IBM Software Group

IBM WebSphere eXtreme Scale V7.0

HTTP session management

@business on demand.

© 2009 IBM Corporation
Updated July 6, 2009

This presentation will cover WebSphere® eXtreme Scale V7.0 HTTP session management.

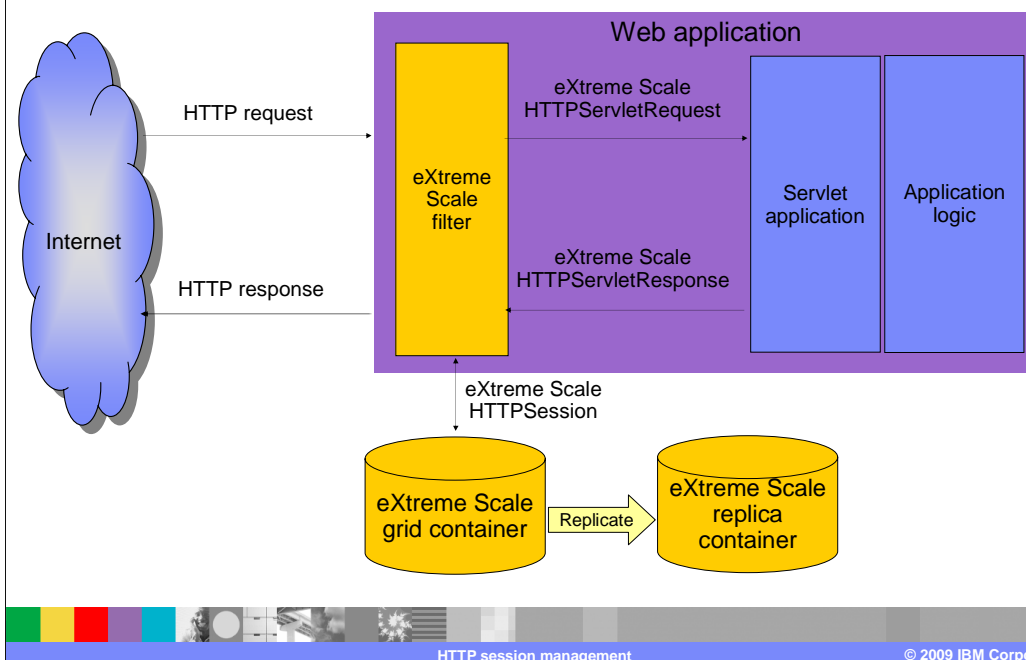# HTTP session persistence

- eXtreme Scale servlet filter to manage HTTP sessions
  - ▸ Upgrade existing session persistence mechanism
  - ▸ Can be inserted into any Web application
  - ▸ Enhanced session persistence
  - ▸ Advanced session replication
    - HTTP sessions can be replicated across servers
- Independent of the WebSphere cell infrastructure
  - ▸ Older versions of WebSphere products
  - ▸ Other servers
    - Geronimo, JBoss, Other J2EE 1.4 or higher

2

HTTP session management   © 2009 IBM Corporation

WebSphere eXtreme Scale can be used to manage your HTTP session data. You can override the default session manager in the base application server to provide HTTP session management capabilities for an associated application. The eXtreme Scale session manager will fully manage the life cycle of HTTP sessions that belong to the application.

The eXtreme Scale session manager is compatible with any J2EE 1.4 or higher servlet container, so it can support older versions of WebSphere Application Server and other vendors' server environments.

eXtreme Scale HTTP session configuration

IBM Software Group

Web application

HTTP request

eXtreme Scale HTTPServletRequest

eXtreme Scale filter

Servlet application

Application logic

HTTP response

eXtreme Scale HTTPServletResponse

Internet

eXtreme Scale HTTPSession

eXtreme Scale grid container

Replicate

eXtreme Scale replica container

This diagram summarizes the operation of the eXtreme Scale session filter. This filter intercepts every request to the Web application. Before passing control to the application servlet or JSP, it wraps the HTTPServletRequest and HTTPServletResponse objects that the application uses to access the request's session state in an eXtreme Scale HTTPSession object. The eXtreme Scale HTTPSession object overrides the default session manager's implementation, so there is no data duplication between the two session managers. The eXtreme Scale session filter normally stores session state in a grid container in the local process. This provides low latency read and write operations when running with asynchronous replication. The filter will ensure that the session data is persisted to the grid and the grid will ensure the data remains highly available.

WXS70_HTTP_Session_Management.ppt

HTTP session replication

- **Session located in the JVM that creates it**
  - ▸ Persisted to eXtreme Scale container

- **Session changes replicated to backup partition**
  - ▸ A shard in a separate process

- **Automatic failover**
  - ▸ Session data does not exist locally
    - State is automatically retrieved from remote servers
  - ▸ This can happen when servers fail

- **Replication zones**
  - ▸ Hints to ensure replicas are

4

HTTP session objects in the eXtreme Scale grid can be replicated for high availability of session data. The primary data location is the Java™ Virtual Machine that originally created the session object. As the application makes changes to the session data, the grid replicates the changes to one or more shards located in separate server processes. The failure of one server in a cluster will not affect the session. If a session object is not found locally, perhaps due to a loss of affinity, the grid will automatically retrieve the session state from a designated backup server. This prevents the loss of session data, and provides for high-speed recovery in the event of a server failure

Configuration

- splicer.properties file
  - ▸ Input to splicer utility
  - ▸ Provides ObjectGrid configuration details
- "Splice" HTTP servlet filter into a Web application
  - ▸ Run script
    ```
    "addObjectGridFilter.[bat|sh]
        <location of ear/war file>
        <location of splicer.properties file>"
    ```
  - ▸ Modifies WEB-INF/web.xml
    - Part of application
    - Adds initialization parameters to tune HTTP session management
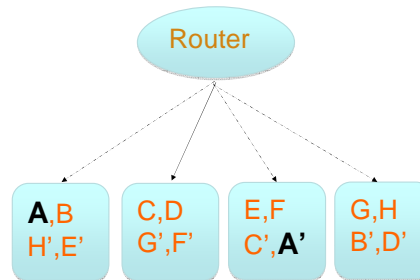- Deploy application

HTTP session management

© 2009 IBM Corporation

To activate the HTTP servlet filter provided by WebSphere eXtreme Scale you must "splice" the filter into your application. eXtreme Scale includes a script, addObjectGridFilter, to add the filter to your application. The script takes two parameters: the absolute path to the application to be spliced, and the location of a file that contains various configuration properties. The splicer utility modifies the application's deployment descriptor, web.xml, with additional parameters which control how the filter interacts with the WebSphere eXtreme Scale grid.

Follow the normal procedure to deploy the new application to your application server. The servlet filter will be started as part of the application initialization.
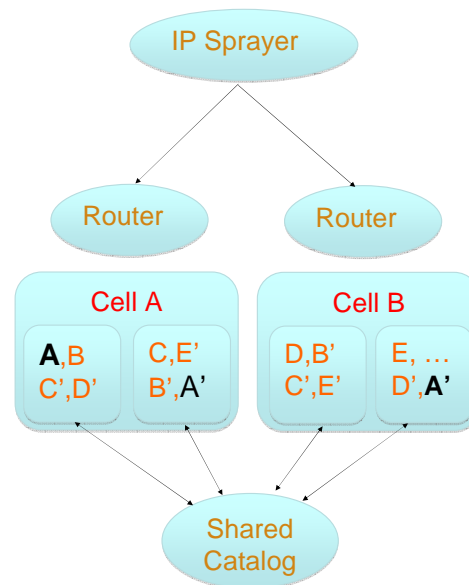
HTTP session topology

- Router sends new requests to any server

- Server persists session to a local grid instance
  - ▶ Session cookie for router affinity

- Session changes are replicated
  - ▶ Plug-in replicates to local primary partition
  - ▶ Grid replicates to remote backups

- Failover to any server
  - ▶ Will retrieve session from replica partition
  - ▶ One additional remote procedure call using normal routers

Router

| A,B H',E' | C,D G',F' | E,F C',A' | G,H B',D' |

6

HTTP session management   © 2009 IBM Corporation

A router, such as the IBM HTTP Server or On Demand Router, will route new requests to an available server. A "new" request is one that is not already associated with a session. The eXtreme Scale servlet filter will create a session object in a local grid instance, and associate that session instance with the session ID. The router will use the session ID to ensure subsequent requests for the session are sent to the same server instance. When session affinity is used, then the ObjectGrid session can be accessed directly in memory with no network cost. This allows very large sessions to be used efficiently.

Multi-cell HTTP session topology

- Multiple cells share a single catalog server

- Shared catalog can be:
  - ▸ Its own cell
  - ▸ Running in both deployment managers
  - ▸ Running in stand-alone JVMs

- Deploy the application to each cell

- Zone rules can be used to influence placement

IP Sprayer

Router   Router

Cell A   Cell B

| **A**,B C',D' | C,E' B',A' | | D,B' C',E' | E, … D',**A'** |

Shared Catalog

eXtreme Scale can also act as a mechanism to persist HTTP session objects across multiple WebSphere cells. You use a shared catalog to distribute primary and replica shards, and an IP sprayer to route requests between cells. Additionally, zone rules can be used to influence placement for the required availability. This diagram shows how zones can be used to place at least one backup for a session in a different cell from the primary. If Cell 'A' becomes unavailable, all of it's sessions can fail-over to Cell 'B'.

Summary

- WebSphere eXtreme Scale can act as a mechanism to persist HTTP session objects
  - ▸ Implemented as a simple servlet filter

- Allows for session persistence across an eXtreme Scale grid
  - ▸ Session objects held in cheap JVM memory
  - ▸ Session changes are replicated to ensure fault tolerance and high availability

- When session affinity is used, session data can be accessed directly in memory

8

© 2009 IBM Corporation

WebSphere eXtreme Scale contains a servlet session filter that can be used to persist and host your HTTP session data in the grid. Session changes are replicated to other application servers in the grid to ensure fault tolerance and high availability. When session affinity is used, the session data can be accessed directly in memory with no network cost.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

J2EE, Java, JSP, JVM, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.