  – Actualice cada contenedor de eXtreme Scale que se ejecuta en una infraestructura OSGi para utilizar las nuevas versiones de servicio.

– Cambie el estado de la cuadrícula para que esté en línea, lo que permite que continúen las transacciones.

El proceso de actualización es idempotent, de modo que si un cliente no puede completar ninguna tarea, la operación se retrotrae. Si un cliente no puede realizar la retrotracción o se interrumpe durante el proceso de actualización, se puede emitir el mismo mandato de nuevo y continúa en el paso adecuado.

Si el cliente no puede continuar y el proceso se reinicia desde otro cliente, utilice la opción `-force` para permitir que el cliente realice la actualización. El mandato **`osgiUpdate`** impide que varios clientes actualicen el mismo conjunto de correlaciones simultáneamente. Para obtener más detalles sobre el mandato **`osgiUpdate`**, consulte "Actualización de servicios OSGi para plug-ins de eXtreme Scale con **`xscmd`**".

Actualización de servicios OSGi para plug-ins de eXtreme Scale con `xscmd`

WebSphere eXtreme Scale soporta la actualización de paquetes de plug-in de servidor de contenedor mientras la cuadrícula está activa. Este soporte permite a los administradores realizar actualizaciones de aplicación y adiciones sin necesidad de reiniciar los procesos de cuadrícula.

Antes de empezar

Complete los pasos siguientes antes de actualizar paquetes OSGi de eXtreme Scale a una nueva versión:

1. Inicie los servidores de eXtreme Scale en una infraestructura OSGi soportada.
2. Separe todos los plug-ins de eXtreme Scale en paquetes y éstos deben utilizar clasificaciones de servicio para identificar cada versión de los plug-ins.
3. Especifique objetos de memoria caché como tipo de primitivos Java como `byte[]`, Integer o String, o se deben almacenar utilizando un plug-in MapSerializerPlugin. Los objetos de datos se almacenan en el paquete eXtreme Scale y no se actualizan. Sólo se actualizan los plug-ins que interactúan con los datos.
4. Diseñe los datos de objeto de memoria caché de forma que sean compatibles con las versiones. Los plug-ins nuevos deben poder interactuar con los datos creados por los plug-ins antiguos.
5. Diseñe plug-ins para escuchar los sucesos ObjectGridLifecycle y BackingMapLifecycle para renovar las referencias a otros plug-ins o los metadatos que los plug-ins pueden tener para que se puedan renovar cuando se actualiza.
6. El proceso de actualización OSGi de eXtreme Scale sólo afecta a los servidores. Debe actualizar de forma independiente los clientes que utilizan plug-ins.

Acerca de esta tarea

Sin la habilitación OSGi, si un administrador necesita actualizar plug-ins de aplicación u objetos de memoria caché, se debe actualizar uno a a uno cada nodo de cuadrícula, lo que produce estrés en la red, la memoria y la utilización de cpu. Esto es necesario puesto que los objetos Java de memoria caché y plug-ins se almacenan directamente en la cuadrícula. Cuando las clases se actualizan sin reiniciar los procesos, los plug-ins de cuadrícula tienen conflictos porque cada clase tiene un ClassLoader diferente.

El producto eXtreme Scale incluye el programa de utilidad xscmd y MBeans lo que permite a los administradores ver todos los paquetes de plug-in instalados en infraestructura OSGi de alojamiento de cada contenedor de cuadrícula y elegir qué revisión va a utilizar. Cuando se utiliza xscmd para actualizar los plug-ins a una nueva clasificación, se inmoviliza la cuadrícula y se drenan todas las transacciones, se actualizan los plug-ins y la cuadrícula se activa de nuevo. Si se produce un error durante el proceso de actualización, el proceso se retrotrae y la clasificación antigua se restaura.

Procedimiento

1. Cree una versión del paquete, aumentando el número de versión en el manifiesto de paquete y aumentando la clasificación para cada servicio de plug-in eXtreme Scale. Si la versión de paquete original es `Bundle-Version: 1.0.0`, la siguiente versión se puede definir como `Bundle-Version: 1.1.0`.

   Si la clasificación de servicio original es `ranking="1"`, la siguiente clasificación se puede definir como `ranking="2"`.

   Importante: Las clasificaciones de servicio de OSGi deben ser enteros.
2. Copie el nuevo paquete en cada nodo de infraestructura OSGi que aloja un servidor de contenedor de eXtreme Scale.
3. Instale el nuevo paquete en la infraestructura OSGi. Se le asigna al paquete un identificador de paquete; por ejemplo:

   ```
   osgi> install <URL a paquete>
   ```
4. Inicie el nuevo paquete utilizando el identificador de paquete asignado; por ejemplo:

   ```
   osgi> start <id>
   ```

   Después de que se haya iniciado el nuevo paquete, el rastreador de servicio OSGi de eXtreme Scale detecta el paquete y lo deja disponible para la actualización.
5. Utilice el mandato **xscmd -c osgiAll** para verificar que cada servidor de contenedor ve el nuevo paquete. El mandato **osgiAll** consulta en todos los contenedores de la cuadrícula todos los servicios a los que se hace referencia en el archivo XML de descriptor de ObjectGrid y visualiza todas las clasificaciones que están disponibles; por ejemplo:

   ```
   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. Utilice el mandato **xscmd -c osgiCheck** para verificar que una o más clasificaciones de servicio son destinos de actualización válidos; por ejemplo:

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```

7. Si el mandato **osgiCheck** no ha encontrado errores resultantes, suspenda el equilibrador del servicio de colocación para evitar movimientos de fragmentos, si se produjera una anomalía durante el proceso de actualización. Para suspender la colocación, utilice el mandato **xscmd -c suspendBalancing** para cada cuadrícula de objeto y conjunto de correlaciones afectados por la actualización; por ejemplo:

   ```
   xscmd -c suspendBalancing -g MyGrid -ms MyMapSet
   ```

8. Después de que se haya suspendido el equilibrio para cada cuadrícula de objeto y conjunto de correlaciones, utilice el mandato **xscmd -c osgiCheck** de nuevo para verificar que una o más clasificaciones de servicio son destinos de actualización válidos; por ejemplo:

   ```
   xscmd -c osgiCheck -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2

   CWXSI0040I: The command osgiCheck has completed successfully.
   ```

9. Después de que se haya suspendido el equilibrio para la cuadrícula de objeto y el conjunto de correlaciones, utilice el mandato **osgiUpdate** para actualizar el servicio en todos los servidores para una cuadrícula de objeto y un conjunto de correlaciones; por ejemplo:

   ```
   xscmd -c osgiUpdate  -sr
   mySerializerServiceFactory;2,myLoaderServiceFactory;2 -g MyGrid -ms MyMapSet
   ```

10. Verifique que la actualización se haya realizado satisfactoriamente; por ejemplo:

    ```
    La actualización ha sido satisfactoria para las siguientes clasificaciones
    de servicio:
    Service                    Ranking
    -------                    -------
    mySerializerServiceFactory  2
    myLoaderServiceFactory      2
    ```

11. Después de verificar que la clasificación se ha actualizado satisfactoriamente, habilite el equilibrado de nuevo, utilizando el mandato **xscmd -c resumeBalancing**; por ejemplo:

    ```
    xscmd -c resumeBalancing -g MyGrid -ms MyMapSet
    ```

12. Detenga y desinstale el paquete antiguo en cada infraestructura OSGi que aloja el contenedor de eXtreme Scale. Por ejemplo, entre el código siguiente en la consola de Eclipse Equinox:

    ```
    osgi> stop <id>
    osgi> uninstall <id>
    ```

Resultados

El paquete eXtreme Scale se ha actualizado a una versión nueva.

Control de la colocación

Puede utilizar varias opciones diferentes para controlar cuándo se colocan fragmentos en varios servidores de contenedor en la configuración. Durante el arranque, puede elegir retardar la colocación de fragmentos. Cuando ejecuta todos los servidores de contenedor, es posible que necesite suspender, reanudar o cambiar la colocación mientras mantiene los servidores.

Procedimiento

Control de la colocación durante el arranque
Puede controlar cuándo empiezan a colocarse los fragmentos mientras se está

iniciando el entorno. De forma predeterminada existe un cierto control. Si no realiza ninguna acción para controlar la colocación, los fragmentos empiezan a colocarse inmediatamente. Cuando los fragmentos se colocan inmediatamente, es posible que no se coloquen de manera uniforme ya que se inician los siguientes servidores de contenedor y se ejecutan operaciones de colocación adicionales para equilibrar la distribución.

- Suspenda temporalmente el equilibrio de fragmentos para evitar la colocación de fragmentos inmediata cuando se inician los servidores de contenedor.

  Antes de iniciar los servidores de contenedor, utilice el mandato **xscmd -c suspendBalancing** para detener el equilibrio de fragmentos de una cuadrícula de datos y un conjunto de correlaciones específicos. Después de que se hayan iniciado los servidores de contenedor, puede utilizar el mandato **xscmd -c resumeBalancing** para empezar la colocación de fragmentos en los servidores de contenedor.

- 7.1.1+ Configure la propiedad **placementDeferralInterval**.

  La propiedad **placementDeferralInterval** minimiza el número de ciclos de colocación de fragmentos en los servidores de contenedor. La colocación de fragmentos se desencadena en el intervalo de tiempo definido.

  Establezca la propiedad **placementDeferralInterval** en el archivo de propiedades de servidor para el servidor de catálogo. Si está utilizando la API de servidor incorporado, utilice el método setPlacementDeferralInterval en la interfaz CatalogServerProperties. Esta propiedad establece un número de milisegundos antes de que se coloquen los fragmentos en los servidores de contenedor. El valor predeterminado de esta propiedad es 15 segundos. Con el valor predeterminado, cuando se inicia un servidor de contenedor, la colocación no se inicia hasta después de que haya transcurrido el tiempo especificado en la propiedad. Si varios servidores de contenedor se inician en sucesión, se restablece el temporizador de intervalo de aplazamiento si se inicia un nuevo servidor de contenedor dentro del intervalo proporcionado. Por ejemplo, si se inicia un segundo servidor de contenedor 10 segundos después del primer servidor de contenedor, la colocación no se inicia hasta 15 segundos después de que se haya iniciado el segundo servidor de contenedor. Sin embargo, si se inicia un tercer servidor de contenedor 20 segundos después del segundo servidor de contenedor, la colocación ya habrá empezado en los dos primeros servidores de contenedor.

  Cuando los servidores de contenedor dejan de estar disponibles, se desencadena la colocación tan pronto como el servidor de catálogo conoce el suceso para que la recuperación se pueda producir lo más rápidamente posible.

  Puede utilizar los siguientes consejos para determinar si el valor de aplazamiento de colocación se ha establecido en un periodo de tiempo correcto:

  – Mientras inicia simultáneamente los servidores de contenedor, examine los mensajes CWOBJ1001 en el archivo `SystemOut.log` de cada servidor de contenedor. La indicación de fecha y hora de estos mensajes en cada archivo de registro de servidor de contenedor indica la hora de inicio de servidor de contenedor real. Puede considerar la posibilidad de ajustar la propiedad **placementDeferralInterval** para incluir más inicios de servidor de contenedor. Por ejemplo, si el primer servidor de contenedor se inicia 90 segundos antes del último servidor de contenedor, puede establecer la propiedad en 90 segundos.
  – Fíjese cuánto tiempo transcurre antes de que se produzcan los mensajes CWOBJ1511 después de los mensajes CWOBJ1001. Este intervalo de tiempo puede indicar si el aplazamiento se ha producido satisfactoriamente.

  - Si está utilizando un entorno de desarrollo, tenga en cuenta la duración del intervalo cuando está probando la aplicación.
- Configure el atributo **numInitialContainers**.

  Si ha utilizado anteriormente el atributo **numInitialContainers**, puede continuar utilizando el atributo. Sin embargo, se sugiere utilizar los mandatos **xscmd -c suspendBalancing** y **xscmd -c resumeBalancing** seguidos de **placementDeferralInterval** en lugar del atributo **numInitialContainers** para controlar la colocación. El atributo **numInitialContainers** especifica el número de servidores de contenedor que son necesarios antes de que se produzca la colocación inicial para los fragmentos de este elemento mapSet. El atributo **numInitialContainers** está en el archivo XML de descriptor de política de despliegue. Si ha establecido **numInitialContainers** y **placementDeferralInterval**, tenga en cuenta que hasta que se satisfaga el valor **numInitialContainers**, no se producirá ninguna colocación, independientemente del valor de la propiedad **placementDeferralInterval**.

Control de la colocación tras el arranque inicial

- Fuerce que se produzca la colocación.

  Puede utilizar el mandato **xscmd -c triggerPlacement -g my_OG -ms my_Map_Set**, donde *my_OG* y *my_Map_Set* se establecen en los valores para la cuadrícula de datos y el conjunto de correlaciones, con el fin de forzar que se produzca la colocación en un momento en el que, de lo contrario, es posible que la colocación no se produjera. Por ejemplo, puede ejecutar este mandato cuando el periodo de tiempo especificado por la propiedad **placementDeferralInterval** aún no ha transcurrido o cuando se suspende el equilibrio.
- Reasigne un fragmento primario.

  Utilice el mandato **xscmd -c swapShardWithPrimary** para asignar un fragmento de réplica para que sea el nuevo fragmento primario. El fragmento primario anterior se convierte en una réplica.
- Vuelva a equilibrar los fragmentos primario y de réplica.

  Utilice el mandato **xscmd -c balanceShardTypes** para ajustar la proporción de fragmentos primarios y de réplica a fin de que sea equitativa entre los servidores de contenedor en ejecución de la configuración. La proporción es coherente dentro de un fragmento de cada servidor de contenedor.
- Suspenda o reanude la colocación.

  Utilice el mandato **xscmd -c suspendBalancing** o el mandato **xscmd -c resumeBalancing** para detener e iniciar el equilibrio de fragmentos de una cuadrícula de datos y un conjunto de correlaciones específicos. Cuando el equilibrado se ha suspendido, se pueden seguir ejecutando las siguientes acciones de colocación:
  - Se puede producir la promoción de fragmentos cuando fallan los servidores de contenedor.
  - Intercambio de roles de fragmento con el mandato **xscmd -c swapShardWithPrimary**.
  - Equilibrio desencadenado por colocación de fragmento con el mandato **xscmd -c triggerPlacement -g myOG -ms myMapSet**.

Qué hacer a continuación

Puede supervisar la colocación en el entorno con el mandato **xscmd -c placementServiceStatus**.

Gestión de la disponibilidad del ObjectGrid

El estado de disponibilidad de una instancia de ObjectGrid determina qué peticiones se pueden procesar en un momento dado. Puede utilizar la interfaz StateManager para establecer y recuperar el estado de una instancia de ObjectGrid.

Acerca de esta tarea

Existen cuatro estados de disponibilidad para una instancia determinada de ObjectGrid.

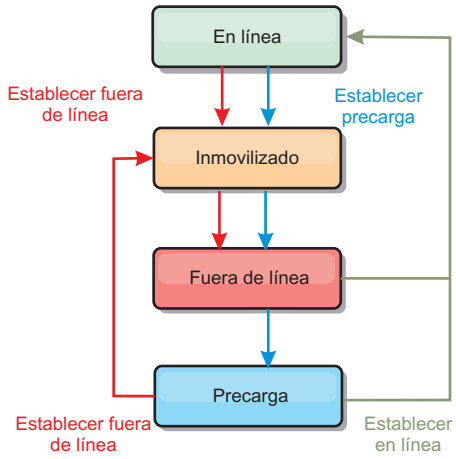


Figura 51. Estados de disponibilidad de una instancia de ObjectGrid

ONLINE
El estado ONLINE es el estado de disponibilidad predeterminado de un ObjectGrid. Un ObjectGrid ONLINE puede procesar las peticiones de un cliente típico de eXtreme Scale. Sin embargo, las solicitudes de un cliente de precarga se rechazan mientras el ObjectGrid está ONLINE.

QUIESCE
El estado QUIESCE es de transición. Un ObjectGrid que está en QUIESCE pronto pasará al estado OFFLINE. Mientras está en el estado QUIESCE, un ObjectGrid puede procesar transacciones pendientes. Sin embargo, las nuevas transacciones se rechazan. un ObjectGrid puede permanecer en QUIESCE hasta 30 segundos. Una vez transcurrido este intervalo, el estado de disponibilidad pasa a ser OFFLINE.

OFFLINE
El estado OFFLINE produce el rechazo de todas las transacciones que se envían al ObjectGrid.

PRELOAD
El estado PRELOAD se puede utilizar para cargar datos en ObjectGrid desde un cliente de precarga. Mientras ObjectGrid está en estado PRELOAD, sólo un cliente de precarga puede confirmar transacciones respecto a ObjectGrid. Todas las demás transacciones se rechazan.

Se rechaza una solicitud si un ObjectGrid no está en el estado de disponibilidad adecuado para dar soporte a esta solicitud. Se genera una excepción AvailabilityException siempre que se rechaza una solicitud.

Procedimiento

1. Establezca el estado inicial de un ObjectGrid con el archivo XML de configuración de ObjectGrid.

   Puede utilizar el atributo **initialState** en un ObjectGrid para indicar su estado inicial. Normalmente, cuando un ObjectGrid finaliza su inicialización, está disponible para el direccionamiento. El estado podrá cambiarse más adelante para impedir que el tráfico se direccione a un ObjectGrid. Si es necesario inicializar el ObjectGrid, pero no está inmediatamente disponible, puede utilizar el atributo **initialState**.

   El atributo initialState se establece en el archivo XML de configuración de ObjectGrid. El estado predeterminado es ONLINE. Los valores válidos son:

   - ONLINE (valor predeterminado)
   - PRELOAD
   - OFFLINE

   Consulte Archivo XML de descriptor ObjectGrid para obtener más información sobre el atributo **initialState**.

   Si el atributo initialState se establece en un ObjectGrid, el estado se debe establecer explícitamente de nuevo en línea o el ObjectGrid continuará estando no disponible. Se genera una excepción AvailabilityException si el ObjectGrid no está en el estado ONLINE.

   Consulte la documentación de la API AvailabilityState para obtener más información.

   Utilización del atributo initialState para la precarga

   Si el ObjectGrid se precarga con datos, puede haber un periodo de tiempo entre el momento en que el ObjectGrid está disponible y el cambio a un estado de precarga para bloquear el tráfico del cliente. Para evitar este periodo de tiempo, el estado inicial en un ObjectGrid se puede establecer en PRELOAD. El ObjectGrid finalizará toda la inicialización necesaria, pero bloqueará el tráfico hasta que el estado cambie y permita que se produzca la precarga.

   Los estados PRELOAD y OFFLINE bloquean el tráfico, pero debe utilizarse el estado PRELOAD si desea iniciar una precarga.

   Migración tras error y equilibrio

   Si una cuadrícula de datos de réplica pasa a ser una cuadrícula de datos primaria, la réplica no utiliza el valor **initialState**. Si la cuadrícula de datos primaria se traslada para realizar un reequilibrado, el valor **initialState** no se utiliza porque los datos se copian en la nueva ubicación primaria antes de que se complete el traslado. Si la réplica no se configura, la cuadrícula primaria pasa al valor **initialState** si se produce una migración tras error y se debe colocar una nueva cuadrícula primaria.

2. Cambie el estado de disponibilidad con la interfaz StateManager.

   Utilice la interfaz StateManager para establecer el estado de disponibilidad de un ObjectGrid. Para definir el estado de disponibilidad de un ObjectGrid que se ejecuta en los servidores, pase un cliente ObjectGrid correspondiente a la interfaz StateManager. El siguiente código demuestra cómo cambiar el estado de disponibilidad de un ObjectGrid.

   ```
   ClientClusterContext client = ogManager.connect("localhost:2809", null, null);
   ObjectGrid myObjectGrid = ogManager.getObjectGrid(client, "myObjectGrid");
   StateManager stateManager = StateManagerFactory.getStateManager();
   stateManager.setObjectGridState(AvailabilityState.OFFLINE, myObjectGrid);
   ```

Cada fragmento del ObjectGrid realiza la transición del estado deseado cuando se llama al método setObjectGridState en la interfaz StateManager. Cuando se devuelve el método, todos los fragmentos del ObjectGrid deben estar en el estado adecuado.

Utilice un plug-in ObjectGridEventListener para cambiar el estado de disponibilidad de un ObjectGrid del lado del servidor. Cambie sólo el estado de disponibilidad de un ObjectGrid del lado del servidor, si ObjectGrid tiene una única partición. Si el ObjectGrid tiene varias particiones, se llama al método shardActivated en cada fragmento primario, que genera llamadas superfluas para cambiar el estado del ObjectGrid

```
public class OGListener implements ObjectGridEventListener,
   ObjectGridEventGroup.ShardEvents {
    public void shardActivated(ObjectGrid grid) {
            StateManager stateManager = StateManagerFactory.getStateManager();
            stateManager.setObjectGridState(AvailabilityState.PRELOAD, grid);
    }
}
```

Puesto que QUIESCE es un estado de transición, no puede utilizar la interfaz StateManager para colocar un ObjectGrid en el estado QUIESCE. Un ObjectGrid pasa a través de este estado al estado OFFLINE.

3. Recupere el estado de disponibilidad.

   Utilice el método getObjectGridState de la interfaz StateManager para recuperar el estado de disponibilidad de un ObjectGrid determinado.

   ```
   StateManager stateManager = StateManagerFactory.getStateManager();
   AvailabilityState state = stateManager.getObjectGridState(inventoryGrid);
   ```

   El método getObjectGridState elige un primario aleatorio dentro de ObjectGrid y devuelve su AvailabilityState. Como todos los fragmentos de ObjectGrid deben estar en el mismo estado de disponibilidad o en transición al mismo estado de disponibilidad, este método proporciona un resultado aceptable para el estado de disponibilidad actual de ObjectGrid.

Gestión de anomalías del centro de datos

Cuando el centro de datos entra en un escenario de anomalía, considere sustituir el quórum de manera que no se ignoren los sucesos del servidor de contenedor. Puede utilizar el programa de utilidad **xscmd** para consultar sobre las tareas de quórum y ejecutarlas, por ejemplo, el estado de quórum y la sustitución de quórum.

Antes de empezar

- Configure el mecanismo de quórum para que tenga el mismo valor en todos los servidores de catálogo. Si desea más información, consulte Configuración del mecanismo de quórum.
- El quórum es el número mínimo de servidores de catálogo que es necesario para realizar operaciones de colocación para la cuadrícula de atos y es el conjunto completo de servidores de catálogo, a menos que se configure un número menor. WebSphere eXtreme Scale espera perder el quórum por los siguientes motivos:
  - El miembro de la JVM del servicio de catálogo falla
  - Caída de la red
  - Pérdida del centro de datos

  El siguiente mensaje indica que se ha perdido el quórum. Consulte este mensaje en los registros del servicio de catálogo.

```
CWOBJ1254W: El servicio de catálogo está esperando el quórum.
```

Acerca de esta tarea

Sustituir quórum en un escenario de anomalía de centro de datos solamente. Cuando se sustituye el quórum, las instancias de servidor de catálogo supervivientes se pueden utilizar. Se notificará a todos los supervivientes que alteren temporalmente el quórum.

Procedimiento

- Consultar estado de quórum con el programa de utilidad **xscmd**.

  ```
  xscmd -c showQuorumStatus -cep cathost:2809
  ```

  Utilice esta opción para visualizar el estado de quórum de una instancia de servicio de catálogo. Se visualiza una de las salidas siguientes:

  - `El quórum está inhabilitado`: los servidores de catálogo se ejecutan en modalidad de quórum inhabilitado. La modalidad de quórum inhabilitado es una modalidad de desarrollo o de un solo centro de datos. No utilice la modalidad de quórum inhabilitado para configuraciones de varios centros de datos.
  - `El quórum está habilitado y el servidor de catálogo tiene quórum`: el quórum está habilitado y el sistema funciona con normalidad.
  - `El quórum está habilitado pero el servidor de catálogo está esperando el quórum`: el quórum está habilitado y se ha perdido.
  - `El quórum está habilitado y se ha sustituido`: el quórum está habilitado y se ha sustituido.
  - `El estado del quórum no está autorizado`: cuando se produce una caída de la red, se divide el servicio de catálogo en dos particiones, A y B. El servidor de catálogo A tiene el quórum sustituido. La partición de la red se resuelve y el servidor de la partición B no está autorizado, lo que requiere un reinicio de la JVM. También se produce si la JVM del catálogo en B se reinicia durante la caída de la red y ésta se soluciona.
- Sustituir quórum con el programa de utilidad **xscmd**.

  ```
  xscmd -c overrideQuorum -cep cathost:2809
  ```

  La ejecución de este mandato obliga a los servidores de catálogo supervivientes a volver a establecer un quórum.
- Diagnostique el quórum con el programa de utilidad **xscmd**.
  - **Visualizar una lista de grupos principales:**

    Utilice la opción **-c listCoreGroups** para visualizar una lista de todos los grupos principales del servidor de catálogo.

    ```
    xscmd -c listCoreGroups -cep cathost:2809
    ```
  - **Desmontar servidores:**

    Utilice la opción **–c teardown** para eliminar un servidor manualmente de la cuadrícula de datos. Normalmente no es necesario eliminar un servidor de la cuadrícula. Los servidores se eliminan automáticamente cuando se detectan como anómalos, pero se proporciona el mandato para su utilización siguiendo las indicaciones del personal de soporte de IBM. Consulte “Detención correcta de servidores con el programa de utilidad **xscmd**” en la página 409 para obtener información sobre la utilización de este mandato.

    ```
    xscmd –c teardown server1,server2,server3 -cep cathost:2809 –g Grid
    ```
  - **Visualizar la tabla de direccionamiento:**

Utilice la opción **-c routetable** para visualizar la tabla de direccionamiento actual simulando una nueva conexión de cliente a la cuadrícula de datos. También valida la tabla de direccionamiento confirmando que todos los servidores de contenedor reconocen su rol en la tabla de direccionamiento como, por ejemplo, qué tipo de fragmento para qué partición.

```
xscmd -c routetable -cep cathost:2809 –g myGrid
```

– **Comprobar los tamaños de correlación:**

  Utilice la opción **-c showMapSizes** para verificar que la distribución de claves es uniforme entre los fragmentos de la clave. Si algunos servidores de contenedor tienen más claves que otros, es probable que la función hash en los objetos de clave tenga una mala distribución.

```
xscmd -c showMapSizes -cep cathost:2809 -g myGrid -ms myMapSet
```

– **Establecer series de rastreo:**

  Utilice la opción **-c setTraceSpec** para establecer los valores de rastreo para todas las JVM que coinciden con el filtro especificado para el mandato **xscmd**. Este valor cambia solo los valores de rastreo, hasta que se utiliza otro mandato o las JVM modificadas fallan o se detienen.

```
xscmd -c setTraceSpec -spec ObjectGrid*=event=enabled -cep cathost:1099
–g myGrid –hf host1
```

  Esta serie habilita el rastreo para todas las JVM en el servidor con el nombre de host especificado, en este caso `host1`.

– **Visualizar fragmentos no asignados:**

  Utilice la opción **-c showPlacement -sf U** para visualizar la lista de fragmentos que no se pueden colocar en la cuadrícula de datos. Los fragmentos no se pueden colocar cuando el servicio de colocación tiene una restricción que evita la colocación. Por ejemplo, si inicia las JVM en un único servidor físico mientras está en modalidad de producción, sólo se colocarán los fragmentos primarios. Las réplicas no se asignan hasta que se inician las JVM en un segundo servidor físico. El servicio de colocación coloca réplicas sólo en las JVM con distintas direcciones IP a las de las JVM que alojan los fragmentos primarios. No tener ninguna JVM en una zona también puede provocar que los fragmentos se queden sin asignar.

```
xscmd -c showPlacement -sf U  -cep cathost:2809 –g myGrid
```

Administración con beans gestionados (MBeans)

Puede utilizar varios tipos distintos de MBeans JMX (Java Management Extensions) para administrar y supervisar despliegues. Cada MBean hace referencia a una entidad específica como, por ejemplo, una correlación, una cuadrícula de datos, un servidor o un servicio.

Las interfaces de MBean JMX yWebSphere eXtreme Scale

Cada MBean tiene métodos get que representan valores de atributos. Estos métodos get no se pueden invocar directamente desde el programa. La especificación JMX trata a los atributos de forma distinta de las operaciones. Puede ver atributos con una consola JMX de proveedor, así como realizar operaciones en el programa o con una consola JMX de proveedor.

Paquete com.ibm.websphere.objectgrid.management

Consulte la documentación de la API para obtener una visión general y las especificaciones de programación detalladas de todos los MBeans disponibles:Paquete com.ibm.websphere.objectgrid.management.

Acceso a beans gestionados (MBeans) mediante la herramienta wsadmin

Puede utilizar el programa de utilidad wsadmin proporcionado en WebSphere Application Server para acceder a la información de beans gestionados (MBeans).

Procedimiento

Ejecute la herramienta wsadmin desde el directorio `bin` en la instalación de WebSphere Application Server. En el siguiente ejemplo se recupera una vista de la colocación de fragmentos actual en un eXtreme Scale dinámico. Puede ejecutar la herramienta wsadmin desde cualquier instalación donde se esté ejecutando eXtreme Scale. No necesita ejecutar la herramienta wsadmin en el servicio de catálogo.

```
$ 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>
```

Acceso a beans gestionados (MBeans) mediante programación

Puede conectarse a MBeans con aplicaciones Java. Estas aplicaciones utilizan las interfaces del paquete com.ibm.websphere.objectgrid.management.

Acerca de esta tarea

Los métodos mediante programación para acceder a MBeans varían según el tipo de servidor al que se vaya a conectar.

- Conectar a un servidor MBean de servidor de catálogo autónomo
- Conectar a un servidor MBean de contenedor
- Conectar a un servidor MBean de servicio de catálogo alojado en WebSphere Application Server
- Conectar a un servidor MBean de servicio de catálogo con la seguridad habilitada

Procedimiento

- **Conectar a un servidor MBean de servicio de catálogo autónomo:**

El programa de ejemplo siguiente se conecta a un servidor MBean de servicio de catálogo autónomo y devuelve una serie con formato XML que lista cada uno de los servidores de contenedor junto con sus fragmentos asignados para un ObjectGrid y MapSet determinados.

```
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;

/**
 * Recopila la información de colocación del servidor de catálogo para un ObjectGrid determinado.
 */
public final class CollectPlacementPlan {
    private static String hostName = "localhost";

    private static int port = 1099;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Se conecta al servicio de catálogo de ObjectGrid para recuperar información de colocación y
     * muestra dicha información.
     *
     * @param args
     * @throws Exception
     *             Si hay un problema al conectar al servidor MBean de servicio de catálogo.
     */
    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();
            }
        }
    }

}
```

Figura 52. CollectPlacementPlan.java

Notas sobre el programa de ejemplo:

– El valor **JMXServiceURL** del servicio de catálogo tiene siempre el formato siguiente: `service:jmx:rmi:///jndi/rmi://<host>:<puerto>/objectgrid/`

MBeanServer, donde <host> es el host en el que se ejecuta el servicio de catálogo y <puerto> es el puerto de servicio JMX que se proporciona con la opción **-JMXServicePort** al iniciar el servicio de catálogo. Si no se especifica ningún puerto, el valor predeterminado es 1099.

– Para que estén habilitadas las estadísticas de ObjectGrid o correlación, debe especificar la propiedad siguiente en el archivo de propiedades del servidor al iniciar un contenedor de ObjectGrid: statsSpec=all=enabled

– Para inhabilitar la ejecución de los MBeans en los servidores de contenedor, especifique la propiedad siguiente en el archivo de propiedades del servidor: enableMBeans=false.

A continuación se muestra un ejemplo de la salida. Esta salida indica que los dos servidores de contenedor están activos. El servidor de contenedor Container-0 aloja cuatro fragmentos primarios. El servidor de contenedor Container-1 aloja una réplica síncrona para cada uno de los fragmentos primarios en el servidor de contenedor Container-0x. En esta configuración, se configuran dos réplicas síncronas y una réplica asíncrona. Como resultado, el servidor de contenedor Unassigned queda con los fragmentos restantes. Si se inician dos servidores de contenedor adicionales, el servidor de contenedor Unassigned no se visualiza.

```
<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>
```

- **Conectar a un servidor MBean de contenedor:**

  Los servidores de contenedor alojan MBeans para consultar información sobre las correlaciones individuales y las instancias de ObjectGrid que se ejecutan en el servidor de contenedor. El programa de ejemplo siguiente muestra el estado de cada uno de los servidores de contenedor que aloja el servidor de catálogo con la dirección JMX de localhost:1099:

```
package com.ibm.websphere.sample.xs.admin;

import java.util.List;
import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectInstance;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Recopila estado de colocación de cada uno de los contenedores disponibles directamente.
 */
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();
            }
        }
    }
}
```

Figura 53. CollectContainerStatus.java

El programa de ejemplo muestra el estado de servidor de contenedor correspondiente a cada contenedor. A continuación se muestra un ejemplo de la salida:

```
<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>
```

- **Conectar a un servidor MBean de servicio de catálogo alojado en WebSphere Application Server:**

  El método para acceder mediante programación a MBeans en WebSphere Application Server es ligeramente distinto a acceder a los MBeans en una configuración autónoma.

1. Cree y configure un programa Java para conectarse al servidor MBean. A continuación se muestra un programa de ejemplo:

```
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;

/**
 * Recopila la información de colocación del servidor de catálogo en ejecución en un gestor de despliegue para un ObjectGrid determinado.
 */
public final class CollectPlacementPlanWAS {
    private static String hostName = "localhost";

    private static int port = 9809;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Se conecta al servicio de catálogo para recuperar información de colocación y visualizarla.
     *
     * @param args
     * @throws Exception
     *             Si hay un problema al conectar al servidor MBean de servicio de catálogo.
     */
    public static void main(String[] args) throws Exception {

        // conectar al puerto de rutina de carga del gestor de despliegue
        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();
            }
        }
    }

}
```

Figura 54. CollectPlacementPlan.java

2. Ejecute el mandato siguiente.

```
"$JAVA_HOME/bin/java" "$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:$WAS_HOME/lib/WMQ/java/lib" \
 -Djava.naming.provider.url=<un_URL_IIOP_o_un_URL_corbaloc_al_nombre_de_máquina_servidor_aplicaciones> \
 -Djava.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory \
 -Dserver.root="$WAS_HOME" "$CLIENTSAS" "$CLIENTSSL" $USER_INSTALL_PROP \
 -classpath "$WAS_CLASSPATH":<lista_de_jars_y_clases_de_la_aplicación> \
 <nombre_clase_completamente_calificado_para_ejecutar> <parámetros_aplicación>
```

Este mandato asume que el script *raíz_was*/bin/setupCmdLine.sh se ha ejecutado para establecer las variables correctamente. Un ejemplo del formato del valor de la propiedad java.naming.provider.url es corbaloc:iiop:1.0@<host>:<puerto>/NameService.

- **Conéctese a un servidor MBean de servicio de catálogo con la seguridad habilitada:**

  Para obtener más información sobre cómo conectar al MBean de servicio de catálogo con la seguridad habilitada, consulte “Seguridad JMX (Java Management Extensions)” en la página 517.

Qué hacer a continuación

Para ver más ejemplos sobre cómo visualizar estadísticas y realizar operaciones administrativas con MBeans, consulte la aplicación de ejemplo **xsadmin**. Puede mirar el código fuente de la aplicación de ejemplo xsadmin en el archivo *inicio_wxs*/samples/xsadmin.jar de una instalación autónoma, o en el archivo *inicio_wxs*/xsadmin.jar en una instalación de WebSphere Application Server. Consulte Ejemplo: Programa de utilidad **xsadmin** para obtener más información sobre las operaciones que puede completar con la aplicación de ejemplo **xsAdmin**.

También puede encontrar más información sobre los MBeans en el paquete com.ibm.websphere.objectgrid.management.

Capítulo 8. Supervisión

Puede utilizar la consola de supervisión, las API, los MBeans, los registros y los programas de utilidad que se incluyen para supervisar el rendimiento del entorno de aplicación.

Visión general de las estadísticas

Las estadísticas de WebSphere eXtreme Scale se calculan basándose en un árbol de estadísticas internas. La API StatsAccessor, los módulos Performance Monitoring Infrastructure (PMI) y la API MBean se crean a partir del árbol interno.

La figura siguiente muestra la configuración general de estadísticas de WebSphere eXtreme Scale.

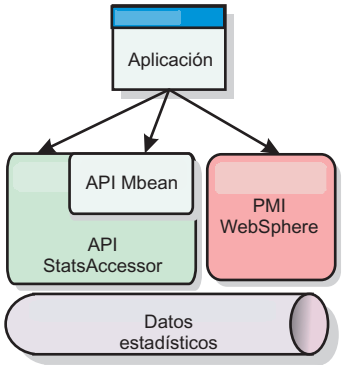


Figura 55. Visión general de las estadísticas

Cada una de estas API ofrecen una visión del árbol de estadísticas, pero se utilizan por distintos motivos:

- **API Statistics**: la API Statistics permite a los desarrolladores acceso directo a las estadísticas, lo que proporciona soluciones de integración de estadísticas flexibles y personalizables, como los MBeans personalizados o el registro cronológico.
- **MBean API**: la API de MBean es un mecanismo basado en especificaciones para realizar la supervisión. La API MBean utiliza la API Statistics y se ejecuta de forma local en la máquina virtual Java (JVM) del servidor. Las estructuras de API y MBean se han diseñado para integrarse fácilmente con otros programas de utilidad. Utilice la API MBean cuando ejecute una cuadrícula de objetos distribuida.
- **Módulos PMI (Performance Monitoring Infrastructure) de WebSphere Application Server**: utilice PMI si está ejecutando WebSphere eXtreme Scale dentro de WebSphere Application Server. Estos módulos proporcionan una vista del árbol de estadísticas internas.

API de estadísticas

De forma muy parecida a una correlación de árbol, hay una correspondiente vía de acceso y clave que se utiliza para recuperar un módulo específico, o en este caso el nivel de agregación o granularidad. Por ejemplo, suponga que siempre hay un nodo raíz arbitrario en el árbol y que las estadísticas se están recopilando para una

correlación llamada "payroll," que pertenece a un ObjectGrid denominado "accounting." Por ejemplo, para acceder al módulo para el nivel de agregación o granularidad de una correlación, podría pasar una String[] de las vías de acceso. En este caso equivaldría a String[] {root, "accounting", "payroll"}, ya que cada String representaría la vía de acceso del nodo. La ventaja de esta estructura es que un usuario puede especificar la matriz para cualquier código en la vía de acceso y obtener el nivel de agregación para ese nodo. Por lo tanto, si pasa String[] {root, "accounting"} le proporcionará estadísticas de correlaciones, pero para toda la cuadrícula de "accounting." Esto ofrece al usuario la capacidad de especificar tipos de estadísticas para supervisar y que nivel de agregación es necesario para la aplicación.

Módulos PMI de WebSphere Application Server

WebSphere eXtreme Scale incluye módulos de estadísticas para utilizarlos con la PMI de WebSphere Application Server. Cuando se aumenta un perfil de WebSphere Application Server con WebSphere eXtreme Scale, los scripts de aumento integran automáticamente los módulos de WebSphere eXtreme Scale en los archivos de configuración de WebSphere Application Server. Con PMI, puede habilitar e inhabilitar módulos de estadísticas, agregar estadísticas automáticamente en distinta granularidad, e incluso trazar un gráfico con los datos utilizando el Tivoli Performance Viewer incorporado. Si desea más información, consulte “Supervisión con PMI de WebSphere Application Server” en la página 463.

Integración de productos de proveedores con beans gestionados (MBean)

Las API eXtreme Scale y los beans gestionados se han diseñado para permitir una fácil integración con las aplicaciones de supervisión de terceros. JConsole o MC4J son algunos ejemplos de consolas JMX (Java Management Extensions) ligeras que pueden utilizarse para analizar información sobre una topología de eXtreme Scale. También puede utilizar las API de programación para grabar implementaciones de adaptador en la instantánea o realizar un seguimiento de eXtreme Scale. WebSphere eXtreme Scale incluye una aplicación de supervisión de ejemplo que admite funciones de supervisión preconfiguradas que se pueden utilizar como plantilla para grabar programas de utilidad de supervisión personalizados más avanzados.

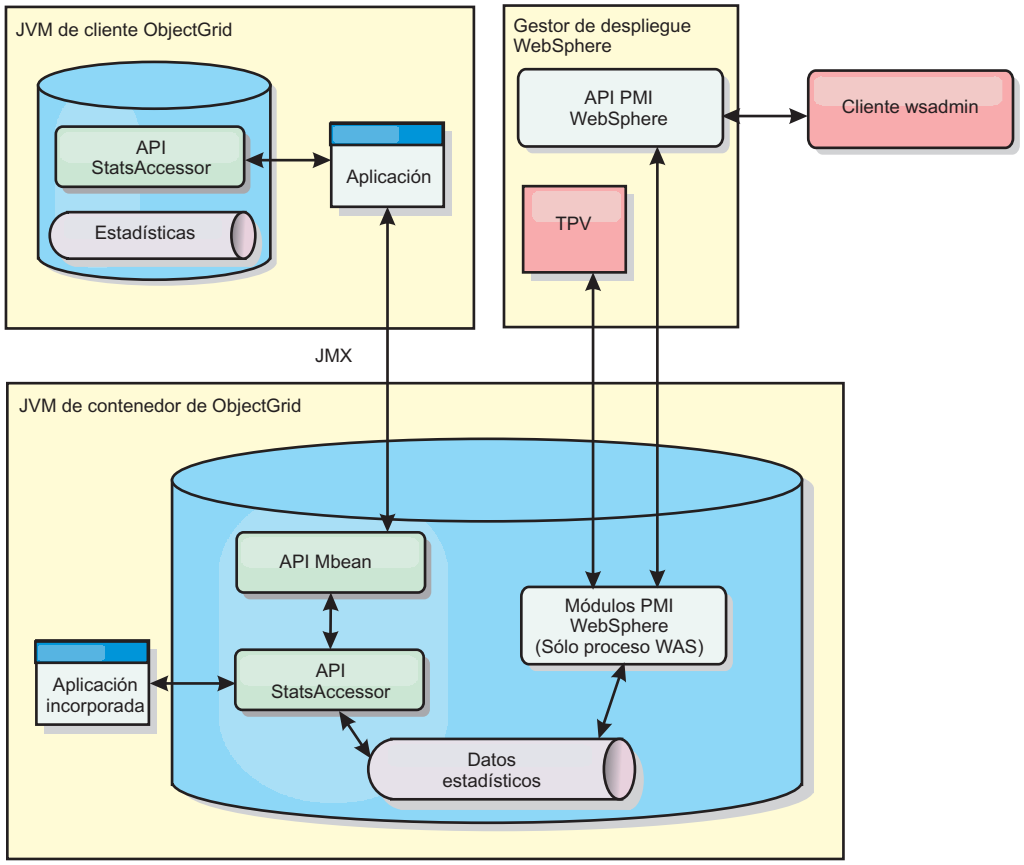


Figura 56. Visión general de MBean

Si desea más información, consulte Ejemplo: Programa de utilidad **xsadmin**. Para obtener más información sobre la integración con aplicaciones de proveedores específicos, consulte los siguientes temas:

- Supervisión de eXtreme Scale con el agente IBM Tivoli Monitoring
- "Supervisión de eXtreme Scale con Hyperic HQ" en la página 486
- "Supervisión de aplicaciones de eXtreme Scale con CA Wily Introscope" en la página 483

Supervisión con la consola web

Con la consola web, puede representar gráficos de las estadísticas actuales e históricas. Esta consola proporciona algunos gráficos configurados previamente para visiones generales de alto nivel y tiene una página de informes personalizados que puede utilizar para crear gráficos de las estadísticas disponibles. Puede utilizar las posibilidades de representación gráfica en la consola de supervisión de WebSphere eXtreme Scale para ver el rendimiento general de las cuadrículas de datos del entorno.

Inicio e inicio de sesión en la consola web

Inicie el servidor de la consola ejecutando el mandato **startConsoleServer** e iniciando sesión en el servidor con el ID de usuario y la contraseña predeterminados.

Antes de empezar

- **Requisitos del sistema**
  - Utilice un sistema AIX, Linux o Windows para ejecutar la consola web.
  - Instale un servidor autónomo de WebSphere eXtreme Scale en el sistema que aloja el servidor de la consola. Si desea más información, consulte "Instalación de WebSphere eXtreme Scale o WebSphere eXtreme Scale Client autónomos" en la página 193.
  - El sistema del servidor de la consola debe poder conectarse al servicio de catálogo. El servicio de catálogo también debe poder conectarse al servidor de la consola web.
- **Requisitos del navegador web**

  Utilice uno de los navegadores siguientes con la consola web:
  - Mozilla Firefox, versión 3.5.x y posteriores
  - Mozilla Firefox, versión 3.6.x y posteriores
  - Microsoft Internet Explorer, versión 7 o 8

Procedimiento

1. Opcional: Si desea ejecutar el servidor de la consola en un puerto que no sea el puerto predeterminado, edite el archivo *raíz_intal_wxs*`/ObjectGrid/console/config/zero.config`. El puerto predeterminado del servidor de la consola es `7080` para HTTP y `7443` para HTTPS. Puede editar las propiedades siguientes para cambiar los valores predeterminados:

   ```
   /config/http/port = 7080
   /config/https/port = 7443
   ```

   Si edita estos valores una vez que el servidor de la consola ya se ha iniciado, reinicie el servidor para que se utilicen los nuevos números de puerto.
2. Inicie el servidor de consola. El script **`startConsoleServer.bat|sh`** para iniciar el servidor de la consola se encuentra en el directorio *raíz_intal_wxs*`/ObjectGrid/bin` de la instalación.
3. Inicie la sesión en la consola.
   a. Desde el navegador web, vaya a `https://su.host.consola:7443`, sustituyendo `su.host.consola` por el nombre de host del servidor en el que ha instalado la consola.
   b. Inicie la sesión en la consola.
      - **ID de usuario:** `admin`
      - **Contraseña:** `admin`

      Se visualiza la página de bienvenida de la consola.
4. Edite la configuración de la consola. Pulse **Valores** > **Configuración** para revisar la configuración de la consola. La configuración de la consola incluye información como:
   - Serie de rastreo del cliente WebSphere eXtreme Scale, como `*=all=disabled`
   - Nombre y contraseña del administrador
   - Dirección de correo electrónico del administrador

Qué hacer a continuación

- Conecte los servidores de catálogo a la consola web para iniciar las estadísticas de seguimiento. Si desea más información, consulte "Conexión de la consola web a servidores de catálogo" en la página 445.

- Si necesita detener el servidor de la consola web, ejecute el script **stopConsoleServer.bat|sh**. Este script se encuentra en el directorio *raíz_intal_wxs*/ObjectGrid/bin de la instalación.

Conexión de la consola web a servidores de catálogo

Para empezar a visualizar estadísticas en la consola web, en primer lugar debe conectarse a los servidores de catálogo que desea supervisar. Se requieren pasos adicionales si los servidores de catálogo tienen la seguridad habilitada.

Antes de empezar

- El servidor de la consola web debe estar en ejecución. Si desea más información, consulte "Inicio e inicio de sesión en la consola web" en la página 443.
- Debe tener como mínimo un servidor de catálogo en ejecución al que desee conectarse. Si desea más información, consulte "Inicio de un servicio de catálogo autónomo" en la página 395.

Procedimiento

1. Si los servidores de catálogo tienen SSL (Secure Sockets Layer) habilitado, debe configurar un archivo de propiedades de almacén de claves, almacén de confianza y cliente. Habilita SSL para un servidor de catálogo estableciendo el atributo transportType en SSL-Required en Archivo de propiedades de servidor .
   a. Configure un almacén de claves y un almacén de confianza y, a continuación, intercambie o importe de forma cruzada los certificados públicos. Por ejemplo, podría copiar el almacén de confianza y el almacén de claves en una ubicación en el servidor que está en ejecución en la consola web.
   b. Edite el archivo de propiedades de cliente en el servidor de la consola web para incluir las propiedades de la configuración SSL. Por ejemplo, podría editar el archivo *raíz_intal_wxs*/ObjectGridProperties/sampleclient.properties. Son necesarias las propiedades siguientes para conexiones SSL de salida desde la consola web:

```
#--------------------------------------------------------------------------------
# Configuración SSL
#
# - contextProvider      (IBMJSSE2, IBMJSSE, IBMJSSEFIPS, etc.)
# - protocol             (SSL, SSLv2, SSLv3, TLS, TLSv1, etc.)
# - keyStoreType         (JKS, JCEK, PKCS12, etc.)
# - trustStoreType       (JKS, JCEK, PKCS12, etc.)
# - keyStore             (vía de acceso completa al archivo del almacén de claves)
# - trustStore           (vía de acceso completa al archivo de almacén de confianza)
# - alias                (serie que especifica el alias de certificado a utilizar desde el almacén de claves)
# - keyStorePassword     (serie que especifica la contraseña para el almacén de claves, codificada o no)
# - trustStorePassword  (serie que especifica la contraseña para el almacén de confianza,
#                        codificada o no)
#
# Descomentar estas propiedades para establecer la configuración SSL.
#--------------------------------------------------------------------------------
#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
```

Importante: Windows Si utiliza Windows, en las vías de acceso debe colocar entre caracteres de escape las barras inclinadas (\). Por ejemplo, si desea utilizar la vía de acceso C:\opt\ibm, especifique C:\\opt\\ibm en el archivo de propiedades.

2. Establezca y mantenga las conexiones a los servidores de catálogo que desea supervisar. Repita los pasos siguientes para añadir cada servidor de catálogo a la configuración.
   a. Pulse **Valores** > **Servidores de catálogo eXtreme Scale**.
   b. Añada un servidor de catálogo nuevo.

      1) Pulse el icono de añadir ( ) para registrar un servidor de catálogo existente.
      2) Proporcione información, como el nombre de host y el puerto de escucha. Consulte "Planificación de puertos de red" en la página 64 para obtener más información sobre la configuración de puerto y los valores predeterminados.
      3) Pulse **Aceptar**.
      4) Verifique que el servidor de catálogo se ha añadido al árbol de navegación.
3. Agrupe los servidores de catálogo que ha creado en un dominio de servicio de catálogo. Debe crear un dominio de servicio de catálogo cuando la seguridad esté habilitada en los servidores de catálogo, ya que los valores de seguridad se configuran en el dominio de servicio de catálogo.
   a. Pulse la página **Valores** > **Dominios de eXtreme Scale**.
   b. Añada un nuevo dominio de servicio de catálogo.

      1) Pulse el icono de añadir ( ) para registrar un dominio de servicio de catálogo. Especifique un nombre para el dominio de servicio de catálogo.
      2) Después de crear el dominio de servicio de catálogo, puede editar las propiedades. A continuación se muestran las propiedades del dominio de servicio de catálogo:

         Nombre
         Indica el nombre de host del dominio, asignado por el administrador.

         Servidores de catálogo
         Enumera uno o varios servidores de catálogo que pertenecen al dominio seleccionado. Puede añadir los servidores de catálogo que ha creado en el paso anterior.

         Clase generator
         Especifica el nombre de la clase que implementa la interfaz CredentialGenerator. Esta clase se utiliza para obtener credenciales para los clientes. Si especifica un valor en este campo, el valor altera temporalmente la propiedad **crendentialGeneratorClass** en el archivo `client.properties`.

         Propiedades de generator
         Especifica las propiedades para la clase de implementación CredentialGenerator. Las propiedades se establecen en el objeto con el método setProperties(String). El valor credentialGeneratorprops sólo se utiliza si el valor de la propiedad credentialGeneratorClass no es nula. Si especifica un valor en este campo, el valor altera temporalmente la propiedad **credentialGeneratorProps** en el archivo `client.properties`.

Vía de acceso de las propiedades de cliente de eXtreme Scale
Especifica la vía de acceso del archivo de propiedades de cliente que ha editado para incluir propiedades SSL en un paso anterior. Por ejemplo, podría indicar el archivo `c:\ObjectGridProperties\sampleclient.properties`. Si desea evitar que la consola intente utilizar conexiones SSL, puede suprimir el valor de este campo. Después de establecer la vía de acceso, la consola utiliza una conexión no segura.

3) Pulse **Aceptar**.
4) Verifique que el dominio se ha añadido al árbol de navegación.

Para visualizar información sobre un dominio de servicio de catálogo existente, pulse el nombre del dominio de servicio de catálogo en el árbol de navegación de la página **Valores** > **Dominios de eXtreme Scale**.

4. Consulte el estado de conexión. El campo **Dominio actual** indica el nombre del dominio de servicio de catálogo que se utiliza actualmente para visualizar información en la consola web. El estado de conexión se visualiza junto al nombre del dominio de servicio de catálogo.

Visualización de estadísticas con la consola web

Puede supervisar estadísticas y otra información de rendimiento con la consola web.

Antes de empezar

Antes de poder visualizar estadísticas con la consola web, debe completar las siguientes tareas:

1. Inicie el servidor de la consola web. Si desea más información, consulte “Inicio e inicio de sesión en la consola web” en la página 443.
2. Conecte los servidores de catálogo al servidor de la consola web. Si desea más información, consulte “Conexión de la consola web a servidores de catálogo” en la página 445.
3. Ejecute las aplicaciones y cuadrículas de datos activos en los servidores que gestiona el dominio de servicio de catálogo.

Acerca de esta tarea

Después de crear las cuadrículas de datos y configurar las aplicaciones para utilizar las cuadrículas de datos, espere unos instantes para que las estadísticas estén disponibles. Por ejemplo, con una cuadrícula de datos dinámica, no están disponibles las estadísticas hasta que un WebSphere Application Server en que se ejecuta una memoria caché dinámica se conecta a la memoria caché dinámica. En general, espere hasta un minuto después de un cambio principal en la configuración para ver los cambios en las estadísticas.

Consejo: Para ver más información específica sobre cualquier punto de datos en un gráfico, puede desplazar el puntero del ratón sobre el punto de datos.

Procedimiento

- Para ver las estadísticas del servidor actual, pulse **Supervisar** > **Visión general de servidor**.
- Para ver el rendimiento de todas las cuadrículas de datos, pulse **Supervisar** > **Visión general del dominio de cuadrícula de datos**.

- Para ver cuadrículas de datos individuales, pulse **Supervisar** > **Visión general de cuadrícula de datos** > ***nombre_cuadrícula_datos***. En esta página se muestra un resumen que incluye el número de entradas de memoria caché, el promedio de tiempo de transacción y el promedio de rendimiento.
- Para ver más detalles sobre una cuadrícula de datos concreta, pulse **Supervisar** > **Detalles de cuadrícula de datos**. Se muestra un árbol con todas las cuadrículas de datos de la configuración. Puede detallar más una cuadrícula de datos específica para visualizar las correlaciones que forman parte de dicha cuadrícula de datos. Puede pulsar un nombre de cuadrícula de datos o una correlación para obtener más información.
- Para seleccionar qué estadísticas prefiere que contenga su informe personalizado, pulse **Supervisar** > **Informes personalizados**.

  Utilice esta vista para construir gráficos de datos detallados de las distintas estadísticas. Utilice el árbol para explorar las cuadrículas de datos y los servidores disponibles y sus estadísticas asociadas. Al pulsar con el ratón o pulsar la tecla Intro en un nodo que hace referencia a los datos que se pueden representar en el gráfico se abre un menú. Cree un gráfico nuevo que contiene las estadísticas o añada las estadísticas a un gráfico existente con las estadísticas compatibles.Si desea más información, consulte "Supervisión de informes personalizados" en la página 453.

Estadísticas de la consola web

En función de la vista que utilice en la consola web, podrá visualizar distintas estadísticas sobre la configuración. Estas estadísticas incluyen la memoria utilizada, las cuadrículas de datos más utilizadas y el número de entradas de memoria caché.

Visión general del dominio de la cuadrícula de datos

Las estadísticas de la visión general del dominio de la cuadrícula de datos se visualizan en la página **Supervisor** > **Visión general del dominio de la cuadrícula de datos**. Pulse uno de los separadores siguientes para obtener más información sobre el dominio de la cuadrícula de datos:

Separador Capacidad utilizada
: En el gráfico **Distribución actual de la capacidad utilizada de la cuadrícula de datos**, se visualiza una imagen de la **Agrupación total**, y los **Consumidores con mayor uso de capacidad**. Sólo se muestran las 25 primeras cuadrículas de datos. En el gráfico **Capacidad utilizada a lo largo del tiempo** se visualiza el número de bytes que consume la cuadrícula de datos.

Separador Rendimiento promedio
: El gráfico **Cinco cuadrículas de datos más activas por promedio de**

tiempo de transacción en milisegundos contiene una lista de las cinco principales memorias caché de datos, organizadas por promedio de tiempo de transacción. El gráfico **Rendimiento promedio a lo largo del tiempo** visualiza el rendimiento promedio, máximo y mínimo durante la última hora, día y semana.

Separador Promedio de tiempo de transacción
El gráfico **Cinco cuadrículas de datos más lentas** visualiza datos sobre las cuadrículas de datos más lentas. El gráfico **Promedio de tiempo de transacción a lo largo del tiempo** visualiza el tiempo promedio, máximo y mínimo de transacción en la última hora, día y semana.

Visión general de la cuadrícula de datos

Para visualizar estadísticas de una cuadrícula de datos individual, pulse **Supervisor** > **Visión general de la cuadrícula de datos** > *nombre_cuadrícula_datos*.

Resumen actual en los últimos 30 segundos
Visualiza el número actual de entradas de memoria caché, el tiempo promedio de transacción, el rendimiento promedio y la proporción de coincidencias de memoria caché de la cuadrícula de datos seleccionada.

Separador Capacidad utilizada
El gráfico **Resumen actual en los últimos 30 segundos** visualiza el número de entradas de memoria caché y la capacidad utilizada en bytes en un rango de tiempo especificado.

Separador Uso de memoria caché
El gráfico **Uso de memoria caché** ayuda a visualizar el número de consultas satisfactorias a la memoria caché y visualiza los intentos de memoria caché, las coincidencias de memoria caché y la proporción de coincidencias de memoria caché en un rango de tiempo especificado.

Separador Rendimiento promedio
El gráfico **Rendimiento promedio frente a Promedio de tiempo de transacción** visualiza el tiempo de transacción y el rendimiento durante un rango de tiempo especificado.

Detalles de cuadrícula de datos

Las estadísticas de cuadrícula de datos se visualizan en la página **Supervisor** > **Detalles de la cuadrícula de datos**. Puede mirar los datos para una cuadrícula seleccionada y las correlaciones que se encuentran en esa cuadrícula.

Resumen actual en los últimos 30 segundos
Visualiza la capacidad utilizada actual, el número de entradas de memoria caché, el rendimiento promedio y el promedio de tiempo de transacción para la cuadrícula de datos seleccionada.

Distribución actual de la capacidad utilizada de la cuadrícula de datos de objetos de eXtreme Scale
Consulte una agrupación total, que incluye la capacidad por zona y la capacidad total de cada zona. Solo se muestran las 25 primeras correlaciones ObjectGrid. También puede visualizar los consumidores con mayor utilización de capacidad por cada correlación.

Distribución actual de la capacidad utilizada de la zona
Muestra una agrupación total, que incluye la agrupación total y los consumidores con mayor utilización de capacidad en la zona de la

cuadrícula de datos seleccionada. También puede visualizar los consumidores con mayor capacidad utilizada por cada zona.

Estadísticas de correlación:

Resumen actual en los últimos 30 segundos
Visualiza la capacidad utilizada actual, el número de entradas de memoria caché, el rendimiento promedio y el promedio de tiempo de transacción de la correlación seleccionada.

Distribución actual de la capacidad utilizada de la partición
Visualiza una partición, que incluye la agrupación total y los consumidores con mayor utilización de capacidad. Sólo se muestran las 25 primeras particiones. También puede visualizar los consumidores con mayor utilización de capacidad por cada partición.

Visión general del servidor

Las estadísticas del servidor se visualizan en la página **Supervisor** > **Visión general del servidor**.

Distribución actual de la memoria utilizada del servidor
Este gráfico se compone de dos vistas. **Agrupación total** visualiza la cantidad actual de memoria utilizada (real) en el tiempo de ejecución del servidor. **Consumidores con mayor utilización de memoria** desglosa la memoria utilizada por servidor; sin embargo, solo se visualizan los 25 servidores con mayor utilización de memoria.

Memoria total con el tiempo
Muestra el uso de memoria real durante la ejecución del servidor.

Memoria utilizada con el tiempo
Muestra la cantidad de memoria utilizada durante la ejecución del servidor.

Informes personalizados: Estadísticas del dominio de servicio de catálogo

Puede visualizar estadísticas del dominio de servicio de catálogo creando un informe personalizado. Pulse **Supervisar** > **Informes personalizados**.

Promedio de tiempo de transacción (mseg.)
Muestra el promedio de tiempo necesario para completar una transacción de este dominio.

Promedio de rendimiento de transacción (trans./seg.)
Muestra el número promedio de transacciones por segundo de este dominio.

Tiempo máximo de transacción (mseg.)
Muestra el tiempo invertido por la transacción que *más* tiempo consume de este dominio.

Tiempo mínimo de transacción (mseg.)
Muestra el tiempo invertido por la transacción que *menos* tiempo consume de este dominio.

Tiempo total de transacción (mseg.)
Muestra el tiempo total invertido en las transacciones de este dominio, desde el momento en que se inicializó el dominio.

Informes personalizados: Estadísticas del servidor de contenedor

Puede visualizar las estadísticas del servidor de contenedor creando un informe personalizado. Pulse **Supervisar** > **Informes personalizados**.

Promedio de tiempo de transacción (mseg.)
Muestra el promedio de tiempo necesario para que este servidor de catálogo complete una transacción.

Promedio de rendimiento de transacción (trans./seg.)
Muestra el número promedio de transacciones por segundo de este servidor de catálogo.

Tiempo máximo de transacción (mseg.)
Muestra el tiempo invertido por la transacción que *más* tiempo consume de este servidor de catálogo.

Tiempo mínimo de transacción (mseg.)
Muestra el tiempo invertido por la transacción que *menos* tiempo consume de este servidor de catálogo.

Tiempo total de transacción (mseg.)
Muestra el tiempo total invertido en las transacciones de este servidor de catálogo, desde el momento en que se inicializó este servidor de catálogo.

Número total de entradas en memoria caché
Muestra el número actual de objetos almacenados en memoria caché en las cuadrículas supervisadas por este servidor de catálogo.

Proporción de coincidencias (porcentaje)
Muestra la proporción de coincidencias (ratio de coincidencias) de la cuadrícula de datos seleccionada. Es deseable una proporción de coincidencias alta. La proporción de coincidencias indica cómo ayuda la cuadrícula a evitar el acceso al almacén persistente.

Bytes utilizados
Muestra el consumo de memoria por parte de esta correlación. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

Número mínimo de bytes utilizados
Muestra el punto de menos consumo de memoria por parte de este servicio de catálogo y sus correlaciones. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

Número máximo de bytes utilizados
Muestra el punto de más consumo de memoria por parte de este servicio de catálogo y sus correlaciones. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

Número total de coincidencias
Muestra el número total de veces en que se han encontrado los datos solicitados en la correlación, con lo que se evita tener que acceder al almacén persistente.

Número total de operaciones get
Muestra el número total de veces que la correlación ha tenido que acceder al almacén persistente para obtener los datos.

Almacenamiento dinámico libre (MB)
Muestra la cantidad de almacenamiento dinámico real disponible para la JVM que el servidor de catálogo utiliza.

Almacenamiento dinámico total
Muestra la cantidad de almacenamiento dinámico disponible para la JVM que el servidor de catálogo está utilizando.

Número de procesadores disponibles
Muestra el número de procesadores disponibles en este servidor de catálogo y sus correlaciones. Para una mayor estabilidad, ejecute los servidores al 60 % de carga del procesador y los almacenamientos dinámicos de la JVM al 60 % de carga de almacenamiento dinámico. Los picos de utilización pueden conducir al uso del procesador a un 80–90%, aunque de forma habitual no debe ejecutar los servidores a niveles más altos que éstos.

Tamaño de almacenamiento dinámico máximo (MB)
Muestra la cantidad máxima de almacenamiento dinámico disponible para la JVM que el servidor de catálogo utiliza.

Memoria utilizada
Muestra la memoria utilizada en la JVM que este servidor de catálogo utiliza.

Informes personalizados: Estadísticas de la cuadrícula de datos

Puede visualizar estadísticas de cuadrícula de datos creando un informe personalizado. Pulse **Supervisar** > **Informes personalizados**.

Promedio de tiempo de transacción (mseg.)
Muestra el promedio de tiempo necesario para completar las transacciones relacionadas con esta cuadrícula.

Promedio de rendimiento de transacción (trans./seg.)
Muestra el número promedio de transacciones por segundo completadas por esta cuadrícula.

Tiempo máximo de transacción (mseg.)
Muestra el tiempo invertido por la transacción que *más* tiempo consume completada por esta cuadrícula.

Tiempo mínimo de transacción (mseg.)
Muestra el tiempo invertido por la transacción que *menos* tiempo consume completada por esta cuadrícula.

Tiempo total de transacción (mseg.)
Muestra la cantidad total de tiempo de proceso de transacciones de esta cuadrícula.

Informes personalizados: Estadísticas de la correlación

Puede visualizar estadísticas de la correlación creando un informe personalizado. Pulse **Supervisar** > **Informes personalizados**.

Número total de entradas en memoria caché
Muestra el número actual de objetos almacenados en memoria caché en esta correlación.

Proporción de coincidencias (porcentaje)
Muestra la proporción de coincidencias (ratio de coincidencias) de la

correlación seleccionada. Es deseable una proporción de coincidencias alta. La proporción de coincidencias indica cómo ayuda la correlación a evitar el acceso al almacén persistente.

Bytes utilizados
Muestra el consumo de memoria por parte de esta correlación. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

Número mínimo de bytes utilizados
Muestra el consumo mínimo (en bytes) de esta correlación. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

Número máximo de bytes utilizados
Muestra el consumo máximo (en bytes) de esta correlación. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

Número total de coincidencias
Muestra el número total de veces en que se han encontrado los datos solicitados en la correlación, con lo que se evita tener que acceder al almacén persistente.

Número total de operaciones get
Muestra el número total de veces que la correlación ha tenido que acceder al almacén persistente para obtener los datos.

Almacenamiento dinámico libre (MB)
Muestra la cantidad de almacenamiento dinámico actual disponible para esta correlación, en la JVM que el servidor de catálogo utiliza.

Almacenamiento dinámico total (MB)
Muestra la cantidad de almacenamiento dinámico total disponible para esta correlación, en la JVM que el servidor de catálogo está utilizando. Para una mayor estabilidad, ejecute los servidores al 60 % de carga del procesador y los almacenamientos dinámicos de la JVM al 60 % de carga de almacenamiento dinámico. Los picos de utilización pueden conducir al uso del procesador a un 80–90%, aunque de forma habitual no debe ejecutar los servidores a niveles más altos que éstos.

Número de procesadores disponibles
Muestra el número de procesadores disponibles en esta correlación. Para una mayor estabilidad, ejecute los servidores al 60 % de carga del procesador y los almacenamientos dinámicos de la JVM al 60 % de carga de almacenamiento dinámico. Los picos de utilización pueden conducir al uso del procesador a un 80–90%, aunque de forma habitual no debe ejecutar los servidores a niveles más altos que éstos.

Tamaño de almacenamiento dinámico máximo (MB)
Muestra la cantidad de almacenamiento dinámico máxima disponible para esta correlación, en la JVM que el servidor de catálogo está utilizando.

Memoria utilizada (MB)
Muestra la cantidad de memoria utilizada en esta correlación.

Supervisión de informes personalizados

Puede crear informes personalizados para guardar diversos gráficos que contengan estadísticas sobre los dominios de servicio de catálogo, cuadrículas de datos y

servidores de contenedor en el entorno. Puede guardar los informes personalizados y cargarlos para visualizarlos de nuevo posteriormente.

Antes de empezar

Antes de poder visualizar estadísticas con la consola web, debe completar las siguientes tareas:

1. Inicie el servidor de la consola web. Si desea más información, consulte "Inicio e inicio de sesión en la consola web" en la página 443.
2. Conecte los servidores de catálogo al servidor de la consola web. Si desea más información, consulte "Conexión de la consola web a servidores de catálogo" en la página 445.
3. Ejecute las aplicaciones y cuadrículas de datos activos en los servidores que gestiona el dominio de servicio de catálogo.

Procedimiento

- Cree un informe personalizado.
  1. Pulse **Supervisar** > **Informes personalizados**. Se muestra en formato de árbol una lista de los dominios de eXtreme Scale que ha definido. Puede expandir cada uno de estos dominios para visualizar las estadísticas disponibles que puede añadir al informe personalizado.
  2. Añada gráficos con las estadísticas de las que desea realizar un seguimiento.

     Las estadísticas disponibles las indica el icono de gráfico (). Pulse una de las estadísticas de las que desee realizar un seguimiento. Elija **Añadir a gráfico nuevo** o **Añadir a gráfico existente**. En función de su selección, la estadística seleccionada se visualiza en un separador de gráfico nuevo o en el gráfico existente seleccionado. Sólo puede añadir una métrica a un gráfico existente si la métrica ya está en el gráfico y la nueva métrica utiliza las mismas unidades.
- Guarde un informe personalizado. Al guardar el informe personalizado se guardan las estadísticas de todos los separadores que ha creado. Para guardar el informe, pulse **Guardar**.
- Cargue un informe personalizado. Pulse **Cargar** y elija el informe personalizado guardado que desea visualizar.

Supervisión con archivos CSV

Puede permitir que se escriban datos de supervisión en archivos CSV. Estos archivos CSV pueden contener información sobre la JVM, la correlación o la instancia de ObjectGrid.

Acerca de esta tarea

Si permite que se escriban datos de supervisión en archivos CSV, puede descargar y analizar datos históricos para un servidor de contenedor o catálogo individual. Se empiezan a recopilar los datos al iniciar el servidor con las propiedades del servidor que habilitan los archivos CSV. A continuación, puede descargar los archivos CSV en cualquier momento y utilizar los archivos como elija.

Procedimiento

1. Actualice el archivo de propiedades del servidor con las propiedades siguientes relacionadas con la habilitación de los archivos CSV.

```
parameter=valor predeterminado
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
```

Para obtener más información sobre estas propiedades, consulte Archivo de propiedades de servidor .

2. Reinicie el servidor para que se apliquen los cambios realizados en el archivo de propiedades del servidor.
3. Descargue el archivo CSV. El archivo CSV se escribe en el directorio *nombre_servidor*/logs.
4. Importe el archivo CSV en el programa que utilice para procesar los datos, por ejemplo, como una hoja de cálculo.

Qué hacer a continuación

Para obtener más información sobre los datos contenidos en los archivos CSV, consulte "Definiciones de estadísticas de archivos CSV".

Definiciones de estadísticas de archivos CSV

Los archivos CSV que puede descargar para un servidor incluyen estadísticas que puede utilizar para crear gráficos históricos u otra información.

Registro de estadísticas de máquina virtual Java (JVM)

TimeStamp
Especifica la fecha y hora de la instantánea de estadísticas que se ha realizado para la JVM.

ServerName
Especifica el nombre de servidor de la JVM.

Hostname
Especifica el nombre de host de la JVM.

DomainName
Especifica el dominio de servicio de catálogo al que pertenece la JVM.

FreeMemory
Especifica el número de bytes disponibles para la JVM.

MaxMemory
Especifica el número máximo de bytes que se pueden asignar para la JVM.

TotalMemory
Muestra el uso de memoria real durante la ejecución del servidor.

AvailProcs
Muestra el número de procesadores disponibles en este servidor de catálogo y sus correlaciones. Para una mayor estabilidad, ejecute los servidores al 60 % de carga del procesador y los almacenamientos dinámicos de la JVM al 60 % de carga de almacenamiento dinámico. Los picos de utilización pueden conducir al uso del procesador a un 80–90%, aunque de forma habitual no debe ejecutar los servidores a niveles más altos que éstos.

Registro de estadísticas de correlación

TimeStamp
Especifica la fecha y hora de la instantánea de estadísticas que se ha realizado para la correlación.

MapName
Especifica el nombre de la correlación.

OgName
Especifica el nombre de la cuadrícula de datos a la que pertenece esta correlación.

PartitionId
Especifica el ID de la partición.

MapSetName
Especifica el conjunto de correlaciones al que pertenece esta correlación.

HitRate
Muestra la proporción de coincidencias (ratio de coincidencias) de la correlación seleccionada. Es deseable una proporción de coincidencias alta. La proporción de coincidencias indica cómo la cuadrícula de datos está ayudando a evitar el acceso al almacén persistente.

Count Indica un recuento de los ejemplos de datos que se han recopilado desde el inicio del servidor. Por ejemplo, un valor de 100 indica que la entrada tiene una entrada de ejemplo 100 que se ha recopilado desde el inicio del servidor.

TotalGetCount
Muestra el número total de veces que la correlación ha tenido que acceder al almacén persistente para obtener los datos.

TotalHitCount
Muestra el número total de veces en que se han encontrado los datos solicitados en la correlación, con lo que se evita tener que acceder al almacén persistente.

StartTime
Especifica la hora de inicio de los contenedores desde la última llamada de restablecimiento. Los restablecimientos se producen cuando el servidor se inicia o reinicia.

LastCount
Especifica el periodo de tiempo desde la realización del último muestreo de datos.

LastTotalGetCount
Indica el número total actual de operaciones get desde la memoria caché menos el número de operaciones get del periodo anterior.

LastTotalHitCount
Indica el número total actual de coincidencias desde la memoria caché menos el número de coincidencias del periodo anterior.

UsedBytes
Muestra el consumo de memoria por parte de esta correlación. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

MinUsedBytes
Muestra el punto de menos consumo de memoria por parte de este servicio de catálogo y sus correlaciones. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

MaxUsedBytes
Muestra el punto de más consumo de memoria por parte de este servicio de catálogo y sus correlaciones. Las estadísticas de bytes utilizados son precisas solo cuando se utilizan objetos simples o la modalidad de copia COPY_TO_BYTES.

LastUsedBytes
Indica el valor de UsedBytes actual menos el valor de UsedBytes del periodo de colección de estadísticas anterior.

SampleLen
Indica la duración, en milisegundos, del periodo durante el que se ha realizado el muestreo de los datos.

Registro de estadísticas de ObjectGrid

Count Indica un recuento de los ejemplos de datos que se han recopilado desde el inicio del servidor. Por ejemplo, un valor de 100 indica que la entrada tiene una entrada de ejemplo 100 que se ha recopilado desde el inicio del servidor.

TimeStamp
Especifica la fecha y hora de la instantánea de estadísticas que se ha realizado para la cuadrícula de datos.

OgName
Especifica el nombre de la cuadrícula de datos.

PartitionId
Especifica el ID de partición.

Hostname
Especifica el nombre de host.

DomainName
Especifica el dominio de servicio de catálogo al que pertenece esta cuadrícula de datos.

MaxTime
Muestra el tiempo invertido por la transacción que *más* tiempo consume de este servidor de catálogo.

MinTime
Muestra el tiempo invertido por la transacción que *menos* tiempo consume de este servidor de catálogo.

MeanTime
Especifica el tiempo promedio utilizado en una transacción.

TotalTime
Muestra el tiempo total invertido en las transacciones de este servidor de catálogo, desde el momento en que se inicializó este servidor de catálogo.

AvgTransTime
Muestra el promedio de tiempo necesario para que este servidor de catálogo complete una transacción.

AvgThroughPut
Muestra el número promedio de transacciones por segundo de este servidor de catálogo.

SumOfSquares
Especifica el valor de la suma de cuadrados para el tiempo de transacción. Este valor mide la desviación de la media en el punto de tiempo especificado.

SampleLen
Indica la duración, en milisegundos, del periodo durante el que se ha realizado el muestreo de los datos.

LastCount
Especifica el periodo de tiempo desde la realización del último muestreo.

LastTotalTime
Especifica el tiempo total actual menos el tiempo total anterior para el ejemplo de datos.

StartTime
Indica la hora a la que se ha iniciado la recopilación de las estadísticas desde la última restauración de los datos. Los datos se restablecen cuando se reinicia el servidor.

Supervisión con la API de estadísticas

La API de estadísticas es la interfaz directa al árbol de estadísticas internas. De forma predeterminada las estadísticas están inhabilitadas, aunque pueden habilitarse estableciendo una interfaz StatsSpec. Una interfaz StatsSpec define cómo WebSphere eXtreme Scale debe supervisar estadísticas.

Acerca de esta tarea

Puede utilizar la API StatsAccessor local para consultar los datos y acceder a las estadísticas sobre cualquier instancia de ObjectGrid que está en la misma Máquina virtual Java (JVM) que el código de ejecución. Si desea más información sobre las interfaces específicas, consulta la documentación de la API. Utilice los pasos siguientes para habilitar la supervisión del árbol de estadísticas internas.

Procedimiento

1. Recupere el objeto StatsAccessor. La interfaz StatsAccessor sigue el patrón singleton. Por lo tanto, aparte de los problemas relacionados con el cargador de clases, debe existir una instancia de StatsAccessor para cada JVM . Esta clase hace las veces de interfaz principal para todas las operaciones de estadísticas locales. El siguiente código es un ejemplo sobre cómo recuperar la clase del descriptor de acceso. Llame a esta operación antes de cualquier otra llamada de ObjectGrid.

```
public class LocalClient
{
```

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

        // Recuperar un descriptor de contexto para StatsAccessor
        StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

    }

}
```

2. Establezca la interfaz StatsSpec de cuadrícula de datos. Establezca esta JVM para recopilar todas las estadísticas sólo en el nivel de ObjectGrid. Debe asegurarse de que una aplicación habilite todas las estadísticas que puedan ser necesarias antes de empezar las transacciones. En el siguiente ejemplo se establece la interfaz StatsSpec utilizando un campo contante estático y una serie de especificación. El uso de un campo constante estático es más fácil porque el campo ya ha definido la especificación. No obstante, si utiliza una serie de especificación, podrá habilitar cualquier combinación de estadísticas que sea necesaria.

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

      // Recuperar un descriptor de contexto para StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Establecer la especificación a través del campo estático
      StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
      accessor.setStatsSpec(spec);

      // Establecer la especificación a través de la serie de especificación
      StatsSpec spec = new StatsSpec("og.all=enabled");
      accessor.setStatsSpec(spec);

}
```

3. Envíe las transacciones ala cuadrícula para obligar a que se recopilen los datos para la supervisión. Para recopilar datos útiles para las estadísticas, debe enviar las transacciones a la cuadrícula de datos. El siguiente extracto de código inserta un registro en MapA, que es un ObjectGridA. Dado que las estadísticas están en un nivel de ObjectGrid, cualquier correlación dentro de ObjectGrid genera los mismos resultados.

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

      // Recuperar un descriptor de contexto para StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Establecer la especificación a través del campo estático
      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");

      // Insertar unidad
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();
}
```

4. Consulte una StatsFact utilizando la API StatsAccessor. Cada vía de acceso de estadísticas está asociada a una interfaz StatsFact. La interfaz StatsFact es un

marcador genérico que se utiliza para organizar y contener un objeto StatsModule. Para poder acceder al módulo de estadísticas real, se debe recuperar el objeto StatsFact.

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

      // Recuperar un descriptor de contexto para StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Establecer la especificación a través del campo estático
      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");

      // Insertar unidad
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // Recuperar StatsFact

      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

}
```

5. Interactúe con el objeto StatsModule. El objeto StatsModule está incluido dentro de la interfaz StatsFact. Puede obtener una referencia al módulo utilizando la interfaz StatsFact. Como la interfaz StatsFact es una interfaz genérica, debe convertir el módulo devuelto en el tipo de StatsModule esperado. Puesto que esta tarea recopila estadísticas de eXtreme Scale, el objeto StatsModule devuelto se convierte en el tipo OGStatsModule. Una vez que se ha convertido el módulo, tendrá acceso a todas las estadísticas disponibles.

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

      // Recuperar un descriptor de contexto para StatsAccessor
      StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

      // Establecer la especificación a través del campo estático
      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");

      // Insertar unidad
      session.begin();
      map.insert("SomeKey", "SomeValue");
      session.commit();

      // Recuperar StatsFact
      StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

      // Recuperar módulo y hora
      OGStatsModule module = (OGStatsModule)fact.getStatsModule();
      ActiveTimeStatistic timeStat =
```

```
        module.getTransactionTime("Default", true);
          double time = timeStat.getMeanTime();

    }
```

Módulos de estadísticas

WebSphere eXtreme Scale utiliza un modelo de estadísticas interno para rastrear y filtrar datos, que es la estructura subyacente que utilizan todas las vistas de datos para recopilar instantáneas de estadísticas. Puede utilizar varios métodos para recuperar la información de módulos de estadísticas.

Visión general

Las estadísticas en WebSphere eXtreme Scale se rastrean y contienen dentro de módulos StatsModules. Dentro del modelo de estadísticas, existen varios tipos de módulos de estadísticas:

OGStatsModule
: Proporciona estadísticas para una instancia de ObjectGrid, incluidos tiempos de respuesta de transacciones.

MapStatsModule
: Proporciona estadísticas para una sola correlación, incluido el número de entradas y la proporción de coincidencias.

QueryStatsModule
: Proporciona estadísticas para consultas, incluido la creación del plan y los tiempos de ejecución.

AgentStatsModule
: Proporciona estadísticas para los agentes de API de DataGrid, incluidos los tiempos de serialización y los tiempos de ejecución.

HashIndexStatsModule
: Proporciona estadísticas para los tiempos de ejecución de mantenimiento y consulta de HashIndex.

SessionStatsModule
: Proporciona estadísticas para el plug-in del gestor de sesiones HTTP.

Para obtener más información sobre los módulos de estadísticas, consulte el paquete com.ibm.websphere.objectgrid.stats de la documentación de la API.

Estadísticas en un entorno local

El modelo se organiza como un árbol n-ario (una estructura de árbol con el mismo grado para todos los nodos) compuesto por todos los tipos de StatsModule mencionados en la lista anterior. Debido a esta estructura de organización, cada nodo del árbol se representa mediante la interfaz StatsFact. La interfaz StatsFact puede representar un módulo individual o un grupo de módulos a efectos de agregación. Por ejemplo, si varios nodos finales en el árbol representan objetos MapStatsModule concretos, el nodo StatsFact padre relativo a estos nodos contiene estadísticas agregadas para todos los módulos hijos. Después de captar un objeto StatsFact, podrá utilizar la interfaz para recuperar el correspondiente StatsModule.

De forma muy parecida a una correlación de árbol, utilice una correspondiente vía de acceso o clave para recuperar un StatsFact específico. La vía de acceso es un valor String[] que consta de cada nodo que está junto a la vía de acceso al hecho solicitado. Por ejemplo, ha creado un ObjectGrid denominado ObjectGridA, que

contiene dos correlaciones: MapA y MapB. La vía de acceso a StatsModule para MapA podría parecerse al siguiente `[ObjectGridA, MapA]`. La vía de acceso a las estadísticas agregadas para las dos correlaciones sería: `[ObjectGridA]`.

Estadísticas en un entorno distribuido

En un entorno distribuido, los módulos de estadísticas se recuperan utilizando una vía de acceso distinta. Dado que un servidor puede contener varias particiones, el árbol de estadísticas necesita realizar un seguimiento de la partición a la que pertenece cada módulo. Como resultado, la vía de acceso para consultar un objeto StatsFact concreto es distinto. Utilizando el ejemplo anterior, aunque añadiendo que las correlaciones existen dentro de la partición 1, la vía de acceso es `[1, ObjectGridA, MapA]` para recuperar ese objeto StatsFact para MapA.

Supervisión con el programa de utilidad `xscmd`

El programa de utilidad **xscmd** sustituye al programa de utilidad **xsadmin** de ejemplo como una herramienta de administración y supervisión completamente soportada. Con el programa de utilidad **xscmd** puede visualizar información textual sobre la topología de WebSphere eXtreme Scale.

Antes de empezar

- Para que el programa de utilidad **xscmd** visualice los resultados, debe haber creado una topología de cuadrícula de datos. Los servidores de catálogo y los servidores de contenedor deben estar iniciados. Si desea más información, consulte "Inicio y detención de los servidores autónomos" en la página 395.
- Consulte "Administración con el programa de utilidad **xscmd**" en la página 415 para obtener más información sobre cómo iniciar el programa de utilidad **xscmd**.

Acerca de esta tarea

Puede utilizar el programa de utilidad **xscmd** para visualizar el diseño actual y el estado específico de la cuadrícula de datos como, por ejemplo, contenido de la cuadrícula. En este ejemplo, el diseño de la cuadrícula de datos de esta tarea consta de una única cuadrícula de datos *ObjectGridA* con una correlación *MapA* que pertenece al conjunto de correlaciones *MapSetA*. Este ejemplo muestra cómo puede visualizar todos los contenedores activos de una cuadrícula de datos e imprimir métricas filtradas relacionadas con el tamaño de correlación de la correlación *MapA*. Para ver todas las posibles opciones del mandato, ejecute el programa de utilidad **xscmd** sin argumentos o con la opción **-help**.

Procedimiento

Supervise el entorno con el programa de utilidad **xscmd**.

- Para habilitar estadísticas para todos los servidores, ejecute el mandato siguiente:
  - UNIX `./xscmd.sh -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
  - Windows `xscmd.bat -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
- Para visualizar todos los servidores de contenedor en línea para una cuadrícula de datos, ejecute el mandato siguiente:
  - UNIX `./xscmd.sh -c showPlacement -g ObjectGridA -ms MapSetA`
  - Windows `xscmd.bat -c showPlacement -g ObjectGridA -ms MapSetA`

  Se visualiza toda la información de contenedores.

Atención: Para obtener esta información cuando Transport Layer Security/Secure Sockets Layer (TLS/SSL) está habilitado, debe iniciar los servidores de catálogo y contenedor con el puerto de servicio JMX establecido. Para establecer el puerto de servicio JMX, puede utilizar la opción **-JMXServicePort** en el script **startOgServer** o puede llamar al método setJMXServicePort en la interfaz ServerProperties.

- Para visualizar información sobre las correlaciones para la cuadrícula de datos ObjectGridA, ejecute el mandato siguiente:
  - UNIX `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA`
  - Windows `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA`
- Para conectarse al servicio de catálogo y visualizar información sobre la correlación MapA par el dominio de servicio de catálogo entero, ejecute el mandato siguiente:
  - 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`

  El programa de utilidad **xscmd** se conecta al servidor MBean que se ejecuta en un servidor de catálogo. Al conectarse a un único servidor de catálogo, puede recuperar información sobre el dominio de servicio de catálogo entero. Un servidor de catálogo puede ejecutarse como proceso autónomo, WebSphere Application Server o incorporado en un proceso de aplicación personalizado. Utilice la opción **-cep** para especificar el nombre de host y puerto del servicio de catálogo. Si incluye una lista de los servidores de catálogo para la opción **-cep**, los servidores de catálogo deben estar dentro del mismo dominio de servicio de catálogo. Puede recuperar estadísticas para un dominio de servicio de catálogo a la vez.
- Para visualizar la colocación configurada y de tiempo de ejecución de la configuración, ejecute uno de los mandatos siguientes:
  - `xscmd -c placementServiceStatus`
  - `xscmd -c placementServiceStatus -g ObjectGridA -ms MapSetA`
  - `xscmd -c placementServiceStatus -ms MapSetA`
  - `xscmd -c placementServiceStatus -g ObjectGridA`

  Puede hacer que el mandato visualice información de colocación para toda la configuración, una sola cuadrícula de datos, un único conjunto de correlaciones o una combinación de una cuadrícula de datos y un conjunto de correlaciones.

Supervisión con PMI de WebSphere Application Server

WebSphere eXtreme Scale da soporte a PMI (Performance Monitoring Infrastructure) cuando se ejecuta en un servidor de aplicaciones WebSphere Application Server o WebSphere Extended Deployment. PMI recopila los datos de rendimiento de aplicaciones en tiempo de ejecución y proporciona interfaces que dan soporte a aplicaciones externas para supervisar datos de rendimiento. Puede utilizar la consola administrativa o la herramienta wsadmin para acceder a los datos de supervisión.

Antes de empezar

Puede utilizar PMI para supervisar el entorno cuando utiliza WebSphere eXtreme Scale junto con WebSphere Application Server.

Acerca de esta tarea

WebSphere eXtreme Scale utiliza la característica PMI personalizada de WebSphere Application Server para añadir su propia instrumentación PMI. Con este enfoque, puede habilitar e inhabilitar WebSphere eXtreme Scale PMI con la consola administrativa o con las interfaces JMX (Java Management Extensions) de la herramienta wsadmin. Además, puede acceder a las estadísticas de WebSphere eXtreme Scale con las interfaces PMI y JMX estándares que utilizan las herramientas de supervisión, incluido Tivoli Performance Viewer.

Procedimiento

1. Habilite eXtreme Scale PMI. Debe habilitar PMI para ver las estadísticas de PMI. Si desea más información, consulte "Habilitación de PMI".
2. Recupere las estadísticas de eXtreme Scale PMI. Vea el rendimiento de las aplicaciones deeXtreme Scale con Tivoli Performance Viewer. Si desea más información, consulte "Recuperar estadísticas de PMI" en la página 466.

Qué hacer a continuación

Para obtener más información sobre la herramienta wsadmin, consulte "Acceso a beans gestionados (MBeans) mediante la herramienta wsadmin" en la página 435.

Habilitación de PMI

Puede utilizar PMI (Performance Monitoring Infrastructure) de WebSphere Application Server para habilitar o inhabilitar estadísticas a cualquier nivel. Por ejemplo, puede elegir que se habiliten las estadísticas de proporción de coincidencias de correlación para una correlación determinada pero no así las estadísticas de número de entradas ni las estadísticas de tiempo de actualización por lotes del cargador. Puede habilitar PMI en la consola de administración o con scripts.

Antes de empezar

El servidor de aplicaciones se debe haber iniciado y debe tener instalada una aplicación habilitada para eXtreme Scale. Para habilitar PMI con el uso de scripts, también debe poder iniciar la sesión y utilizar la herramienta wsadmin. Para obtener más información sobre la herramienta wsadmin, consulte el tema Herramienta wsadmin en el Information Center de WebSphere Application Server.

Acerca de esta tarea

Utilice WebSphere Application Server PMI para proporcionar un mecanismo granular con el que poder habilitar o inhabilitar estadísticas a cualquier nivel. Por ejemplo, puede elegir que se habiliten las estadísticas de proporción de coincidencias de correlación para una correlación determinada pero no así las estadísticas de número de entradas ni las estadísticas de tiempo de actualización por lotes del cargador. Esta sección muestra cómo utilizar la consola administrativa y los scripts wsadmin para habilitar PMI de ObjectGrid.

Procedimiento

- **Habilite PMI en la consola administrativa.**
  1. En la consola administrativa, pulse **Supervisión y ajuste** > **Performance Monitoring Infrastructure** > *nombre_servidor*.

2. Verifique que Performance Monitoring Infrastructure (PMI) se ha seleccionado. De forma predeterminada este valor está habilitado. Si el valor no está habilitado, seleccione el recuadro de selección y reinicie el servidor.
3. Pulse **Personalizado**. En el árbol de configuración, seleccione el módulo de correlaciones ObjectGrid y ObjectGrid. Habilite las estadísticas para cada módulo.

La categoría de tipo de transacción para estadísticas de ObjectGrid se crea en el tiempo de ejecución. Sólo puede ver las subcategorías de las estadísticas de ObjectGrid y de correlación en el separador **Tiempo de ejecución**.

- **Habilite PMI con el uso de scripts.**
  1. Abra un indicador de línea de mandatos. Vaya al directorio *raíz_was*/bin. Especifique **wsadmin** para iniciar la herramienta de línea de mandatos wsadmin.
  2. Modifique la configuración del tiempo de ejecución de PMI de eXtreme Scale. Verifique que PMI se ha habilitado para el servidor mediante los siguientes mandatos:

     ```
     wsadmin>set s1 [$AdminConfig getid /Cell:CELL_NAME/Node:NODE_NAME/Server:
     APPLICATION_SERVER_NAME/]
     wsadmin>set pmi [$AdminConfig list PMIService $s1]
     wsadmin>$AdminConfig show $pmi.
     ```

     Si PMI no se ha habilitado, ejecute los mandatos siguientes para habilitar PMI:

     ```
     wsadmin>$AdminConfig modify $pmi {{enable true}}
     wsadmin>$AdminConfig save
     ```

     Si necesita habilitar PMI, reinicie el servidor.
  3. Establezca las variables para cambiar el conjunto de estadísticas por un conjunto personalizado utilizando los siguientes mandatos:

     ```
     wsadmin>set perfName [$AdminControl completeObjectName type=Perf,
     process=APPLICATION_SERVER_NAME,*]
     wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
     wsadmin>set params [java::new {java.lang.Object[]} 1]
     wsadmin>$params set 0 [java::new java.lang.String custom]
     wsadmin>set sigs [java::new {java.lang.String[]} 1]
     wsadmin>$sigs set 0 java.lang.String
     ```
  4. Establezca el conjunto de estadísticas que desea personalizar utilizando el siguiente mandato:

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName setStatisticSet $params $sigs
     ```
  5. Establezca las variables para habilitar las estadísticas de PMI de objectGridModule utilizando los siguientes mandatos:

     ```
     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. Establezca la serie de estadísticas utilizando el siguiente mandato:

     ```
     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. Establezca la serie de estadísticas utilizando el siguiente mandato:

```
wsadmin>$AdminControl invoke_jmx $perfOName setCustomSetString $params2 $sigs2
```

Estos pasos habilitan el PMI de tiempo de ejecución de eXtreme Scale, pero no modifican la configuración de PMI. Si reinicia el servidor de aplicaciones, los valores de PMI se pierden excepto para la habilitación de PMI principal.

Ejemplo

Puede efectuar los siguientes pasos para habilitar las estadísticas de PMI para la aplicación de ejemplo:

1. Inicie la aplicación utilizando la dirección web `http://host:puerto/ObjectGridSample`, donde el host y el puerto son el nombre del host y el número de puerto HTTP del servidor en el que se ha instalado el ejemplo.
2. En la aplicación de ejemplo, pulse ObjectGridCreationServlet, y luego pulse los botones de acción 1, 2, 3, 4 y 5 para generar acciones para ObjectGrid y correlaciones. No cierre esta página de servlet ahora.
3. En la consola administrativa, pulse **Supervisión y ajuste** > **Performance Monitoring Infrastructure** > *nombre_servidor* Pulse la pestaña **Tiempo de ejecución**.
4. Pulse el botón de selección **Personalizado**.
5. Expanda el módulo de correlaciones de ObjectGrid en el árbol de tiempo de ejecución y pulse el enlace clusterObjectGrid. Bajo el grupo de correlaciones de ObjectGrid, hay una instancia de ObjectGrid llamada clusterObjectGrid, y bajo el grupo clusterObjectGrid existen cuatro correlaciones: contadores, empleados, oficinas y sitios. En la instancia de ObjectGrids, existe la instancia de clusterObjectGrid y bajo dicha instancia hay un tipo de transacción llamado DEFAULT.
6. Puede habilitar las estadísticas que desee. Por ejemplo, puede habilitar una cantidad de entradas de correlación para la correlación de empleados y un tiempo de respuesta de transacción para el tipo de transacción DEFAULT.

Qué hacer a continuación

Una vez que se ha habilitado PMI, puede ver las estadísticas PMI con la consola administrativa o con el uso de scripts.

Recuperar estadísticas de PMI

Al recuperar estadísticas de PMI, podrá ver el rendimiento de las aplicaciones eXtreme Scale.

Antes de empezar

- Habilite el rastreo de estadísticas de PMI para el entorno. Si desea más información, consulte “Habilitación de PMI” en la página 464.
- En las vías de acceso de esta tarea se da por supuesto que se recuperan estadísticas de la aplicación de ejemplo, aunque puede utilizar estas estadísticas para cualquier otra aplicación con pasos parecidos.
- Si utiliza la consola administrativa para recuperar estadísticas, debe poder iniciar la sesión en la consola administrativa. Si utiliza scripts, debe poder iniciar la sesión en wsadmin.

Acerca de esta tarea

Puede recuperar estadísticas de PMI y verlas en Tivoli Performance Viewer efectuando los pasos en la consola administrativa o con scripts.

- Pasos en la consola administrativa
- Pasos en los scripts

Para obtener más información sobre las estadísticas que pueden recuperarse, consulte "Módulos PMI" en la página 468.

Procedimiento

- Recupere estadísticas de PMI en la consola administrativa.
  1. En la consola administrativa, pulse **Supervisión y ajuste** > **Performance Viewer** > **Actividad actual**
  2. Seleccione el servidor que desee supervisar utilizando Tivoli Performance Viewer y luego habilite la supervisión.
  3. Pulse el servidor para ver la página de Performance Viewer.
  4. Expanda el árbol de configuración. Pulse **Correlaciones de ObjectGrid** > **clusterObjectGrid** seleccione **employees**. Expanda **ObjectGrids** > **clusterObjectGrid** y seleccione **DEFAULT**.
  5. En la aplicación de ejemplo de ObjectGrid, vaya al servlet ObjectGridCreationServlet, pulse el botón 1 y luego rellene las correlaciones. Puede ver las estadísticas en el visor.
- Recupere estadísticas de PMI con scripts.
  1. En un indicador de línea de mandatos, vaya al directorio *raíz_was*/bin. Escriba `wsadmin` para iniciar la herramienta wsadmin.
  2. Establezca las variables para el entorno utilizando los siguientes mandatos:

     ```
     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. Establezca las variables para obtener estadísticas de mapModule utilizando los siguientes mandatos:

     ```
     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. Obtenga estadísticas de mapModule utilizando el siguiente mandato:

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params $sigs
     ```

  5. Establezca las variables para obtener estadísticas de objectGridModule utilizando los siguientes mandatos:

     ```
     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. Obtenga las estadísticas de objectGridModule utilizando el siguiente mandato:

     ```
     wsadmin>$AdminControl invoke_jmx $perfOName getStatsString $params2 $sigs2
     ```

Resultados

Puede ver las estadísticas en Tivoli Performance Viewer.

Módulos PMI

Puede supervisar el rendimiento de las aplicaciones con los módulos PMI (Performance Monitoring Infrastructure).

objectGridModule

objectGridModule contiene una estadística de tiempo: el tiempo de respuesta de la transacción. Una transacción se define como la duración entre la llamada del método Session.begin y la llamada del método Session.commit. Este intervalo de tiempo se considera el tiempo de respuesta de la transacción. El elemento raíz de objectGridModule, "root", hace las veces de punto de entrada de las estadísticas de WebSphere eXtreme Scale. Este elemento raíz tiene ObjectGrids como sus elementos hijo, que tienen tipos de transacción como elementos hijo. La estadística de tiempo de respuesta está asociado a cada tipo de transacción.

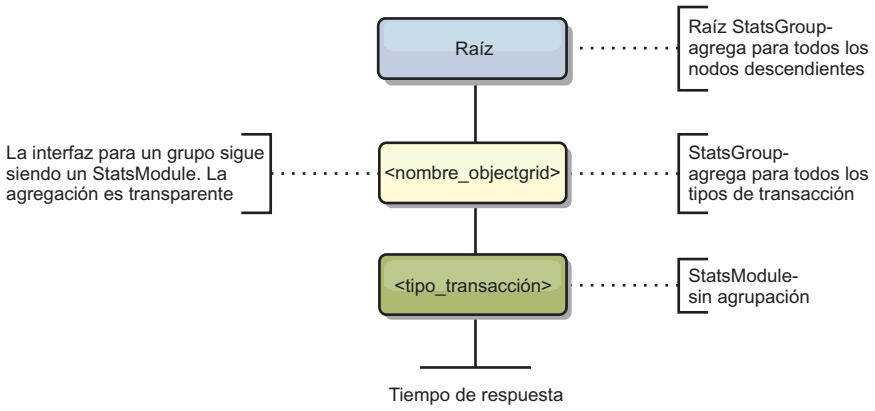


Figura 57. Estructura del módulo ObjectGridModule

El siguiente diagrama muestra un ejemplo de la estructura de ObjectGridModule. En este ejemplo, existen dos instancias de ObjectGrid en el sistema: el ObjectGrid A y el ObjectGrid B. La instancia A de ObjectGrid tiene dos tipos de transacciones: la A y la predeterminada. La instancia de ObjectGrid B tiene sólo el tipo de transacción predeterminado.

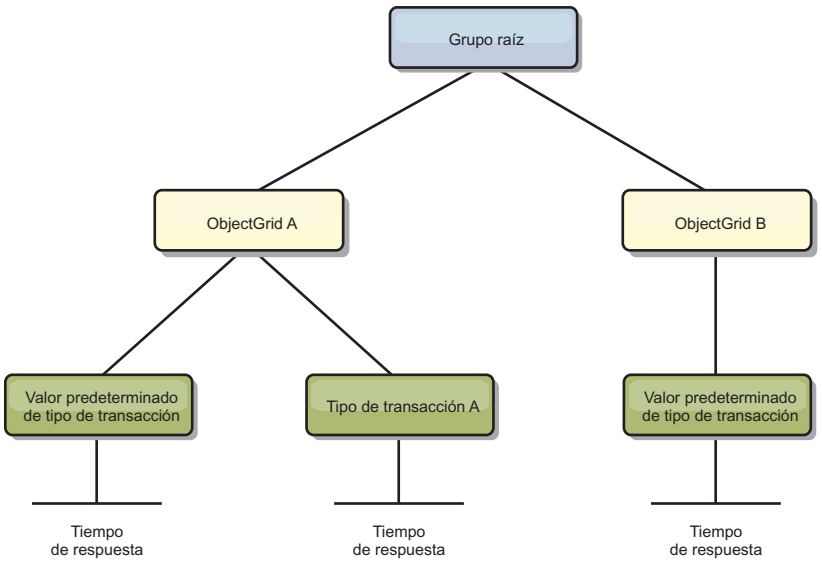


Figura 58. Ejemplo de estructura del módulo ObjectGridModule

Los tipos de transacción los definen los desarrolladores de transacciones porque conocen los tipos de transacciones que utilizan sus aplicaciones. El tipo de transacción se establece utilizando el siguiente método Session.setTransactionType(String):

```
/**
* Establece el tipo de transacción para futuras transacciones.
*
* Después de llamar a este método, todas las transacciones futuras tendrán el mismo
* tipo hasta que se establezca otro tipo de transacción. Si no se establece ningún
* tipo de transacción , se utiliza el tipo de transacción TRANSACTION_TYPE_DEFAULT
* predeterminado.
*
* Los tipos de transacción se usan principalmente para fines de seguimiento de datos
* estadísticos.
* Los usuarios pueden definir previamente los tipos de transacciones que se
* ejecutan en una aplicación.
La idea es clasificar las transacciones con las mismas características en una
* categoría (tipo), de modo que una estadística de tiempo de respuesta se pueda
* utilizar para realizar un seguimiento de cada tipo de transacción.
*
* Este seguimiento resulta útil cuando la aplicación tiene tipos diferentes de
* transacciones.
* Entre ellos, algunos tipos de transacciones, como las transacciones de
* actualización, tardan más en procesarse que otras transacciones como, por
* ejemplo, las de sólo lectura. Utilizando el tipo de transacción, se puede
* realizar un seguimiento de las transacciones diferentes por estadísticas
* diferentes, por lo que las estadísticas resultan más útiles.
*
* @param tranType el tipo de transacción para las transacciones futuras.
*/
void setTransactionType(String tranType);
```

El ejemplo siguiente establece el tipo de transacción en `updatePrice`:

```
// Establecer el tipo de transacción en updatePrice
// El período de tiempo entre session.begin() y session.commit() se reflejará
// en la estadística de tiempo "updatePrice".
session.setTransactionType("updatePrice");
session.begin();
map.update(stockId, new Integer(100));
session.commit();
```

La primera línea indica que el tipo de transacción subsiguiente es updatePrice. Existe una estadística updatePrice en la instancia de ObjectGrid que se corresponde a la sesión del ejemplo. Utilizando las interfaces JMX (Java Management Extensions), puede obtener el tiempo de respuesta de la transacción para las transacciones updatePrice. También puede obtener la estadística agregada para todos los tipos de transacciones en la instancia de ObjectGrid especificada.

mapModule

El mapModule contiene tres estadísticas que están relacionadas con las correlaciones de eXtreme Scale:

- **Proporción de coincidencias de la correlación** -*BoundedRangeStatistic*: efectúa un seguimiento de la proporción de coincidencias de una correlación. La proporción de coincidencias es un valor flotante entre 0 y 100 inclusive, que representa el porcentaje de coincidencias de una correlación en relación con las operaciones get de la correlación.
- **Número de entradas** - *CountStatistic*: efectúa un seguimiento del número de entradas de la correlación.
- **Tiempo de respuesta de la actualización por lotes del cargador**-*TimeStatistic*: efectúa un seguimiento del tiempo de respuesta que se utiliza para la operación de actualización por lotes del cargador.

El elemento raíz de mapModule, "root", hace las veces de punto de entrada para las estadísticas de la correlación de ObjectGrid. Este elemento raíz tiene ObjectGrids como elementos hijo, que tienen correlaciones como sus elementos hijo. Cada instancia de correlación tiene tres estadísticas listadas. En el diagrama siguiente se muestra la estructura de mapModule:

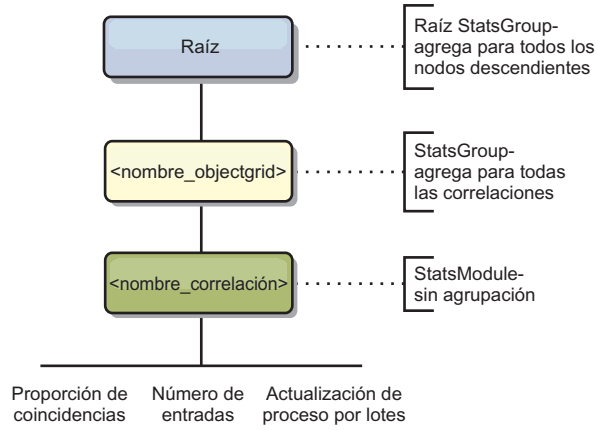
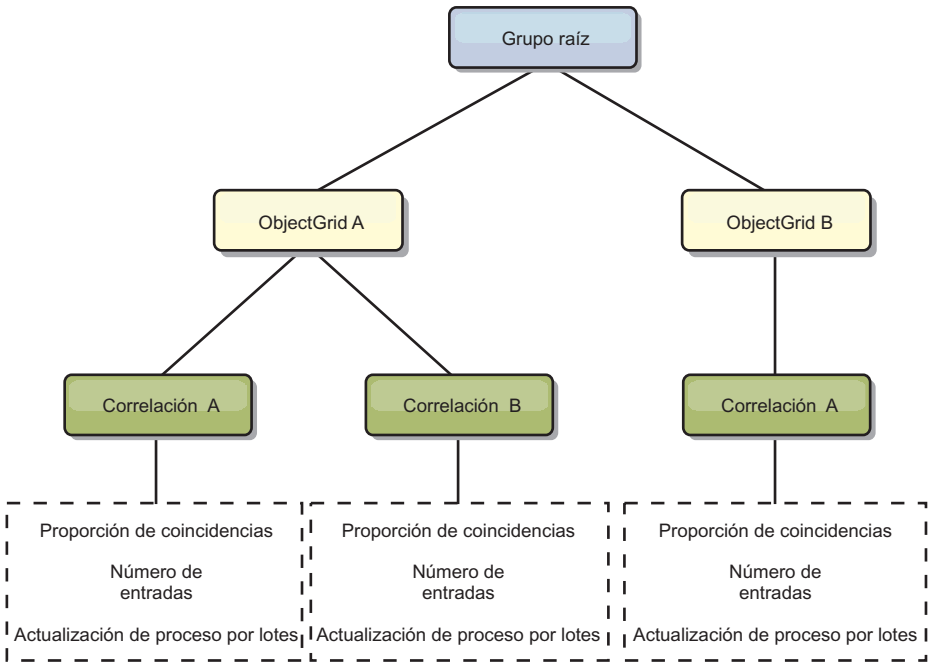


Figura 59. Estructura de mapModule

El siguiente diagrama muestra un ejemplo de la estructura de mapModule:

Figura 60. Ejemplo de la estructura del módulo mapModule



hashIndexModule

hashIndexModule contiene las siguientes estadísticas relacionadas con los índices de nivel de correlación:

- **Recuento de búsquedas** - *CountStatistic*: el número de invocaciones para la operación de búsqueda de índices.
- **Recuento de colisiones** - *CountStatistic*: el número de colisiones para la operación de búsqueda.
- **Recuento de anomalías** - *CountStatistic*: el número de anomalías para una operación de búsqueda.
- **Recuento de resultados** - *CountStatistic*: el número de claves devueltas de la operación de búsqueda.
- **Recuento de actualizaciones de proceso por lotes** - *CountStatistic*: el número de actualizaciones de proceso por lotes realizadas en relación con este índice. Cuando se cambia la correlación correspondiente en algún modo, el índice llamará a su método doBatchUpdate(). Esta estadística le indicará con que frecuencia se cambia o actualiza el índice.
- **Periodo de tiempo de la operación de búsqueda**-*TimeStatistic*: el intervalo de tiempo que la operación de búsqueda tarda en llevarse a cabo

El elemento raíz de hashIndexModule, "root", hace las veces de punto de entrada de las estadísticas de HashIndex. Este elemento raíz tiene ObjectGrids como elementos hijo, ObjectGrids tienen correlaciones como elementos hijo, que finalmente tienen HashIndexes como elementos hijo y nodos finales del árbol. Cada instancia de HashIndex tiene tres estadísticas listadas. En el diagrama siguiente se muestra la estructura de hashIndexModule:

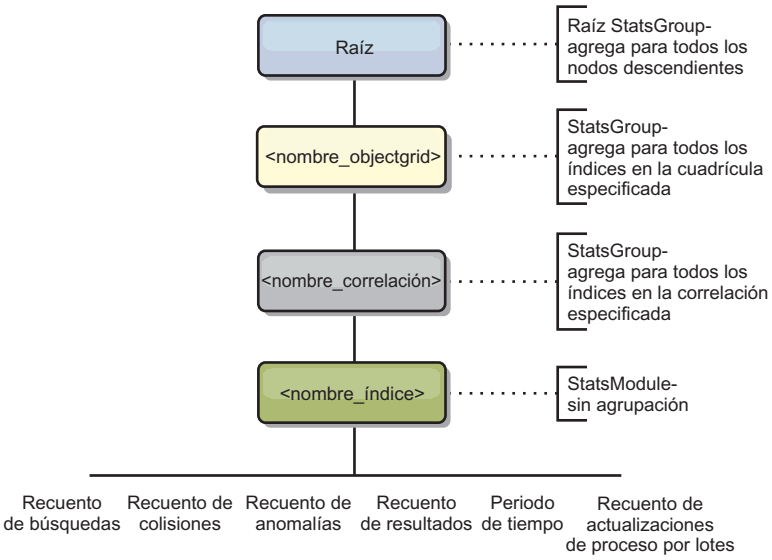


Figura 61. Estructura del módulo hashIndexModule

El siguiente diagrama muestra un ejemplo de la estructura de hashIndexModule:

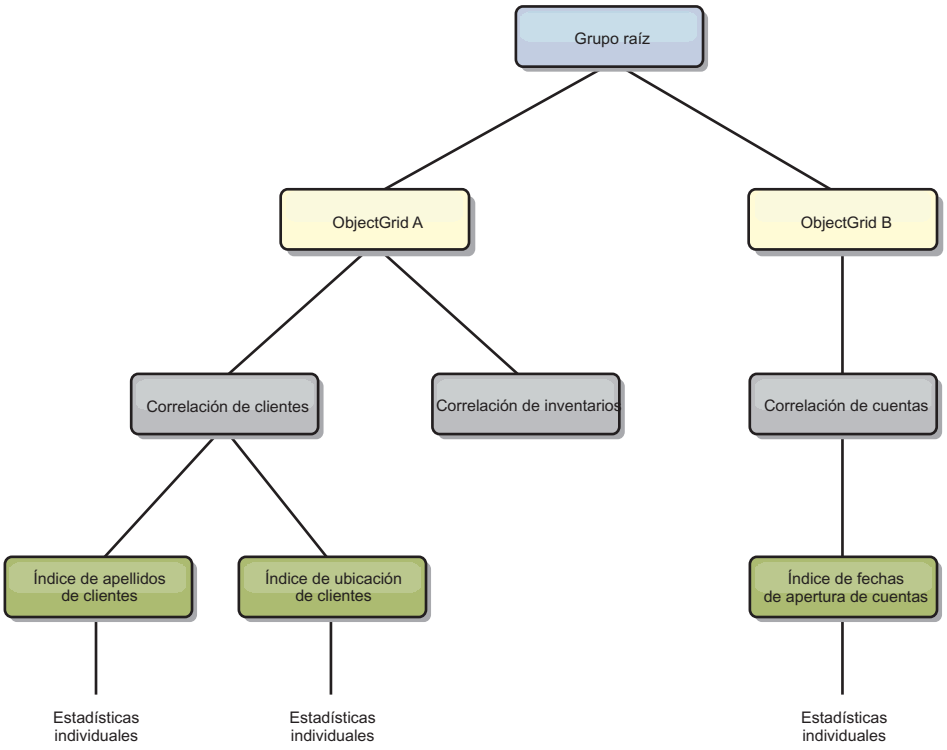


Figura 62. Ejemplo de estructura del módulo hashIndexModule

agentManagerModule

agentManagerModule contiene estadísticas relacionadas con los agentes de nivel de correlación:

- **Periodo de tiempo de reducción** - *TimeStatistic*: el intervalo de tiempo para que el agente termine la operación de reducción.

- **Periodo de tiempo total** - *TimeStatistic*: el intervalo de tiempo para que el agente complete todas las operaciones.
- **Periodo de tiempo de serialización de agente** - *TimeStatistic*: el intervalo de tiempo para serializar el agente.
- **Periodo de tiempo de inflación de agente** - *TimeStatistic*: el intervalo de tiempo que se tarda en inflar el agente en el servidor.
- **Periodo de tiempo de serialización de resultados** - *TimeStatistic*: el intervalo de tiempo para serializar los resultados de un agente.
- **Periodo de tiempo de inflación de resultados** - *TimeStatistic*: el intervalo de tiempo para inflar los resultados del agente.
- **Recuento de anomalías** - *CountStatistic*: el número de veces que el agente ha fallado.
- **Recuento de invocaciones** - *CountStatistic*: el número de veces que se ha invocado AgentManager.
- **Recuento de particiones** - *CountStatistic*: el número de particiones a las que se envía el agente.

El elemento raíz de agentManagerModule, "root", hace las veces de punto de entrada de las estadísticas de AgentManager. Este elemento raíz tiene ObjectGrids como elementos hijo, ObjectGrids tiene correlaciones como elementos hijo, que por último tienen instancias de AgentManager como elementos hijo y nodos finales del árbol. Cada instancia de AgentManager tiene estadísticas.

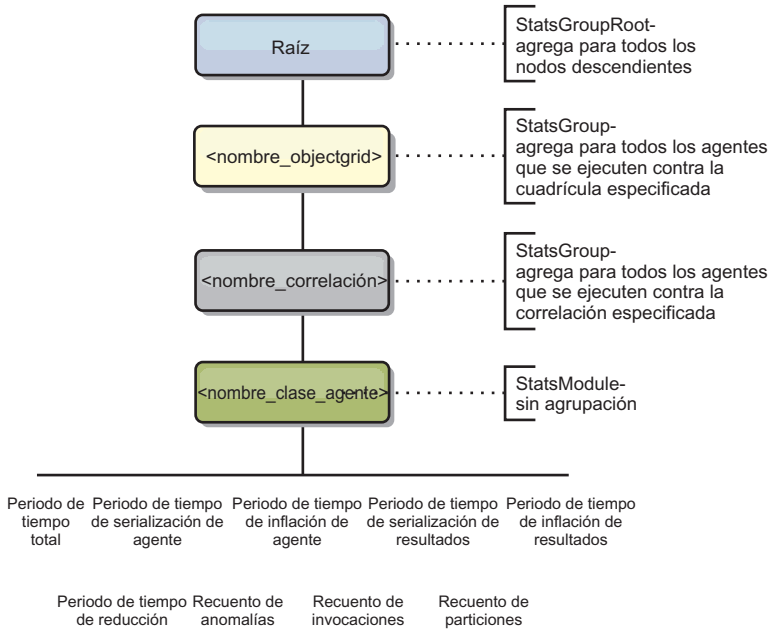


Figura 63. Estructura de agentManagerModule

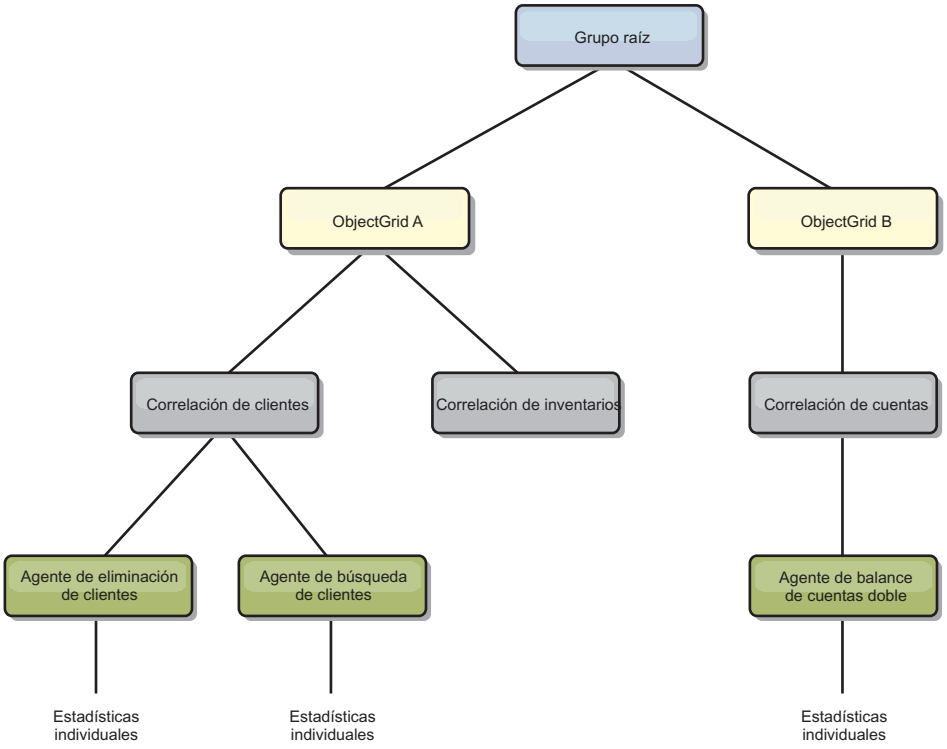


Figura 64. Ejemplo de la estructura de agentManagerModule

queryModule

queryModule contiene estadísticas relacionadas con las consultas de eXtreme Scale:

- **Tiempo de creación de plan** - *TimeStatistic*: el intervalo de tiempo para crear el plan de consulta.
- **Tiempo de ejecución** - *TimeStatistic*: el intervalo de tiempo para ejecutar la consulta.
- **Recuento de ejecuciones** - *CountStatistic*: el número de veces que se ha ejecutado la consulta.
- **Recuento de resultados** - *CountStatistic*: el recuento para cada conjunto de resultados de cada ejecución de consulta.
- **FailureCount** - *CountStatistic*: el número de veces que la consulta ha fallado.

El elemento raíz de queryModule, "root", hace las veces de punto de entrada de las estadísticas de Query. Este elemento raíz tiene ObjectGrids como elementos hijo, que tienen objetos Query como elementos hijo y nodos finales del árbol. Cada instancia de consulta tiene tres estadísticas listadas.

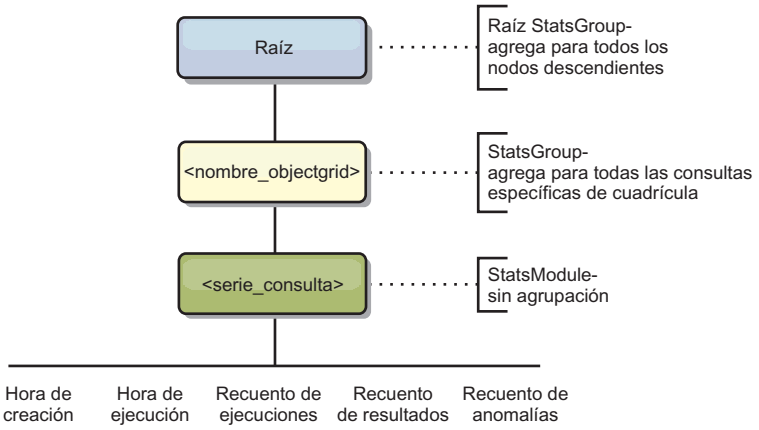


Figura 65. Estructura de queryModule

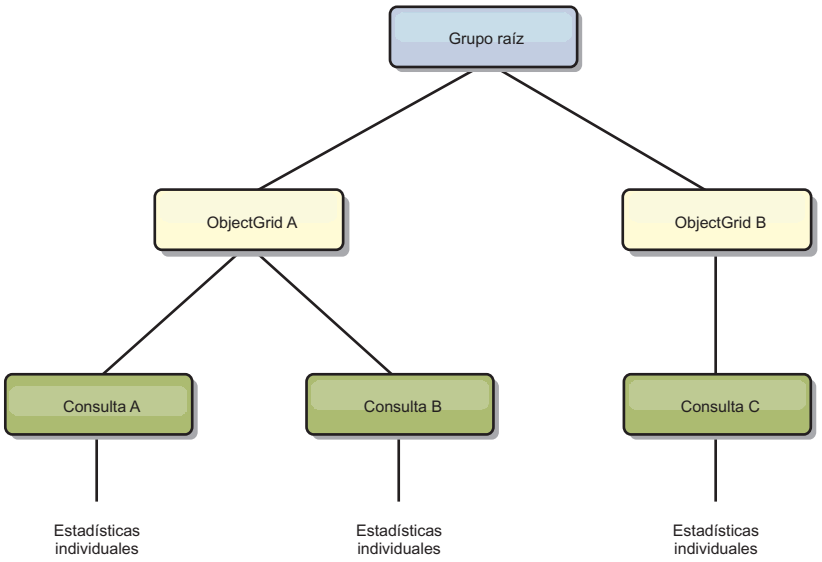


Figura 66. Ejemplo de la estructura de queryModule de QueryStats.jpg

Acceso a beans gestionados (MBeans) mediante la herramienta wsadmin

Puede utilizar el programa de utilidad wsadmin proporcionado en WebSphere Application Server para acceder a la información de beans gestionados (MBeans).

Procedimiento

Ejecute la herramienta wsadmin desde el directorio `bin` en la instalación de WebSphere Application Server. En el siguiente ejemplo se recupera una vista de la colocación de fragmentos actual en un eXtreme Scale dinámico. Puede ejecutar la herramienta wsadmin desde cualquier instalación donde se esté ejecutando eXtreme Scale. No necesita ejecutar la herramienta wsadmin en el servicio de catálogo.

```
$ 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>
```

Supervisión con beans gestionados (MBeans)

Puede utilizar los beans gestionados (MBeans) para rastrear las estadísticas en el entorno.

Antes de empezar

Para que se registren los atributos, debe habilitar las estadísticas. Puede habilitar las estadísticas de una de las formas siguientes:

- **Con el archivo de propiedades del servidor:**

  Puede habilitar las estadísticas en el archivo de propiedades del servidor con una entrada de clave-valor de `statsSpec=<StatsSpec>`. A continuación, algunos ejemplos de valores posibles:

  – Para habilitar todas las estadísticas, utilice `statsSpec=og.all=enabled`.

  – Para habilitar sólo las estadísticas de ObjectGrid, utilice `statsSpec=og.all=enabled`. Para ver una descripción de todas las especificaciones estadísticas posibles, consulte la API StatsSpec en la documentación de la API.

  Si desea más información sobre el archivo de propiedades de servidor, consulte Archivo de propiedades de servidor .

- **Con un bean gestionado:**

  Puede habilitar las estadísticas utilizando el atributo StatsSpec en el MBean ObjectGrid. Para obtener más información, consulte el tema API StatsSpec de la documentación de la API.

- **A través de programa:**

  También puede habilitar las estadísticas a través de programas con la interfaz StatsAccessor, que se recupera con la clase StatsAccessorFactory. Utilice la interfaz de un entorno de cliente o cuando necesite supervisar una cuadrícula de datos que se ejecute en el proceso actual.

Procedimiento

- **Acceda a estadísticas de MBeans mediante la herramienta wsadmin.**

  Para obtener más información, consulte “Acceso a beans gestionados (MBeans) mediante la herramienta wsadmin” en la página 435.

- **Acceda a estadísticas de MBeans mediante programación.**

Para obtener más información, consulte “Acceso a beans gestionados (MBeans) mediante programación” en la página 435.

Ejemplo

Si desea ver un ejemplo sobre cómo utilizar los beans gestionados, consulte Ejemplo: Programa de utilidad **xsadmin**.

Supervisión con herramientas de proveedor

WebSphere eXtreme Scale se puede supervisar utilizando varias soluciones populares de supervisión empresarial. Los agentes de plug-in se incluyen para IBM Tivoli Monitoring e Hyperic HQ, que supervisan WebSphere eXtreme Scale utilizando los beans de gestión a los que se puede acceder públicamente. CA Wily Introscope utiliza la instrumentación de métodos Java para capturar las estadísticas.

Supervisión con IBM Tivoli Enterprise Monitoring Agent for WebSphere eXtreme Scale

IBM Tivoli Enterprise Monitoring Agent es una solución de supervisión con muchas características que puede utilizar para supervisar bases de datos, sistemas operativos y servidores en entornos distribuidos y de host. WebSphere eXtreme Scale incluye un agente personalizado que puede utilizar para realizar introspecciones de los beans de gestión de eXtreme Scale. Esta solución funciona de forma eficaz tanto para los despliegues de eXtreme Scale autónomo, como para los despliegues de WebSphere Application Server.

Antes de empezar

- Instale WebSphere eXtreme Scale versión 7.0.0 o posterior.

  Además, se deben habilitar las estadísticas para recopilar los datos estadísticos de los servidores WebSphere eXtreme Scale. En los apartados “Supervisión con beans gestionados (MBeans)” en la página 476 y Ejemplo: Programa de utilidad **xsadmin** se describen varias opciones para habilitar las estadísticas.
- Instale IBM Tivoli Monitoring versión 6.2.1 con el fixpack 2 o posterior.
- Instale el agente de sistema operativo Tivoli en cada servidor o sistema principal en el que se ejecuten los servidores eXtreme Scale.
- Instale el agente WebSphere eXtreme Scale, que puede descargar de forma gratuita desde el sitio IBM Open Process Automation Library (OPAL).

Complete los pasos siguientes para instalar y configurar Tivoli Monitoring Agent:

Procedimiento

1. Instale Tivoli Monitoring Agent for WebSphere eXtreme Scale.

   Descargue la imagen de instalación de Tivoli y extraiga sus archivos en un directorio temporal.
2. Instale los archivos de soporte de aplicaciones de eXtreme Scale.

   Instale el soporte de aplicaciones de eXtreme Scale en cada uno de los siguientes despliegues.
   - Tivoli Enterprise Portal Server (TEPS)
   - Enterprise Desktop Client (TEPD)
   - Tivoli Enterprise Monitoring Server (TEMS)

a. Desde el directorio temporal que ha creado, inicie una nueva ventana de mandato y ejecute el archivo ejecutable apropiado para la plataforma. El script de instalación detecta automáticamente el tipo de despliegue Tivoli (TEMS, TEPD o TEPS). Puede instalar cualquier tipo en un único host o en varios hosts; y los tres tipos de despliegue requieren la instalación de los archivos de soporte de aplicaciones del agente eXtreme Scale.

b. En la ventana del **Instalador**, verifique que las selecciones de los componentes Tivoli desplegados son correctas. Pulse **Siguiente**.

c. Si se le solicita, someta el nombre de host y las credenciales administrativas. Pulse **Siguiente**.

d. Seleccione **Monitoring Agent for WebSphere eXtreme Scale**. Pulse **Siguiente**.

e. Se le notificará qué acciones de instalación se llevarán a cabo. Pulse **Siguiente**, y podrá ver el progreso de la instalación hasta su finalización.

Después de completar el procedimiento, se instalan todos los archivos de soporte de aplicaciones necesarios para el agente de WebSphere eXtreme Scale.

3. Instale el agente en cada uno de los nodos de eXtreme Scale.

   Instale un agente de sistema operativo Tivoli en cada uno de los sistemas. No tendrá que configurar ni iniciar este agente. Utilice la misma imagen de instalación del paso anterior para ejecutar el archivo ejecutable específico de la plataforma.

   Como directriz es necesario instalar sólo un agente por host. Cada agente es capaz de dar soporte a muchas instancias de servidores de eXtreme Scale. Para obtener un mejor rendimiento, utilice una instancia de agente para supervisar aproximadamente 50 servidores de eXtreme Scale.

   a. Desde la pantalla de bienvenida del asistente de instalación, pulse **Siguiente** para abrir la pantalla para especificar la información de la vía de acceso de instalación.

   b. Para el campo **Directorio de instalación de Tivoli Monitoring**, escriba o vaya a: `C:\IBM\ITM` (o `/opt/IBM/ITM`). Para el campo **Ubicación para soporte instalable**, verifique que el valor visualizado es correcto y pulse **Siguiente**.

   c. Seleccione los componentes que desea añadir como, por ejemplo, **Realizar una instalación local de la solución** y pulse **Siguiente**.

   d. Seleccione las aplicaciones para las que añade soporte seleccionando la aplicación como, por ejemplo, **Monitoring Agent for WebSphere eXtreme Scale**, y pulse **Siguiente**.

   e. Podrá ver el progreso hasta que el soporte de aplicación se añada correctamente.

   Nota: Repita estos pasos en cada uno de los nodos de eXtreme Scale. También puede utilizar una instalación silenciosa. Consulte el Centro de información de IBM Tivoli Monitoring si desea más información sobre la instalación silenciosa.

4. Configure el agente WebSphere eXtreme Scale.

   Es necesario configurar cada uno de los agentes instalados para supervisar cualquier servidor de catálogo, servidor de eXtreme Scale o ambos.

   Los pasos para configurar las plataformas Windows y UNIX son diferentes. La configuración de la plataforma Windows se completa con la interfaz de usuario de **Manage Tivoli Monitoring Services**. La configuración de las plataformas UNIX se basan en la línea de mandatos.

   Windows Utilice los pasos siguientes para configurar inicialmente el agente en Windows

a. Desde la ventana **Manage Tivoli Enterprise Monitoring Services**, pulse **Inicio** > **Todos los programas** > **IBM Tivoli Monitoring** > **Manage Tivoli Monitoring Services**.
b. Pulse con el botón derecho del ratón **Monitoring Agent for WebSphere eXtreme Scale** y seleccione **Configurar utilizando valores predeterminados**, que abre una ventana para crear una instancia exclusiva del agente.
c. Elija un nombre exclusivo: por ejemplo, `instance1`, y pulse **Siguiente**.

- Si planifica supervisar los servidores eXtreme Scale autónomos, complete los pasos siguientes:
  a. Actualice los parámetros Java, asegúrese de que el valor de **Java Home** es correcto. Los argumentos de JVM se pueden dejar vacíos. Pulse **Siguiente**.
  b. Seleccione el tipo de **Tipo de conexión de servidor MBean**, Utilice `Servidor compatible con JSR-160` para los servidores eXtreme Scale autónomos. Pulse **Siguiente**.
  c. Si la seguridad está habilitada, actualice los valores **ID de usuario** y **Contraseña**. Deje el valor **URL de servicio JMX** tal cual. Altere temporalmente este valor más adelante. Deje el campo **Información de vía de acceso de clase JMX** tal cual. Pulse **Siguiente**.

  Para configurar los servidores para el agente en Windows, complete los pasos siguientes:
  a. Configure las instancias de subnodo de los servidores eXtreme Scale en el panel **Servidores de cuadrícula de WebSphere eXtreme Scale**. Si no existe ningún servidor de contenedor en el sistema, pulse **Siguiente** para seguir con el panel de servicio de catálogo.
  b. Si existen varios servidores de contenedor eXtreme Scale en el sistema, configure el agente para supervisar cada uno de los servidores.
  c. Puede añadir tantos servidores eXtreme Scale como necesite, si sus nombres y puertos son exclusivos, pulsando **Nuevo**. (Cuando se inicia un servidor eXtreme Scale, se debe especificar un valor JMXPort.)
  d. Tras configurar los servidores de contenedor, pulse **Siguiente**, que le llevará al panel **Servidores de catálogos de WebSphere eXtreme Scale**.
  e. Si no tiene ningún servidor de catálogo, pulse **Aceptar**. Si tiene servidores de catálogos, añada una nueva configuración para cada servidor, tal como ha hecho con los servidores de contenedor. De nuevo, elija un nombre exclusivo, preferentemente el mismo nombre que se utiliza cuando se inicia el servicio de catálogo. Pulse **Aceptar** para finalizar.
- Si planifica supervisar los servidores para el agente en los servidores eXtreme Scale que se incorporan dentro de un proceso WebSphere Application Server, complete los pasos siguientes:
  a. Actualice los parámetros Java, asegúrese de que el valor de **Java Home** es correcto. Los argumentos de JVM se pueden dejar vacíos. Pulse **Siguiente**.
  b. Seleccione el **Tipo de conexión de servidor MBean**. Seleccione la versión de WebSphere Application Server que sea apropiada para el entorno. Pulse **Siguiente**.
  c. Asegúrese de que la información de WebSphere Application Server del panel es correcta. Pulse **Siguiente**.
  d. Añada sólo una definición de subnodo. Dé un nombre a la definición del subnodo, pero actualice la definición del puerto. Dentro del entorno WebSphere Application Server, el agente de nodo puede recopilar los

datos de todos los servidores de aplicaciones gestionados por el agente de nodo que se ejecuta en el sistema. Pulse **Siguiente**.

e. Si no existe ningún servidor de catálogo en el entorno, pulse **Aceptar**. Si tiene servidores de catálogos, añada una nueva configuración para cada servidor de catálogo, de forma similar a los servidores de contenedor. Elija un nombre exclusivo para el servicio de catálogos, preferentemente, el mismo nombre que utilice al iniciar el servicio de catálogos. Pulse **Aceptar** para finalizar.

Nota: No es necesario que los servidores de contenedor utilicen una ubicación compartida con el servicio de catálogos.

Ahora que el agente y los servidores están configurados y listos, en la ventana siguiente, pulse con el botón derecho del ratón `instance1` para iniciar el agente.

UNIX Para configurar el agente en la plataforma UNIX en la línea de mandatos, complete los pasos siguientes:

A continuación aparece un ejemplo para servidores autónomos que utiliza un tipo de conexión compatible con JSR160. El ejemplo muestra tres contenedores eXtreme Scale en el host único (rhea00b02) y las direcciones del receptor JMX son 15000,15001 y 15002 respectivamente. No hay ningún servidor de catálogo.

La salida del programa de utilidad de configuración aparece en *`cursiva monoespaciado`*, mientras que la respuesta del usuario aparece en **negrita monoespaciado**. (Si no era necesario ninguna respuesta del usuario, el valor predeterminado se seleccionó pulsando la tela Intro.)

```
rhea00b02 # ./itmcmd config -A xt
Se ha iniciado la configuración del agente...
Especifique el nombre de instancia (el valor predeterminado es: ):  inst1
¿Editar los valores de "Monitoring Agent for WebSphere eXtreme Scale"? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):
¿Editar valores 'Java'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):
Directorio inicio de Java (el valor predeterminado es: C:\Archivos de programa\IBM\Java50):  /opt/OG61/java
Nivel de rastreo Java [ 1=Error, 2=Aviso, 3=Información, 4=Depuración mínima, 5=Depuración media, 6=Depuración máxima,
    7=Todos ] (el valor predeterminado es: 1):
Argumentos de JVM (el valor predeterminado es: ):
¿Editar valores de 'Conexión'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):
Tipo de conexión de servidor MBean [ 1=Servidor compatible con JSR-160, 2=WebSphere Application Server versión 6.0,
3=WebSphere Application Server versión 6.1, 4=WebSphere Application Server versión 7.0 ] (el valor predeterminado es: 1): 1
¿Editar valores de 'Servidor compatible con JSR-160'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):
ID de usuario JMX (el valor predeterminado es: ):
Especificar contraseña JMX (el valor predeterminado es: ):
Vuelva a especificar : contraseña JMX ( el valor predeterminado es: ):
URL de servicio de JMX (el valor predeterminado es: service:jmx:rmi:///jndi/rmi://localhost:port/objectgrid/MBeanServer):
----------------------------------------
Información de la classpath JMX
Vía de acceso base de JMX (el valor predeterminado es: ):
Vía de acceso de clases de JMX (el valor predeterminado es: ):
Directorios JAR de JMX (el valor predeterminado es: ):
¿Editar valores de 'Servicio de catálogo de WebSphere eXtreme Scale'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):  2
¿Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1): 1
¿No hay disponible ningún valor de 'Servidores de cuadrícula de WebSphere eXtreme Scale'?
Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale', [1=Añadir, 2=Editar, 3=Suprimir, 4=Siguiente, 5=Salir]
(el valor predeterminado es: 4): 1
Servidores de cuadrícula de WebSphere eXtreme Scale (el valor predeterminado es: ): rhea00b02_c0
URL del servicio JMX (el valor predeterminado es: service:jmx:rmi:///jndi/rmi://localhost:<puerto>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15000/objectgrid/MBeanServer

Valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale': WebSphere eXtreme Scale Grid Servers=ogx
Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale', [1=Añadir, 2=Editar, 3=Suprimir, 4=Siguiente, 5=Salir]
(el valor predeterminado es: 4): 1
Servidores de cuadrícula de WebSphere eXtreme Scale (el valor predeterminado es: ): rhea00b02_c1
URL del servicio JMX (el valor predeterminado es: service:jmx:rmi:///jndi/rmi://localhost:<puerto>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer

Valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale': WebSphere eXtreme Scale Grid Servers= rhea00b02_c1
Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale', [1=Añadir, 2=Editar, 3=Suprimir, 4=Siguiente, 5=Salir]
(el valor predeterminado es: 4): 1
Servidores de la cuadrícula de WebSphere eXtreme Scale (el valor predeterminado es: ): rhea00b02_c2
URL de servicio JMX (el valor predeterminado es: service:jmx:rmi:///jndi/rmi://localhost:<puerto>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer
```

```
Valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale': WebSphere eXtreme Scale Grid Servers= rhea00b02_c2
Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale', [1=Añadir, 2=Editar, 3=Suprimir, 4=Siguiente, 5=Salir]
(el valor predeterminado es: 4): 5

¿Se conectará este agente a TEMS? [1=SÍ, 2=NO] (el valor predeterminado es: 1):
Nombre de host TEMS (el valor predeterminado es: rhea00b00):

Protocolo de red [ip, sna, ip.pipe o ip.spipe] (el valor predeterminado es: ip.pipe):

     Ahora seleccione el siguiente número de protocolo entre uno de los siguientes:
     - ip
     - sna
     - ip.spipe
     - 0 para ninguno
Network Protocol 2 (el valor predeterminado es: 0):
Número de puerto IP.PIPE (el valor predeterminado es: 1918):
Especifique el nombre de KDC_PARTITION (el valor predeterminado es: null):

¿Configurar la conexión para un TEMS secundario? [1=SÍ, 2=NO] (el valor predeterminado es: 2):
Especifique el Nombre de red principal opcional o 0 para "ninguno" (el valor predeterminado es: 0):
Se ha completado la configuración del agente...
```

El ejemplo anterior crea una instancia de agente denominada “inst1” y actualiza los valores de Java Home. Se configuran los servidores de contenedor eXtreme Scale, pero no se configura el servicio de catálogos.

Nota: El procedimiento anterior crea un archivo de texto con el siguiente formato en el directorio: `<instalación_ITM>/config/<host>_xt_<nombre instancia>.cfg`.

Ejemplo: `rhea00b02_xt_inst1.cfg`

Es mejor editar este archivo con el editor de texto sin formato que elija. A continuación, aparece un ejemplo del contenido de dicho archivo:

```
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 }  ]]
```

A continuación aparece un ejemplo que muestra una configuración en un despliegue de WebSphere Application Server:

```
rhea00b02 # ./itmcmd config -A xt
Se ha iniciado la configuración del agente...
Especifique el nombre de instancia (el valor predeterminado es: ):  inst1
¿Editar los valores de "Monitoring Agent for WebSphere eXtreme Scale"? [ 1=Sí, 2=No ] (el valor predeterminado es: 1): 1
¿Editar valores 'Java'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1): 1
Inicio de Java (el valor predeterminado es: C:\Archivos de programa\IBM\Java50): /opt/WAS61/java
Nivel de rastreo Java [ 1=Error, 2=Aviso, 3=Información, 4=Depuración mínima, 5=Depuración media, 6=Depuración máxima,
    7=Todos ] (el valor predeterminado es: 1):
Argumentos de JVM (el valor predeterminado es: ):
¿Editar valores de 'Conexión'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):
Tipo de conexión de servidor MBean [ 1=Servidor compatible con JSR-160, 2=WebSphere Application Server versión 6.0,
3=WebSphere Application Server
versión 6.1, 4=WebSphere Application Server versión 7.0 ]
(el valor predeterminado es: 1): 4
¿Editar valores de 'WebSphere Application Server versión 7.0'? [ 1=Sí, 2=No ]
(el valor predeterminado es: 1):ID de usuario WAS
(el valor predeterminado es: ):
Escribir la contraseña WAS (el valor predeterminado es: ):
Volver a escribir: contraseña WAS (el valor predeterminado es: ):
Nombre de host WAS (el valor predeterminado es: localhost): rhea00b02
Puerto WAS (el valor predeterminado es: 2809):
Protocolo de conector WAS [ 1=rmi, 2=soap ] (el valor predeterminado es: 1):
Nombre de perfil WAS (el valor predeterminado es: ): default
---------------------------------------
Información de classpath WAS
Vías de acceso básicas de WAS (el valor predeterminado es:
```

```
C:\Archivos de programa\IBM\WebSphere\AppServer;/opt/IBM/WebSphere/AppServer):
/opt/WAS61
Classpath WAS (el valor predeterminado es:
runtimes/com.ibm.ws.admin.client_6.1.0.jar;runtimes/com.ibm.ws.ejb.thinclient_7.0.0.jar):
Directorios JAR de WAS (el valor predeterminado es: lib;plugins):
¿Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):
¿No hay ningún valor disponible de 'Servidores de cuadrícula de WebSphere
eXtreme Scale'?
Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale', [1=Añadir, 2=Editar, 3=Suprimir, 4=Siguiente, 5=Salir]
(el valor predeterminado es: 4): 1
Servidores de cuadrícula de WebSphere eXtreme Scale (el valor predeterminado
es: ): rhea00b02
URL de servicio JMX (el valor predeterminado es:
service:jmx:rmi:///jndi/rmi://localhost:<puerto>/objectgrid/MBeanServer):

Valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale': Servidores de cuadrícula de WebSphere eXtreme Scale=rhea00b02
Editar valores de 'Servidores de cuadrícula de WebSphere eXtreme Scale', [1=Añadir, 2=Editar, 3=Suprimir, 4=Siguiente, 5=Salir]
(el valor predeterminado es: 4): 5
¿Editar valores de 'Servicio de catálogo de WebSphere eXtreme Scale'? [ 1=Sí, 2=No ] (el valor predeterminado es: 1):  2
¿Este agente se conectará a un TEMS? [1=SÍ, 2=NO] (el valor predeterminado es: 1):
Nombre de host de TEMS (el valor predeterminado es: rhea00b02):

Protocolo de red [ip, sna, ip.pipe o ip.spipe] (el valor predeterminado es: ip.pipe):

     Ahora seleccione el siguiente número de protocolo entre uno de los siguientes:
     - ip
     - sna
     - ip.spipe
     - 0 para ninguno
Network Protocol 2 (el valor predeterminado es: 0):
Número de puerto IP.PIPE (el valor predeterminado es: 1918):
Especifique el nombre de KDC_PARTITION (el valor predeterminado es: null):

¿Configurar la conexión para un TEMS secundario? [1=SÍ, 2=NO] (el valor predeterminado es: 2):
Especifique el Nombre de red principal opcional o 0 para "ninguno" (el valor predeterminado es: 0):
Ha finalizado la configuración del agente...
rhea00b02 #
```

Para los despliegues de WebSphere Application Server, no es necesario que cree varios subnodos. El agente eXtreme Scale se conecta al agente de nodo para recopilar toda la información de los servidores de aplicaciones de los que es responsable.

`SECTION=CAT` indica una línea de servicio de catálogo mientras que `SECTION=OGS` indica una línea de configuración de servidor de eXtreme Scale.

5. Configure el puerto JMX para todos los servidores de contenedor eXtreme Scale.

   Cuando se inician los servidores contenedor eXtreme Scale, sin especificar el argumento **-JMXServicePort**, se asigna un servidor MBean a un puerto dinámico. El agente necesita saber con anticipación con qué puerto se va a comunicar. El agente no funciona con puertos dinámicos.

   Cuando inicie los servidores, deberá especificar el argumento `-JMXServicePort <número_puerto>` cuando inicie el servidor eXtreme Scale utilizando el mandato `startOgServer.sh | .bat`. Ejecutar este mandato garantiza que el servidor del proceso está a la escucha de un puerto estático definido previamente.

   Para ver los ejemplos anteriores en una instalación UNIX, se deben iniciar dos servidores eXtreme Scale con los puertos establecidos:

   a. "-JMXServicePort" "15000" (para rhea00b02_c0)
   b. "-JMXServicePort" "15001" (para rhea00b02_c1)
   a. Inicie el agente eXtreme Scale.

      Dando por supuesto que se ha creado la instancia `inst1`, como en el ejemplo anterior, emita los siguientes mandatos.

      1) `cd <instalación_ITM>/bin`
      2) `itmcmd agent –o inst1 start xt`

   b. Detenga el agente eXtreme Scale.

Dando por supuesto que se ha creado la instancia “inst1”, como en el ejemplo anterior, emita los siguientes mandatos.

1) `cd <instalación_ITM>/bin`
2) `itmcmd agent –o inst1 stop xt`

6. Habilite las estadísticas para todos los servidores de contenedor eXtreme Scale.

   El agente utiliza MBeans de estadísticas de eXtreme Scale para grabar las estadísticas. Se debe habilitar la especificación de estadísticas de eXtreme Scale utilizando uno de los métodos siguientes.

   - Configure las propiedades de servidor para habilitar todas las estadísticas cuando se inician los servidores de contenedor: all=enabled.
   - Utilice el programa de utilidad de ejemplo de xsadmin para habilitar las estadísticas para todos los contenedores activos con los parámetros -setstatsspec all=enabled.

Resultados

Después de configurar e iniciar todos los servidores, los datos de MBeans se visualizan en la consola de IBM Tivoli Portal. Los espacios de trabajo definidos previamente muestran gráficos y métricas de datos en cada nivel de nodo.

Están definidos los siguientes espacios de trabajo: el nodo de **Servidores de cuadrícula de eXtreme Scale** para todos los nodos supervisados.

- Vista de transacciones de eXtreme Scale
- Vista de fragmento primario de eXtreme Scale
- Vista de memoria de eXtreme Scale
- Vista de ObjectMap de eXtreme Scale

También puede configurar su propio espacio de trabajo. Si desea más información, consulte la información sobre cómo personalizar los espacios de trabajo en el centro de información de IBM Tivoli Monitoring.

Supervisión de aplicaciones de eXtreme Scale con CA Wily Introscope

CA Wily Introscope es un producto de gestión de otro proveedor que puede utilizar para detectar y diagnosticar problemas de rendimiento en entornos de aplicaciones de empresa. eXtreme Scale incluye detalles sobre cómo configurar CA Wily Introscope para realizar introspecciones en las partes de selección del tiempo de ejecución de eXtreme Scale para ver y validar rápidamente las aplicaciones eXtreme Scale. CA Wily Introscope funciona de forma eficaz tanto para los despliegues autónomos, como para los despliegues de WebSphere Application Server.

Visión general

Para supervisar aplicaciones de eXtreme Scale con CA Wily Introscope, debe poner valores en los archivos ProbeBuilderDirective (PBD) que le proporcionan acceso a la información de supervisión para eXtreme Scale.

Atención: Los puntos de instrumentación para Introscope podrían cambiar con cada fixpack o release. Al instalar un nuevo fixpack o release, consulte la documentación para ver cualquier cambio en los puntos de instrumentación.

Puede configurar los archivos CA Wily Introscope ProbeBuilderDirective (PBD) para supervisar las aplicaciones de eXtreme Scale. CA Wily Introscope es un producto de gestión de aplicaciones con el que puede detectar, desencadenar y diagnosticar de forma proactiva los problemas en los entornos complejos, compuestos y de aplicación web.

Valores de archivos PBD para supervisar el servicio de catálogo

Puede utilizar uno o más de los siguientes valores en el archivo PBD para supervisar el servicio de catálogo.

```
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}"
```

Clases para supervisar el servicio de catálogo

HAControllerImpl
La clase HAControllerImpl maneja sucesos de comentarios y ciclo de vida del grupo principal. Puede supervisar esta clase para obtener una indicación de los cambios y estructura del grupo principal.

ServerAgent
La clase ServerAgent se ocupa de comunicar sucesos de grupos principales con el servicio de catálogo. Puede supervisar las diversas llamadas de pulsaciones para encontrar sucesos importantes.

PlacementServiceImpl
La clase PlacementServiceImpl coordina los contenedores. Puede utilizar los métodos en esta clase para supervisar sucesos de colocación y unión de servidores.

BalanceGridEventListener
La clase BalanceGridEventListener controla el liderazgo del catálogo. Puede supervisar esta clase para obtener una indicación de qué servicio de catálogo actúa actualmente como líder.

Valores de archivos PBD para supervisar los contenedores

Puede utilizar uno o más de los siguientes valores del archivo PBD para supervisar los contenedores.

```
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}"
```

Clases para supervisar los contenedores

ShardImpl

La clase ShardImpl tiene el método processMessage. El método processMessage es el método para las solicitudes de cliente. Con este método, puede obtener los recuentos de solicitudes y tiempos de respuesta del lado del servidor. Observando los recuentos a lo largo de todos los servidores y supervisando la utilización del almacenamiento dinámico, puede determinar si la cuadrícula está equilibrada.

CheckpointIterator

La clase CheckpointIterator tiene la llamada al método activateListener que coloca los primarios en modalidad de igual. Cuando los primarios se colocan en modalidad de igual, al réplica está actualizada con el primario una vez el método finaliza. Cuando una réplica se regenera a partir de un primario completo, esta operación puede tardar bastante tiempo. El sistema no se recupera totalmente hasta que la operación finaliza, de modo que puede utilizar esta clase para supervisar el progreso de la operación.

CommittedLogSequenceListenerProxy

La clase CommittedLogSequenceListenerProxy tiene dos métodos de interés. El método applyCommitted se ejecuta para cada transacción y sendApplyCommitted se ejecuta cuando la réplica extrae información. La proporción de la frecuencia en que estos dos métodos se ejecutan puede darle algún indicio de hasta qué punto la réplica puede mantener el ritmo del primario.

Valores de archivos PBD para supervisar los clientes

Puede utilizar uno o más de los siguientes valores en el archivo PBD para supervisar los clientes.

```
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}"
```

Clases para supervisar los clientes

ORBClientCoreMessageHandler
La clase ORBClientCoreMessageHandler es responsable de enviar solicitudes de aplicación a los contenedores. Puede supervisar el método sendMessage para el tiempo de respuesta del cliente y el número de solicitudes.

ClusterStore
La clase ClusterStore mantiene la información de direccionamiento en el lado del cliente.

BaseMap
La clase BaseMap tiene el método evictMapEntries que se invoca cuando el desalojador desea eliminar entradas de la correlación.

SelectionServiceImpl
La clase SelectionServiceImpl toma las decisiones de direccionamiento. Si el cliente toma decisiones relacionadas con la migración tras error, puede utilizar esta clase para ver las acciones que se han completado de las decisiones.

ObjectGridImpl
La clase ObjectGridImpl tiene el método getSession que puede supervisar para ver el número de solicitudes para este método.

Supervisión de eXtreme Scale con Hyperic HQ

Hyperic HQ es una solución de supervisión de otro proveedor que está disponible de forma gratuita como una solución de código abierto o como un producto de empresa. WebSphere eXtreme Scale incluye un plug-in que permite a los agentes de Hyperic HQ descubrir los servidores de contenedor eXtreme Scale y crear informes y agregar estadísticas utilizando los beans de gestión de eXtreme Scale. Puede utilizar Hyperic HQ para supervisar los despliegues de eXtreme Scale autónomos.

Antes de empezar

- Este conjunto de instrucciones es para Hyperic versión 4.0. Si tiene una versión más reciente de Hyperic, consulte la documentación de Hyperic si desea más información como, por ejemplo, los nombres de vía de acceso y cómo iniciar agentes y servidores.
- Descargue las instalaciones de agente y servidor de Hyperic. Debe estar en ejecución una instalación de servidor. Para detectar todos los servidores eXtreme Scale, un agente Hyperic debe estar en ejecución en cada máquina en la que se ejecuta un servidor eXtreme Scale. Consulte el sitio web de Hyperic si desea información de descarga y soporte de documentación.
- Debe tener acceso a los archivos `objectgrid-plugin.xml` `yhqplugin.jar`. Estos archivos se encuentran en el directorio *raíz_intal_wxs*`/hyperic/etc`.

Acerca de esta tarea

Mediante la integración de eXtreme Scale con el software de supervisión Hyperic HQ, puede supervisar y visualizar mediante gráficos las métricas sobre el

rendimiento del entorno. Configure esta integración utilizando una implementación de plug-in en cada agente.

Procedimiento

1. Inicie los servidores eXtreme Scale. El plug-in Hyperic consulta los procesos locales para conectarse al Máquinas virtuales Java que ejecuta eXtreme Scale. Para conectarse correctamente a Máquinas virtuales Java, todos los servidores deben estar iniciados con la opción **-jmxServicePort**. Si desea información sobre cómo iniciar los servidores con la opción **-jmxServicePort**, consulte "Script **startOgServer**" en la página 401.
2. Coloque el archivo `extremescale-plugin.xml` y el archivo `wxshyperic.jar` en los servidores apropiados de plug-in de servidor y agente de la configuración de Hyperic. Para integrar con Hyperic, las instalaciones de cliente y de servidor deben tener acceso a los archivos de plug-in y JAR (Java Archive). Aunque el servidor puede intercambiar dinámicamente configuraciones, debe completar la integración antes de iniciar cualquiera de los servicios.
   a. Coloque el archivo `extremescale-plugin.xml` en el directorio `plugin` del servidor, que está en la siguiente ubicación:

      inicio_hyperic`/server_home/hq-engine/server/default/deploy/hq.ear/hq-plugins`
   b. Coloque el archivo `extremescale-plugin.xml` en el directorio `plugin` del agente, que se encuentra en la siguiente ubicación:

      inicio_agente`/bundles/gent-4.0.2-939/pdk/plugins`
   c. Coloque el archivo `wshyperic.jar` en el directorio `lib` del agente, que está en la siguiente ubicación

      inicio_agente`/bundles/gent-4.0.2-939/pdk/lib`
3. Configure el agente. El archivo `agent.properties` sirve como punto de configuración para el tiempo de ejecución del agente. Esta propiedad está en el directorio *agent_home*`/conf`. Las siguientes claves son opcionales, pero son importantes para el plug-in de eXtreme Scale:
   - `autoinventory.defaultScan.interval.millis=<`*tiempo_en_milisegundos*`>`

     Establece el intervalo en milisegundos entre los descubrimientos de agente.
   - `log4j.logger.org.hyperic.hq.plugin.extremescale.XSServerDetector=DEBUG`

     : Habilita las sentencias de depuración verbosa desde el plug-in de eXtreme Scale.
   - `username=<`*nombre_usuario*`>`: establece el nombre de usuario de JMX (Java Management Extensions) si la seguridad está habilitada.
   - `password=<`*contraseña*`>`: establece la contraseña de JMX si la seguridad está habilitada.
   - `sslEnabled=<`*true*`|`*false*`>`: indica al plug-in si debe utilizar o no SSL (Secure Sockets Layer). El valor es `false` de forma predeterminada.
   - `trustPath=<`*vía_acceso*`>`: establece la vía de acceso de confianza para la conexión SSL.
   - `trustType=<`*tipo*`>`: establece el tipo de confianza para la conexión SSL.
   - `trustPass=<`*contraseña*`>`: establece la contraseña de confianza para la conexión SSL.
4. Inicie el descubrimiento del agente. Los agentes de Hyperic envían información de descubrimientos y métricas al servidor. Utilice el servidor para personalizar vistas de datos y objetos de inventario lógico de grupo para generar

información útil. Después de que el servidor está disponible, debe ejecutar el script de inicio o iniciar el servicio de Windows para el agente:

- Linux *inicio_agente*/bin/hq-agent.sh start
- Windows Inicie el agente con el servicio Windows.

Después de iniciar los agentes, los servidores se detectan y los grupos se configuran. Puede iniciar la sesión en la consola de servidor y elegir qué recursos añadir a la base de datos de inventario para el servidor. La consola del servidor está en el siguiente URL de forma predeterminada: `http://<nombre_host_servidor>:7080/`

5. Se deben habilitar las estadísticas para que Hyperic reúna datos estadísticos.

   Utilice la acción de control **SetStatsSpec** en la consola de Hyperic para eXtreme Scale. Navegue hasta el recurso, luego utilice la lista desplegable **Acción de control** en la página con separadores **Control** para especificar un valor de `SetStatsSpec` con `ALL=enabled` en el recuadro de texto **Argumentos de control**.

   El filtro establecido en la consola de Hyperic no detecta los servidores de catálogo. Consulte la información sobre la propiedad **statsSpec** en el apartado Archivo de propiedades de servidor , que permite las estadísticas en cuanto se inician los contenedores. En los apartados "Supervisión con beans gestionados (MBeans)" en la página 476 y Ejemplo: Programa de utilidad **xsadmin** se describen varias opciones para habilitar las estadísticas.
6. Supervise los servidores con la consola de Hyperic. Después de que se añadan los servidores al modelo de inventario, sus servicios ya no son necesarios.
   - **Vista del panel de instrumentos**: cuando visualizó los sucesos de detección de recursos, había iniciado la sesión en la vista del panel de instrumentos principal. La vista del panel de instrumentos es una vista genérica que hace las veces de centro de mensajes que se puede personalizar. Puede exportar gráficos u objetos de inventario a este panel de instrumentos principal.
   - **Vista de recursos**: puede consultar y ver todo el modelo de inventario desde esta página. Una vez que se han añadido los servicios, podrá ver cada uno de los servidores de eXtreme Scale etiquetados y listados correctamente bajo la sección de servidores. Puede pulsar los servidores individuales para ver las métricas básicas.
7. Vea todo el inventario del servidor en la página Ver recurso. En esta página, puede seleccionar varios servidores ObjectGrid y agruparlos juntos. Tras agrupar un conjunto de recursos, sus métricas comunes se pueden representar en un gráfico solapándose para mostrar las coincidencias y las diferencias entre los miembros del grupo. Para mostrar una coincidencia, seleccione las métricas en la pantalla del grupo de servidores. La métrica se visualiza en el área de gráficos. Para mostrar una coincidencia para todos los miembros del grupo, pulse el nombre de métrica subrayada. Puede exportar cualquiera de las gráficas, vistas de nodo y gráficas comparativas al panel de instrumentos principal con el menú **Herramientas**.

Supervisión de la información de eXtreme Scale en DB2

Cuando se utiliza JPALoader o JPAEntityLoader con DB2 como base de datos de programa de fondo, se puede pasar información específica de eXtreme Scale a DB2. Puede visualizar esta información mediante una herramienta de supervisor de rendimiento como por ejemplo DB2 Performance Expert para supervisar las aplicaciones eXtreme Scale que acceden a la base de datos.

Antes de empezar

Consulte “Recopilación de rastreo” en la página 536 para obtener más información sobre los distintos métodos de establecer el rastreo que puede utilizar.

Acerca de esta tarea

Cuando el cargador se configura para utilizar DB2 como base de datos de programa de fondo, la siguiente información de eXtreme Scale se puede pasar a DB2 para fines de supervisión:

- **Usuario:** especifica el nombre del usuario que se autentica en eXtreme Scale. Cuando la autenticación básica no se utiliza, se utilizan los principales de la autenticación.
- **Nombre de la estación de trabajo:** especifica el nombre de host, la IP del servidor de contenedor de eXtreme Scale.
- **Nombre de aplicación:** especifica el nombre del ObjectGrid, nombre de unidad de persistencia (si se ha establecido).
- **Información de contabilidad:** especifica el ID de hebra, el tipo de transacción, el ID de transacción y la serie de conexión.

Lea la información sobre DB2 Performance Expert para aprender a supervisar el acceso a la base de datos.

Procedimiento

- Para habilitar toda la información del cliente de eXtreme Scale, defina las series de rastreo siguientes:

  ```
  ObjectGridClientInfo*=event=enabled
  ```

- Para habilitar toda la información excepto la de usuario, utilice uno de los valores siguientes:
  –

  ```
  ObjectGridClientInfo*=event=enabled,ObjectGridClientInfoUser=event=disabled
  ```

  o bien
  –

  ```
  ObjectGridClientInfo=event=enabled
  ```

Resultados

Después de activar la función de rastreo, se visualizarán los datos en la herramienta de supervisor de rendimiento como, por ejemplo, DB2 Performance Expert.

Ejemplo

En el ejemplo siguiente, el usuario `bob` se autentica como un usuario eXtreme Scale. La aplicación accede a la cuadrícula de datos `mygrid` mediante la unidad de persistencia `DB2Hibernate`. El servidor de contenedor se denomina `XS_Server1`. A continuación se muestra la información resultante:

- **Usuario**=`bob`
- **Nombre de estación de trabajo**=`XS_Server1,192.168.1.101`
- **Nombre de aplicación**=`mygrid,DB2Hibernate`
- **Información de contabilidad**=`1, DEFAULT,FE7954BD-0126-4000-E000-2298094151DB,com.ibm.db2.jcc.t4.b@71787178`

En el ejemplo siguiente, el usuario bob se autentica utilizando una señal de WebSphere Application Server. La aplicación accede a la cuadrícula de datos mygrid mediante el nombre de unidad de persistencia DB2OpenJPA. El servidor de contenedor se denomina XS_Server2. A continuación se muestra la información resultante:

- **Usuario**

  ```
  =acme.principal.UserPrincipal[Bob],acme.principal.
  GroupPrincipal[admin]
  ```

- **Nombre de estación de trabajo**=XS_Server2,192.168.1.102
- **Nombre de aplicación**=mygrid,DB2OpenJPA
- **Información de contabilidad**=188,DEFAULT,FE72BC63-0126-4000-E000-851C092A4E33,com.ibm.ws.rsadapter.jdbc.WSJccSQLJConnection@2b432b43

Capítulo 9. Ajuste del rendimiento

Puede ajustar los valores de su entorno para aumentar el rendimiento global de su entorno de WebSphere eXtreme Scale.

Ajuste de los valores de red y de los sistemas operativos

El ajuste de red puede reducir el retardo de la pila del protocolo de control de transmisiones (TCP) modificando los valores de conexión y puede mejorar el rendimiento modificando los almacenamientos intermedios de TCP.

Sistemas operativos

Un sistema Windows necesita menos ajustes, mientras que un sistema Solaris necesita más ajustes. La siguiente información pertenece a cada sistema especificado y podría mejorar el rendimiento de WebSphere eXtreme Scale. Deberá realizar los ajustes de acuerdo con su red y su carga de aplicaciones.

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
```

Propiedades ORB

Las propiedades del intermediario de solicitud de objetos (ORB) modifican el comportamiento de transporte de la cuadrícula de datos. Estas propiedades se pueden establecer con un archivo `orb.properties`, como valores en la consola administrativa de WebSphere Application Server o como propiedades personalizadas en el ORB en la consola administrativa de WebSphere Application Server.

orb.properties

El archivo `orb.properties` se encuentra en el directorio `java/jre/lib`. Cuando modifica el archivo `orb.properties` en un directorio `java/jre/lib` de WebSphere Application Server, las propiedades de ORB se actualizan en el agente de nodo y cualquier otra máquina virtual Java (JVM) que utilice el entorno de ejecución Java (JRE). Si no desea este comportamiento, utilice propiedades personalizadas o los valores de ORB de la consola administrativa de WebSphere Application Server.

Valores predeterminados de WebSphere Application Server

WebSphere Application Server tiene algunas propiedades definidas en el ORB de forma predeterminada. Estos valores se encuentran en los servicios de contenedor del servidor de aplicaciones y el gestor de despliegue. Estos valores predeterminados sustituyen a los valores que crea en el archivo `orb.properties`. Para cada propiedad descrita, consulte la sección **Dónde se especifica** para determinar la ubicación para definir el valor sugerido.

Valores del descriptor de archivo

Para sistemas UNIX y Linux, existe un límite para el número de archivos abiertos que se permite por proceso. El sistema operativo especifica el número de archivos abiertos permitidos. Si este valor se ha establecido en un valor demasiado bajo, se produce un error de asignación de memoria en AIX y se registran demasiados archivos abiertos.

En la ventana del terminal del sistema UNIX, establezca este valor en un valor superior al valor del sistema predeterminado. Para grandes máquinas SMP con clones, establezca este valor en ilimitado.

Para configuraciones AIX, establezca este valor en `-1` (ilimitado) con el mandato: `ulimit -n -1`.

Para configuraciones Solaris, establezca este valor en `16384` con el mandato: `ulimit -n 16384`.

Para visualizar el valor actual, utilice el mandato: `ulimit –a`.

Valores básicos

Los siguientes valores son una buena base, pero no necesariamente los mejores valores para todos los entornos. Comprenda los valores a fin de poder tomar una decisión correcta sobre qué valores son adecuados para su entorno.

```
com.ibm.CORBA.RequestTimeout=30
com.ibm.CORBA.ConnectTimeout=10
com.ibm.CORBA.FragmentTimeout=30
com.ibm.CORBA.LocateRequestTimeout=10
```

```
com.ibm.CORBA.ThreadPool.MinimumSize=256
com.ibm.CORBA.ThreadPool.MaximumSize=256
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ConnectionMultiplicity=1
com.ibm.CORBA.MinOpenConnections=1024
com.ibm.CORBA.MaxOpenConnections=1024
com.ibm.CORBA.ServerSocketQueueDepth=1024
com.ibm.CORBA.FragmentSize=0
com.ibm.CORBA.iiop.NoLocalCopies=true
com.ibm.CORBA.NoLocalInterceptors=true
```

Descripciones de propiedad

Valores de tiempo de espera

Los siguientes valores están relacionados con la cantidad de tiempo que espera el ORB antes de abandonar una solicitud de operaciones. Utilice estos valores para evitar que se cree un exceso de hebras en una situación anómala.

Tiempo de espera de solicitud

Nombre de propiedad: com.ibm.CORBA.RequestTimeout

Valor válido: valor entero para número de segundos.

Valor sugerido: 30

Dónde se especifica: consola administrativa de WebSphere Application Server

Descripción: Indica cuántos segundos espera cualquier solicitud una respuesta antes de abandonar. Esta propiedad influye en la cantidad de tiempo que tarda el cliente en fallar si se produce una caída de la red. Si establece este propiedad en un valor demasiado bajo, las solicitudes podrían exceder el tiempo de espera sin querer. Considere atentamente el valor de esta propiedad para evitar tiempos de espera involuntarios.

Tiempo de espera de conexión

Nombre de propiedad: com.ibm.CORBA.ConnectTimeout

Valor válido: valor entero para número de segundos.

Valor sugerido: 10

Dónde se especifica: el archivo `orb.properties`

Descripción: indica cuántos segundos espera un intento de conexión de socket antes de abandonar. Esta propiedad, al igual que el tiempo de espera de solicitud, puede influir en el tiempo que tarda un cliente en fallar si se produce una caída de la red. En general, establezca esta propiedad en un valor menor al valor de tiempo de espera de solicitud, ya que el periodo de tiempo para establecer conexiones es relativamente constante.

Tiempo de espera de fragmento

Nombre de propiedad: com.ibm.CORBA.FragmentTimeout

Valor válido: valor entero para número de segundos.

Valor sugerido: 30

Dónde se especifica: el archivo `orb.properties`

Descripción: indica cuántos segundos espera una solicitud de fragmento antes de abandonar. Esta propiedad es similar a la propiedad de tiempo de espera de solicitud.

Valores de agrupación de hebras

Estas propiedades limitan el tamaño de la agrupación de hebras a un número específico de hebras. Las hebras son utilizadas por el ORB para derivar las solicitudes de servidor después de que se reciban en el socket. Si se establecen los valores de estas propiedades en valores demasiado bajos, aumentará la profundidad de la cola de sockets y posiblemente los tiempos de espera excedidos.

Multiplicidad de conexión

Nombre de propiedad: com.ibm.CORBA.ConnectionMultiplicity

Valor válido: valor entero correspondiente al número de conexiones entre el cliente y el servidor. El valor predeterminado es 1. Establecer un valor mayor establece la multiplexación entre varias conexiones.

Valor sugerido: 1

Dónde se especifica: el archivo `orb.properties`**Descripción:** permite al ORB utilizar varias conexiones a cualquier servidor. En teoría, si se establece este valor, se promueve el paralelismo sobre las conexiones. En la práctica, el rendimiento no saca partido de la definición de la multiplicidad de conexiones. No establezca este parámetro.

Conexiones abiertas

Nombres de propiedad: com.ibm.CORBA.MinOpenConnections, com.ibm.CORBA.MaxOpenConnections

Valor válido: valor entero correspondiente al número de conexiones.

Valor sugerido: 1024

Dónde se especifica: consola administrativa de WebSphere Application Server**Descripción:** especifica un número máximo y mínimo de conexiones abiertas. El ORB mantiene una memoria caché de conexiones que se han establecido con clientes. Estas conexiones se depuran cuando se proporciona este valor. La depuración de conexiones podría causar un bajo rendimiento en la cuadrícula de datos.

Con posibilidad de crecimiento

Nombre de propiedad: com.ibm.CORBA.ThreadPool.IsGrowable

Valor válido: booleano; se establece en `true` o `false`.

Valor sugerido: `false`

Dónde se especifica: el archivo `orb.properties`**Descripción:** si se establece en `true`, el tamaño de la agrupación de hebras que utiliza el ORB para las solicitudes de entrada puede crecer a un tamaño superior al que soporta la agrupación. Si el tamaño de la agrupación se excede, se crean nuevas hebras para manejar la solicitud, pero las hebras no se agrupan. Evite el crecimiento de la agrupación de hebras estableciendo el valor en `false`.

Profundidad de cola de socket de servidor

Nombre de propiedad: com.ibm.CORBA.ServerSocketQueueDepth

Valor válido: valor entero correspondiente al número de conexiones.

Valor sugerido: `1024`

Dónde se especifica: el archivo `orb.properties`**Descripción:** especifica la longitud de la cola de las conexiones de entrada de clientes. El ORB pone en cola las conexiones de entrada de clientes. Si la cola está llena, se rechazan las conexiones. El rechazo de las conexiones podría causar un bajo rendimiento en la cuadrícula de datos.

Tamaño de fragmento

Nombre de propiedad: com.ibm.CORBA.FragmentSize

Valor válido: número entero que especifica el número de bytes. El valor predeterminado es 1024.

Valor sugerido: `0`

Dónde se especifica: el archivo `orb.properties`**Descripción:** especifica el tamaño máximo de paquete que utiliza el ORB al enviar una solicitud. Si una solicitud es mayor que el límite de tamaño de fragmento, dicha solicitud se divide en fragmentos de solicitud que se envían de forma separada y se vuelven a ensamblar en el servidor. Fragmentar las solicitudes es útil en las redes no fiables donde es posible que los paquetes se tengan que volver a enviar. Sin embargo, si la red está disponible, la división de las solicitudes en fragmentos podría causar proceso innecesario.

Sin copias locales

Nombre de propiedad: com.ibm.CORBA.iiop.NoLocalCopies

Valor válido: booleano; se establece en `true` o `false`.

Valor sugerido: `true`

Dónde se especifica: la consola administrativa de WebSphere Application Server, el valor **Pasar por referencia**. **Descripción:** especifica si se pasa el ORB por referencia. El ORB utiliza la invocación pasar por valor de forma predeterminada. La invocación de Pasar por valor genera costes adicionales de basura y serialización a la vía de acceso cuando se inicia localmente una interfaz. Mediante la definición de este valor en true, el ORB utiliza un método "pasar por referencia" que es más eficaz que la invocación "pasar por valor".

Sin interceptores locales

Nombre de propiedad: com.ibm.CORBA.NoLocalInterceptors

Valor válido: booleano; se establece en `true` o `false`.

Valor sugerido: `true`

Dónde se especifica: el archivo `orb.properties` **Descripción:** especifica si el ORB inicia los interceptores de solicitud, incluso cuando se realizan solicitudes locales (internas de proceso). Los interceptores que utiliza WebSphere eXtreme Scale son para el manejo de seguridad y rutas, que no son necesarios si la solicitud se maneja en el proceso. Los interceptores que se mueven entre procesos sólo son necesarios para las operaciones de llamada de procedimiento remoto (RPC). Estableciendo los interceptores no locales, puede evitar el proceso adicional que presenta la utilización de interceptores locales.

Atención: Si utiliza seguridad de WebSphere eXtreme Scale, establezca el valor de la propiedad com.ibm.CORBA.NoLocalInterceptors en `false`. La infraestructura de seguridad utiliza interceptores para la autenticación.

Ajuste de las máquinas virtuales Java

Debe tener en cuenta varios aspectos específicos sobre el ajuste de la máquina virtual Java (JVM) para conseguir el mejor rendimiento posible de WebSphere eXtreme Scale. En la mayoría de los casos, se requieren pocos valores de JVM especiales, o ninguno. Si se almacenan muchos objetos en la cuadrícula de datos, ajuste el tamaño del almacenamiento dinámico en un valor adecuado para evitar quedarse sin memoria.

7.1.1+ Configurando eXtremeMemory, puede almacenar objetos en memoria nativa en lugar de hacerlo en el almacenamiento dinámico Java. La configuración de eXtremeMemory habilita eXtremeIO, un nuevo mecanismo de transporte. Si mueve los objetos fuera del almacenamiento dinámico de Java, evitará las pausas de recogida de basura, lo que hará que el rendimiento sea más constante y los tiempos de respuesta sean predecibles. Para obtener más información, consulte "Configuración de IBM eXtremeMemory e IBM eXtremeIO" en la página 277.

Plataformas probadas

La prueba de rendimiento se ha producido principalmente en sistemas AIX (de 32 vías), Linux (cuatro vías) y Windows (ocho vías). Con sistemas AIX de gama alta, puede probar escenarios con un gran número de hebras para identificar los puntos de contención y corregirlos.

Recogida de basura

WebSphere eXtreme Scale crea objetos temporales asociados a cada transacción como, por ejemplo, una petición y una respuesta y una secuencia de registro. Puesto que estos objetos afectan a la eficacia de la recogida de basura, es muy importante ajustar la recogida de basura.

Todas las JVM modernas utilizan algoritmos de recogida de basura paralelos, lo que significa que si se utilizan más núcleos se puede reducir las pausas en la recogida de basura. Un servidor físico con ocho núcleos tiene una recogida de basura más rápida que un servidor físico con cuatro núcleos.

Cuando la aplicación debe gestionar una gran cantidad de datos para cada partición, la recogida de basura podría ser un factor. Un escenario principalmente de lectura funciona incluso con almacenamientos intermedios grandes (20 GB o más) si se utiliza un recopilador generacional. Sin embargo, después de que se llene el almacenamiento dinámico de tenencia, se produce una pausa proporcional al tamaño del almacenamiento dinámico activo y al número de procesadores en el sistema. Esta pausa puede ser grande en sistemas más pequeños con almacenamientos dinámicos grandes.

Máquina virtual IBM para la recogida de basura de Java

Para la máquina virtual IBM para Java, utilice el recopilador **optavgpause** para escenarios con un índice alto de actualización (100% de entradas de modificación de transacciones). El recopilador **gencon** funciona mucho mejor que el recopilador **optavgpause** para escenarios donde los datos se actualizan con relativa poca frecuencia (10% del tiempo o menos). Experimente con los dos tipos de recolectores para ver cuál funciona mejor en su escenario. Realice la ejecución con la recogida de basura detallada activada para comprobar el porcentaje de tiempo que se emplea en la recogida de basura. Se han dado casos en los que se empleaba el 80% del tiempo en la recogida de basura hasta que se arregló el problema.

Utilice el parámetro **-Xgcpolicy** para cambiar el mecanismo de recogida de basura. El valor del parámetro **-Xgcpolicy** se puede establecer en: **-Xgcpolicy:gencon** o **-Xgcpolicy:optavgpause**, en función de la recogida de basura que desea utilizar.

- En una configuración de WebSphere Application Server, establezca el parámetro **-Xgcpolicy** en la consola administrativa. Pulse **Servidores** > **Servidores de aplicaciones** > **nombre_servidor** > **Definición de proceso** > **Máquina virtual Java**. Añada el parámetro en el campo **Argumentos de JVM genéricos**.
- En una configuración autónoma, pase el parámetro **-jvmArgs** en el script **startOgServer** para especificar la recogida de basura. El parámetro **-jvmArgs** debe ser el último parámetro que se pasa al script.

Otras opciones de recogida de basura

Atención: Si utiliza una JVM Sun, es posible que sean necesarios ajustes en la recogida de basura predeterminada y en la política de ajuste.

WebSphere eXtreme Scale soporta WebSphere Real Time Java. Con WebSphere Real Time Java, la respuesta del proceso de transacción de WebSphere eXtreme Scale es más coherente y predecible. Como resultado, el impacto de la recogida de basura y la planificación de hebras se minimiza considerablemente. El impacto se reduce hasta el nivel de que la desviación estándar del tiempo de respuesta es menor que el 10% del Java habitual.

Rendimiento de la JVM

WebSphere eXtreme Scale se puede ejecutar en distintas versiones de Java Platform, Standard Edition. WebSphere eXtreme Scale soporta Java SE Versión 5 y posterior. Para obtener un rendimiento y una productividad de desarrollador mejores, utilice Java SE 5 o posterior para aprovechar las anotaciones y la recogida de basura mejorada. WebSphere eXtreme Scale funciona en máquinas virutales Java de 32 o de 64 bits.

WebSphere eXtreme Scale se prueba con un subconjunto de las máquinas virtuales disponibles, sin embargo, la lista soportada no es exclusiva. Puede ejecutar WebSphere eXtreme Scale en cualquier JVM de proveedor en la Edición 5 o posterior. Sin embargo, si se produce un problema con una JVM de proveedor, debe ponerse en contacto con el proveedor de JVM para solicitar soporte. Si es posible, utilice la JVM del tiempo de ejecución de WebSphere en cualquier plataforma que dé soporte a WebSphere Application Server.

Para la mayoría de los escenarios en los que se utiliza WebSphere eXtreme Scale, Java SE Versión 6 de la JVM tiene un mejor rendimiento que la Edición 5. En general, utilice la versión más reciente disponible de Java Platform, Standard Edition para obtener el mejor rendimiento.

Tamaño de almacenamiento dinámico

La recomendación es almacenamientos dinámicos de entre 1 y 2 GB con una JVM por cada cuatro núcleos. El número óptimo del tamaño de almacenamiento dinámico depende de los factores siguientes:

- El número de objetos activos en el almacenamiento dinámico.
- La complejidad de los objetos activos del almacenamiento dinámico.
- El número de núcleos disponibles para la JVM.

Por ejemplo, una aplicación que almacena matrices de bytes de 10 K puede ejecutar un almacenamiento dinámico más grande que una aplicación que utiliza gráficos complejos de objetos POJO.

Número de hebras

El número de hebras depende de unos pocos factores. Existe un límite en el número de hebras que puede gestionar un solo fragmento. Un fragmento es una instancia de una partición, y puede ser un fragmento primario o de réplica. Con más fragmentos para cada JVM, tiene más hebras con cada fragmento adicional, lo que proporciona más vías de acceso simultáneas a los datos. Cada fragmento es tan simultáneo como es posible aunque hay un límite para la simultaneidad.

Requisitos de Object Request Broker (ORB)

IBM SDK incluye una implementación de IBM ORB que se ha probado con WebSphere Application Server y WebSphere eXtreme Scale. Para facilitar el proceso de soporte, utilice una JVM proporcionada por IBM. Otras implementaciones de JVM utilizan un ORB diferente. El ORB de IBM sólo se proporciona con máquinas virtuales IBM-provided Java. WebSphere eXtreme Scale requiere un ORB en funcionamiento para poder funcionar. Puede utilizar WebSphere eXtreme Scale con ORB de otros proveedores. Sin embargo, si tiene un problema con un proveedor de ORB, debe ponerse en contacto con el proveedor del ORB para obtener soporte. La implementación del IBM ORB es compatible con las máquinas virtuales Java de otros proveedores y se puede sustituir, si es necesario.

Ajuste de orb.properties

En el laboratorio, se ha utilizado el archivo siguiente en cuadrículas de datos de hasta 1500 JVM. El archivo `orb.properties` se encuentra en la carpeta `lib` del entorno de ejecución.

```
# Propiedades de IBM JDK para ORB
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton

# Interceptores de WS
org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer

# Propiedades de plugins y ORB de WS
com.ibm.CORBA.ForceTunnel=never
com.ibm.CORBA.RequestTimeout=10
com.ibm.CORBA.ConnectTimeout=10

# Necesario cuando muchas JVM se conectan al catálogo a la vez
com.ibm.CORBA.ServerSocketQueueDepth=2048

# Los clientes y el servidor de catálogo pueden tener sockets abiertos para todas las JVM
com.ibm.CORBA.MaxOpenConnections=1016

# Agrupación de hebras para el manejo de solicitudes de entrada, aquí 200 hebras
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ThreadPool.MaximumSize=200
com.ibm.CORBA.ThreadPool.MinimumSize=200
com.ibm.CORBA.ThreadPool.InactivityTimeout=180000

# No se dividen las peticiones/respuestas grandes en fragmentos menores
com.ibm.CORBA.FragmentSize=0
```

Ajuste del valor de intervalo de pulsación para la detección de migración tras error

Puede configurar la cantidad de tiempo entre las comprobaciones de sistema para los servidores que han fallado con el valor de intervalo de pulsaciones.

Acerca de esta tarea

La configuración de la migración tras error varía en función del tipo de entorno que utiliza. Si utiliza un entorno autónomo, puede configurar una migración tras error con la línea de mandatos. Si utiliza un entorno WebSphere Application Server Network Deployment, debe configurar la migración tras error en la consola de administración de WebSphere Application Server Network Deployment.

Procedimiento

- Configure la migración tras error para los entornos autónomos.

  Puede configurar los intervalos de pulsación en la línea de mandatos utilizando el parámetro **-heartbeat** en el archivo de script **startOgServer**. Establezca este parámetro en uno de los siguientes valores:

  Tabla 30. Intervalos de pulsaciones

Valor	Acción	Descripción
0	Típica (valor predeterminado)	Las migraciones tras error se detectan normalmente en 30 segundos.
-1	Agresiva	Las migraciones tras error se detectan normalmente en 5 segundos.
1	Relajada	Las migraciones tras error se detectan normalmente en 180 segundos.

  Un intervalo de pulsaciones agresivo puede ser útil cuando los procesos y la red son estables. Si la red o los procesos no se han configurado de forma óptima, es posible que las pulsaciones se pierdan, lo que comportará en una detección de anomalía falsa.

- Configure la migración tras error para los entornos WebSphere Application Server.

  Puede configurar WebSphere Application Server Network Deployment versión 6.0.2 y posterior para permitir a WebSphere eXtreme Scale que realice la migración tras error muy rápidamente. El tiempo de migración tras error predeterminado para las anomalías graves es aproximadamente de 200 segundos. Una anomalía grave es un bloqueo del servidor o sistema físico, una desconexión del cable de red o un error del sistema operativo. Las anomalías debidas a cuelgues del proceso o a anomalías leves normalmente realizan la migración tras error en menos de un segundo. La detección de anomalías correspondientes a anomalías leves sucede cuando el sistema operativo cierra automáticamente los sockets de red del proceso inactivo para el servidor que aloja el proceso.

  Configuración de pulsaciones de grupo principal

  WebSphere eXtreme Scale que se ejecuta en un proceso WebSphere Application Server hereda las características de migración tras error de los valores del grupo principal del servidor de aplicaciones. Las siguientes secciones describen cómo configurar los valores de pulsación del grupo principal para distintas versiones de WebSphere Application Server Network Deployment:

  – **Actualice los valores de grupo principal para WebSphere Application Server Network Deployment versión 6.x y 7.x:**

    Especifique el intervalo de pulsación en segundos en las versiones de WebSphere Application Server de la versión 6.0 a la versión 6.1.0.12 o en milisegundos a partir de la versión 6.1.0.13. También debe especificar el número de pulsaciones que faltan. Este valor indica cuántas pulsaciones pueden perderse antes de que se considere anómala una Máquina virtual Java

(JVM) de igual. El tiempo de detección de anomalías graves es aproximadamente el producto del intervalo de pulsaciones y el número de pulsaciones perdidas.

Estas propiedades se especifican utilizando las propiedades personalizadas en el grupo principal a través de la consola administrativa de WebSphere. Consulte Propiedades personalizadas del grupo principal para obtener detalles sobre la configuración. Estas propiedades deben especificarse para todos los grupos principales que la aplicación utiliza:

- El intervalo de pulsación se especifica utilizando la propiedad personalizada IBM_CS_FD_PERIOD_SEC para segundos o la propiedad personalizada IBM_CS_FD_PERIOD_MILLIS para milisegundos (requiere la Versión 6.1.0.13 o posterior).
- El número de pulsaciones perdidas se especifica utilizando la propiedad personalizada IBM_CS_FD_CONSECUTIVE_MISSED.

El valor predeterminado para la propiedad IBM_CS_FD_PERIOD_SEC es 20 y para la propiedad IBM_CS_FD_CONSECUTIVE_MISSED es 10. Si se especifica la propiedad IBM_CS_FD_PERIOD_MILLIS, altera temporalmente cualquier conjunto de propiedades personalizadas IBM_CS_FD_PERIOD_SEC. Los valores de estas propiedades son valores enteros positivos.

Utilice los siguientes valores para conseguir un tiempo de detección de anomalías de 1500 ms para los servidores WebSphere Application Server Network Deployment versión 6.x:

- Establezca `IBM_CS_FD_PERIOD_MILLIS = 750` (WebSphere Application Server Network Deployment V6.1.0.13 y posterior)
- Establezca `IBM_CS_FD_CONSECUTIVE_MISSED = 2`

– **Actualice los valores de grupo principal para WebSphere Application Server Network Deployment versión 7.0**

WebSphere Application Server Network Deployment versión 7.0 proporciona dos valores de grupo principal que se pueden ajustar para aumentar o reducir la detección de migración tras error:

- **Periodo de transmisión de pulsación.** El valor predeterminado es 30000 milisegundos.
- **Periodo de tiempo de espera de pulsación.** El valor predeterminado es 180000 milisegundos.

Si desea más detalles sobre cómo cambiar estos valores, consulte el centro de información de WebSphere Application Server Network Deployment: Valores de descubrimiento y detección de errores.

Utilice los valores siguientes para conseguir un tiempo de detección de anomalías de 1500 ms para los servidores WebSphere Application Server Network Deployment versión 7:

- Establezca el periodo de transmisión de pulsaciones en 750 milisegundos.
- Establezca el periodo de tiempo de espera de pulsaciones en 1500 milisegundos.

Qué hacer a continuación

Cuando estos valores se modifican para proporcionar tiempos de migración tras error cortos, se debe tener en cuenta algunas cuestiones relativas al ajuste del sistema. En primer lugar, Java no es un entorno de tiempo real. Es posible que las hebras se demoren si JVM está sufriendo tiempos de recogida de basura de larga duración. Las hebras también podrían demorarse si la máquina que aloja la JVM tiene mucha carga (debido a la propia JVM o a otros procesos que se ejecutan en la

máquina). Si las hebras se retrasan, es posible que las pulsaciones no se envíen a tiempo. En el peor de los casos, podrían demorarse el tiempo de migración tras error necesario. Si las hebras se demoran, se producen detecciones de anomalías falsas. El sistema se debe ajustar y se debe modificar su tamaño para asegurarse de que las detecciones de anomalías falsas no se producen en un entorno de producción. La mejor manera de garantizarlo es utilizando una carga adecuada durante la fase de prueba.

Nota: La versión actual de eXtreme Scale soporta WebSphere Real Time.

Ajuste de la recopilación de basura con WebSphere Real Time

El uso de WebSphere eXtreme Scale con WebSphere Real Time aumenta la coherencia y la previsibilidad con un coste de rendimiento en comparación con la política de recogida de basura predeterminada empleada en el Java™ SE Runtime Environment (JRE) de IBM estándar. La proporción de coste frente a beneficios puede variar. WebSphere eXtreme Scale crea muchos objetos temporales que se asocian con cada transacción. Estos objetos temporales se ocupan de peticiones, respuestas, secuencias de registro y sesiones. Sin WebSphere Real Time, el tiempo de respuesta de la transacción puede ascender hasta miles de milisegundos. Sin embargo, el uso de WebSphere Real Time con WebSphere eXtreme Scale puede aumentar la eficacia de la recogida de basura y reducir el tiempo de respuesta en un 10% del tiempo de respuesta de la configuración autónoma.

WebSphere Real Time en un entorno autónomo

Puede utilizar WebSphere Real Time con WebSphere eXtreme Scale. Mediante la habilitación de WebSphere Real Time, puede obtener una recogida de basura más predecible junto con un tiempo de respuesta estable y coherente y un rendimiento de transacciones en un entorno autónomo de eXtreme Scale.

Ventajas de WebSphere Real Time

WebSphere eXtreme Scale crea muchos objetos temporales que se asocian con cada transacción. Estos objetos temporales se ocupan de peticiones, respuestas, secuencias de registro y sesiones. Sin WebSphere Real Time, el tiempo de respuesta de la transacción puede ascender hasta miles de milisegundos. Sin embargo, el uso de WebSphere Real Time con WebSphere eXtreme Scale puede aumentar la eficacia de la recogida de basura y reducir el tiempo de respuesta en un 10% del tiempo de respuesta de la configuración autónoma.

Habilitación de WebSphere Real Time

Instale WebSphere Real Time y el WebSphere eXtreme Scale autónomo en los sistemas en los que tiene previsto ejecutar eXtreme Scale. Establezca la variable de entorno JAVA_HOME para indicar un Java SE Runtime Environment (JRE) estándar.

Establezca la variable de entorno JAVA_HOME para indicar al WebSphere Real Time instalado. A continuación, habilite WebSphere Real Time del modo siguiente.

1. Edite el archivo de instalación autónomo `objectgridRoot/bin/setupCmdLine.sh | .bat` eliminando el comentario de la siguiente línea.

   ```
   WXS_REAL_TIME_JAVA="-Xrealtime -Xgcpolicy:metronome
   -Xgc:targetUtilization=80"
   ```

2. Guarde el archivo.

Ahora, ha habilitado WebSphere Real Time. Si desea inhabilitar WebSphere Real Time, puede volver a añadir el comentario a la misma línea.

Procedimientos recomendados

WebSphere Real Time permite a las transacciones eXtreme Scale tener un tiempo de respuesta más predecible. Los resultados muestran que la desviación de un tiempo de respuesta de una transacción eXtreme Scale mejora significativamente con WebSphere Real Time, en comparación con el Java estándar con su recogida de basura predeterminada. La habilitación de WebSphere Real Time con eXtreme Scale es óptima si la estabilidad y el tiempo de respuesta de la aplicación son esenciales.

Los mejores procedimientos descritos en esta sección explican cómo hacer más eficaz a WebSphere eXtreme Scale a través del ajuste y de las prácticas de código, en función de la carga esperada.

- Establezca el nivel correcto de uso de procesador para la aplicación y la recogida de basura.

  WebSphere Real Time proporciona la capacidad para controlar el uso del procesador, de forma que el impacto de la recogida de basura en la aplicación está controlado y minimizado. Utilice el parámetro `-Xgc:targetUtilization=NN` para especificar el `NN` porcentaje del procesador que es utilizado por la aplicación cada 20 segundos. El valor predeterminado para WebSphere eXtreme Scale es 80%, pero puede modificar el script en el archivo `objectgridRoot/bin/setupCmdLine.sh` para definir un número distintos como, por ejemplo, `70`, que proporciona más capacidad de procesador a la recogida de basura. Despliegue los suficientes servidores para mantener la carga del procesador por debajo del 80% para las aplicaciones.
- Establezca un tamaño mayor de memoria de almacenamiento dinámico.

  WebSphere Real Time utiliza más memoria que el Java típico, así que planifique WebSphere eXtreme Scale con una memoria de almacenamiento dinámico grande y establezca el tamaño del almacenamiento dinámico cuando inicie los servidores y contenedores de catálogo con el parámetro `–jvmArgs –XmxNNNM` en el mandato **`ogStartServer`**. Por ejemplo, podría utilizar el parámetro `–jvmArgs –Xmx500M` para iniciar los servidores de catálogo y utilizar el tamaño de memoria apropiado para iniciar los contenedores. Puede establecer el tamaño de la memoria en un 60-70% del tamaño de datos esperado por JVM. Si no establece este valor, se podría generar un error OutOfMemoryError. De forma opcional, también puede utilizar el parámetro `–jvmArgs –Xgc:noSynchronousGCOnOOM` para impedir el comportamiento nondeterministic cuando la JVM agota la memoria.
- Ajuste las hebras para la recogida de basura.

  WebSphere eXtreme Scale crea muchos objetos temporales asociados a cada transacción y a hebras de llamada de procedimiento remoto (RPC). La recogida de basura tiene ventajas de rendimiento si el sistema tiene los suficientes ciclos de procesador. El número predeterminado de hebras es 1. Puede cambiar el número de hebras con el argumento `–Xgcthreads n`. El valor sugerido de este argumento es el número de núcleos que están disponibles con consideración del número de máquinas virtuales Java por sistema.
- Ajuste el rendimiento para las aplicaciones de corta ejecución con WebSphere eXtreme Scale.

  WebSphere Real Time se ajusta para las aplicaciones de larga ejecución. Normalmente, debe ejecutar las transacciones continuas de WebSphere eXtreme Scale durante dos horas para obtener datos de rendimiento fiables. Puede utilizar el parámetro `–Xquickstart` para mejorar el rendimiento de las

aplicaciones de corta ejecución. Este parámetro indica al compilador JIT (just-in-time) que utilice el nivel inferior de optimización.

- Minimice la cola de cliente de WebSphere eXtreme Scale y la transmisión del cliente de WebSphere eXtreme Scale.

  La principal ventaja de utilizar WebSphere eXtreme Scale con WebSphere Real Time es tener un tiempo de respuesta de transacción muy fiable, que normalmente tiene varios tiempos de mejoras de magnitud de orden en la desviación del tiempo de respuesta de transacción. Las peticiones de cliente en cola y la transmisión de solicitud de cliente a través de otro software impacta en el tiempo de respuesta que está más allá del control de WebSphere Real Time y WebSphere eXtreme Scale. Debe cambiar las hebras y los parámetros de sockets para mantener una carga fija sin problemas sin ningún retardo significativo y reducir la profundidad de la cola.

- Escriba aplicaciones WebSphere eXtreme Scale para utilizar las hebras de WebSphere Real Time.

  Sin modificar la aplicación, puede obtener un tiempo de respuesta de transacción de WebSphere eXtreme Scale muy fiable con varias mejoras de magnitud de orden en la desviación del tiempo de respuesta. Puede explotar de forma adicional la ventaja de hebras de las aplicaciones transaccionales de la hebra Java regular en RealtimeThread que proporciona un mejor control en la prioridad de la hebras y una planificación del control.

  Actualmente, la aplicación incluye el siguiente código.

  ```
  public class WXSCacheAppImpl extends Thread implements WXSCacheAppIF
  ```

  De forma opcional, puede sustituir este código por lo siguiente.

  ```
  public class WXSCacheAppImpl extends RealtimeThread implements
  WXSCacheAppIF
  ```

WebSphere Real Time en WebSphere Application Server

Puede utilizar WebSphere® Real Time con eXtreme Scale in un entorno de WebSphere Application Server Network Deployment versión 7.0. Mediante la habilitación de WebSphere Real Time, puede obtener una recogida de basura más predecible junto con un tiempo de respuesta y un rendimiento de transacciones estable y coherente.

Ventajas

El uso de WebSphere eXtreme Scale con WebSphere Real Time aumenta la coherencia y la previsibilidad con un coste de rendimiento en comparación con la política de recogida de basura predeterminada empleada en el Java™ SE Runtime Environment (JRE) de IBM estándar. La proporción de coste frente a beneficios puede variar en función de varios criterios. A continuación se enumeran algunos de los criterios principales.

- Prestaciones del servidor - Memoria disponible, velocidad y tamaño de la CPU y velocidad y uso de la red
- Cargas del servidor – Carga sostenida de la CPU, carga máxima de la CPU
- Configuración de Java – Tamaños de almacenamiento dinámico, uso de destino, hebras de recogida de basura
- Configuración de modalidad de copia de WebSphere eXtreme Scale – Matriz de bytes frente a almacenamiento POJO
- Cuestiones específicas de la aplicación – Uso de hebras, requisitos de respuesta y tolerancia, tamaño de los objetos, etc.

Además de esta política de recogida de basura cíclica disponible en WebSphere Real Time, hay políticas de recogida de basura opcionales disponibles en el IBM Java™ SE Runtime Environment (JRE) estándar. Estas políticas, optthruput (predeterminada), gencon, optavgpause y subpool, están expresamente diseñadas para solucionar requisitos y entornos de aplicación distintos. Para obtener más información sobre estas políticas,, consulte el apartado “Ajuste de las máquinas virtuales Java” en la página 496. Según los requisitos, los recursos y las restricciones de la aplicación y el entorno, el uso de una o varias de estas políticas de recogida de basura como prototipo puede garantizar que cumpla sus requisitos y determine una política óptima.

Prestaciones con WebSphere Application Server Network Deployment

1. A continuación se indican algunas versiones soportadas.
   - WebSphere Application Server Network Deployment versión 7.0.0.5 y superior.
   - WebSphere Real Time V2 SR2 para Linux y superior. Consulte IBM WebSphere Real Time V2 para Linux para obtener más información.
   - WebSphere eXtreme Scale versión 7.0.0.0 y superior.
   - Sistemas operativos Linux de 32 y 64 bits.
2. Los servidores WebSphere eXtreme Scale no pueden compartir ubicación un Dmgr de WebSphere Application Server.
3. Real Time no soporta DMgr.
4. Real Time no soporta los agentes de nodo WebSphere.

Habilitación de WebSphere Real Time

Instale WebSphere Real Time y WebSphere eXtreme Scale en los sistemas en los que tenga previsto ejecutar eXtreme Scale. Actualice WebSphere Real Time Java a SR2.

Puede especificar los valores de la JVM para cada servidor mediante la consola de WebSphere Application Server versión 7.0 tal como se indica a continuación.

Seleccione **Servidores** > **Tipos de servidor** > **Servidores de aplicaciones WebSphere** > **<servidor instalado necesario>**

En la página resultante, seleccione "Definición de proceso".

En la página siguiente, pulse Máquina virtual Java en la parte superior de la columna de la derecha. (Aquí, puede definir tamaños de almacenamiento dinámico, la recogida de basura y otros distintivos para cada servidor).

Defina los distintivos siguientes en el campo "Argumentos de JVM genéricos":

```
-Xrealtime -Xgcpolicy:metronome  -Xnocompressedrefs -Xgc:targetUtilization=80
```

Aplique y guarde los cambios.

Para utilizar Real Time en WebSphere Application Server 7.0 con servidores eXtreme Scale incluyendo los distintivos de JVM anteriores, debe crear una variable de entorno JAVA_HOME.

Defina JAVA_HOME tal como se indica a continuación.

1. Expanda "Entorno".

2. Seleccione "Variables de WebSphere".
3. Asegúrese de que "Todos los ámbitos" esté marcado debajo de “Mostrar ámbito".
4. Seleccione el servidor necesario en la lista desplegable. (No seleccione DMgr ni servidores de agente de nodo).
5. Si la variable de entorno JAVA_HOME no está en la lista, seleccione "Nueva" y especifique JAVA_HOME como nombre de la variable. En el campo "Valor", escriba el nombre de vía de acceso completo para Real Time.
6. Aplique y guarde los cambios.

Procedimientos recomendados

Para conocer un conjunto de procedimientos recomendados, consulte la sección sobre los procedimientos recomendados en “Ajuste de la recopilación de basura con WebSphere Real Time” en la página 501. Hay algunas modificaciones importantes que se deben tener en cuenta en esta lista de procedimientos recomendados para un entorno de WebSphere eXtreme Scale autónomo al realizar el despliegue en un entorno de WebSphere Application Server Network Deployment.

Debe colocar cualquier parámetro adicional de la línea de mandatos de la JVM en la misma ubicación que los parámetros de la política de recogida de basura especificados en la sección anterior.

Un objetivo inicial aceptable para cargas de procesador sostenidas es del 50% con picos de corta duración que lleguen hasta el 75%. Además de esto, debe añadir capacidad adicional para poder ver una degradación mensurable de la previsibilidad y la coherencia. Puede aumentar un poco el rendimiento si está dispuesto a tolerar tiempos de respuesta más largos. Superar un umbral del 80% suele conllevar una degradación considerable de la coherencia y la previsibilidad.

Ajuste del proveedor de la memoria caché dinámica

El proveedor de la memoria caché dinámica WebSphere eXtreme Scale soporta los siguientes parámetros de configuración para el ajuste de rendimiento.

Acerca de esta tarea

- **com.ibm.websphere.xs.dynacache.ignore_value_in_change_event**: Cuando registra un escucha de sucesos de cambio con el proveedor de memoria caché dinámica y genera una instancia de ChangeEvent, hay una sobrecarga asociada a la deserialización de la entrada de memoria caché para que el valor se pueda poner en ChangeEvent. Si se establece este parámetro opcional de la instancia de memoria caché en `true` se omite la deserialización de la entrada de memoria caché al generar ChangeEvents. El valor devuelto es nulo, en el caso de una operación de eliminación, o una matriz de bytes que contiene el formato serializado del objeto. Las instancias InvalidationEvent llevan una penalización de rendimiento similar, que se puede evitar estableciendo com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent en `true`.
- **com.ibm.websphere.xs.dynacache.enable_compression**: De forma predeterminada, el proveedor de memoria caché dinámica de eXtreme Scale comprime las entradas de caché en la memoria para aumentar la densidad de memoria caché, lo que puede ahorrar una cantidad significativa de memoria para aplicaciones como almacenamiento en memoria caché de servlet. Si sabe

que la mayor parte de los datos de memoria caché no se pueden comprimir, considere la posibilidad de establecer este valor en `false`.

Capítulo 10. Seguridad

WebSphere eXtreme Scale puede proteger el acceso a los datos, incluida la posibilidad de integración con proveedores de datos externos. Entre los aspectos de seguridad se incluyen la autenticación, la autorización y la seguridad en el transporte, en la cuadrícula de datos, local y en JMX (MBean).

Autenticación de cliente de aplicaciones

La autenticación del cliente de aplicaciones consiste en la habilitación de la seguridad de cliente-servidor y la autenticación de credenciales, y en la configuración de un autenticador y un generador de credenciales de sistema.

Habilitación de la seguridad cliente/servidor

Debe habilitar la seguridad tanto en el cliente, como en el servidor, para autenticarse correctamente con ObjectGrid.

Habilitar seguridad de cliente

WebSphere eXtreme Scale proporciona un archivo de ejemplo de propiedades de cliente, el archivo `sampleClient.properties`, en el directorio *raíz_was*/`optionalLibraries/ObjectGrid/properties` para una instalación de WebSphere Application Server, o el directorio `/ObjectGrid/properties` en una instalación de servidores mixtos. Puede modificar este archivo de plantilla con los valores correspondientes. Establezca la propiedad securityEnabled en el archivo `objectgridClient.properties` en `true`. La propiedad securityEnabled indica si la seguridad está habilitada. Cuando un cliente se conecta a un servidor, el valor en el cliente y en el servidor se deben establecer ambos en `true` o ambos en `false`. Por ejemplo, si el servidor conectado está habilitado, el valor de propiedad se debe establecer en `true` en el cliente para que el cliente se conecte al servidor.

La interfaz com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration representa el archivo `security.ogclient.props`. Puede usar la API pública com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory para crear una instancia de esta interfaz con valores predeterminados, o puede crear una instancia al pasar el archivo de propiedades de seguridad de cliente ObjectGrid. El archivo `security.ogclient.props` contiene otras propiedades. Consulte la documentación de la API ClientSecurityConfiguration y la documentación de la API ClientSecurityConfigurationFactory si desea más detalles.

Habilitar la seguridad del servidor

Para habilitar la seguridad en el lado del servidor, puede establecer la propiedad **`securityEnabled`** del archivo `security.xml` en `true`. Utilice un archivo XML de descriptor de seguridad para especificar la configuración de seguridad de la cuadrícula de datos para aislar la configuración de seguridad de nivel de cuadrícula de la configuración sin seguridad.

Habilitación de la autenticación de credenciales

Después de que el cliente de eXtreme Scale recupere el objeto Credential utilizando el objeto CredentialGenerator, el objeto Credential se envía junto con la petición de cliente al servidor eXtreme Scale. El servidor autentica el objeto Credential antes de procesar la solicitud. Si el objeto Credential se ha autenticado correctamente, se devuelve un objeto Subject para representar este objeto Credential. Este objeto Subject se utiliza para autorizar la petición.

Establezca la propiedad **credentialAuthentication** en los archivos de propiedades de cliente y de servidor para habilitar la autenticación de credenciales. Si desea más información, consulte Archivo de propiedades de cliente y Archivo de propiedades de servidor .

La siguiente tabla proporciona qué mecanismos de autenticación utilizar bajo distintos valores.

Tabla 31. Autenticación de credenciales bajo los valores de cliente y servidor

Autenticación de credenciales de cliente	Autenticación de credenciales de servidor	Resultado
No	Nunca	Inhabilitado
No	Soportado	Inhabilitado
No	Necesario	Caso de error
Soportado	Nunca	Inhabilitado
Soportado	Soportado	Habilitado
Soportado	Necesario	Habilitado
Necesario	Nunca	Caso de error
Necesario	Soportado	Habilitado
Necesario	Necesario	Habilitado

Configuración de una autenticador

El servidor eXtreme Scale utiliza el plug-in Authenticator para autenticar el objeto Credential. Una implementación de la interfaz Authenticator obtiene el objeto Credential y, después, lo autentica en un registro de usuarios, por ejemplo, un servidor LDAP (Lightweight Directory Access Protocol), etc. eXtreme Scale no proporciona una configuración de registro. Se debe implementar una conexión a un registro de usuarios y autenticarla en este plug-in.

Por ejemplo, una implementación de Authenticator extrae el ID de usuario y la contraseña de la credencial, los utiliza para conectarse y validar un servidor LDAP y crea un objeto Subject como resultado de la autenticación. La implementación puede utilizar los módulos de inicio de sesión JAAS (Java Authentication and Authorization Service). Como resultado de la autenticación, se devuelve un objeto Subject.

Puede configurar el autenticador en el archivo XML de descriptor de seguridad, tal como se indica en el siguiente ejemplo:

```
<?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>
```

Utilice la opción **-clusterSecurityFile** al iniciar un servidor seguro para establecer el archivo XML de seguridad. Consulte la guía de aprendizaje de seguridad de Java SE security en la *Visión general del producto* para obtener más información.

Configuración de un generador de credenciales del sistema

El generador de credenciales del sistema se utiliza para representar una fábrica de la credencial del sistema.Una credencial del sistema es similar a una credencial del administrador. Puede configurar el elemento SystemCredentialGenerator en el archivo XML de seguridad de catálogo, como se muestra en el ejemplo anterior:

```
<systemCredentialGenerator className ="com.ibm.websphere.objectgrid.security.plugins.builtins.
UserPasswordCredentialGenerator">
          <property name="properties" type="java.lang.String" value="manager manager1"
      description="username password" />
 </systemCredentialGenerator>
```

Por motivos de demostración, el nombre de usuario y la contraseña se almacenan en texto visible. No almacene el nombre de usuario y la contraseña en texto visible en un entorno de producción.

WebSphere eXtreme Scale proporciona un generador de credenciales del sistema predeterminado, que utiliza las credenciales del servidor. Si no especifica explícitamente el generador de credenciales del sistema, se utiliza este generador de credenciales del sistema predeterminado.

Autorización de cliente de aplicaciones

La autorización del cliente de aplicaciones consta de clases de permisos de ObjectGrid, mecanismos de autorización, un periodo de comprobación de permisos y un acceso sólo por parte de la autorización del creador.

Para eXtreme Scale, la autorización se basa en el objeto Subject y los permisos. El producto soporta dos tipos de mecanismos de autorización: Java Authentication and Authorization Service (JAAS) y la autorización personalizado.

Clases de permiso de ObjectGrid

La autorización se basa en permisos. Existen cuatro tipos diferentes de clases de permiso del modo siguiente.

- La clase MapPermission representa permisos para acceder a los datos de las correlaciones ObjectGrid.
- La clase ObjectGridPermission representa permisos para acceder a ObjectGrid.
- La clase ServerMapPermission representa permisos para acceder a las correlaciones ObjectGrid en el lado del servidor desde un cliente.
- La clase AgentPermission representa permisos para iniciar un agente en el lado del servidor.

Para obtener más información sobre las API y los permisos asociados, consulte el tema sobre la programación de la autorización de cliente en the *Guía de programación*.

Período de comprobación de permisos

eXtreme Scale soporta el almacenamiento en memoria caché de los resultados de la comprobación de permisos de correlación con finalidades de rendimiento. Sin este mecanismo, cuando se llama a un método que está en la lista de métodos para la clase de permiso en particular, el tiempo de ejecución llama al mecanismo de autorización configurado para autorizar el acceso. Con este período de comprobación de permisos establecido, el mecanismo de autorización se llama periódicamente en función del período de comprobación de permisos.Para ver una lista de los métodos para cada clase de permiso, consulte el tema sobre la programación de la autorización de cliente en la *Guía de programación*.

La información de autorización de permisos se basa en el objeto Subject. Cuando un cliente intenta acceder a los métodos, el tiempo de ejecución de eXtreme Scale consulta la memoria caché en función del objeto Subject. Si el objeto no se encuentra en la memoria caché, el tiempo de ejecución comprueba los permisos concedidos para este objeto Subject, y luego almacena los permisos en una memoria caché.

El período de comprobación de permisos debe definirse antes de inicializar ObjectGrid. El período de comprobación de permisos puede configurarse de dos modos:

Puede utilizar el archivo XML de ObjectGrid para definir un ObjectGrid y establecer el periodo de comprobación de permisos. En el siguiente ejemplo, el periodo de comprobación de permisos se establece en 45 segundos:

```
<objectGrids>
 <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
 authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS"
 permissionCheckPeriod="45">
  <bean id="bean id="TransactionCallback"
className="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
...
</objectGrids>
```

Si desea crear un ObjectGrid con API, llame al siguiente método para establecer el periodo de comprobación de permisos. Este método sólo puede llamarse antes de inicializar la instanciel ObjectGrid. Este método se aplica sólo al modelo de programación local de eXtreme Scale cuando cree una instancia directamente de ObjectGrid.

```
/**
 * Este método toma un único parámetro que indica con qué frecuencia
 * desea comprobar el permiso utilizado para permitir un acceso de cliente. Si el
 * parámetro es 0 cada llamada única get/put/update/remove/evict
 * solicita al mecanismo de autorización, autorización JAAS o personalizada,
 * comprobar si el objeto Subject actual tiene permiso. Esto podría ser
 * muy costoso desde el punto de vista del rendimiento en función de
 * la implementación de autorización, pero si necesita comprobar el
 * mecanismo de autorización, establezca el parámetro en 0.
 * De forma alternativa, si el parámetro es > 0, indica el número
 * de segundos que tarda en almacenar en la memoria caché un conjunto de
permisos antes de volver al
 * mecanismo de autorización para que los actualice. Este valor proporciona un
 * mejor rendimiento, pero si los permisos del programa de fondo
 * se cambian durante este tiempo, ObjectGrid puede
 * permitir o denegar el acceso aunque el proveedor de seguridad
 * del programa de fondo se haya modificado.
 *
 * @param period periodo de comprobación de servicio en segundos.
 */
void setPermissionCheckPeriod(int period);
```

Autorización de sólo acceso de creador

La autorización de sólo acceso de creador garantiza que sólo el usuario (representado por los objetos Principal asociados a él) que inserta la entrada en la correlación ObjectGrid pueda acceder (leer, actualizar, invalidar y eliminar) a la entrada.

El modelo de autorización de la correlación ObjectGrid existente se basa en el tipo de acceso, no en las entradas de datos. En otras palabras, un usuario tiene un tipo determinado de acceso (leer, grabar, insertar, suprimir o invalidar) para todos los datos de la correlación o para ninguno. No obstante, eXtreme Scale no autoriza a los usuarios la entrada individual de los datos. Esta característica ofrece una nueva manera de autorizar a los usuarios las entradas de datos.

En un escenario donde diferentes usuarios pueden acceder a distintos conjuntos de datos, este modelo puede ser de utilidad. Cuando el usuario carga los datos del almacén persistente en las correlaciones ObjectGrid, el acceso puede autorizarse desde el almacén persistente. En este caso, no es necesario realizar otra autorización en la capa de correlación ObjectGrid. Sólo debe asegurarse de que la persona que carga los datos en la correlación pueda acceder a ella mediante la habilitación de la característica de sólo acceso de creador.

Valores del atributo modalidad de sólo creador:

`disabled`
La característica de sólo acceso de creador está inhabilitada.

`complement`
La característica de sólo acceso de creador está habilitada para complementar la autorización de correlaciones. En otras palabras, la autorización de correlaciones y, también, la característica de sólo acceso de creador entran en vigor. Por lo tanto, puede limitar las operaciones a los datos. Por ejemplo, el creador no puede invalidar los datos.

`supersede`
La característica de sólo acceso de creador está habilitada para reemplazar la autorización de correlaciones. En otras palabras, la característica de sólo acceso de creador reemplaza la autorización de correlaciones; no se produce ninguna autorización de correlaciones.

Puede configurar esta característica de dos modos:

Mediante un archivo XML:

Puede utilizar el archivo XML de ObjectGrid para definir un ObjectGrid y establecer la modalidad de sólo acceso de creador en `disabled`, `complement` o `supersede`, tal como se indica en el siguiente ejemplo:

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
         accessByCreatorOnlyMode="supersede"
        <bean id="TransactionCallback"
              classname="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
    ...
</objectGrids>
```

A través de programa:

Si desea crear un ObjectGrid mediante programa, puede llamar al siguiente método para establecer la modalidad de sólo acceso de creador. La llamada a este

método sólo se aplica al modelo de programación de eXtreme Scale local cuando se crea directamente una instancia de ObjectGrid:

```
/**
 * Establezca la modalidad de sólo acceso de creador.
 * Si habilita esta modalidad se asegura de que sólo el usuario (representado
 * por los principales asociados a éste), que inserta el registro en la correlación,
 * pueda acceder (leer, actualizar, invalidar y eliminar) al registro.
 * La modalidad de sólo acceso de creador puede inhabilitarse, o puede complementar
 * el modelo de autorización ObjectGrid, o puede reemplazar el modelo de autorización
 * ObjectGrid. El valor predeterminado es la modalidad inhabilitada:
 * {@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 acceso mediante la modalidad de creador.
 *
 * @since WAS XD 6.1 FIX3
*/
void setAccessByCreatorOnlyMode(int accessByCreatorOnlyMode);
```

Con el propósito de ilustrar con más detalle, considere un escenario en el que una cuenta de correlaciones de ObjectGrid está en una cuadrícula de banca, y Manager1 y Employee1 son los dos usuarios. La política de autorización de eXtreme Scale otorga todos los permisos de acceso a Manager1, pero sólo otorga un permiso de acceso de lectura a Employee1. La política JAAS para la autorización de correlación ObjectGrid se muestra en el siguiente ejemplo:

```
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"
};
```

Considere cómo la modalidad de sólo acceso de creador afecta a la autorización:

- **disabled** Si la característica de sólo acceso de creador está inhabilitada, la autorización de correlaciones no cambia. El usuario "Manager1" puede acceder a todos los datos de la correlación de cuenta "account". El usuario "Employee1" puede leer e insertar todos los datos de la correlación pero no puede actualizarlos, invalidarlos ni eliminar ningún dato de la correlación.
- **complement** Si la característica de sólo acceso de creador está habilitada con la opción complementaria "complement", entrarán en vigor la autorización de correlaciones y la autorización de sólo acceso de creador. El usuario "Manager1" puede acceder a los datos de la correlación de cuenta "account", pero sólo si el usuario los ha cargado en la correlación. El usuario "Employee1" puede leer los datos de la correlación de cuenta "account", pero sólo si ese usuario los ha cargado en la correlación. No obstante, este usuario no puede actualizar, invalidar ni eliminar ningún dato en la correlación.
- **supersede** Si la característica de sólo acceso de creador está habilitada con la opción de reemplazar "supersede", no se aplicará la autorización de correlaciones. La autorización de sólo acceso de creador será la única política de autorización. El usuario "Manager1" tiene el mismo privilegio que en la modalidad "complement": este usuario puede acceder a los datos de la correlación de cuenta "account" sólo si ese mismo usuario ha cargado los datos en la correlación. No obstante, el usuario "Employee1" ahora tiene acceso completo a los datos de la correlación "account" si este usuario los ha cargado en la correlación. En otras palabras, la política de autorización definida en la política JAAS (Java Authentication and Authorization Service) no se aplicará.

Autenticación de la cuadrícula de datos

Puede utilizar el plug-in de gestor de señales seguro para habilitar la autenticación de servidor a servidor, que requiere la implementación de la interfaz SecureTokenManager.

El método generateToken(Object) toma un objeto y, a continuación, genera una señal que los otros no pueden entender. El método verifyTokens(byte[]) realiza el proceso inverso: convierte la señal en el objeto original.

Una implementación sencilla de SecureTokenManager utiliza un algoritmo de codificación sencillo como, por ejemplo, un algoritmo XOR, para codificar el objeto en un formato serializado y, a continuación, utilizar el algoritmo de decodificación correspondiente para descifrar la señal. Esta implementación no es segura y es fácil quebrantarla.

Implementación predeterminada de WebSphere eXtreme Scale

WebSphere eXtreme Scale proporciona una implementación disponible de forma inmediata para esta interfaz. Esta implementación predeterminada utiliza un par de claves para firmar y verificar la firma y utiliza una clave secreta para cifrar el contenido. Cada servidor tiene un almacén de claves de tipo JCKES donde se almacena el par de claves, una clave privada y una clave pública, y un clave secreta. El almacén de claves tiene que ser de tipo JCKES para poder almacenar las claves secretas. Estas claves se utilizan para cifrar y firmar o verificar la serie secreta en el envío. Además, la señal se asocia con un tiempo de caducidad. En el extremo receptor, los datos se verifican, se descifran y se comparan con la serie secreta del receptor. Los protocolos de comunicación SSL (Secure Sockets Layer) no son obligatorios para la autenticación entre un par de servidores porque las claves privadas y públicas sirven para ese mismo propósito. No obstante, si la comunicación del servidor no está cifrada, los datos podrían robarse con sólo observar la comunicación. Como la señal caduca pronto, la amenaza de ataque de reproducción se minimiza. Esta posibilidad disminuye en gran medida si todos los servidores se despliegan detrás de un cortafuegos.

La desventaja de este enfoque es que los administradores de WebSphere eXtreme Scale deben generar claves y transportarlas a todos los servidores, que pueden provocar una violación de seguridad durante el transporte.

Seguridad de la cuadrícula de datos

La seguridad de la cuadrícula de datos garantiza que un servidor que se una tenga las credenciales adecuadas, de manera que un servidor malintenciado no se pueda unir a la cuadrícula de datos. La seguridad de la cuadrícula de datos utiliza un mecanismo de serie secreta compartida.

Todos los servidores WebSphere eXtreme Scale, incluidos los servidores de catálogo, acuerdan una serie secreta compartida. Cuando un servidor se une a la cuadrícula de datos, se solicita que proporcione la serie secreta. Si la serie secreta del servidor que se une coincide con la serie del servidor presidente o el servidor de catálogo, se acepta el servidor que se une. Si la serie no coincide, se rechaza la solicitud de unión.

El envío de una serie secreta en texto normal no es seguro. La infraestructura de seguridad de WebSphere eXtreme Scale proporciona un plug-in de gestor de señales seguras para permitir al servidor proteger este secreto antes de enviarlo.

Debe decidir cómo implementar la operación segura. WebSphere eXtreme Scale proporciona una implementación directa, en la que la operación segura se implementa para cifrar y firmar el secreto.

La serie secreta se establece en el archivo `server.properties`. Consulte Archivo de propiedades de servidor si desea más información sobre la propiedad authenticationSecret.

Plug-in SecureTokenManager

Un plug-in de gestor de señales seguras se representa mediante la interfaz com.ibm.websphere.objectgrid.security.plugins.SecureTokenManager.

Si desea más información sobre el plug-in SecureTokenManager, consulte la documentación de la API SecureTokenManager.

El método generateToken(Object) toma un objeto y, a continuación, genera una señal que no los otros no pueden entender. El método verifyTokens(byte[]) realiza el proceso inverso: el método convierte la señal en el objeto original.

Una implementación sencilla de SecureTokenManager utiliza un algoritmo de codificación sencillo, como un algoritmo exclusivo o (XOR), para codificar el objeto en un formato serializado y, a continuación, utilizar el algoritmo de decodificación correspondiente para descifrar la señal. Esta implementación no es segura.

WebSphere eXtreme Scale proporciona una implementación disponible de forma inmediata para esta interfaz.

La implementación predeterminada utiliza un par de claves para firmar y verificar la firma, y utiliza una clave secreta para cifrar el contenido. Cada servidor tiene un almacén de claves de tipo JCKES donde se almacena el par de claves, una clave privada y una clave pública, y un clave secreta. El almacén de claves tiene que ser de tipo JCKES para poder almacenar las claves secretas.

Estas claves se utilizan para cifrar y firmar o verificar la serie secreta en el envío. Además, la señal se asocia con un tiempo de caducidad. En el extremo receptor, los datos se verifican, se descifran y se comparan con la serie secreta del receptor. Los protocolos de comunicación SSL (Secure Sockets Layer) no son obligatorios para la autenticación entre un par de servidores porque las claves privadas y públicas sirven para ese mismo propósito. No obstante, si la comunicación del servidor no está cifrada, los datos podrían robarse con sólo observar la comunicación. Como la señal caduca pronto, la amenaza de ataque de reproducción se minimiza. Esta posibilidad disminuye en gran medida si todos los servidores se despliegan detrás de un cortafuegos.

La desventaja de este enfoque es que los administradores de WebSphere eXtreme Scale deben generar claves y transportarlas a todos los servidores, que puede provocar una violación de seguridad durante el transporte.

Scripts de ejemplo para crear propiedades de gestor de señales seguras predeterminadas

Como se ha indicado en la sección anterior, puede crear un almacén de claves que contenga un par de claves para firmar y verificar la firma y una clave secreta para cifrar el contenido.

Por ejemplo, puede utilizar el mandato keytool de JDK 6 para crear las claves tal como se indica a continuación:

```
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
```

Estos dos mandatos crean a un par de claves "keypair1" y una clave secreta "seckey1". Luego puede configurar lo siguiente en el archivo de propiedades del servidor:

```
secureTokenKeyStore=key1.jck
secureTokenKeyStorePassword=key111
secureTokenKeyStoreType=JCEKS
secureTokenKeyPairAlias=keypair1
secureTokenKeyPairPassword=keypair1
secureTokenSecretKeyAlias=seckey1
secureTokenSecretKeyPassword=seckey1
secureTokenCipherAlgorithm=DES
secureTokenSignAlgorithm=RSA
```

Configuración

Consulte Propiedades de servidor si desea más información sobre las propiedades que utiliza para configurar el gestor de señales seguras.

Transport Layer Security (TLC) y Secure Sockets Layer (SSL)

WebSphere eXtreme Scale soporta TCP/IP y TLS/SSL (Transport Layer Security/Secure Sockets Layer) para la comunicación segura entre clientes y servidores.

Habilitar TLS/SSL en ambas direcciones

TLS/SSL a veces está habilitado en una dirección. Por ejemplo, el certificado público de servidor se importa en el almacén de confianza del cliente, pero el certificado público del cliente no se importa en el almacén de confianza del servidor. Sin embargo, WebSphere eXtreme Scale utiliza ampliamente agentes de cuadrícula de datos. Una característica de un agente de cuadrícula de datos es que cuando el servidor responde al cliente, crea una conexión nueva. A continuación, el servidor eXtreme Scale actúa como cliente. Por lo tanto, debe importar el certificado público de cliente en el almacén de confianza del servidor.

Habilitación de la seguridad de transporte para Sun JDK

WebSphere eXtreme Scale requiere IBM Java Secure Sockets Extension (IBMJSSE) o IBM Java Secure Sockets Extension 2 (IBMJSSE2). Los proveedores IBMJSSE e IBMJSSE2 contienen una implementación de referencia que da soporte a los protocolos SSL y TLS (seguridad de la capa de transporte) y una infraestructura de interfaz de programación de aplicaciones (API).

El SDK de Sun no proporciona proveedores IBM JSSE ni IBM JSSE2, por lo que no se puede habilitar la seguridad de transporte c on un JDK de Sun. Para realizar este trabajo, se requiere un JDK de Sun proporcionado con WebSphere Application Server. El JDK de Sun proporcionado con WebSphere Application Server contiene los proveedores IBM JSSE e IBM JSSE2.

Consulte “Configuración de un intermediario de solicitud de objetos personalizado” en la página 291 para obtener información sobre cómo utilizar un JDK no IBM para WebSphere eXtreme Scale. Si se configura `-Djava.endorsed.dirs`, apunta tanto al directorio `objectgridRoot/lib/endorsed` como al directorio `JRE/lib/endorsed`. El directorio `objectgridRoot/lib/endorsed` se necesita para utilizar IBM ORB y el directorio `JRE/lib/endorsed` se necesita para cargar os proveedores IBM JSSE e IBM JSSE2.

Trabaje con el paso 4 de la guía de aprendizaje de seguridad de *Visión general del producto* para obtener información sobre cómo configurar los parámetros SSL necesrios, para crear almacenes de claves y almacenes de confianza y para iniciar servidores seguros en WebSphere eXtreme Scale.

Configuración de tipos de transporte seguro

La seguridad de la capa de transporte (TLS) proporciona comunicación segura entre el cliente y el servidor. El mecanismo de comunicación que se utiliza depende del valor del parámetro **`transportType`** que se especifica en los archivos de configuración de cliente y servidor.

Acerca de esta tarea

Cuando se utiliza SSL (Secure Sockets Layer), se deben proporcionar los parámetros de configuración SSL tanto en el lado del cliente como en el lado del servidor. En un entorno Java SE, la configuración de SSL se realiza en los archivos de propiedad de cliente o servidor. Si el cliente o el servidor está en WebSphere Application Server, puede utilizar los valores de transporte existentes CSIV2 de WebSphere Application Server para los servidores y clientes de contenedor. Si desea más información, consulte “Integración de la seguridad con WebSphere Application Server” en la página 525.

Tabla 32. Protocolo de transporte a utilizar bajo los valores de transporte de cliente y de transporte de servidor.

Si los valores de transportType son distintos entre el cliente y el servidor, el protocolo resultante puede variar o generar un error.

Propiedad transportType de cliente	Propiedad transportType de servidor	Protocolo resultante
TCP/IP	TCP/IP	TCP/IP
TCP/IP	SSL-supported	TCP/IP
TCP/IP	SSL-required	Error
SSL-supported	TCP/IP	TCP/IP
SSL-supported	SSL-supported	SSL (si SSL falla, TCP/IP)
SSL-supported	SSL-required	SSL
SSL-required	TCP/IP	Error
SSL-required	SSL-supported	SSL
SSL-required	SSL-required	SSL

Procedimiento

1. Para establecer la propiedad **`transportType`** en la configuración de seguridad de cliente, consulte Archivo de propiedades de cliente .
2. Para establecer la propiedad **`transportType`** en la configuración de seguridad de servidor de catálogo y contenedor, consulte Archivo de propiedades de servidor .

Configuración de los parámetros SSL (Secure Sockets Layer) para clientes o servidores

La forma de configurar los parámetros SSL varía entre clientes y servidores.

Acerca de esta tarea

TLS/SSL a veces está habilitado en una dirección. Por ejemplo, se importa el certificado público de servidor al almacén de confianza de cliente, pero no se importa el certificado público de cliente al almacén de confianza de servidor. Sin embargo, WebSphere eXtreme Scale utiliza ampliamente agentes de cuadrícula de datos. Una característica de un agente de cuadrícula de datos consiste en que cuando el servidor revuelve respuestas al cliente, crea una conexión. A continuación, el servidor eXtreme Scale actúa como cliente. Por lo tanto, debe importar el certificado público de cliente en el almacén de confianza del servidor.

Procedimiento

- Configure los parámetros SSL de cliente.

  Utilice una de las opciones siguientes para configurar los parámetros SSL en el cliente:

  – Cree un objeto com.ibm.websphere.objectgrid.security.config.SSLConfiguration utilizando la clase de fábrica com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory.

  – Configure los parámetros en el archivo `client.properties`. A continuación, puede establecer el archivo de propiedades como una propiedad de cliente de JVM o bien puede utilizar las API de WebSphere eXtreme Scale. Pase el archivo de propiedades al método ClientSecurityConfigurationFactory.getClientSecurityConfiguration(String) del cliente y utilice el objeto devuelto como un parámetro para el método ObjectGridManager.connect(String, ClientSecurityConfiguration, URL).

- Configure los parámetros SSL de servidor.

  Los parámetros SSL de los servidores se configuran mediante el archivo `server.properties`. Para iniciar un servidor de contenedor o catálogo con un archivo de propiedades específico, utilice el parámetro **`-serverProps`** del script **`startOgServer`**. Para obtener más información sobre los parámetros SSL que puede establecer para servidores eXtreme Scale, consulte Propiedades del servidor de seguridad .

Seguridad JMX (Java Management Extensions)

Puede proteger las invocaciones de beans gestionados (MBean) en un entorno distribuido.

Para obtener más información sobre los MBeans disponibles, consulte el apartado “Administración con beans gestionados (MBeans)” en la página 434.

En la topología de despliegue distribuido, los MBeans se alojan directamente en los servidores de catálogo y servidores de contenedor. En general, la seguridad JMX en una topología distribuida sigue la especificación de seguridad JMX tal como se indica en la especificación Java Management Extensions (JMX). Consta de las tres partes siguientes:

1. Autenticación: el cliente remoto debe autenticarse en el servidor conector.
2. Control de accesos: el control de accesos de MBean limita quién puede acceder a la información de MBean y quién puede realizar las operaciones de MBean.

3. Transporte seguro: el transporte entre el cliente y el servidor JMX se puede proteger utilizando TLS/SSL.

Autenticación

JMX proporciona métodos para que los servidores de tipo conector autentiquen los clientes remotos. Para el conector RMI, la autenticación se completa proporcionando un objeto que implementa la interfaz JMXAuthenticator cuando se crea el servidor de conector. Así, eXtreme Scale implementa esta interfaz JMXAuthenticator para utilizar el plug-in Authenticator de ObjectGrid para autenticar los clientes remotos. Consulte “Guía de aprendizaje de seguridad de Java SE - Paso 2” en la página 73la guía de aprendizaje de seguridad en *Visión general del producto* para ver los detalles sobre cómo eXtreme Scale autentica un cliente.

El cliente JMX sigue las API de JMX para proporcionar credenciales y establecer conexión con el servidor conector. La infraestructura JMX pasa la credencial al servidor conector, y después llama a la implementación de JMXAuthenticator para obtener la autenticación. Como se ha descrito anteriormente, la implementación de JMXAuthenticator delega la autenticación a la implementación de ObjectGrid Authenticator.

Observe el ejemplo siguiente que describe cómo establecer conexión con un servidor conector mediante una credencial:

```
javax.management.remote.JMXServiceURL jmxUrl = new JMXServiceURL(
        "service:jmx:rmi:///jndi/rmi://localhost:1099/objectgrid/MBeanServer");

    environment.put(JMXConnector.CREDENTIALS, new UserPasswordCredential("admin", "xxxxxx"));

    // Crear JMXCconnectorServer
    JMXConnector cntor = JMXConnectorFactory.newJMXConnector(jmxUrl, null);

    // Conectar e invocar una operación en MBeanServer remoto
    cntor.connect(environment);
```

En el ejemplo anterior, se proporciona un objeto UserPasswordCredential con el ID de usuario establecido en `admin` y la contraseña establecida en `xxxxx`. Este objeto UserPasswordCredential se establece en la correlación de entorno, que se utiliza en el método JMXConnector.connect(Map). A continuación, este objeto UserPasswordCredential lo pasa al servidor la infraestructura JMX y, finalmente, se pasa a la infraestructura de autenticación de ObjectGrid para la autenticación.

El modelo de programación de cliente cumple la especificación JMX de manera estricta.

Control de acceso

Un servidor MBean JMX puede tener acceso a información confidencial y realizar operaciones confidenciales. JMX ofrece el control de acceso necesario que identifica qué clientes pueden acceder a la información y qué clientes pueden llevar a cabo las operaciones. El control de accesos se crea en el modelo de seguridad Java estándar definiendo los permisos que controlan el acceso al servidor MBean y sus operaciones.

Para el control de accesos o la autorización de la operación JMX, eXtreme Scale se basa en el soporte JAAS proporcionado por la implementación de JMX. En un punto dado de la ejecución de un programa, hay un conjunto de permisos actuales que una hebra de ejecución contiene. Cuando dicha hebra llama a una operación de la especificación JMX, estos permisos se denominan permisos mantenidos.

Cuando se realiza una operación JMX, se realiza una comprobación de seguridad para verificar que el permiso necesario está implicado en el permiso mantenido.

La definición de política MBean sigue el formato de la política Java. Por ejemplo, la política siguiente otorga a todos los firmantes y a todas las bases de código el derecho a recuperar la dirección JMX del servidor para PlacementServiceMBean, pero con una restricción para el dominio com.ibm.websphere.objectgrid.

```
grant {
    permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
     [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

Puede utilizar el siguiente ejemplo de política para completar la autorización basada en la identidad de cliente remoto. La política otorga al mismo permiso MBean como se muestra en el ejemplo anterior, excepto que sólo para los usuarios con el nombre X500Principal como 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";
}
```

Las políticas Java sólo se comprueban si el gestor de seguridad está activo. Inicie los servidores de catálogo y los servidores de contenedor con el argumento JVM -Djava.security.manager para aplicar el control de acceso de operaciones MBean.

Transporte seguro

El transporte entre el cliente JMX y el servidor puede protegerse mediante TLS/SSL. Si el valor transportType del servidor de catálogo o servidor de contenedor está establecido en SSL_Required o SSL_Supported, utilice SSL para conectarse al servidor JMX.

Para utilizar SSL, debe configurar el almacén de confianza, el tipo de almacén de confianza y la contraseña del almacén de confianza en el cliente MBean utilizando las propiedades del sistema -D:

1. -Djavax.net.ssl.trustStore=TRUST_STORE_LOCATION
2. -Djavax.net.ssl.trustStorePassword=TRUST_STORE_PASSWORD
3. -Djavax.net.ssl.trustStoreType=TRUST_STORE_TYPE

Si utiliza com.ibm.websphere.ssl.protocol.SSLSocketFactory como la fábrica de sockets SSL en el archivo *inicio_java*/jre/lib/security/java.security, utilice las propiedades siguientes:

1. -Dcom.ibm.ssl.trustStore=TRUST_STORE_LOCATION
2. -Dcom.ibm.ssl.trustStorePassword=TRUST_STORE_PASSWORD
3. -Dcom.ibm.ssl.trustStoreType=TRUST_STORE_TYPE

Para obtener esta información cuando Transport Layer Security/Secure Sockets Layer (TLS/SSL) está habilitado, debe iniciar los servidores de catálogo y contenedor con el puerto de servicio JMX establecido. Para establecer el puerto de servicio JMX, puede utilizar la opción **-JMXServicePort** en el script **startOgServer** o puede llamar al método setJMXServicePort en la interfaz ServerProperties.

Para habilitar el transporte seguro JMX para el servidor de contenedor, debe establecer el puerto de servicio JMX. Es necesario establecer el puerto de servicio JMX cuando utilice Transport Layer Security/Secure Sockets Layer (TLS/SSL) y desee visualizar información de servidor de contenedor del servidor de catálogo. Por ejemplo, es necesario el puerto cuando se utiliza el mandato **xscmd -c showMapSizes**. Utilice uno de los métodos siguientes para establecer el puerto de servicio JMX:

- Utilice la opción **-JMXServicePort** en el script **startOgServer**.
- Si utiliza un servidor incorporado, llame al método setJMXServicePort en la interfaz ServerProperties para establecer el puerto de servicio JMX.

Debe utilizar un número de puerto distinto para cada JVM de la configuración. Si desea utilizar JMX/RMI, especifique explícitamente la opción **-JMXServicePort** y el número de puerto, incluso si desea utilizar el valor de puerto predeterminado.

Integración de la seguridad con proveedores externos

Para proteger los datos, el producto se puede integrar con varios proveedores de seguridad.

WebSphere eXtreme Scale puede integrarse con una implementación de seguridad externa. Esta implementación externa debe proporcionar servicios de autenticación y autorización para WebSphere eXtreme Scale. WebSphere eXtreme Scale tiene puntos de plug-in para integrarse con una implementación de seguridad.WebSphere eXtreme Scale se ha integrado satisfactoriamente con los componentes siguientes:

- Lightweight Directory Access Protocol (LDAP)
- Kerberos
- Seguridad de ObjectGrid
- Tivoli Access Manager
- JAAS (Java Authentication and Authorization Service)

eXtreme Scale utiliza el proveedor de seguridad para las siguientes tareas:

- Autenticación de clientes en servidores.
- Autorización de clientes para acceder a determinados artefactos de eXtreme Scale o para especificar qué puede hacerse con los artefactos de eXtreme Scale.

eXtreme Scale tiene los siguientes tipos de autorizaciones:

Autorización de correlaciones
Los clientes o grupos pueden estar autorizados para realizar operaciones de inserción, lectura, actualización o supresión en correlaciones.

Autorización de ObjectGrid
Se puede autorizar a los clientes o grupos para realizar consultas de objetos o entidades en objectGrids.

Autorización de agentes de DataGrid
Los clientes o grupos pueden estar autorizados para permitir que se desplieguen los agentes DataGrid en un ObjectGrid.

Autorización de correlaciones del lado de servidor
Los clientes o grupos pueden estar autorizados para duplicar una correlación de servidor en el lado del cliente o para crear un índice dinámico en la correlación del servidor.

Autorización de administración
Los clientes o grupos pueden estar autorizados para realizar tareas de administración.

Nota: Si ya tenía habilitada la seguridad para el programa de fondo, recuerde que estos valores de seguridad ya no serán suficiente para proteger los datos. Los valores de seguridad de la base de datos u otro almacén de datos no se transfiere en absoluto a la memoria caché. Debe proteger de forma separada los datos que ahora están almacenados en la memoria caché utilizando el mecanismo de seguridad de eXtreme Scale, incluido la seguridad de autenticación, autorización y nivel de transporte.

Restricción: No utilice un Development Kit o Runtime Environment de la versión 1.6 y superior cuando también esté utilizando la seguridad de la capa de transporte SSL con una configuración autónoma de WebSphere eXtreme Scale. La versión 1.6 y posterior no da soporte a las interfaces de programación de aplicaciones de WebSphere eXtreme Scale Versión 7.1. Utilice la versión 1.5 o anterior para configuraciones que requieran seguridad de transporte SSL para instalaciones autónomas de eXtreme Scale. Esta restricción solo es aplicable cuando se utiliza seguridad SSL en una configuración autónoma de eXtreme Scale. La versión 1.6 y posterior está soportada para configuraciones de transporte que no sean SSL.

Protección del servicio de datos REST

Proteja varios aspectos del servicio de datos REST. El acceso al servicio de datos REST de eXtreme Scale se puede proteger mediante autenticación y autorización. El acceso solo lo pueden controlar las reglas de configuración con ámbito de servicio, denominadas reglas de acceso. La tercera cosa a tener en cuenta es la seguridad del transporte.

Acerca de esta tarea

El acceso al servicio de datos REST de eXtreme Scale se puede proteger mediante autenticación y autorización. La autenticación y autorización se llevan a cabo realizando la integración con la seguridad de eXtreme Scale.

El acceso lo pueden controlar también las reglas de configuración con ámbito de servicio, denominadas reglas de acceso. Existen dos tipos de reglas de acceso: derechos de operación de servicio que controlan las operaciones CRUD que permite el servicio y derechos de acceso de entidad que controlan las operaciones CRUD que se permiten para un tipo de entidad determinado.

Para conexiones entre el cliente web y el servicio REST, se proporciona seguridad de transporte mediante la configuración del contenedor que los aloja. Y se proporciona seguridad de transporte mediante la configuración de cliente de eXtreme Scale (para conexiones de servicio REST a cuadrícula de datos de eXtreme Scale).

Procedimiento

- Autenticación y autorización de control

  El acceso al servicio de datos REST de eXtreme Scale se puede proteger mediante autenticación y autorización. La autenticación y la autorización se realizan mediante la integración con la seguridad de eXtreme Scale.

El servicio de datos REST de eXtreme Scale utiliza seguridad de eXtreme Scale, para autenticación y autorización, para controlar qué usuarios pueden acceder al servicio y las operaciones que se permite realizar a un usuario mediante el servicio. El servicio de datos REST de eXtreme Scale utiliza una credencial global configurada, con usuario y contraseña, y una credencial derivada de un reto HTTP BASIC que se envía con cada transacción a la cuadrícula de datos de eXtreme Scale donde se realiza la autenticación y autorización.

1. Configure la autenticación y autorización del cliente eXtreme Scale en la cuadrícula Consulte el apartado “Integración de la seguridad con proveedores externos” en la página 520 para obtener detalles sobre cómo configurar la autenticación y autorización del cliente eXtreme Scale.
2. Configure el cliente de eXtreme Scale, que utiliza el servicio REST, para la seguridad.

   El servicio de datos REST de eXtreme Scale invoca la biblioteca del cliente eXtreme Scale al comunicarse con la cuadrícula eXtreme Scale. Por lo tanto, se debe configurar el cliente eXtreme Scale para la seguridad de eXtreme Scale.

   La autenticación del cliente eXtreme Scale se habilita mediante las propiedades del archivo de propiedades del cliente de objectgrid. Como mínimo, se deben habilitar los atributos siguientes al utilizar la seguridad del cliente con el servicio REST:

   ```
   securityEnabled=true
   credentialAuthentication=Supported [-or-] Required
   credentialGeneratorProps=user:pass [-or-] {xor encoded user:pass}
   ```

   Recuerde: El usuario y la contraseña especificados en la propiedad credentialGeneratorProps se deben correlacionar con un ID en el registro de autenticación y tener derechos de política de ObjectGrid suficientes para conectar a ObjectGrids y crearlos.

   Un archivo de política de cliente de objectgrid de ejemplo se encuentra en *inicio_servicioRest*/security/security.ogclient.properties. Véase también el apartado Archivo de propiedades de cliente .
3. Configure el servicio de datos REST de eXtreme Scale para la seguridad.

   El archivo de propiedades de configuración de servicio de datos REST de eXtreme Scale debe contener las entradas siguientes para la integración con la seguridad de eXtreme Scale:

   ```
   ogClientPropertyFile=nombre_archivo
   ```

   ogClientPropertyFile es la ubicación del archivo de propiedades que contiene las propiedades del cliente de ObjectGrid mencionadas en el paso anterior. El servicio REST utiliza este archivo para inicializar el cliente eXtreme Scale a fin de comunicarse con la cuadrícula cuando está habilitada la seguridad.

   ```
   loginType=basic [-or-] none
   ```

   La propiedad loginType configura el servicio REST para el tipo de inicio de sesión. Si se especifica el valor de none, el ID de usuario y la contraseña “global” definidos por credentialGeneratorProps se enviarán a la cuadrícula para cada transacción. Si se especifica un valor de basic, el servicio REST presentará un reto HTTP BASIC al cliente solicitándole credenciales que enviará a cada transacción al comunicarse con la cuadrícula.

   Para obtener más información sobre las propiedades ogClientPropertyFile y loginType, consulte Archivo de propiedades del servicio de datos REST.

- Aplique las reglas de acceso.

  El acceso se puede controlar también mediante reglas de configuración con ámbito de servicio, conocidas como reglas de acceso. Existen dos tipos de reglas

de acceso, los derechos de operación de servicio que controlan las operaciones CRUD permitidas por el servicio y los derechos de acceso de entidad que controlan las operaciones CRUD permitidas para un tipo de entidad concreto.

El servicio de datos REST de eXtreme Scale permite de modo opcional reglas de acceso que se pueden configurar para limitar el acceso al servicio y a las entidades del servicio. Estas reglas de acceso se especifican en el archivo de propiedades de derechos de acceso del servicio REST. El nombre de este archivo se especifica en el archivo de propiedades de servicio de datos REST mediante la propiedad wxsRestAccessRightsFile. Para obtener más información sobre esta propiedad, consulte Archivo de propiedades del servicio de datos REST. Este archivo es un archivo de propiedades Java típico con pares de clave y valor. Existen dos tipos de reglas de acceso, los derechos de operación de servicio que controlan las operaciones CRUD permitidas por el servicio y los derechos de acceso de entidad que controlan las operaciones CRUD permitidas para un tipo de entidad concreto.

1. Configure los derechos de operación de servicio.

   Los derechos de operación de servicio especifican los derechos de acceso que se aplican a todos los ObjectGrids expuestos mediante el servicio REST o a todas las entidades de un ObjectGrid individual como se ha especificado.

   Utilice la sintaxis siguiente.

   ```
   serviceOperationRights=derecho_operación_servicio
   serviceOperationRights.nombre_cuadrícula -OR- *=derecho_operación_servicio
   ```

   donde

   - serviceOperationRights puede tener uno de los valores siguientes [NONE, READSINGLE, READMULTIPLE, ALLREAD, ALL]
   - serviceOperationRights.*nombre_cuadrícula -OR-* * implica que el derecho de acceso se aplica a todos los ObjectGrids, si no se puede proporcionar el nombre de un ObjectGrid concreto.

   Por ejemplo:

   ```
   serviceOperationsRights=ALL
   serviceOperationsRights.*=NONE
   serviceOperationsRights.EMPLOYEEGRID=READSINGLE
   ```

   El primer ejemplo especifica que se permiten todas las operaciones de servicio para todos los ObjectGrids expuestos por el servicio REST. El segundo ejemplo es similar al primero porque se aplica también a todos los ObjectGrids expuestos por el servicio REST, no obstante, especifica el derecho de acceso como NONE, que significa que no se permite ninguna operación de servicio en los ObjectGrids. El último ejemplo especifica cómo controlar las operaciones de servicio para una cuadrícula concreta, aquí sólo se lee qué resultados se permiten en un solo registro para todas las entidades del EMPLOYEEGRID.

   El valor predeterminado que supone el servicio REST es `serviceOperationsRights=ALL`, lo que significa que se permiten todas las operaciones para todos los ObjectGrids expuestos por este servicio. Esto es distinto a la implementación de Microsoft, para la que el valor predeterminado es `NONE`, así que no se permiten operaciones en el servicio REST.

   Importante: Los derechos de operaciones de servicio se evalúan en el orden en que se especifican en este archivo, de modo que el último derecho especificado alterará temporalmente los derechos que le preceden.

2. Configure los derechos de acceso de entidad.

Los derechos de conjunto de entidades especifican los derechos de acceso que se aplican a entidades ObjectGrid concretas expuestas mediante el servicio REST. Estos derechos proporcionan un modo de imponer un control más estrecho y refinado en entidades ObjectGrid individuales en comparación con los derechos de operación de servicio.

Utilice la sintaxis siguiente.

```
entitySetRights.nombre_cuadrícula.nombre_entidad=derecho_conjunto_entidades
```

donde

– *derecho_conjunto_entidades* puede ser uno de estos derechos.

Tabla 33. Derechos de acceso de entidad. Valores admitidos.

Derecho de acceso	Descripción
NONE	Deniega todos los derechos para acceder a los datos
READSINGLE	Permite leer elementos de datos individuales
READMULTIPLE	Permite leer los conjuntos de datos
ALLREAD	Permite leer uno o varios conjuntos de datos
WRITEAPPEND	Permite crear nuevos elementos de datos en los conjuntos de datos
WRITEREPLACE	Permite sustituir los datos
WRITEDELETE	Permite suprimir los elementos de datos de los conjuntos de datos
WRITEMERGE	Permite fusionar los datos
ALLWRITE	Permite grabar (por ejemplo, crear, sustituir, fusionar o suprimir) los datos
ALL	Permite crear, leer, actualizar y suprimir los datos

– *nombre_entidad* es el nombre de un ObjectGrid concreto en el servicio REST.

– *nombre_cuadrícula* es el nombre de una entidad concreta en el ObjectGrid especificado.

Nota: Si se especifican los derechos de operación de servicio y los derechos de conjunto de entidades para un ObjectGrid respectivo y sus entidades, se impondrá lo más limitante de esos derechos, como se ilustra en los ejemplos siguientes. Recuerde también que los derechos del conjunto de entidades se evalúan en el orden en que se han especificado en el archivo. El último derecho especificado alterará temporalmente los derechos que lo preceden.

Ejemplo 1: Si se ha especificado serviceOperationsRights.NorthwindGrid=READSINGLE y entitySetRights.NorthwindGrid.Customer=ALL. Se impondrá READSINGLE para la entidad Customer.

Ejemplo 2: Si se ha especificado serviceOperationsRights.NorthwindGrid=ALLREAD y entitySetRights.NorthwindGrid.Customer=ALLWRITE solo se permitirán las lecturas para todas las entidades de NorthwindGrid. No obstante, para Customer sus derechos de conjunto de entidades impedirán las lecturas (dado que tiene especificado ALLWRITE) y de ahí realmente la entidad Customer tendrá de derecho de acceso NONE.

• Proteja los transportes.

Para conexiones entre el cliente web y el servicio web, se proporciona seguridad de transporte mediante la configuración del contenedor que lo aloja. Para conexiones entre el servicio REST y la cuadrícula de eXtreme Scale, se proporciona seguridad de transporte mediante la configuración de cliente de eXtreme Scale.

1. Proteja la conexión del cliente y el servicio REST. El entorno de contenedor de host proporciona la seguridad de transporte de esta conexión, no en eXtreme Scale.
2. Proteja la conexión del servicio REST y la cuadrícula eXtreme Scale. La seguridad de transporte de esta conexión se configura en eXtreme Scale. Consulte "Transport Layer Security (TLC) y Secure Sockets Layer (SSL)" en la página 515.

Integración de la seguridad con WebSphere Application Server

Cuando WebSphere eXtreme Scale se despliega en un entorno de WebSphere Application Server, puede simplificar el flujo de autenticación y la configuración de la seguridad de la capa de transporte desde WebSphere Application Server.

Flujo de autenticación simplificado

Cuando los clientes y los servidores eXtreme Scale se ejecutan en WebSphere Application Server y en el mismo dominio de seguridad, puede utilizar la infraestructura de seguridad de WebSphere Application Server para propagar las credenciales de autenticación de cliente en el servidor eXtreme Scale. Por ejemplo, si un servlet actúa como un cliente de eXtreme Scale para conectarse a un servidor eXtreme Scale en el mismo dominio de seguridad, y el servlet ya ha sido autenticado, es posible propagar la señal de autenticación del cliente (servlet) al servidor y, a continuación, utilice la infraestructura de seguridad de WebSphere Application Server para volver a convertir la señal de autenticación a las credenciales de cliente.

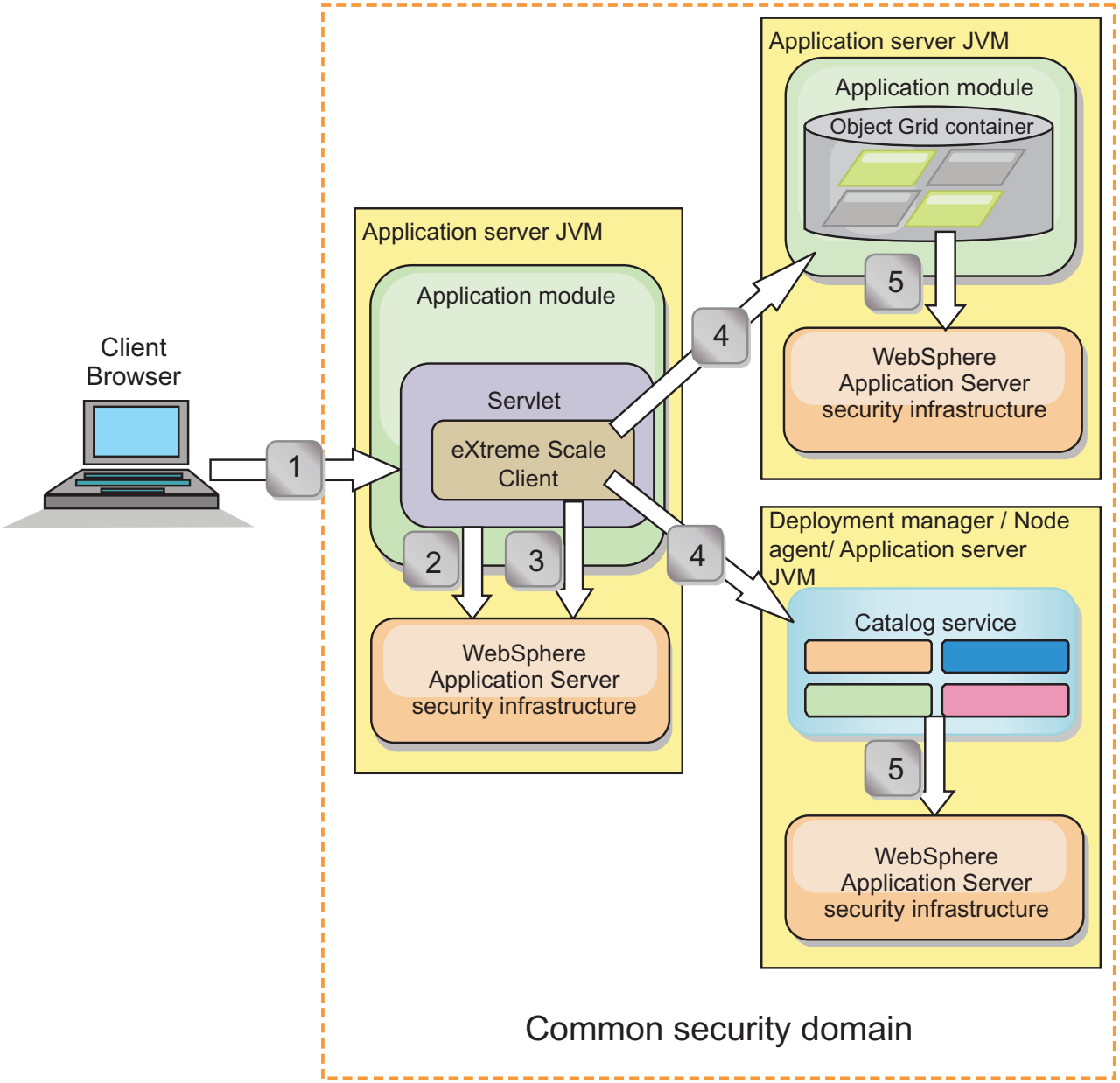


Figura 67. Flujo de autenticación para servidores en el mismo dominio de seguridad

En el diagrama anterior, los servidores de aplicaciones se encuentran en el mismo dominio de seguridad. Un servidor de aplicaciones aloja la aplicación web, que es también un cliente de eXtreme Scale. El otro servidor de aplicaciones aloja el servidor de contenedor. El gestor de despliegue o la máquina virtual Java (JVM) del agente del nodo aloja el servicio de catálogo. Las flechas del diagrama indican cómo fluye el proceso de autenticación:

1. Un usuario de aplicación empresarial utiliza un navegador web para iniciar la sesión en el primer servidor de aplicaciones con un nombre de usuario y una contraseña.
2. El primer servidor de aplicaciones envía el nombre de usuario y la contraseña del cliente a la infraestructura de seguridad de WebSphere Application Server para la autenticación en el registro de usuarios. Por ejemplo, este registro de usuarios podría ser un servidor LDAP. Como resultado, la información de seguridad se almacena en la hebra del servidor de aplicaciones.

3. El archivo JavaServer Pages (JSP) actúa como un cliente de eXtreme Scale para recuperar la información de seguridad de la hebra del servidor. El archivo JSP llama a la infraestructura de seguridad de WebSphere Application Server para obtener las señales de seguridad que representan el usuario de aplicación empresarial.
4. El cliente de eXtreme Scale, o archivo JSP, envía las señales de seguridad con la solicitud al servidor de contenedor y al servicio de catálogo alojado en las otras JVM. El servidor de catálogo y el servidor de contenedor utilizan las señales de seguridad de WebSphere Application Server como una credencial de cliente de eXtreme Scale.
5. Los servidores de catálogo y contenedor envían las señales de seguridad a la infraestructura de seguridad de WebSphere Application Server para convertir las señales de seguridad en información de seguridad de usuario. Esta información de seguridad de usuario la representa un objeto Subject, que contiene los principales, las credenciales públicas y las credenciales privadas. Esta conversión se puede producir porque los servidores de aplicaciones que alojan el cliente, el servidor de catálogo y el servidor de contenedor de eXtreme Scale comparten las mismas señales LTPA (Lightweight Third-Party Authentication) de WebSphere Application Server.

Integración de autenticación

Integración de seguridad distribuida con WebSphere Application Server:

Para el modelo distribuido, utilice las clases siguientes:

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredentialGenerator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenAuthenticator
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSTokenCredential

Para ver ejemplos sobre cómo utilizar estas clases, consulte “Guía de aprendizaje: Integrar la seguridad de WebSphere eXtreme Scale con WebSphere Application Server” en la página 86.

En el lado del servidor, utilice el autenticador WSTokenAuthentication para autenticar el objeto WSTokenCredential.

Integración de seguridad local con WebSphere Application Server:

Para el modelo de ObjectGrid local, utilice las clases siguientes:

- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectSourceImpl
- com.ibm.websphere.objectgrid.security.plugins.builtins. WSSubjectValidationImpl

Para obtener más información sobre estas clases, consulte Programación de la seguridad local. Puede configurar la clase WSSubjectSourceImpl como el plug-in SubjectSource y la clase WSSubjectValidationImpl como el plug-in SubjectValidation.

Soporte de seguridad de la capa de transporte en WebSphere Application Server

Cuando un cliente, servidor de contenedor o servidor de catálogo de eXtreme Scale se ejecuta en un proceso de WebSphere Application Server, la seguridad de transporte de eXtreme Scale la gestionan los valores de transporte CSIV2 de

WebSphere Application Server. Para el cliente o servidor de contenedor de eXtreme Scale, no debe utilizar las propiedades de cliente o servidor de eXtreme Scale para configurar los valores SSL. Todos los valores SSL se debe especificar en la configuración de WebSphere Application Server.

No obstante, el servidor de catálogo es un poco diferente. El servidor de catálogo tiene sus propias vías de acceso de transporte de propietario que no pueden gestionar los valores de transporte CSIV2 de WebSphere Application Server. Por lo tanto, las propiedades de SSL se deben seguir configurando en el archivo de propiedades de servidor correspondiente al servidor de catálogo. Si desea más información, consulte "Guía de aprendizaje: Integrar la seguridad de WebSphere eXtreme Scale con WebSphere Application Server" en la página 86.

Configuración de la seguridad de cliente en un dominio de servicio de catálogo

Al configurar la seguridad de cliente en un dominio de servicio de catálogo, puede definir propiedades predeterminadas de configuración de autenticación de cliente. Estas propiedades se utilizan cuando un archivo de propiedades de cliente no se encuentra en la máquina virtual Java (JVM) que aloja el cliente o cuando el cliente no especifica mediante programación propiedades de seguridad. Si existe un archivo de propiedades de cliente, las propiedades que especifique en la consola sustituirán los valores del archivo. Puede sustituir estas propiedades mediante la especificación de un archivo `splicer.properties` con la propiedad personalizada com.ibm.websphere.xs.sessionFilterProps o empalmando el archivo EAR de la aplicación.

Antes de empezar

- Debe conocer la implementación CredentialGenerator que está utilizando para autenticar clientes en la cuadrícula de datos remota. Puede utilizar una de las implementaciones proporcionadas por WebSphere eXtreme Scale: UserPasswordCredentialGenerator o WSTokenCredentialGenerator.

  También puede utilizar una implementación personalizada de la interfaz CredentialGenerator. La implementación personalizada debe estar en la classpath del cliente de tiempo de ejecución y del servidor. Si está configurando un escenario de sesiones HTTP con WebSphere Application Server, debe colocar la implementación de la classpath del gestor de despliegue y la classpath del servidor de aplicaciones en el que se está ejecutando el cliente.
- Debe tener definido un dominio de servicio de catálogo. Consulte el apartado "Creación de dominios de servicio de catálogo en WebSphere Application Server" en la página 257 para obtener más información.

Acerca de esta tarea

Debe configurar la seguridad de cliente en el dominio de servicio de catálogo cuando ha habilitado la autenticación de credenciales en el lado del servidor, configurando uno de los escenarios siguientes:

- La política de seguridad del lado del servidor tiene la propiedad **`credentialAuthentication`** establecida en `Required`.
- La política de seguridad del lado del servidor tiene la propiedad **`credentialAuthentication`** establecida en `Supported` y se ha especificado un **`authorizationMechanism`** en el archivo XML de ObjectGrid.

En estos escenarios, se debe proporcionar una credencial desde el cliente. La credencial que se proporciona desde el cliente se recupera del método

getCredential en una clase que implementa la interfaz CredentialGenerator. En un escenario de configuración de sesión HTTP, el tiempo de ejecución debe conocer la implementación CredentialGenerator que se utilizará para generar una credencial que se proporciona a la cuadrícula de datos remota. Si no especifica la clase de implementación CredentialGenerator que se utilizará, la cuadrícula de datos remota rechazará solicitudes del cliente porque el cliente no se podrá autenticar.

Procedimiento

Defina las propiedades de seguridad de cliente. En la consola administrativa de WebSphere Application Server, pulse **Administración del sistema** > **WebSphere eXtreme Scale** > **Dominios de servicio de catálogo** > *nombre_dominio_servicio_catálogo* > **Propiedades de seguridad de cliente**. Especifique las propiedades de seguridad de cliente en la página y guarde los cambios. Consulte "Propiedades de la seguridad de cliente" en la página 272 para ver una lista de las propiedades que puede establecer.

Resultados

Las propiedades de seguridad de cliente que ha configurado en el dominio de servicio de catálogo se utilizan como valores predeterminados. Los valores que especifica sustituyen a las propiedades definidas en los archivos `client.properties`.

Qué hacer a continuación

Configure las aplicaciones para utilizar WebSphere eXtreme Scale para la gestión de sesiones. Si desea más información, consulte "Empalmar automáticamente aplicaciones para la gestión de sesiones HTTP en WebSphere Application Server" en la página 305.

Habilitación de la seguridad local

WebSphere eXtreme Scale proporciona varios puntos finales de seguridad para integrar mecanismos personalizados. En el modelo de programación local, la principal función se seguridad es la autorización, que no tiene soporte de autenticación. Debe realizar la autenticación de forma independiente desde la autenticación que ya existe de WebSphere Application Server. Sin embargo, puede utilizar los plug-ins proporcionados para obtener y validar objetos Subject.

Acerca de esta tarea

Puede habilitar la seguridad local con el archivo de descriptor XML de ObjectGrid o mediante programación.

Procedimiento

Habilite la seguridad local con el archivo XML de descriptor de ObjectGrid. El archivo `secure-objectgrid-definition.xml` que se utiliza en la aplicación empresarial de ObjectGridSample se muestra en el siguiente ejemplo. Establezca el atributo securityEnabled en `true` para habilitar la seguridad.

```
<objectGrids>
    <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
          authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS">
    ...
</objectGrids>
```

Qué hacer a continuación

Inicie los servidores de contenedor y catálogo con la seguridad habilitada.

Inicio y detención de servidores seguros

La seguridad se habilita especificando configuraciones específicas de seguridad al iniciar y detener los servidores.

Inicio de servidores seguros en un entorno autónomo

Para iniciar servidores autónomos seguros, proporciona los archivos de configuración adecuados especificando parámetros en el mandato **startOgServer**.

Antes de empezar

Si está utilizando un proveedor de seguridad de cliente externo para la autenticación o autorización, defina la variable de entorno *CLIENT_AUTH_LIB*. Abra una ventana de terminal o línea de mandatos y ejecute el mandato correspondiente a su sistema operativo:

- Windows `set CLIENT_AUTH_LIB=<vía_acceso_a_JAR_o_clases_seguridad>`
- UNIX `set CLIENT_AUTH_LIB=<vía_acceso_a_JAR_o_clases_seguridad> export CLIENT_AUTH_LIB`

Cuando se ejecutan los mandatos **startOgServer** y **stopOgServer**, esta variable se añade a la classpath.

Procedimiento

- Inicie los servidores de contenedor seguros.

  El inicio de un servidor de contenedor seguro requiere el siguiente archivo de configuración de seguridad:

  – **Archivo de propiedad de servidor:** el archivo de propiedad de servidor configura las propiedades de seguridad específicas del servidor. Consulte Archivo de propiedades de servidor si desea más detalles.

  Especifique la ubicación de este archivo de configuración proporcionando el argumento siguiente al script **startOgServer**:

  -serverProps
  Especifica la ubicación del archivo de propiedades del servidor, que contiene propiedades de seguridad específicas del servidor. El nombre de archivo especificado para esta propiedad tiene formato de vía de acceso de archivo sencillo, por ejemplo, `../security/server.properties`.

- Inicie los servidores de catálogo seguros.

  Para iniciar un servicio de catálogo seguro, debe tener los siguientes archivos de configuración:

  – **Archivo XML de descriptor de seguridad:** el archivo XML de descriptor de seguridad describe las propiedades de seguridad comunes a todos los servidores, incluidos los servidores de catálogo y los servidores de contenedor. Un ejemplo de propiedad es la configuración de autenticador que representa el mecanismo de autenticación y el registro de usuarios.

  – **Archivo de propiedades del servidor:** el archivo de propiedades del servidor configura las propiedades de seguridad específicas del servidor.

  Especifique la ubicación de estos archivos de configuración proporcionando los argumentos siguientes al script **startOgServer**:

-clusterSecurityFile y -clusterSecurityUrl

Estos argumentos especifican la ubicación del archivo XML de descriptor de seguridad. Utilice el parámetro **-clusterSecurityFile** para especificar un archivo local, o el parámetro **-clusterSecurityUrl** para especificar el URL del archivo `objectGridSecurity.xml`.

-serverProps

Especifica la ubicación del archivo de propiedades del servidor, que contiene propiedades de seguridad específicas del servidor. El nombre de archivo especificado para esta propiedad tiene el formato de vía de acceso de archivo sencillo, por ejemplo, `c:/tmp/og/catalogserver.props`.

Inicio de servidores seguros en WebSphere Application Server

Para iniciar servidores seguros en WebSphere Application Server, debe especificar los archivos de configuración de seguridad en los argumentos genéricos de la JVM (máquina virtual Java).

Procedimiento

- Inicie un servicio de catálogo seguro en WebSphere Application Server.

  Un servidor de catálogo contiene dos niveles distintos de información de seguridad:

  – `-Dobjectgrid.cluster.security.xml.url`: especifica la ubicación del archivo `objectGridSecurity.xml`, que describe las propiedades de seguridad comunes a todos los servidores, incluidos los servidores de catálogo y los servidores de contenedor. Un ejemplo de las propiedades de seguridad definidas es la configuración del autenticador, que representa el registro de usuario y el mecanismo de autenticación. El nombre de archivo especificado para esta propiedad debe estar en un formato de URL como, por ejemplo, `file:///tmp/og/objectGridSecurity.xml`.

  – `-Dobjectgrid.server.props`: especifica el archivo de propiedades del servidor que contiene las propiedades de seguridad específicas del servidor. El nombre de archivo especificado para esta propiedad tiene el formato de vía de acceso de archivo sencillo, por ejemplo, `c:/tmp/og/catalogserver.props`.

  1. En la consola administrativa de WebSphere Application Server, pulse **Administración del sistema**. Pulse el proceso en el que se despliega el servidor de catálogo, por ejemplo, el gestor de despliegue.
  2. Pulse **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**.
  3. Especifique las propiedades en el campo **Argumentos de JVM genéricos**. A continuación se muestra un ejemplo de los valores que podría añadir:

     ```
     -Dobjectgrid.cluster.security.xml.url=file:///tmp/og/objectGridSecurity.xml
     -Dobjectgrid.server.props=/tmp/og/catalog.server.props
     ```

  4. Pulse **Aceptar** y guarde los cambios.

- Inicie el servidor de contenedor seguro en WebSphere Application Server.

  Un servidor de contenedor, al conectarse al servidor de catálogo, hereda la configuración de seguridad que se encuentra en el archivo `objectGridSecurity.xml` como, por ejemplo, la configuración del autenticador o los valores de tiempo de espera de inicio de sesión. Debe definir también propiedades de seguridad específicas del servidor para servidores de contenedor específicos en la propiedad `-Dobjectgrid.server.props`.

  El nombre de archivo especificado para esta propiedad está en formato de vía de acceso de archivo sencillo como, por ejemplo, `c:/tmp/og/server.props`.

Siga los mismos pasos anteriores para añadir la propiedad de seguridad a los argumentos genéricos de la JVM.

1. Abra la página de la máquina virtual Java para el servidor. En la consola administrativa de WebSphere Application Server, pulse **Servidores** > **Servidores de aplicaciones** > *nombre_servidor* > **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java**
2. Escriba la propiedad en el campo **Argumentos de JVM genéricos**. A continuación se muestra un ejemplo de los valores que podría añadir:

   ```
   -Dobjectgrid.server.props=/opt/wxs/security/server2.props
   ```
3. Pulse **Aceptar** y guarde los cambios.

Detención de servidores seguros

La detención de servidores de catálogo o servidores de contenedor seguros necesita un archivo de configuración de seguridad.

Procedimiento

Detenga un servidor de contenedor o de catálogo seguro.
La detención de un servidor seguro requiere el siguiente archivo de configuración de seguridad:

- **Archivo de propiedades de cliente:** el archivo de propiedades de cliente se puede utilizar para configurar las propiedades de seguridad de cliente. Las propiedades de seguridad de cliente son necesarias para que un cliente se conecte a un servidor seguro. Consulte Archivo de propiedades de cliente si desea más detalles.

Especifique la ubicación de estos archivos de configuración proporcionando el siguiente argumento al script **stopOgServer**:

-clientSecurityFile
Especifica la vía de acceso del archivo de propiedades de cliente que define las propiedades de seguridad para el cliente. El nombre de archivo que especifica para esta propiedad es el formato de vía de acceso de archivo sencillo, por ejemplo, `../security/objectGridClient.properties`.

Ejemplo

```
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
```

Configuración de perfiles de seguridad para el programa de utilidad xscmd

Mediante la creación de un perfil de seguridad, puede utilizar parámetros de seguridad guardados para utilizar el programa de utilidad **xscmd** con entornos seguros.

Antes de empezar

Para obtener más información sobre cómo configurar el programa de utilidad **xscmd**, consulte "Administración con el programa de utilidad **xscmd**" en la página 415.

Acerca de esta tarea

Puede utilizar el parámetro **-ssp** *nombre_perfil* o **--saveSecProfile** *nombre_perfil* con el resto del mandato **xscmd** para guardar un perfil de seguridad. El perfil puede contener valores para los nombres de usuario y las contraseñas, los generadores de credenciales, los almacenes de claves, los almacenes de confianza y los tipos de transporte.

El grupo de mandatos **ProfileManagement** del programa de utilidad **xscmd** contiene mandatos para gestionar los perfiles de seguridad.

Procedimiento

- Guarde un perfil de seguridad.

  Para guardar un perfil de seguridad, utilice el parámetro **-ssp** *nombre_perfil* o **--saveSecProfile** *nombre_perfil* con el resto del mandato. Al añadir este parámetro al mandato se guardan los parámetros siguientes:

  ```
  -al,--alias <alias>
  -arc,--authRetryCount <entero>
  -ca,--credAuth <soporte>
  -cgc,--credGenClass <nombre_clase>
  -cgp,--credGenProps <propiedad>
  -cxpv,--contextProvider <proveedor>
  -ks,--keyStore <vía_acceso_archivo>
  -ksp,--keyStorePassword <contraseña>
  -kst,--keyStoreType <tipo>
  -prot,--protocol <protocolo>
  -pwd,--password <contraseña>
  -ts,--trustStore <vía_acceso_archivo>
  -tsp,--trustStorePassword <contraseña>
  -tst,--trustStoreType <tipo>
  -tt,--transportType <tipo>
  -user,--username <nombre_usuario>
  ```

  Los perfiles de seguridad se guardan en el directorio *inicio_usuario*\.xscmd\profiles\security\<*nombre_perfil*>.properties.

- Utilice un perfil de seguridad guardado.

  Para utilizar un perfil de seguridad guardado, añada el parámetro **-sp** *nombre_perfil* o **--securityProfile** *nombre_perfil* al mandato que está ejecutando. Ejemplo de mandato: `xscmd -c listHosts -cep myhost.mycompany.com -sp myprofile`

- Liste los mandatos del grupo de mandatos **ProfileManagement**.

  Ejecute el mandato siguiente: **xscmd -lc ProfileManagement**.

- Liste los perfiles de seguridad existentes.

  Ejecute el mandato siguiente: **xscmd -c listProfiles -v**.

- Visualice los valores que se han guardado en un perfil de seguridad.

  Ejecute el mandato siguiente: **xscmd -c showProfile -pn** *nombre_perfil*.

- Elimine un perfil de seguridad existente.

  Ejecute el mandato siguiente: **xscmd -c RemoveProfile -pn** *nombre_perfil*.

Capítulo 11. Resolución de problemas

Además de los registros y el rastreo, los mensajes y las notas de release que se describen en este apartado, puede utilizar herramientas de supervisión para descubrir cuestiones como, por ejemplo, la ubicación de los datos en el entorno, la disponibilidad de los servidores en la cuadrícula de datos, etc. Si está trabajando en un entorno WebSphere Application Server, podrá utilizar la infraestructura PMI (Performance Monitoring Infrastructure). Si está trabajando en un entorno autónomo, podrá utilizar una herramienta de supervisión de proveedor como, por ejemplo, CA Wily Introscope o Hyperic HQ. También puede utilizar y personalizar el programa de utilidad **xscmd** para visualizar información textual sobre el entorno.

Habilitación del registro

Puede utilizar los registros para supervisar y solucionar problemas del entorno.

Acerca de esta tarea

Los registros se guardan en distintas ubicaciones y formatos en función de la configuración.

Procedimiento

- **Habilite los registros en un entorno autónomo.**

  Con servidores de catálogo autónomos, los registros se encuentran en la ubicación en la que se ejecuta el mandato **startOgServer**. Para los servidores de contenedor, puede utilizar la ubicación predeterminada o establecer una ubicación de registro personalizada:

  – **Ubicación de registro predeterminada:** los registros se encuentran en el directorio donde se ha ejecutado el mandato del servidor. Si inicia los servidores en el directorio *inicio_wxs*/bin, los registros y los archivos de rastreo se encuentran en los directorios logs/<*nombre_servidor*> del directorio bin.

  – **Ubicación de registro personalizada:** para especificar una ubicación alternativa para los registros de servidor de contenedor, cree un archivo de propiedades como, por ejemplo, server.properties, con el contenido siguiente:

  ```
  workingDirectory=<directorio>
  traceSpec=
  systemStreamToFileEnabled=true
  ```

  La propiedad **workingDirectory** es el directorio raíz de los registros y del archivo de rastreo opcional. WebSphere eXtreme Scale crea un directorio con el nombre del servidor de contenedor con un archivo SystemOut.log, un archivo SystemErr.log y un archivo de rastreo. Para utilizar un archivo de propiedades durante el inicio del contenedor, utilice la opción **-serverProps** y proporcione la ubicación del archivo de propiedades de servidor.

- **Habilite los registros en WebSphere Application Server.**

  Consulte WebSphere Application Server: Habilitación e inhabilitación del registro para obtener más información.

- **Recupere los archivos FFDC.**

  Los archivos FFDC sirven para que el servicio de soporte de IBM ayude a realizar la depuración. Es posible que el servicio de soporte IBM solicite estos

archivos cuando se produzca un problema. Estos archivos están en un directorio denominado, `ffdc`, y contienen archivos que se parecen al siguiente:

```
server2_exception.log
server2_20802080_07.03.05_10.52.18_0.txt
```

Qué hacer a continuación

Visualice los archivos de registro en sus ubicaciones especificadas. Los mensajes comunes para buscar en el archivo `SystemOut.log` son mensajes de confirmación de inicio, como el ejemplo siguiente:

```
CWOBJ1001I: ObjectGrid Server catalogServer01 está listo para procesar
solicitudes.
```

Para obtener más información sobre un mensaje específico en los archivos de registro, consulte Mensajes.

Recopilación de rastreo

Puede utilizar el rastreo para supervisar y resolver los problemas del entorno. Debe proporcionar rastreo para que un servidor funcione con el soporte de IBM.

Acerca de esta tarea

La recopilación de rastreo puede ayudar a supervisar y corregir problemas en el entorno de WebSphere eXtreme Scale. La forma de recopilar el rastreo dependerá de su configuración. Consulte "Opciones de rastreo" en la página 537 para ver una lista de las distintas especificaciones de rastreo que puede recopilar.

Procedimiento

- **Recopile el rastreo desde un entorno de WebSphere Application Server.**

  Si los servidores de catálogo y contenedor están en un entorno de WebSphere Application Server, consulte WebSphere Application Server: Cómo trabajar con el rastreo para obtener más información.

- **Recopile el rastreo con el mandato de inicio del servidor de catálogo o contenedor autónomo.**

  Puede establecer el rastreo en un servicio de catálogo o servidor de contenedor utilizando los parámetros **-traceSpec** y **-traceFile** con el mandato **startOgServer**. Por ejemplo:

  ```
  startOgServer.sh catalogServer -traceSpec ObjectGridPlacement=all=enabled -traceFile  /home/user1/logs/trace.log
  ```

  El parámetro **-traceFile** es opcional. Si no establece una ubicación **-traceFile**, el archivo de rastreo va a la misma ubicación que los archivos de registro de salida del sistema.Para obtener más información sobre estos parámetros, consulte "Script **startOgServer**" en la página 401.

- **Recopile el rastreo desde el servidor de contenedor o catálogo autónomo con un archivo de propiedades.**

  Para recopilar rastreo de un archivo de propiedades, cree un archivo como, por ejemplo, un archivo `server.properties`, con el contenido siguiente:

  ```
  workingDirectory=<directorio>
  traceSpec=<especificación_rastreo>
  systemStreamToFileEnabled=true
  ```

  La propiedad **workingDirectory** es el directorio raíz de los registros y del archivo de rastreo opcional. Si el valor **workingDirectory** no está establecido, el directorio de trabajo predeterminado es la ubicación utilizada para iniciar los

servidores, por ejemplo, *inicio_wxs*/bin. Para utilizar un archivo de propiedades durante el inicio del servidor, utilice el parámetro **-serverProps** con el mandato **startOgServer** y proporcione la ubicación del archivo de propiedades del servidor.Para obtener más información sobre el archivo de propiedades del servidor y sobre cómo utilizar el archivo, consulte Archivo de propiedades de servidor .

- **Recopile el rastreo de un cliente autónomo.**

  Puede iniciar la recopilación de rastreo en un cliente autónomo añadiendo propiedades del sistema al script de inicio de la aplicación cliente. En el ejemplo siguiente se especifican los valores de rastreo de la aplicación com.ibm.samples.MyClientProgram:

  ```
  java -DtraceSettingsFile=MyTraceSettings.properties
  -Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager
  -Djava.util.logging.configureByServer=true com.ibm.samples.MyClientProgram
  ```

  Consulte WebSphere Application Server: Habilitación del rastreo en aplicaciones cliente y autónomas para obtener más información.

- **Recopile el rastreo con la interfaz ObjectGridManager.**

  También puede establecer el rastreo durante el tiempo de ejecución en una interfaz ObjectGridManager. Si se establece el rastreo en una interfaz ObjectGridManager, se puede utilizar para obtener el rastreo en un cliente de eXtreme Scale, mientras se conecta a eXtreme Scale y confirma transacciones. Para establecer el rastreo en una interfaz ObjectGridManager, proporcione una especificación de rastreo y un registro de rastreo.

  ```
  ObjectGridManager manager = ObjectGridManagerFactory.getObjectGridManager();
  ...
  manager.setTraceEnabled(true);
  manager.setTraceFileName("logs/myClient.log");
  manager.setTraceSpecification("ObjectGridReplication=all=enabled");
  ```

  Para obtener más información sobre la interfaz ObjectGridManager, consulte la información sobre cómo interactuar con el ObjectGrid mediante la interfaz ObjectGridManager en la *Guía de programación*.

- **Recopile el rastreo en los servidores de contenedor con el programa de utilidad xscmd.**

  Para recopilar el rastreo con el programa de utilidad **xscmd**, utilice el mandato **-c setTraceSpec**. Utilice el programa de utilidad **xscmd** para recopilar el rastreo en un entorno autónomo durante el tiempo de ejecución en lugar de hacerlo durante el arranque. Puede recopilar rastreo en todos los servidores y servicios de catálogo o bien puede filtrar los servidores en función del nombre del ObjectGrid y otras propiedades. Por ejemplo, para recopilar rastreo de ObjectGridReplication con acceso al servidor de servicio de catálogo, ejecute:

  ```
  xscmd -c setTraceSpec "ObjectGridReplication=all=enabled"
  ```

  También puede inhabilitar el rastreo estableciendo la especificación de rastreo en *=all=disabled.

Resultados

Los archivos de rastreo se graban en la ubicación especificada.

Opciones de rastreo

Puede habilitar el rastreo para proporcionar información sobre el entorno al servicio de soporte de IBM.

Sobre el rastreo

El rastreo de WebSphere eXtreme Scale se divide en varios componentes distintos. Puede especificar el nivel de rastreo que se debe utilizar. Los niveles comunes de rastreo son: all, debug, entryExit y event.

A continuación se muestra una serie de rastreo de ejemplo:

```
ObjectGridComponent=level=enabled
```

Puede concatenar valores de rastreo. Utilice el símbolo `*` (asterisco) para especificar un valor comodín como, por ejemplo, `ObjectGrid*=all=enabled`. Si necesita proporcionar un rastreo al servicio de soporte de IBM, se solicita una serie de rastreo específica. Por ejemplo, si hay un problema con la réplica, se puede solicitar la serie de rastreo `ObjectGridReplication=debug=enabled`.

Especificación de rastreo

ObjectGrid
Motor de memoria caché principal general.

ObjectGridCatalogServer
Servicio de catálogo general.

ObjectGridChannel
Comunicaciones de topología de despliegue estática.

ObjectGridClientInfo
Información del cliente DB2

ObjectGridClientInfoUser
Información del usuario de DB2.

ObjectgridCORBA
Comunicaciones de topología de despliegue dinámica.

ObjectGridDataGrid
API de AgentManager.

ObjectGridDynaCache
Proveedor de la memoria caché dinámica de WebSphere eXtreme Scale.

ObjectGridEntityManager
API de EntityManager. Utilícela con la opción Projector.

ObjectGridEvictors
Desalojadores incorporados de ObjectGrid.

ObjectGridJPA
Cargadores JPA (Java Persistence API).

ObjectGridJPACache
Plug-ins de memoria caché JPA.

ObjectGridLocking
Gestor de bloqueos de entradas de memoria caché de ObjectGrid.

ObjectGridMBean
Beans de gestión.

ObjectGridMonitor
Infraestructura de supervisión histórica.

7.1.1+ **ObjectGridNative**
Rastreo de código nativo de WebSphere eXtreme Scale, incluido el código nativo eXtremeMemory.

7.1.1+ **ObjectGridOSGi**
Componentes de integración OSGi de WebSphere eXtreme Scale.

ObjectGridPlacement
Servicio de colocación de fragmentos de servidor de catálogo.

ObjectGridQuery
Consulte de ObjectGrid.

ObjectGridReplication
Servicio de réplica.

ObjectGridRouting
Detalles de direccionamiento de cliente/servidor.

ObjectGridSecurity
Rastreo de seguridad.

7.1.1+ **ObjectGridSerializer**
Infraestructura de plug-in DataSerializer.

ObjectGridStats
Estadísticas de ObjectGrid.

ObjectGridStreamQuery
API de consulta de secuencia.

7.1.1+ **ObjectGridTransactionManager**
Gestor de transacciones de WebSphere eXtreme Scale.

ObjectGridWriteBehind
Escritura diferida de ObjectGrid

7.1.1+ **ObjectGridXM**
Rastreo general de IBM eXtremeMemory.

7.1.1+ **ObjectGridXMEviction**
Rastreo de desalojo de eXtremeMemory.

7.1.1+ **ObjectGridXMTransport**
Rastreo de transporte general de eXtremeMemory.

7.1.1+ **ObjectGridXMTransportInbound**
Rastreo de transporte específico de entrada de eXtremeMemory.

7.1.1+ **ObjectGridXMTransportOutbound**
Rastreo de transporte específico de entrada de eXtremeMemory.

Projector
Motor en la API EntityManager.

QueryEngine
Motor de consulta para la API de consulta de objetos y la API de consulta EntityManager.

QueryEnginePlan
Rastreo del plan de consulta.

7.1.1+ **TCPChannel**
Canal TCP/IP de IBM eXtremeIO.

7.1.1+ **XsByteBuffer**
Rastreo de almacenamiento intermedio de bytes de WebSphere eXtreme Scale.

Análisis de datos de registro y rastreo

Puede utilizar las herramientas de análisis de registro para analizar cómo se realiza su ejecución y solucionar los problemas que se producen en el entorno.

Acerca de esta tarea

Puede generar informes a partir de los archivos de registro y de rastreo en el entorno. Estos informes visuales se pueden utilizar para los fines siguientes:

- **Para analizar el estado y el rendimiento del entorno de ejecución:**
  - Coherencia del entorno de despliegue
  - Frecuencia de registro
  - Topología en ejecución vs. topología configurada
  - Cambios de topología no planificados
  - Estado de quórum
  - Estado de réplica de partición
  - Estadísticas de memoria, rendimiento, uso de procesador, etc.
- **Para resolver problemas del entorno:**
  - Vistas de topología en puntos específicos en el tiempo
  - Estadísticas de memoria, rendimiento, uso del procesador durante anomalías de cliente
  - Niveles de fixpack actuales, valores de ajuste
  - Estado de quórum

Visión general del análisis de registro

Puede utilizar la herramienta **xsLogAnalyzer** como ayuda para la resolución de problemas del entorno.

Todos los mensajes de migración tras error

Visualiza el número total de mensajes de migración tras error como un gráfico a lo largo del tiempo. También muestra una lista de los mensajes de migración tras error, incluidos los servidores que han resultado afectados

Todos los mensajes críticos de eXtreme Scale

Visualiza ID de mensaje junto con las explicaciones y acciones de usuario asociadas, que le pueden evitar perder tiempo buscando mensajes.

Todas las excepciones

Visualiza las cinco primeras excepciones, incluidos los mensajes, el número de veces que éstos se han producido y qué servidores se han visto afectados por la excepción.

Resumen de topología

Visualiza un diagrama de cómo se configura la topología según los archivos de registro. Puede utilizar este resumen para comparar con la configuración real, posiblemente identificando errores de configuración.

Coherencia de la topología: tabla de comparación de intermediario de solicitud de objetos (ORB)

Visualiza valores de ORB en el entorno. Puede utilizar esta tabla como ayuda para determinar si los valores son coherentes en todo el entorno.

Vista de franja horaria del suceso

Visualiza un diagrama de franja horaria de las distintas acciones que se han producido en la cuadrícula de datos, incluidos los sucesos de ciclo de vida, excepciones, mensajes críticos y sucesos de captura de datos en primer error (FFDC).

Ejecución de análisis de registro

Puede ejecutar la herramienta **xsLogAnalyzer** en un conjunto de archivos de registro y rastreo desde cualquier sistema.

Antes de empezar

- Habilite los registros y el rastreo. Consulte "Habilitación del registro" en la página 535 y "Recopilación de rastreo" en la página 536 para obtener más información.
- Recopile los archivos de registro. Los archivos de registro se pueden encontrar en diversas ubicaciones en función de cómo los haya configurado. Si está utilizando los valores de registro predeterminados, puede obtener los archivos de registro de las ubicaciones siguientes:
  - En una instalación autónoma: *raíz_intal_wxs*/bin/logs/<*nombre_servidor*>
  - En una instalación integrada con WebSphere Application Server: *raíz_was*/logs/<*nombre_servidor*>
- Recopile los archivos de rastreo. Los archivos de rastreo pueden estar en diversas ubicaciones en función de cómo los haya configurado. Si está utilizando los valores de rastreo predeterminados, puede obtener los archivos de rastreo en las ubicaciones siguientes:
  - En una instalación autónoma: si no se establece ningún valor de rastreo específico, los archivos de rastreo se graban en la misma ubicación que los archivos de registro de salida del sistema.
  - En una instalación integrada con WebSphere Application Server: *raíz_was*/profiles/*nombre_servidor*/logs.

  Copie los archivos de registro y rastreo en el sistema desde el que está planificando utilizar la herramienta Log Analyzer.
- Si desea crear exploraciones personalizadas en el informe generado, cree un archivo de propiedades de especificaciones de exploración y un archivo de configuración antes de ejecutar la herramienta. Para obtener más información, consulte "Creación de exploradores personalizados para el análisis de registro" en la página 543.

Procedimiento

1. Ejecute la herramienta **xsLogAnalyzer**.

   El script se encuentra en las ubicaciones siguientes:

   - En una instalación autónoma: *raíz_intal_wxs*/ObjectGrid/bin
   - En una instalación integrada con WebSphere Application Server: *raíz_was*/bin

   Consejo: Si los archivos de registro son grandes, tenga en cuenta la posibilidad de utilizar los parámetros **-startTime**, **-endTime** y **-maxRecords** al ejecutar el informe para restringir el número de entradas de registro que se exploran. Si utiliza estos parámetros al ejecutar el informe, será más fácil leer los informes y éstos se ejecutarán de forma más efectiva. Puede ejecutar varios informes en el mismo conjunto de archivos de registro.

   ```
   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
   Especifica la vía de acceso absoluta al directorio de registro que desea evaluar (necesario).

   -outDir
   Especifica un directorio existente para grabar la salida de informe. Si no especifica un valor, el informe se graba en la ubicación raíz de la herramienta **xsLogAnalyzer**.

   -startTime
   Especifica la hora de inicio para realizar la evaluación en los registros. La fecha tiene el formato siguiente:
   año.mes.día.hora.minuto.segundo.milisegundo

   -endTime
   Especifica la hora de finalización para realizar la evaluación en los registros. La fecha tiene el formato siguiente:
   año.mes.día.hora.minuto.segundo.milisegundo

   -trace Especifica una serie de rastreo, por ejemplo ObjectGrid*=all=enabled.

   -maxRecords
   Especifica el número máximo de registros a generar en el informe. El valor predeterminado es 100. Si especifica el valor como 50, se generan los primeros 50 registros para el periodo de tiempo especificado.

2. Abra los archivos generados. Si no ha definido un directorio de salida, los informes se generan en una carpeta denominada report_*fecha_hora*. Para abrir la página principal de los informes, abra el archivo index.html.
3. Utilice los informes para analizar los datos de registro. Utilice las sugerencias siguientes para maximizar el rendimiento de las visualizaciones de informe:
   - Para maximizar el rendimiento de las consultas en los datos de registro, utilice la información más específica que sea posible. Por ejemplo, una consulta para servidor tarda mucho más tiempo en ejecutarse y devuelve más resultados que nombre_host_servidor.
   - Algunas vistas tienen un número limitado de puntos de datos que se visualizan a la vez. Puede ajustar el segmento de tiempo que se está viendo cambiando en la vista los datos actuales, por ejemplo hora de inicio y finalización.

Qué hacer a continuación

Para obtener más información acerca de la resolución de problemas de la herramienta **xsLogAnalyzer** y los informes generados, consulte “Resolución de problemas de análisis de registro” en la página 544.

Creación de exploradores personalizados para el análisis de registro

Puede crear exploradores personalizados para el análisis de registro. Después de configurar el explorador, se generan los resultados en los informes al ejecutar la herramienta **xsLogAnalyzer**. El explorador personalizado explora en las anotaciones cronológicas para obtener registros de sucesos basándose en las expresiones regulares que se han especificado.

Procedimiento

1. Cree un archivo de propiedades de especificaciones de explorador que especifique la expresión general a ejecutar para el explorador personalizado.
   a. Cree y guarde un archivo de propiedades. El archivo debe estar en el directorio *raíz_loganalyzer*/config/custom. Puede utilizar el nombre que desee para el archivo. Puesto que el nuevo explorador utilizará el archivo, resulta útil darle nombre al explorador en el archivo de propiedades, por ejemplo: *mi_especificación_explorador_servidor_nuevo*.properties.
   b. Incluya las propiedades siguientes en el archivo: *mi_especificación_explorador_servidor_nuevo*.properties

      ```
      include.regular_expression = EXPRESIÓN_REGULAR_PARA_EXPLORAR
      ```

      La variable *EXPRESIÓN_REGULAR_PARA_EXPLORAR* es una expresión regular en la que se deben filtrar los archivos de registro.

      Ejemplo: Para explorar en instancias de líneas que contienen las series "xception" y "rror" independientemente del orden, establezca la propiedad **include.regular_expression** en el valor siguiente:

      ```
      include.regular_expression = (xception.+rror)|(rror.+xception)
      ```

      Esta expresión regular hace que se registren sucesos si la serie "rror" va antes o después de la serie "xception".

      Ejemplo:Para explorar en cada línea de los registros las instancias de líneas que contienen las series "xception" de frase o "rror" de frase independientemente del orden, establezca la propiedad **include.regular_expression** en el valor siguiente:

      ```
      include.regular_expression = (xception)|(rror)
      ```

      Esta expresión regular hace que se registren sucesos si la serie "rror" va antes o después de la serie "xception".
2. Cree un archivo de configuración que la herramienta **xsLogAnalyer** utilice para crear el explorador.
   a. Cree y guarde un archivo de propiedades. El archivo debe estar en el directorio *raíz_loganalyzer*/config/custom. Puede dar al archivo el nombre *nombre_explorador*Scanner.config, donde *nombre_explorador* es un nombre exclusivo para el nuevo explorador. Por ejemplo, puede dar al archivo el nombreserverScanner.config
   b. Incluya las propiedades siguientes en el archivo *nombre_explorador*Scanner.config:

```
scannerSpecificationFiles = UBICACIÓN_DE_ARCHIVO_ESPECIFICACIÓN_EXPLORADOR
```

La variable *UBICACIÓN_DE_ARCHIVO_ESPECIFICACIÓN_EXPLORADOR* es la vía de acceso y la ubicación del archivo de especificación que ha creado en el paso anterior. Por ejemplo: *raíz_loganalyzer*/config/custom/mi_especificación_escáner_nueva.properties. Puede también especificar varios archivos de especificación de explorador utilizando una lista separada por punto y coma:

```
scannerSpecificationFiles = UBICACIÓN_DE_ARCHIVO1_ESPECIFICACIÓN_EXPLORADOR;UBICACIÓN_DE_ARCHIVO2_ESPECIFICACIÓN_EXPLORADOR
```

3. Ejecute la herramienta **xsLogAnalyzer**. Para obtener más información, consulte "Ejecución de análisis de registro" en la página 541.

Resultados

Después de ejecutar la herramienta **xsLogAnalyzer**, el informe contiene separadores nuevos para los exploradores personalizados que ha configurado. Cada separador contiene las vistas siguientes:

Gráficos
Gráfico trazado que ilustra los sucesos registrados. Los sucesos se visualizan en el orden en el que se han encontrado.

Tablas Representación tabular de los sucesos registrados.

Informes de resumen

Resolución de problemas de análisis de registro

Utilice la siguiente información de resolución de problemas para diagnosticar y arreglar problemas con la herramienta **xsLogAnalyzer** y los informes generados.

Procedimiento

- **Problema:** Se producen condiciones de falta de memoria cuando se utiliza la herramienta **xsLogAnalyzer** para generar informes. A continuación se muestra un ejemplo de un error que se puede producir: `java.lang.OutOfMemoryError: Se ha superado el límite de sobrecarga de GC`.

  Solución: La herramienta **xsLogAnalyzer** se ejecuta en una máquina virtual Java (JVM). Puede configurar la JVM para aumentar el tamaño de almacenamiento dinámico antes de ejecutar la herramienta **xsLogAnalyzer** especificando algunos valores cuando ejecute la herramienta. Si aumenta el tamaño de almacenamiento dinámico se podrán almacenar más registros de sucesos en la memoria de JVM. Empiece con un valor de 2048M, suponiendo que el sistema operativo tenga suficiente memoria principal. En la misma instancia de línea de mandatos en la que piensa ejecutar la herramienta **xsLogAnalyzer**, establezca el tamaño de almacenamiento dinámico de JVM máximo:

  ```
  java -XmxTAMAÑO_ALMACENAMIENTO_DINÁMICOm
  ```

  El valor de *TAMAÑO_ALMACENAMIENTO_DINÁMICO* puede ser cualquier cualquier entero y representa el número de megabytes que están asignados al almacenamiento dinámico de JVM. Por ejemplo, puede ejecutar `java -Xmx2048m`. Si continúan los mensajes que indican que falta memoria o no tiene los recursos para asignar 2048m o más memoria, limite el número de sucesos que se están manteniendo en el almacenamiento dinámico. Puede limitar el número de sucesos en el almacenamiento dinámico pasando el parámetro **-maxRecords** al mandato **xsLogAnalyzer**.
- **Problema:** Cuando se abre un informe generado desde la herramienta **xsLogAnalyzer**, el navegador se cuelga o no carga la página.

Causa: Los archivos HTML generados son demasiado grandes y el navegador no los puede cargar. Estos archivos son grandes porque el ámbito de los archivos de registro que está analizando es demasiado amplio.

Solución: Considere la posibilidad de utilizar los parámetros **-startTime**, **-endTime** y **-maxRecords** cuando ejecute la herramienta **xsLogAnalyzer** para restringir el número de entradas de registro que se exploran. Si utiliza estos parámetros al ejecutar el informe, será más fácil leer los informes y éstos se ejecutarán de forma más efectiva. Puede ejecutar varios informes en el mismo conjunto de archivos de registro.

Resolución de problemas de la instalación

Utilice esta información para resolver problemas de la instalación.

Procedimiento

- **Problema:** al ejecutar el mandato de instalación desde un sistema remoto como, por ejemplo, `\\mymachine\downloads\`, se visualiza el mensaje siguiente: `CMD.EXE se ha iniciado con la vía de acceso anterior como directorio actual. No se da soporte a vías de acceso UNC. Se toma el directorio predeterminado de Windows.` Como resultado, la instalación no se completa correctamente.

  Solución: correlacione el sistema remoto con una unidad de red. Por ejemplo, en Windows, puede pulsar con el botón derecho del ratón en **Mi sistema** y elegir **Correlacionar unidad de red** e incluir la vía de acceso de convenio de nomenclatura uniforme (UNC) del sistema remoto. A continuación, puede ejecutar el script de instalación satisfactoriamente desde la unidad de red, por ejemplo: `y:\mymachine\downloads\WXS\install.bat`.
- **Problema:** la instalación no se completa satisfactoriamente.

  Solución: consulte los archivos de registro para ver dónde ha fallado la instalación. Cuando la instalación no se completa satisfactoriamente, los registros se encuentran en el directorio *raíz_intal_wxs*`/logs/wxs`.
- **Problema:** se produce una anomalía extremadamente grave durante la instalación.

  Solución: consulte los archivos de registro para ver dónde ha fallado la instalación. Cuando la instalación falla cuando se ha completado parcialmente, los registros normalmente se pueden encontrar en el directorio *raíz_usuario*`/wxs_install_logs/`.
- Windows **Problema:** si va a instalar WebSphere eXtreme Scale Client en Windows, podría ver el texto siguiente en los resultados de la instalación:

  ```
  Finalizado con éxito: Se ha instalado de forma satisfactoria el siguiente
  producto: cliente de WebSphere eXtreme Scale. Algunos pasos de la configuración
  han dado errores. Consulte el siguiente archivo de anotaciones cronológicas para
  obtener más información: <raíz de instalación de WebSphere Application
  Server>\logs\wxs_client\install\log.txt" Revise el archivo de registro
  de instalación (log.txt) y revise el registro de aumento del
  gestor de despliegue.
  ```

  Solución: si ve una anomalía del archivo `iscdeploy.sh`, puede ignorar el error. Este error no causa problemas.

Resolución de problemas de la integración de la memoria caché

Utilice esta información para resolver problemas de la configuración de la integración de la memoria caché, incluidas las configuraciones de memoria caché dinámica y de sesión HTTP.

Procedimiento

- 7.1.1+ **Problema:** los ID de sesión HTTP no se están reutilizando.

  Causa: puede utilizar los ID de sesión. Si crea una cuadrícula de datos para la persistencia de sesión en la versión 7.1.1 o posterior, se habilita automáticamente la reutilización de ID de sesión. Sin embargo, si ha creado configuraciones anteriores, es posible que este valor ya se haya creado con un valor incorrecto.

  Solución: compruebe los valores siguientes para verificar que tiene habilitada la reutilización de ID de sesión HTTP:

  – La propiedad `reuseSessionId` del archivo `splicer.properties` se debe establecer en `true`.

  – El valor de la propiedad personalizada `HttpSessionIdReuse` se debe establecer en `true`. Esta propiedad personalizada podría establecerse en una de las siguientes vías de acceso en la consola administrativa de WebSphere Application Server:

  - **Servidores** > *nombre_servidor* > **Gestión de sesiones** > **Propiedades personalizadas**
  - **Clústeres dinámicos** > *nombre_clúster_dinámico* > **Plantilla de servidor** > **Gestión de sesiones** > **Propiedades personalizadas**
  - **Servidores** > **Tipos de servidor** > **Servidores de aplicaciones WebSphere** > *nombre_servidor* y, a continuación, en Infraestructura del servidor, pulse **Java y gestión de procesos** > **Definición de proceso** > **Máquina virtual Java** > **Propiedades personalizadas**
  - **Servidores** > **Tipos de servidor** > **Servidores de aplicaciones WebSphere** > *nombre_servidor* > **Valores de contenedor web** > **Contenedor web**

  Si actualiza cualquier valor de propiedad personalizada, vuelva a configurar la gestión de sesiones de eXtreme Scale de modo que el archivo `splicer.properties` tenga en cuenta el cambio.

- **Problema:** Cuando se utiliza una cuadrícula de datos para almacenar sesiones HTTP y la carga de transacciones es alta, se visualiza un mensaje CWOBJ0006W en el archivo `SystemOut.log`.

  ```
  CWOBJ0006W: Se ha producido una excepción:
  com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
  java.util.ConcurrentModificationException
  ```

  Este mensaje sólo aparece cuando el parámetro **`replicationInterval`** del archivo `splicer.properties` está establecido en un valor mayor que cero y la aplicación web modifica un objeto de lista que se ha establecido como atributo en HTTPSession.

  Solución: Clone el atributo que contiene el objeto de lista modificado y ponga el atributo clonado en el objeto de sesión.

Resolución de problemas del plug-in de memoria caché JPA

Utilice esta información para resolver problemas de la configuración del plug-in de memoria caché JPA. Estos problemas se pueden producir tanto en configuraciones Hibernate como en configuraciones OpenJPA.

Procedimiento

- **Problema:** se visualiza la siguiente excepción: `CacheException: No se ha podido obtener el servidor ObjectGrid`.

  Con un valor de atributo de **`ObjectGridType`** `EMBEDDED` o `EMBEDDED_PARTITION`, la memoria caché de eXtreme Scale intenta obtener una instancia de servidor en el tiempo de ejecución. En un entorno Java Platform, Standard Edition, se inicia un

servidor eXtreme Scale con el servicio de catálogo incorporado. El servicio de catálogo incorporado intenta estar a la escucha en el puerto 2809. Si ese puerto lo utiliza otro proceso, se produce el error.

Solución: si se especifican puntos finales de servicio de catálogo externo, por ejemplo, con el archivo `objectGridServer.properties`, se produce este error si el nombre de host o puerto se especifica incorrectamente. Corrija el conflicto de puerto.

- **Problema:** se visualiza la siguiente excepción: `CacheException: No se ha podido obtener el ObjectGrid REMOTE para el ObjectGrid REMOTE configurado. objectGridName = [ObjectGridName], PU name = [persistenceUnitName]`

  Este error se produce cuando la memoria caché no puede obtener la instancia de ObjectGrid desde los puntos finales de servicio de catálogo proporcionados.

  Solución: este problema normalmente se produce debido a un nombre de host o puerto incorrecto.

- **Problema:** se visualiza la siguiente excepción: `CacheException: no se puede tener dos PU [nombreUnidadPersistencia_1, nombreUnidadPersistencia_2] configuradas con el mismo ObjectGridName [ObjectGridName] de ObjectGridType EMBEDDED`

  Esta excepción se produce si tiene muchas unidades de persistencia configuradas y las memorias caché de eXtreme Scale de estas unidades se configuran con el mismo nombre de ObjectGrid y valor de atributo de **`ObjectGridType`** `EMBEDDED`. Estas configuraciones de unidades de persistencia podrían estar en los mismos archivos `persistence.xml` o en archivos diferentes.

  Solución: debe verificar que el nombre de ObjectGrid sea exclusivo para cada unidad de persistencia cuando el valor de atributo **`ObjectGridType`** sea `EMBEDDED`.

- **Problema:** se visualiza la siguiente excepción: `CacheException: REMOTE ObjectGrid [ObjectGridName] no incluye las BackingMaps necesarias [nombreCorrelación_1, nombreCorrelación_2,...]`

  Con el tipo de ObjectGrid REMOTE, si el ObjectGrid del lado del cliente obtenido no tiene correlaciones de respaldo de entidad completas para dar soporte a la memoria caché de unidad de persistencia, se produce esta excepción. Por ejemplo, se listan cinco clases de entidad en la configuración de la unidad de persistencia, pero el ObjectGrid obtenido sólo tiene dos BackingMaps. Aunque el ObjectGrid obtenido podría tener 10 BackingMaps, si no se encuentra alguno de las cinco BackingMaps de entidad necesarias en las diez correlaciones de respaldo, aún se produce esta excepción.

  Solución: asegúrese de que la configuración de correlación de respaldo dé soporte a la memoria caché de unidad de persistencia.

Resolución de problemas de administración

Utilice la información siguiente para resolver problemas de administración, incluyendo el inicio de servidores, utilizando el programa de utilidad **`xscmd`**, etc.

Procedimiento

- **Problema:** Faltan los scripts de administración en el directorio *raíz_perfil*`/bin` de una instalación de WebSphere Application Server.

  Causa: Cuando se actualiza la instalación, los nuevos archivos de script no se instalan automáticamente en los perfiles.

  Solución: Si desea ejecutar un script desde el directorio *raíz_perfil*`/bin`, reduzca y vuelva a aumentar el perfil con el último release. Para obtener más

información, consulte Reducción de un perfil utilizando el indicador de mandatos y “Creación y aumento de perfiles para WebSphere eXtreme Scale” en la página 184.

- **Problema:** Cuando se ejecuta un mandato **xscmd**, se muestra el mensaje siguiente en la pantalla:

```
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
```

  Causa: Se ha producido un problema de conexión con el servidor de catálogo.

  Solución: Verifique que los servidores de catálogo se están ejecutando y están disponibles a través de la red. Este mensaje también puede aparecer cuando se ha definido un dominio de servicio de catálogo, pero se están ejecutando menos de dos servidores de catálogo. El entorno no está disponible hasta que se inician dos servidores de catálogo.

Resolución de problemas de onfiguraciones de varios centros de datos

Utilice esta información para resolver los problemas de configuraciones de varios centros de datos, incluido el enlace entre los dominios de servicio de catálogo.

Procedimiento

Problema: faltan datos en uno o varios dominios de servicio de catálogo. Por ejemplo, puede ejecutar el mandato `xscmd -c establishLink`. Cuando comprueba los datos correspondientes a cada dominio de servicio de catálogo enlazado, parece que los datos son distintos, por ejemplo, desde el mandato `xscmd -c showMapSizes`.
Solución: puede solucionar este problema con el mandato `xscmd -c showLinkedPrimaries`. Este mandato imprime cada fragmento primario, incluidos qué primarios foráneos están enlazados.
En el escenario descrito, al ejecutar el mandato `xscmd -c showLinkedPrimaries` podría descubrir que los fragmentos primarios del primer dominio de servicio de catálogo están enlazados con los fragmentos primarios del segundo dominio de servicio de catálogo, pero que el segundo dominio de servicio de catálogo no tiene enlaces al primer dominio de servicio de catálogo. Puede considerar volver a ejecutar el mandato `xscmd -c establishLink` desde el segundo dominio de servicio de catálogo al primer dominio de servicio de catálogo.

Resolución de problemas de los cargadores

Utilice esta información para resolver problemas de los cargadores de base de datos.

Procedimiento

- **Problema:** cuando se utiliza un cargador OpenJPA con DB2 en WebSphere Application Server, se produce una excepción de cursor cerrado.

  Ls siguiente excepción procede de DB2 en el archivo de registro de org.apache.openjpa.persistence.PersistenceException:

```
[jcc][t4][10120][10898][3.57.82] Operación no válida: el conjunto de
resultados está cerrado.
```

Solución: De forma predeterminada, el servidor de aplicaciones configura la propiedad personalizada resultSetHoldability con un valor de 2 (CLOSE_CURSORS_AT_COMMIT). Esta propiedad hace que DB2 cierre su conjunto de resultados/cursor en los límites de transacción. Para eliminar la excepción, cambie el valor de la propiedad personalizada a 1 (HOLD_CURSORS_OVER_COMMIT). Establezca la propiedad personalizada resultSetHoldability en la siguiente vía de acceso en la célula de WebSphere Application Server: **Recursos** > **Proveedor JDBC** > **Proveedor de controlador JDBC de DB2 Universal** > **Orígenes de datos** > *nombre_origen_datos* > **Propiedades personalizadas** > **Nueva**.

- **Problema** DB2 visualiza una excepción: `la transacción actual se ha retrotraído debido a un punto muerto o tiempo de espera excedido. Código de razón "2".. SQLCODE=-911, SQLSTATE=40001, DRIVER=3.50.152`

  Esta excepción se produce debido a un problema de contención de bloqueo cuando realiza la ejecución con OpenJPA con DB2 en WebSphere Application Server. El nivel de aislamiento predeterminado de WebSphere Application Server es Lectura repetitiva (RR), que obtiene bloqueos de larga duración con DB2.**Solución:**

  Establezca el nivel de aislamiento en Lectura confirmada para reducir la contención de bloqueo. Establezca la propiedad personalizada de origen de datos webSphereDefaultIsolationLevel para establecer el nivel de aislamiento en 2(TRANSACTION_READ_COMMITTED) en la siguiente vía de acceso en la célula de WebSphere Application Server: **Recursos** > **Proveedor JDBC** > *proveedor_JDBC* > **Orígenes de datos** > *nombre_origen_datos* > **Propiedades personalizadas** > **Nueva**. Para obtener más información sobre la propiedad personalizada webSphereDefaultIsolationLevel y los niveles de aislamiento de transacción, consulte Requisitos para establecer los niveles de aislamiento para el acceso a datos.

- **Problema:** al utilizar la función de precarga de JPALoader o JPAEntityLoader, el mensaje CWOBJ1511 siguiente no se visualiza para la partición en un servidor de contenedor: `CWOBJ1511I: GRID_NAME:MAPSET_NAME:PARTITION_ID (primario) está abierto para operaciones empresariales.`

  En su lugar, se produce una excepción TargetNotAvailableException en el servidor de contenedor, que activa la partición especificada por la propiedad preloadPartition.

  Solución: establezca el atributo preloadMode en `true` si utiliza un JPALoader o JPAEntityLoader para precargar los datos en la correlación. Si la propiedad preloadPartition de JPALoader y JPAEntityLoader se establece en un valor entre `0` y `número_total_de_particiones - 1`, JPALoader y JPAEntityLoader intentan precargar los datos de la base de datos de respaldo con la correlación. El fragmento de código siguiente ilustra cómo se establece el atributo preloadMode para habilitar la precarga asíncrona:

  ```
  BackingMap bm = og.defineMap( "map1" );
  bm.setPreloadMode( true );
  ```

  También puede establecer el atributo preloadMode mediante un archivo XML, tal como se muestra en el ejemplo siguiente:

  ```
   <backingMap name="map1" preloadMode="true" pluginCollectionRef="map1"
  lockStrategy="OPTIMISTIC"/>
  ```

Resolución de problemas de configuración de XML

Al configurar eXtreme Scale, puede encontrar un comportamiento inesperado de los archivos XML. Las secciones siguientes describen problemas que se pueden producir y sus soluciones.

Procedimiento

- **Problema:** los archivos XML de ObjectGrid y política de despliegue deben coincidir.

  Los archivos XML de política de despliegue y ObjectGrid deben coincidir. Si no tienen nombres de correlaciones y nombres ObjectGrid coincidentes, se producen errores.

  Si la lista de backingMap del archivo XML de ObjectGrid no coincide con la lista de referencias de correlaciones en un archivo XML de política de despliegue, se produce un error en el servidor de catálogo.

  Por ejemplo, el siguiente archivo XML de ObjectGrid y archivo XML de política de despliegue se utiliza para iniciar un proceso de contenedor. El archivo de política de despliegue tiene más referencias a correlaciones que se listan en el archivo XML de ObjectGrid.

  ObjectGrid.xml - ejemplo incorrecto

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

  deploymentPolicy.xml - ejemplo incorrecto

```
<?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>
```

  Mensajes: se produce un mensaje de error en el archivo `SystemOut.log` cuando la política de despliegue no es compatible con el archivo XML de ObjectGrid. Para el ejemplo anterior, se genera el mensaje siguiente:

```
CWOBJ3179E: La correlación ledger a
la que se hace referencia en el mapSet mapSet1 del archivo de descriptor de despliegue de
ObjectGrid accounting no hace referencia a una correlación de copia de seguridad válida
del XML ObjectGrid.
```

  Si a la política de despliegue le faltan referencias de correlación a backingMaps que se enumeran en el archivo XML de ObjectGrid, se genera un mensaje de error en el archivo `SystemOut.log`. Por ejemplo:

```
CWOBJ3178E: La correlación de ledger de ObjectGrid accounting a la que se hace referencia en el XML de ObjectGrid
no se ha encontrado en el archivo de descriptor de despliegue.
```

  Solución: determine qué archivo tiene la lista correcta y modifique el código relevante según corresponda.

- **Problema:** los nombres de ObjectGrid incorrectos entre archivos XML también causan un error.

  Se hace referencia al nombre del ObjectGrid en el archivo XML de ObjectGrid y en el archivo XML de política de despliegue.

Mensaje: se produce una ObjectGridException debido a una excepción de IncompatibleDeploymentPolicyException. A continuación se muestra un ejemplo.

Causado por:
com.ibm.websphere.objectgrid.IncompatibleDeploymentPolicyException: El objectgridDeployment con objectGridName "accountin" no tiene un objectGrid correspondiente en el XML de ObjectGrid.

El archivo XML de ObjectGrid es la lista maestra de nombres de ObjectGrid. Si una política de despliegue tiene un nombre de ObjectGrid que no está incluido en el archivo XML de ObjectGrid, se produce un error.

Solución: compruebe los detalles como por ejemplo la ortografía del nombre de ObjectGrid. Elimine todos los nombres adicionales, o añada los nombres de ObjectGrid que faltan, a los archivos XML de ObjectGrid o de política de despliegue. En el mensaje de ejemplo, se ha escrito incorrectamente el objectGridName como "accountin", en lugar de "accounting".

- **Problema:** a algunos de los atributos en el archivo XML solo se pueden asignar determinados valores. Estos atributos tienen valores aceptables enumerados por el esquema. La siguiente lista proporciona alguno de los atributos:
  - Atributo authorizationMechanism en el elemento objectGrid
  - Atributo copyMode en el elemento backingMap
  - Atributo lockStrategy en el elemento backingMap
  - Atributo ttlEvictorType en el elemento backingMap
  - Atributo type en el elemento property
  - initialState en el elemento objectGrid
  - evictionTriggers en el elemento backingMap

  Si se asigna un valor no válido a uno de estos atributos, no se supera la validación XML. En el siguiente archivo XML de ejemplo, se utiliza un valor de INVALID_COPY_MODE incorrecto:

  Ejemplo de 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>
  ```

  En el registro aparece este mensaje.

  ```
  CWOBJ2403E: El archivo XML no es válido. Se ha detectado un problema
  con < null > en la línea 5. El mensaje de error es cvc-enumeration-valid:
  El valor 'INVALID_COPY_MODE' no es facet-valid respecto a la enumeración
  '[COPY_ON_READ_AND_COMMIT, COPY_ON_READ, COPY_ON_WRITE, NO_COPY, COPY_TO_BYTES]'.
  Debe ser un valor de la enumeración.
  ```

- **Problema:** la falta de atributos o códigos, o que estos sean incorrectos, en un archivo XML causa errores como el ejemplo siguiente en el que el archivo XML de ObjectGrid falta en el código < /objectGrid > de cierre:

  faltan atributos - XML de ejemplo

  ```
  <?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>
  ```

  Mensaje:

  ```
  CWOBJ2403E: El archivo XML no es válido. Se ha detectado un problema con < null >
  en la línea 7. El mensaje de error es: El código final del
  tipo de elemento "objectGrid" debe terminar con un delimitador '>'.
  ```

Se produce una ObjectGridException sobre el archivo XML no válido con el nombre del archivo XML.

Solución: asegúrese de que todos los atributos y códigos necesarios aparezcan en los archivos XML con el formato correcto.

- **Problema:** si un archivo XML se formatea con sintaxis incorrecta o que falta, aparece el mensaje CWOBJ2403E en el registro. Por ejemplo, se visualiza el mensaje siguiente cuando falta una comilla en uno de los atributos XML.

  ```
  CWOBJ2403E: El archivo XML no es válido. Se ha detectado un problema con < null >
  en la línea 7.
  El mensaje de error es: se espera una comilla abierta para el
  atributo "maxSyncReplicas" asociado a un tipo de elemento "mapSet".
  ```

  También se produce un ObjectGridException acerca del archivo XML no válido.

  Solución: se pueden utilizar diversas soluciones para un error de sintaxis XML determinado. Consulte la documentación correspondiente sobre la scritura del script XML.

- **Problema:** la referencia a una colección de plug-ins inexistente hace que un archivo XML no sea válido. Por ejemplo, cuando se utiliza XML para definir plug-ins BackingMap, el atributo pluginCollectionRef del elemento backingMap debe hacer referencia a una backingMapPluginCollection. El atributo pluginCollectionRef debe coincidir con los elementos backingMapPluginCollection.

  Mensaje:

  Si el atributo pluginCollectionRef no coincide con ningún atributo de ID de ninguno de los elementos backingMapPluginConfiguration, se mostrará en el archivo de registro el siguiente mensaje o uno similar.

  ```
  [7/14/05 14:02:01:971 CDT] 686c060e XmlErrorHandl E CWOBJ9002E:
  Este es un mensaje informativo sólo en inglés: Invalid XML file.
  Line: 14; URI: null; Message: Key 'pluginCollectionRef' with
  value 'bookPlugins' not found for identity constraint of element 'objectGridConfig'.
  ```

  Se utiliza el siguiente archivo XML para producir el error. Observe que el nombre del manual BackingMap tiene su atributo pluginCollectionRef establecido en bookPlugins, y la backingMapPluginCollection única tiene un ID de collection1.

  referencia a un XML de atributo no existente - ejemplo

  ```
  <?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>
  ```

  Solución:

  Para corregir el problema, asegúrese de que el valor de cada pluginCollectionRef coincida con el ID de uno de los elementos backingMapPluginCollection. Simplemente cambie el nombre de pluginCollectionRef por collection1 para no recibir este error. De forma alternativa, cambie el ID de la backingMapPluginCollection existente de modo que coincida con pluginCollectionRef, o añada una backingMapPluginCollection adicional con un ID que coincida con pluginCollectionRef para corregir el error.

- **Problema:** IBM Software Development Kit (SDK) Versión 5 contiene una implementación de alguna función JAXP (Java API for XML Processing) con el

fin de que se utilice para la validación XML en un esquema. Cuando se utiliza un SDK que no contiene esta implementación, los intentos de realizar la validación no serán satisfactorios.

Cuando intente validar XML con un SDK que no tiene la implementación necesaria, el registro contiene el siguiente error:

```
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)...
```

El SDK que se utiliza no contiene una implementación de la función JAXP que es necesaria para validar archivos XML con un esquema.

Solución: si desea validar el XML utilizando un SDK que no contiene una implementación JAXP, descargue Apache Xerces e incluya sus archivos de archivado Java (JAR) en la classpath. Para evitar este problema, después de descargar Xerces e incluir los archivos JAR en la classpath, podrá validar el archivo de XML satisfactoriamente.

Resolución de problemas de la seguridad

Utilice esta información para resolver problemas de la configuración de la seguridad.

Procedimiento

- **Problema:** el cliente final de la conexión requiere SSL (Secure Sockets Layer), con el valor de transportType establecido en `SSL-Required`. Sin embargo, el servidor de la conexión no da soporte a SSL, y tiene el valor transportType establecido en `TCP/IP`. Como resultado, la siguiente excepción se encadena a otra excepción en los archivos de registro:

  ```
  java.net.ConnectException: connect: La dirección no es válida en la máquina local, o
  el puerto no es válido en la máquina remota
      en java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:389)
      en java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:250)
      en java.net.PlainSocketImpl.connect(PlainSocketImpl.java:237)
      en java.net.SocksSocketImpl.connect(SocksSocketImpl.java:385)
      en java.net.Socket.connect(Socket.java:540)
      en com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:155)
      en com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:167)
  ```

  La dirección de esta excepción podría ser un servidor de catálogo, un servidor de contenedor o un cliente.

  Solución: consulte “Configuración de tipos de transporte seguro” en la página 516 para una tabla con las configuraciones de seguridad válidas entre clientes y servidores.
- Cuando se utiliza el agente, el cliente envía la llamada del agente al servidor, y el servidor envía la respuesta de nuevo al cliente para acusar recibo de la llamada del agente. Cuando el agente finaliza el proceso, el servidor inicia una conexión para enviar los resultados de agente. Esto hace del servidor de contenedor un cliente desde el punto de vista de la conexión. Por consiguiente, si se configura TLS/SSL, asegúrese de que el certificado público de cliente se importa en el almacén de confianza de servidor.

IBM Support Assistant para WebSphere eXtreme Scale

Puede utilizar IBM Support Assistant para recopilar los datos, analizar los síntomas y acceder a la información sobre el producto.

IBM Support Assistant Lite

IBM Support Assistant Lite para WebSphere eXtreme Scale proporciona una recopilación automática de los datos y soporte de análisis de síntomas para los casos de determinación de problemas.

IBM Support Assistant Lite reduce el tiempo que lleva reproducir un problema con los niveles de rastreo establecidos correctos de fiabilidad, disponibilidad y capacidad de servicio (la herramienta establece automáticamente los niveles de rastreo) para simplificar la determinación de problemas. Si necesita más asistencia, IBM Support Assistant Lite reduce también el esfuerzo necesario para enviar la información de registro adecuada a IBM Support.

IBM Support Assistant Lite se incluye en todas las instalaciones de WebSphere eXtreme Scale Versión 7.1.0

IBM Support Assistant

IBM® Support Assistant (ISA) proporciona un acceso rápido a los recursos del producto, formación y soporte que pueden ayudarle a contestar las preguntas y a resolver los problemas con los productos de software de IBM por sí solo, sin necesidad de ponerse en contacto con IBM Support. Distintos plug-ins específicos del producto le permiten personalizar IBM Support Assistant para los productos concretos que ha instalado. IBM Support Assistant recopila además los datos del sistema, los archivos de registro y otra información para ayudar a IBM Support a determinar la causa de un problema concreto.

IBM Support Assistant es un programa de utilidad para instalarlo en la estación de trabajo, no directamente en el sistema servidor WebSphere eXtreme Scale en sí. Los requisitos de memoria y de recursos para Assistant podrían afectar negativamente al rendimiento del sistema servidor WebSphere eXtreme Scale. Los componentes de diagnóstico portátiles incluidos están diseñados para un impacto mínimo en la operación normal de un servidor.

Puede utilizar IBM Support Assistant para que le ayude de estos modos:

- Para buscar en las fuentes de información y de conocimientos de IBM y no IBM entre varios productos de IBM para contestar una pregunta o solucionar un problema
- Para encontrar información adicional en los recursos web específicos del producto; incluidas las páginas iniciales del producto y de soporte, los foros y los grupos de noticias de clientes, las capacidades y los recursos de formación y la información sobre resolución de problemas y preguntas más frecuentes
- Para ampliar la capacidad para diagnosticar los problemas específicos del producto con herramientas de diagnóstico orientadas disponibles en Support Assistant
- Para simplificar la recopilación de datos de diagnóstico para ayudarle a usted y a IBM a resolver los problemas (recopilando datos generales o específicos del síntoma o producto)

- Para ayudarle a informar de las incidencias de problemas a IBM Support mediante una interfaz en línea personalizada para adjuntar los datos de diagnóstico mencionados anteriormente o cualquier otra información a las incidencias nuevas o existentes.

Finalmente, puede utilizar el recurso actualizador incorporado para obtener soporte de los productos y las capacidades de software adicionales a medida que están disponibles. Para configurar IBM Support Assistant para utilizarlo con WebSphere eXtreme Scale, instale en primer lugar IBM Support Assistant con los archivos proporcionados en la imagen descargada de la página web Visión general de soporte de IBM en: http://www-947.ibm.com/support/entry/portal/Overview/Software/Other_Software/IBM_Support_Assistant. A continuación, utilice IBM Support Assistant para ubicar e instalar las actualizaciones del producto. Puede elegir también instalar los plug-ins disponibles para otro software de IBM en el entorno. Hay disponible más información y la última versión de IBM Support Assistant desde la página web de IBM Support Assistant en la dirección: http://www.ibm.com/software/support/isa/.

Avisos

Las referencias en esta publicación a productos, programas o servicios de IBM no implica que IBM tenga previsto ponerlos a la venta en todos los países en los que IBM opera. Cualquier referencia a un producto, programa o servicio de IBM no pretende indicar ni implica que sólo se pueda utilizar este producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ningún derecho de propiedad intelectual de IBM. La evaluación y la verificación del funcionamiento con otros productos, excepto aquellos expresamente designados por IBM, es responsabilidad del usuario.

IBM puede tener patentes o solicitudes de patentes pendientes que conciernan al tema de este documento. La posesión de este documento no le da ninguna licencia sobre estas patentes. Puede enviar preguntas acerca de licencias por escrito a:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594 Estados Unidos

Los propietarios de licencias de este programa que deseen obtener información sobre el mismo con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, se deben poner en contacto con:

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
Estados Unidos
Attention: Information Requests

Esta información puede estar disponible, bajo las condiciones y los términos adecuados, incluyendo en algunos casos, el pago de una cuota.

Marcas registradas

Los siguientes términos son marcas registradas de IBM Corporation en Estados Unidos yen otros países.

- AIX
- CICS
- Cloudscape
- DB2
- Domino
- IBM
- Lotus
- RACF
- Redbooks
- Tivoli
- WebSphere
- z/OS

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en Estados Unidos y/o en otros países.

LINUX es una marca registrada de Linus Torvalds en Estados Unidos y/o en otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en Estados Unidos y/o en otros países.

UNIX es una marca registrada de The Open Group en Estados Unidos y en otros países.

Otros nombres de compañías, productos y servicios pueden ser marcas registradas o de servicio de terceros.

Índice

H

I

J

L

M

O

P

Q

R

S

T

V

W

X

Z

Impreso en España