

ibm.com



e-business



Best Practice for WebSphere Performance on z/OS HTTP Sessions



Redbooks

International Technical Support Organization

© Copyright IBM Corp. 2003. All rights reserved.

Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Objectives

- HTTP Sessions Overview
- Single Server Session
- Multiple Server Session
- Session Affinity
- WebSphere AE Plugin
- Sessions in DB2 Performance
- Summary



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

HTTP Session Tracking

- Used to maintain a 'session' state in a stateless connection (http)
- Session id is stored together with an session object on the server
- Session is managed with cookie or a modified URL (rewritten)
- FormBasedLogin (LoginToken) requires HTTP sessions
- JSP's create HTTP sessions on default
- Session object has to be externalized and made persistent
 - When multiple servers are used and no session affinity is available or enabled
 - and Application needs high availability for the HTTP sessions
- Persistent session tracking uses DB2
- Session affinity performs better than persistent session in DB2



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

HTTP Sessions and WebSphere for z/OS

- Fully supports the HTTP Session state semantic proposed by the Java Servlet Specification V2.2
- Allows multiple requests in the same session to execute concurrently within the same and among multiple Application Server instances
- Makes use of a DB2 database as the mechanism for serializing access to and sharing HTTP Session State data
- Supports maintaining HTTP session state data in-memory, all request inside a Session has to end in the Server Region where the session was created (Session Affinity)
- Session API defined in the HttpSession interface
 - setAttribute(), getAttribute(), and others
 - See the javax.servlet.http package at <http://java.sun.com/j2ee/j2sdkee/techdocs/api/index.html>
 - getSession() is part of the HttpSession interface
- See also Java Servlet Specification, V2.2
 - <http://java.sun.com/products/servlet/download.html>



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session `MaxInactiveInterval` from JSPs

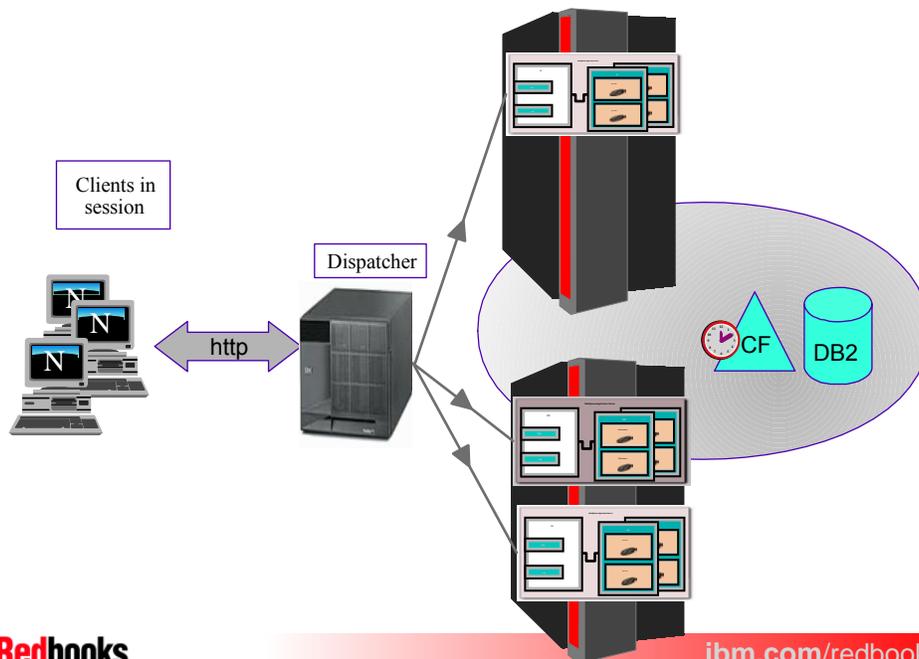
```
<HTML>
<HEAD>
<TITLE>Session MaxInactiveInterval Test</TITLE>
</HEAD>
<BODY>
<H3>Test Buffering and Commit</H3>
<HR>
<%! static int junk = 101; %>
<% out.println("In Jsp");
// session.setMaxInactiveInterval(0);
session.invalidate();
System.out.println("Is the page comitted to the client? +
                   <BR>Ans: "                +
                   response.isCommitted());
%>
<BR>Accessing static variable junk using the implicit page object
<BR>
</BODY>
</HTML>
```



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

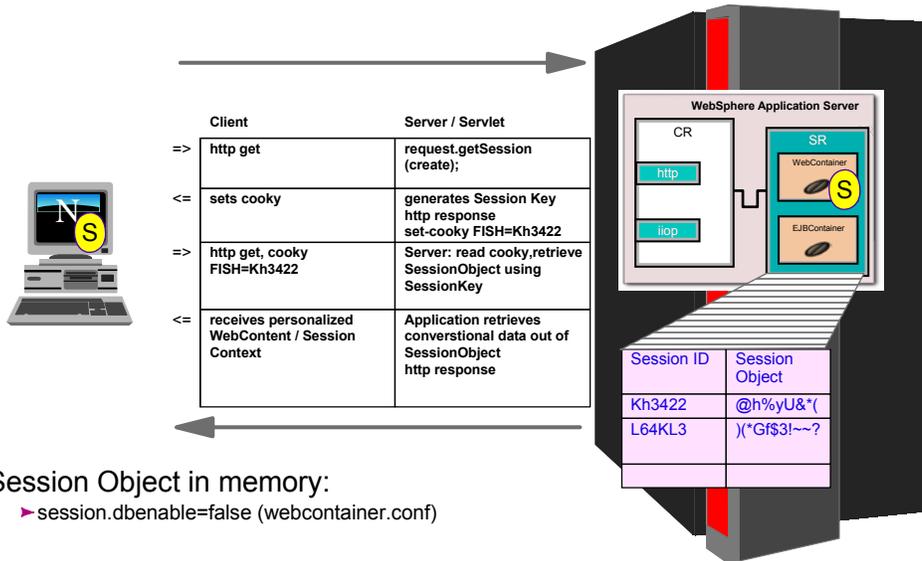
HTTP Sessions on WAS for z/OS



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

HTTP Session Tracking with single SR

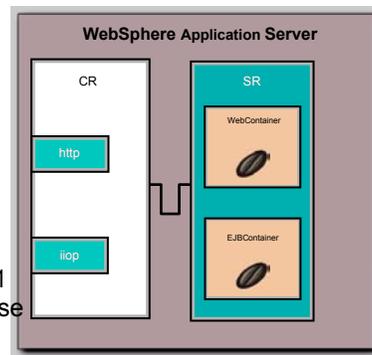


ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Limiting Server Regions

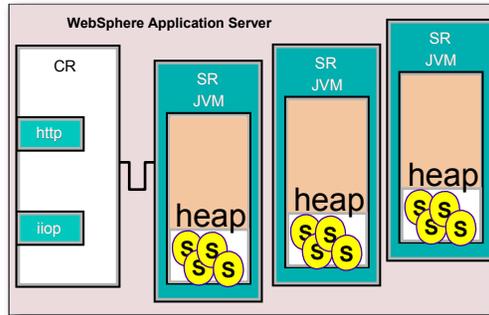
- Single Server Options:
 - ▶ IHS standalone + WebSphere 3.5 WebContainer loaded
 - ▶ IHS standalone + WebSphere 4.0 plugin configured as WebContainer
 - ▶ WebSphere 4/5 runtime with limitation on Server Regions
 - ◆ The use of MAX_SRS=1 /wlm_maximumSRCcount=1 **may interfere** with WLM processing which may cause application to experience difficulties and requires limiting total number WLM service classes
 - ◆ WLM application environment limit



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Objects in Memory



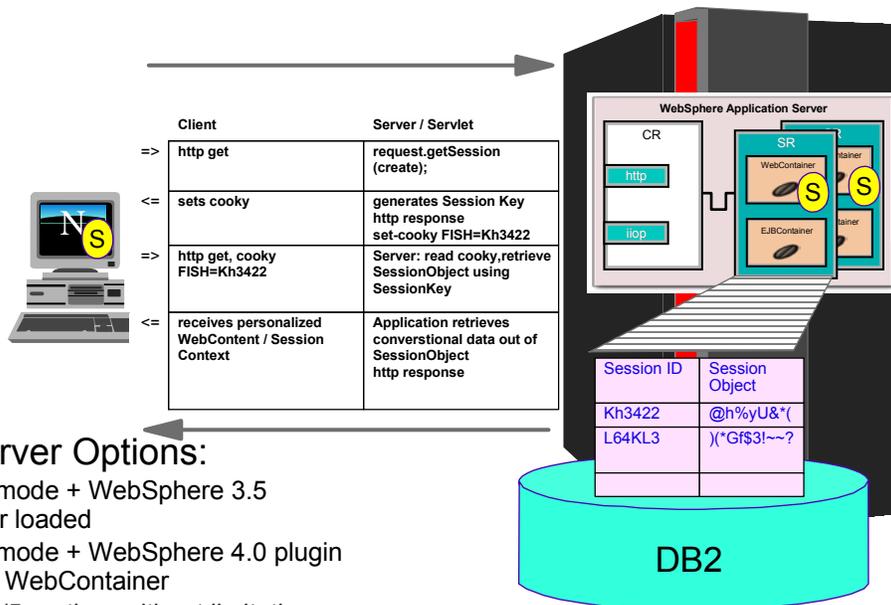
- Session in Memory are using JVM HEAP
 - ▶ Have multiple SRs to distribute session objects
 - ▶ Large Objects will cause GC
 - ▶ control sessions memory usage (and distribution) using
 - ◆ session.invalidationtime=<milliseconds> (server side)
 - ◆ session.cookie.maxage=<milliseconds> (client side)
 - ◆ session.tableoverflowenable=false
 - ◆ session.tablesize=<# session objects>
 - ▶ Allow many SRs if you have large session objects
 - ▶ Multiple SRs and Sessions in memory needs session affinity



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

HTTP Session Tracking with multiple SRs



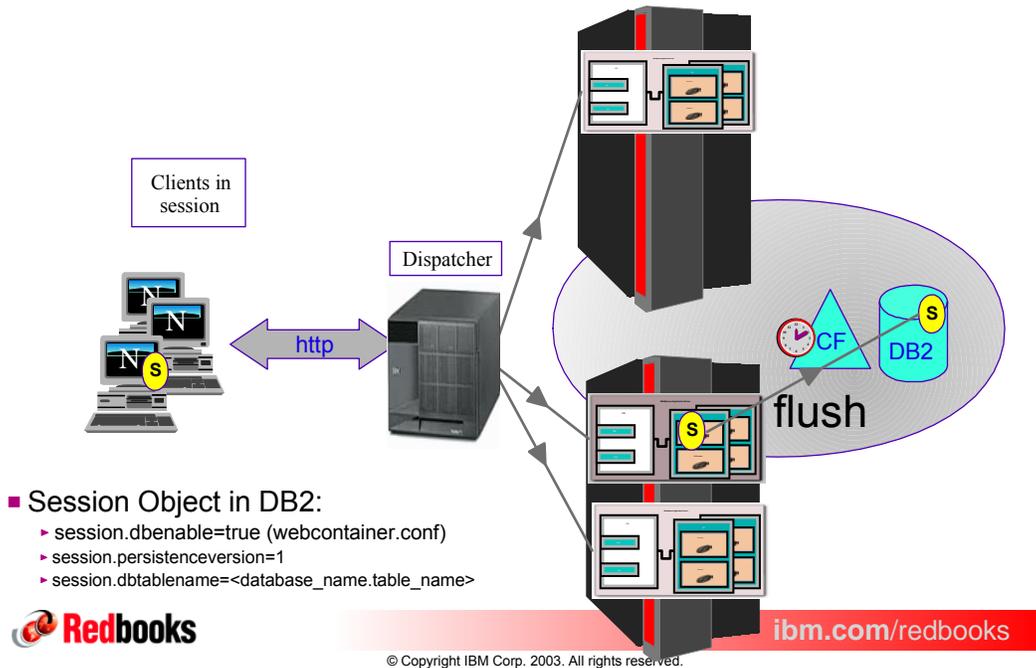
- Multiple Server Options:
 - ▶ IHS scalable mode + WebSphere 3.5 WebContainer loaded
 - ▶ IHS scalable mode + WebSphere 4.0 plugin configured as WebContainer
 - ▶ WebSphere 4/5 runtime without limitation on Server Regions



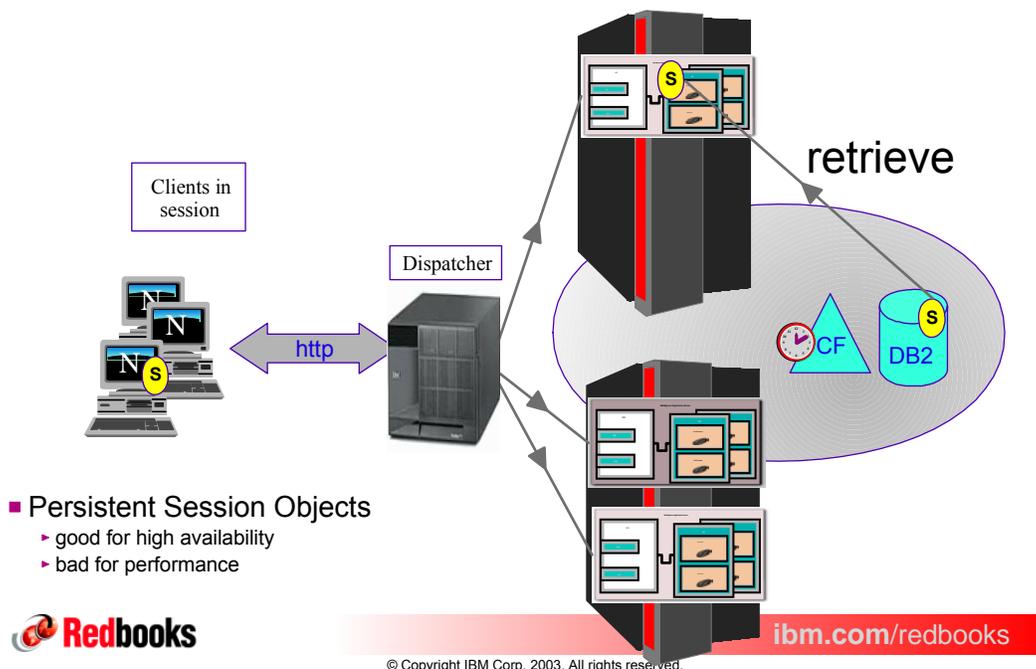
ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

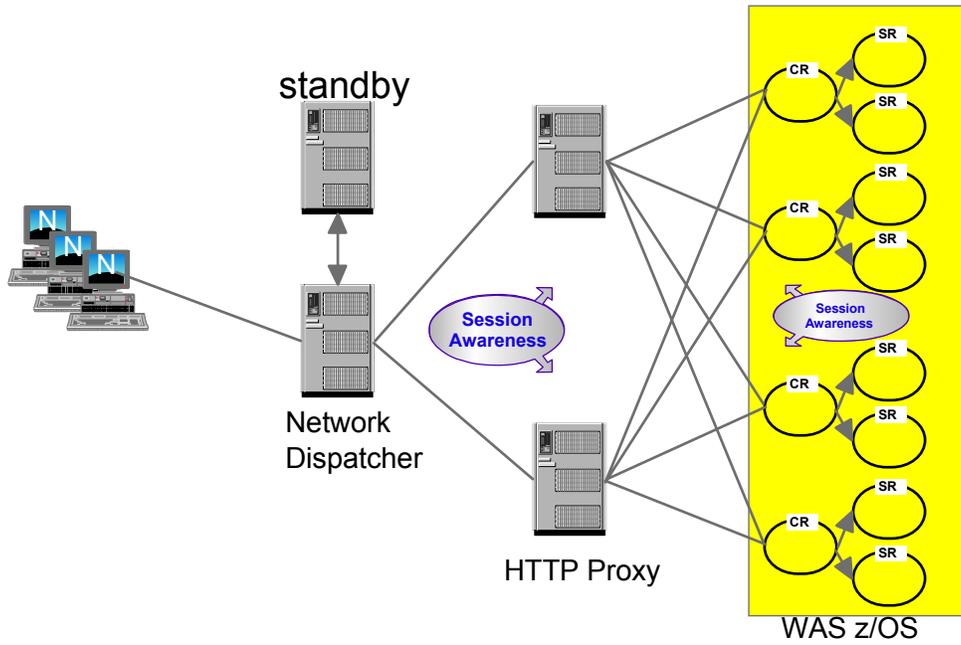
Sharing HTTP Sessions (1)



Sharing HTTP Sessions (2)



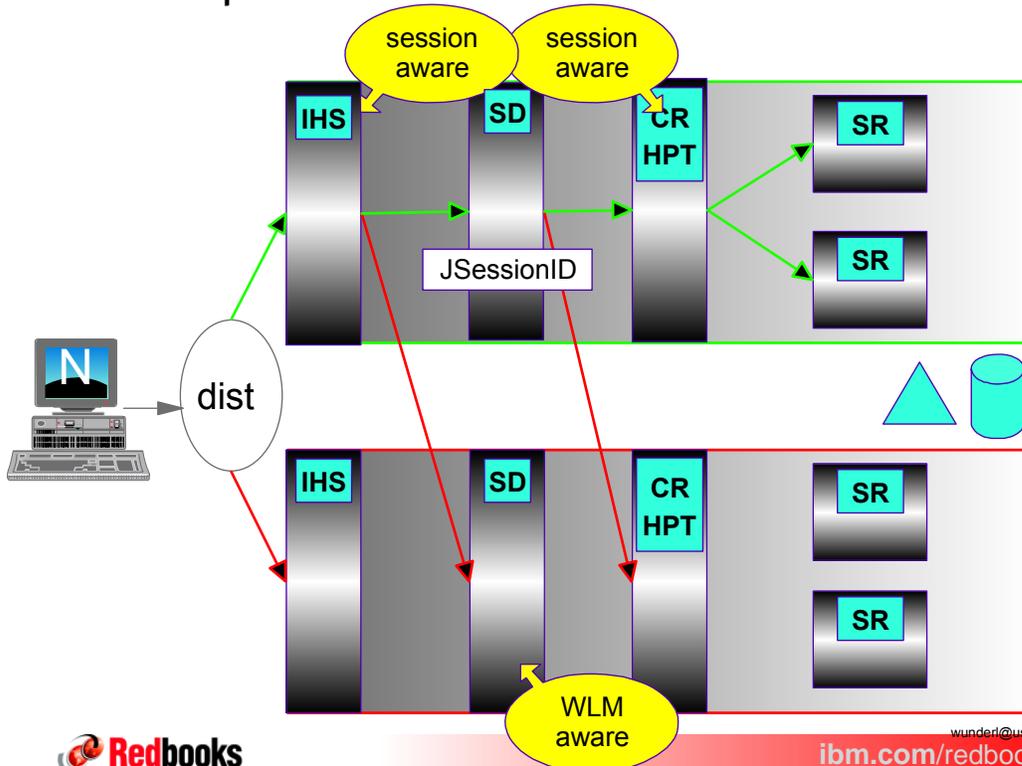
Session Affinity



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

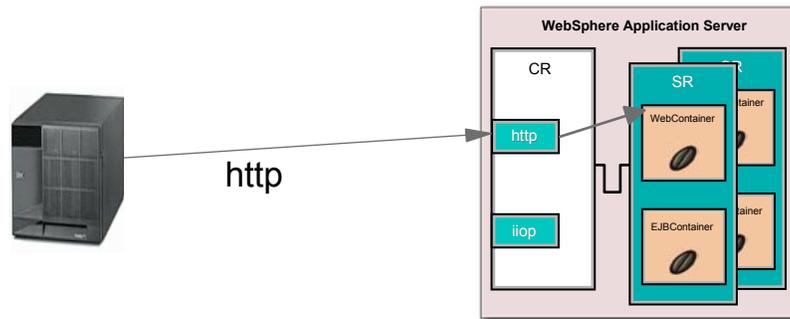
One step further



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Affinity to a specific Server Region inside a J2EE server instance (1)



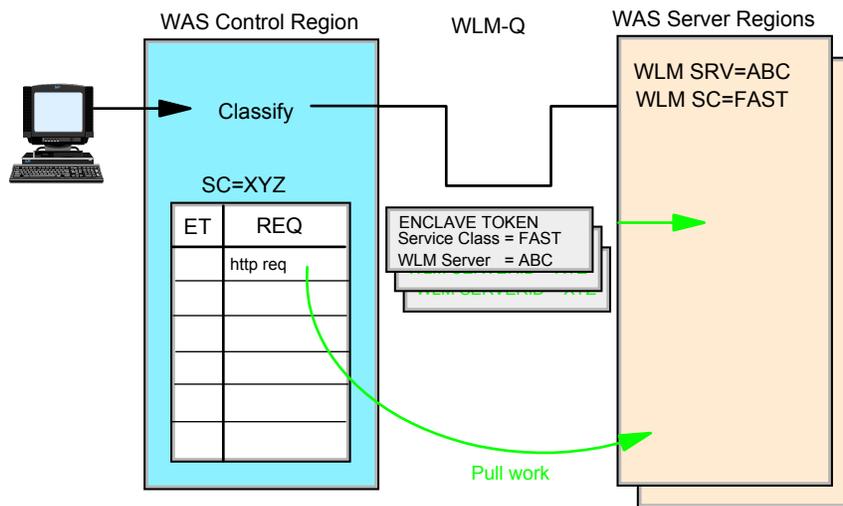
WLM server ID



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WLM Adress Space Management



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Affinity to a specific Server Region inside a server instance (2)

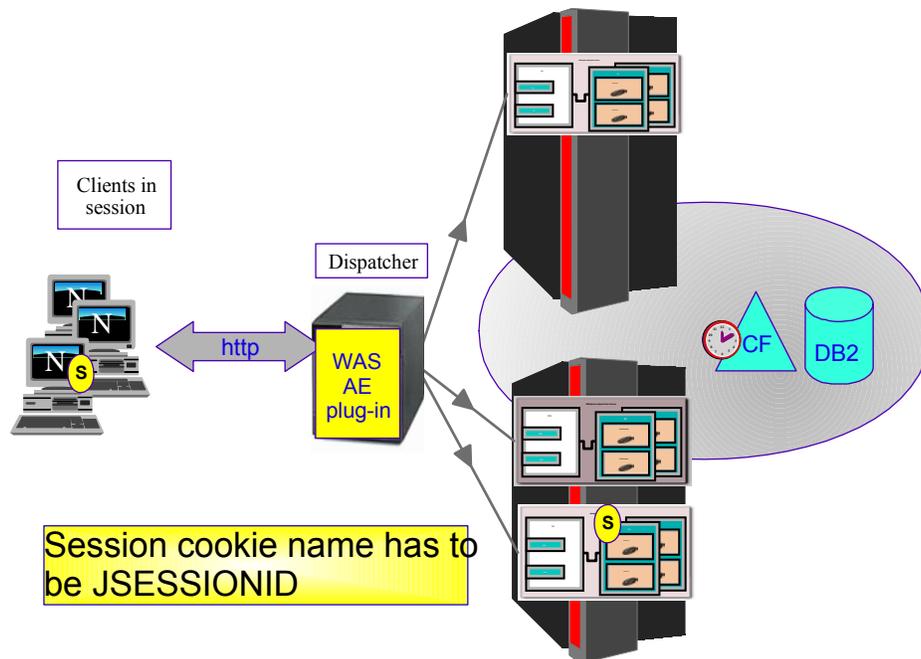
- HTTP Session Affinity in the WAS 4 Server Region
 - Session-in-memory is **only** possible when using cookies and the **HTTP Transport Handler**
 - Within a J2EE server instance, the CR can route requests for a specific HTTP session **back** to the server region maintaining the data for that session
 - The application can be executed within a J2EE server instance for which multiple server regions have been defined even if HTTP session data is being maintained in-memory
- Session Affinity is not possible when using WebContainers in a scaleable IHS
- Session Affinity is not possible when using the IHS and the WAS plugin to locate WebApplications
- One cannot recycle server regions
- NO PORTSHARING!



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WAS Session Affinity with multiple J2EE Server instances (1)



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WAS Session Affinity with multiple J2EE Server instances (2)

- WebSphere AE Web server plugin needed for session affinity, support will come soon
- Supported with WAS for z/OS 4.0.1, delivered as a PTF
- Session affinity based on CloneID (server instance name)
 - ▶ CloneID = <generic_server_name.sever_instance_name>
 - ▶ Generated by WebSphere and put into session cookie
- ▶ A Plugin that will run in the z/OS HTTP Server is planned for shipment with the W401409 maintenance release.

Webserver	Server Operating System				
	Win2K/NT	AIX	HP-UX	Sun Solaris	Linux
IBM HTTP Server (IHS)	✓	✓	✓	✓	✓
Lotus Domino	✓	✓	✓	✓	✓
Apache	✓	✓	✓	✓	✓
iPlanet (Netscape)	✓	✓	✓	✓	✓
Microsoft Internet Info. Server (IIS)	✓				

The various web servers and operating systems on which the affinity plugin runs

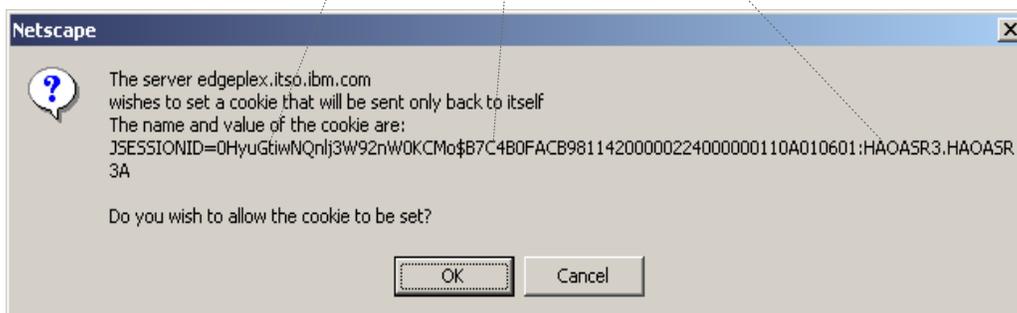


ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

JSESSIONID session cookie

JSESSIONID=sessionID\$wlm\$vrID:cloneID



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Web server plug-in configuration

- Configuration file (plugin-cfg.xml) is now generated by WebSphere

Update web server plugin configuration

The web server plugin configuration file controls what content is transferred from the web server to an application server. This file must be regenerated when server, cluster, HTTP transport, or virtual host alias configurations are changed. The generated plugin-cfg.xml file is placed in the config directory of the WebSphere installation. If your web server is located on a remote machine, you must manually move this file to that machine.

Click the OK button to update the plugin configuration file.

[View or download the current web server plugin configuration file](#)

- Web Server starts plug-in based on httpd.conf configuration (done automatically during plugin installation)

```
LoadModule ibm_app_server_http_module
```

```
C:/WebSphere/AppServer/bin/mod_ibm_app_server_http.dll
```

- httpd.conf points to the plugin-cfg.xml file

```
WebSpherePluginConfig C:\WebSphere\AppServer\config\plugin-cfg.xml
```



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WAS 501 plug-in files

Directory List

Select one or more files with / or action codes. If / is used also select an action from the action bar otherwise your default action will be used. Select with S to use your default action. Cursor select can also be used for quick navigation. See help for details.

```
EUID=0 /WebSphere/BS0F/appserver/config/cells/
Type Perm Changed-EST5EDT Owner -----Size Filename Row 1 of 5
_ Dir 770 2003-10-07 13:51 WDSFSTU 8192 .
_ Dir 770 2003-10-07 13:53 WDSFSTU 8192 ..
_ Dir 770 2003-10-01 19:53 WDSFSTU 8192 cdfsc59
_ File 660 2003-10-07 13:53 WDCFSTU 1843 plugin-cfg.xml
_ File 660 2003-10-07 13:53 WDCFSTU 1847 plugin-cfg-ascii.xml
```



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

plugin-cfg.xml (1)

Warning: this config was done with a non GA plug in

```
<!-- LogLevel can be Trace, Warn, or Error. generates lots of output. -->
<Log LogLevel="Trace"

<VirtualHostGroup Name="default_host">
  <VirtualHost Name="*:80"/>
  <VirtualHost Name="*:9080"/>
</VirtualHostGroup>

<ServerGroup Name="HAplex">
  <ClusterAddress name="haplex1">
    <Transport hostname="202.5.10.10" port="8080"
      protocol="http"/>
  </ClusterAddress>

  <Server CloneID="HAOASR2.HAOASR2A" Name="SC43">
    <Transport Hostname="10.1.6.1" Port="8080" Protocol="http"/>
  </Server>

  <Server CloneID="HAOASR2.HAOASR2B" Name="SC50">
    <Transport Hostname="10.1.6.3" Port="8080" Protocol="http"/>
  </Server>

  <Server CloneID="HAOASR2.HAOASR2C" Name="SC52">
    <Transport Hostname="10.1.6.4" Port="8080" Protocol="http"/>
  </Server>
</ServerGroup>
```

Sysplex
Distributor

WS Server
Name

WS Instance



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

plugin-cfg.xml (2)

```
<VirtualHostGroup Name="default_host">
  <VirtualHost Name="edgeplex.itso.ibm.com"/>
</VirtualHostGroup>

<UriGroup Name="default_host_URIs">
  <Uri Name="/PolicyIVP/*"/>
  <Uri Name="/servlet/*"/>
  <Uri Name="/webapp/examples/*"/>
  <Uri Name="*.jsp"/>
  <Uri Name="*.jsv"/>
  <Uri Name="*.jsw"/>
  <Uri Name="/WebSphereSamples/*"/>
  <Uri Name="/estore/*"/>
</UriGroup>

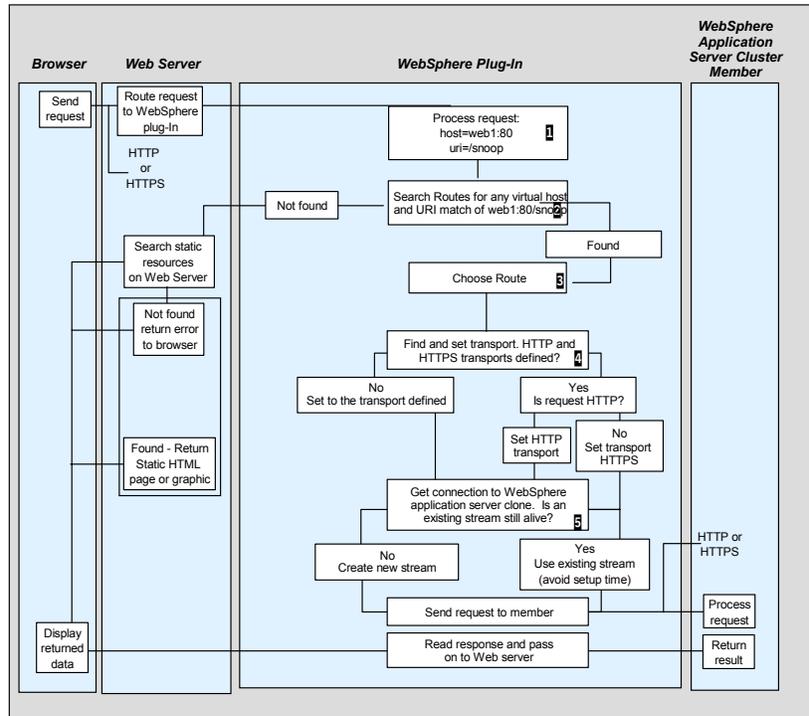
<Route ServerGroup="HAplex" UriGroup="default_host_URIs"
  VirtualHostGroup="default_host"/>
```



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

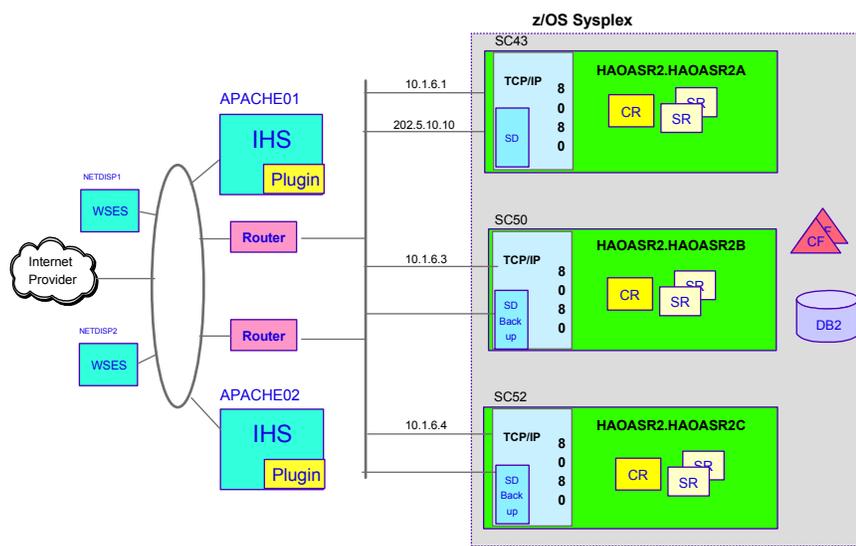
plug-in session handling detail



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

High Available WebSphere Implementation



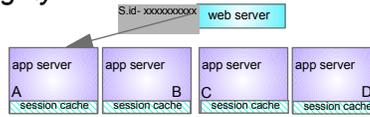
ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

data replication services

1. low-cost, high perf, low integrity

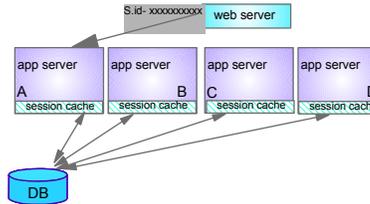
- Affinity established on first access
- Local Access to Cached Session Data



- ✓ High Performance
- ✓ Low cost of ownership (no-DB)
- ✗ No Failover

2. higher-cost, high perf, high integrity

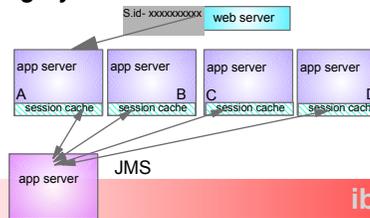
- Affinity (from first access)
- Write through cache
- DB Robustness
- Programmatic Updates
- Immediate or Time-based Updates
- Time-based Invalidation



- ✓ Robust
- ✓ Full Cluster Failover
- ✓ Full App Server CPU Utilization
- ✓ Fully symmetric design for failover
- ✓ Impact of Writes to DB have been minimized (sync)
- ✗ Cost of ownership
- ✗ Cost to avoid DB-SPOF

3. low-cost, high perf, high integrity

- Same as above
- JMS to AppServer running WebSphere Internal Messaging for Persistence



New in V5.0

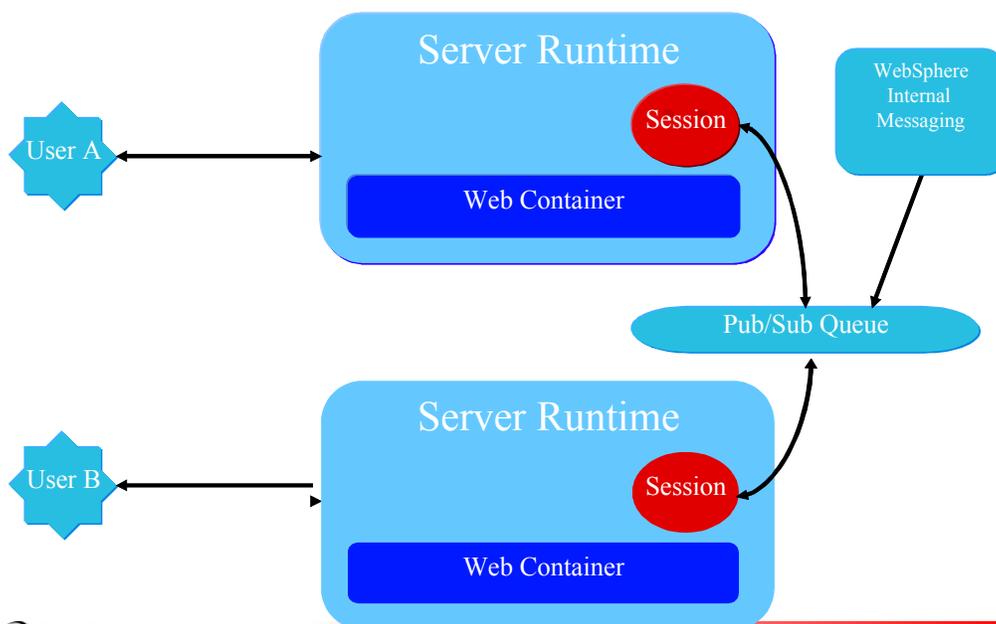
- ✓ Same as above
- ✓ Lower cost of ownership (DB vs App Server)
- ✓ Multiple Message Brokers to avoid SPOF



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Propagation



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Persistence update

- Main performance improvements based on caching DB2 data and relying on affinity to access cache
- Session server region affinity is a prereq
- Database is no longer used to share the data across server regions, but is used as a backup in the event of failover, and as an overflow repository if the cache is full.
- New session persistence implementation requires new DB2 tables and requires using the HTTP Transport (for affinity).
- New session persistence code is known as Persistence Version 2; Old code is Persistence Version 1 and is the default.
- This is set in webcontainer.conf session.persistenceversion property for WAS 4 and is default behaviour in WAS 5
- Session in memory always runs new code, but there are no known external behavior differences in the new implementation so customers won't know the difference.



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WAS5 settings

Session Management

Session manager configuration properties allow you to control the behavior of HTTP session support [1]

Configuration

General Properties	
Session tracking mechanism:	<input type="checkbox"/> Enable SSL ID tracking <input checked="" type="checkbox"/> Enable Cookies <input type="checkbox"/> Enable URL Rewriting <input type="checkbox"/> Enable protocol switch rewriting
Maximum in-memory session count:	<input type="text" value="1000"/> sessions <small>[1] Specifies the maximum number of sessions to maintain in memory.</small>
Overflow:	<input checked="" type="checkbox"/> Allow overflow
Session timeout:	<input type="radio"/> No timeout <input checked="" type="radio"/> Set timeout <input type="text" value="30"/> minutes
Security integration	<input type="checkbox"/> Enable
Serialize session access:	<input type="checkbox"/> Allow serial access Maximum wait time: <input type="text" value="5"/> sec <input checked="" type="checkbox"/> Allow access on timeout

[Application Servers > wdfndfcdsc59 > Web Container > Session Management > Distributed Environment Settings](#)

Distributed Environment Settings

Specify a type for saving session in a distributed environment. [1]

Configuration

General Properties	
Distributed Sessions	<input checked="" type="radio"/> None <input type="radio"/> Database
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

Additional Properties

[Custom Tuning Parameters](#) Specify tuning parameters in a distributed environment



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WAS5 settings

[Application Servers](#) > [wdfndfcdsc59](#) > [Web Container](#) > [Session Management](#) > [Distributed Environment Settings](#) >

Tuning Parameters

Session Manager provides various tuning options for managing session data in a distributed environment. [i](#)

Configuration		Custom Tuning Parameters	
General Properties Tuning level: <ul style="list-style-type: none"> <input type="radio"/> Very high (optimize for performance) Write frequency: Time based: 300 seconds Write contents: Only updated attributes Schedule sessions cleanup: true <input type="radio"/> High Write frequency: Time based: 300 seconds Write contents: All session attributes Schedule sessions cleanup: false <input type="radio"/> Medium Write frequency: End of servlet service Write contents: Only updated attributes Schedule sessions cleanup: false <input type="radio"/> Low (optimize for failover) Write frequency: End of servlet service Write contents: All session attributes Schedule sessions cleanup: false <input checked="" type="radio"/> Custom settings Write frequency: Time based: 10 seconds Write contents: Only updated attributes Schedule sessions cleanup: false 		Configuration Tuning parameters for session management. i	
		General Properties Write frequency: <ul style="list-style-type: none"> <input type="radio"/> End of servlet service <input type="radio"/> Manual update <input checked="" type="radio"/> Time based: <input type="text" value="10"/> seconds 	
		Write contents: <ul style="list-style-type: none"> <input checked="" type="radio"/> Only updated attributes <input type="radio"/> All session attributes 	
		Schedule sessions cleanup: <ul style="list-style-type: none"> <input type="checkbox"/> Specify distributed sessions cleanup schedule First time of day (0-23): <input type="text"/> Second time of day (0-23): <input type="text"/> 	
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>			



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

WAS5 settings

Session Management

Session manager configuration properties allow you to control the behavior of HTTP session support [i](#)

Configuration		
General Properties		
Session tracking mechanism:	<input type="checkbox"/> Enable SSL ID tracking <input checked="" type="checkbox"/> Enable Cookies <input type="checkbox"/> Enable URL Rewriting <input type="checkbox"/> Enable protocol switch rewriting	i Specify a mechanism for HTTP session management.
Maximum in-memory session count:	<input type="text" value="1000"/> sessions	i Specifies the maximum number of sessions to maintain in memory.
Overflow:	<input checked="" type="checkbox"/> Allow overflow	i Whether to allow the number of sessions in memory to exceed the value specified by Max In Memory Session Count property. This is valid only in non-persistent sessions mode.
Session timeout:	<input type="radio"/> No timeout <input checked="" type="radio"/> Set timeout <input type="text" value="30"/> minutes	i Specifies how long a session is allowed to go unused before it will be considered valid no longer. Specify either "Set timeout" or "No timeout." If you select to set the timeout, the value must be at least two minutes, specified in minutes.
Security integration	<input type="checkbox"/> Enable	i When security integration is enabled, the Session Manager will associate the identity of users with their HTTP sessions.
Serialize session access:	<input type="checkbox"/> Allow serial access Maximum wait time: <input type="text" value="5"/> seconds <input checked="" type="checkbox"/> Allow access on timeout	i Serialize session access indicates whether to disallow concurrent session access in a given server (JVM).



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Recovery / WAS 4

session.persistanceversion=2

- **session.reaperinterval=<integer>**
 - ▶ Specifies the interval, in seconds, at which the invalidator will run. So this is the interval in seconds how often invalid session objects will be deleted from **memory**.
 - If this property is not set, or if the specified value is less than 30, the interval will be automatically calculated based on the session.invalidationtime property.
- **session.timebasedwrite=<true|false>**
 - ▶ If true, **database** updates are done on a separate thread on a time interval specified by session.timebasedwriteinterval. If false, database updates are done at the end of every request. True will provide better performance but leaves a larger window for data loss in the even of failover. Default is false.
- **session.timebasedwriteinterval=<integer>**
 - ▶ Specifies the time interval, in seconds, at which session updates get written to the **database** by a background thread. Only applies if timebasedwrite=true. Default is 120.



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Recovery / WAS 4

database parameters

- **session.dbconnections=<integer>**
 - ▶ Specifies the number of DB2 connections held by the session manager for its exclusive use. Holding connections will improve performance. If connections are held, the optimal number is the same as the number of threads in your JVM (default number of threads is 3). The default for this setting is 0 (don't hold any connections).
- **session.usingmultirow=<true|false>**
 - ▶ If true, each session attribute will be written to a separate database row. If false, all data is written in a single row. If most session attributes are referenced on each request, or if all attributes are relatively small, false (single row) will result in better performance. Default is true.
- **session.writeallproperties=<true|false>**
 - ▶ If true, all properties (i.e. attributes) are written to the database even if setAttribute() isn't called. This forces changes to objects already added to the session to always be written to the database. Default is false, which yields better performance.



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session Recovery / WAS4

batched deletion of invalidated objects in the DB

- `session.scheduledinvalidation=<true|false>`
 - ▶ If true, specifies that sessions are to be invalidated at two specific hours of the day.
 - ▶ HttpSession timeouts are not strictly enforced. Instead, all invalidation processing is handled at the configured invalidation times
- `session.scheduledhour1=<integer_between_0_and_22>`
 - ▶ Specifies the first hour of the day (0-22) at which invalidation will occur if `scheduledinvalidation=true`.
- `session.scheduledhour2=<integer_between_1_and_23>`
 - ▶ Specifies the second hour of the day (1-23) at which session invalidation will occur if `scheduledinvalidation=true`.



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Session invalidation trigger

- JSP
 - ▶ `session.setMaxInactiveInterval(3600)`
- Servlet
 - ▶ `session.setMaxInactiveInterval(3600)`
- WebApp / `web.xml`
 - ▶ `session-timeout`
- WebContainer Conf
 - ▶ `session.invalidationtime`
 - ▶ (`session.cookie.maxage` defines how long the cookie lives on the browser side)



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

create table, index and lobs

```

CREATE DATABASE SESSDB
  STOGROUP SYSDEFLT
  CCSID EBCDIC;

CREATE TABLESPACE SESSTS IN SESSDB
  USING STOGROUP SYSDEFLT
  PRIQTY 512
  SECQTY 1024
  LOCKSIZE ROW
  BUFFERPOOL BP32K;

CREATE TABLE SESSDB.SESSIONS (
  ID          VARCHAR(95) NOT NULL ,
  PROPID     VARCHAR(95) NOT NULL ,
  APPNAME    VARCHAR(64) ,
  LISTENERCNT SMALLINT ,
  LASTACCESS DECIMAL(19,0),
  CREATIONTIME DECIMAL(19,0),
  MAXINACTIVETIME INTEGER ,
  USERNAME   VARCHAR(256) ,
  SMALL      VARCHAR(3122) FOR BIT DATA ,
  MEDIUM     VARCHAR(28869) FOR BIT DATA ,
  LARGE      BLOB(2097152),
  SESSROW    ROWID NOT NULL GENERATED ALWAYS
)
IN SESSDB.SESSTS;

CREATE UNIQUE INDEX SESSDB.SESS_INDEX ON
  SESSDB.SESSIONS
  (ID ASC,
  PROPID ASC,
  APPNAME ASC);

CREATE LOB TABLESPACE LOBOBJTS IN SESSDB
  BUFFERPOOL BP32K
  USING STOGROUP SYSDEFLT
  PRIQTY 512
  SECQTY 1024
  LOCKSIZE LOB;

CREATE AUX TABLE SESSDB.SESSIONSAUX
  IN SESSDB.LOBOBJTS
  STORES SESSDB.SESSIONS
  COLUMN LARGE;

CREATE INDEX SESSDB.SESSION_IX ON SESSDB.SESSIONSAUX;

GRANT ALL ON SESSDB.SESSIONS TO CBASRUI;

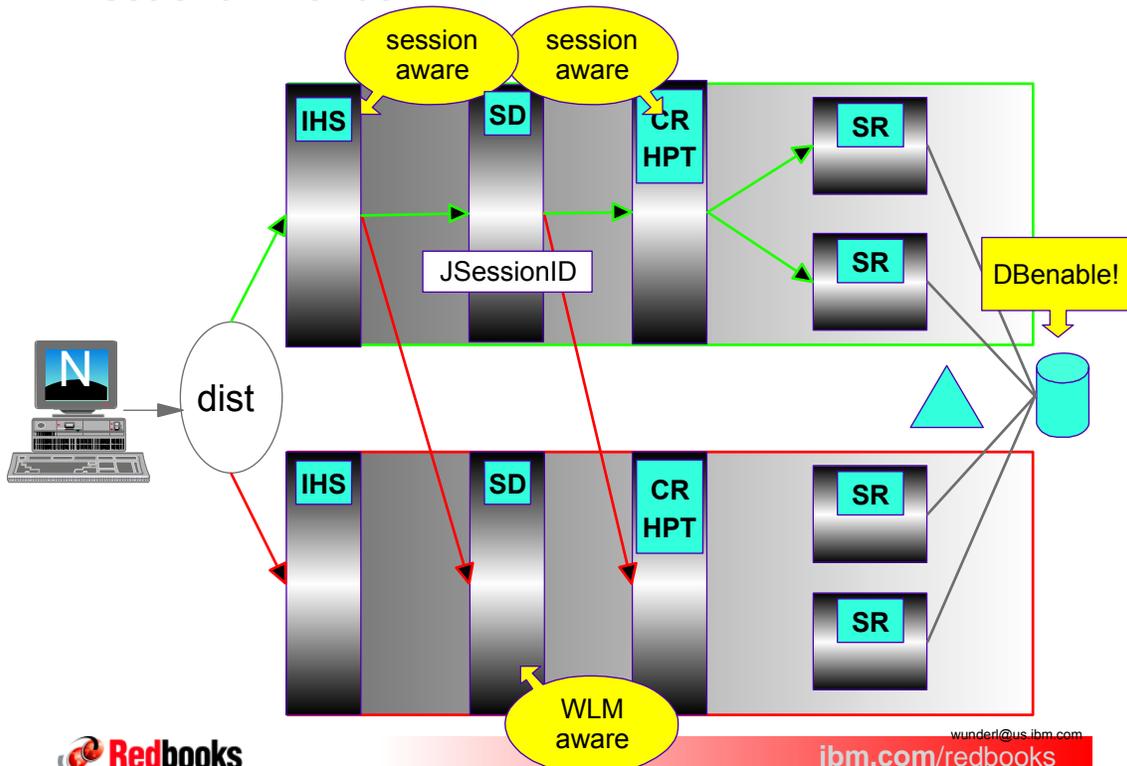
```



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Best of all worlds



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Best practices for session programming

- ▶ **When developing new objects to be stored in the HTTP session, make sure to implement the Serializable interface.** This ensures the object to properly persist session information to the database. An example of this is: `public class MyObject implements java.io.Serializable {...}` Without this extension, the object will not persist correctly and will throw an error.
- ▶ **When adding Java objects to a session, make sure they are in the correct class path.** If Java objects will be added to a session, make sure to place the class files for those objects in the application server class path or in the web application path. In the case of session clustering, this applies to every node in the cluster. Because the HttpSession object is shared among servlets that the user might access, consider adopting a site-wide naming convention to avoid conflicts.
- ▶ **Do not store large Object graphs in HttpSession.** In most applications, each servlet requires only a fraction of the total session data. However, by storing the data in HttpSession as one large object, an application forces WebSphere to process all of it each time.
- ▶ **Release HttpSession objects when you are finished.** HttpSession objects live inside the Web container until:
 - ▶ The application explicitly and programmatically releases it using `javax.servlet.http.HttpSession.invalidate()`; quite often, programmatic invalidation is part of an application logout function.
 - ▶ The application server destroys the allocated HttpSession object when it expires (default is 1800 seconds or 30 minutes). When session persistence is used, the application server can maintain only a certain number of HttpSession objects in memory. When this limit is reached, the application server removes the least recently used session entries from the cache to make a room for new ones. If Allow Overflow is enabled, the product also uses an overflow memory table to cache the entries when there is a racing condition for an entry in the cache. The product makes its best effort to keep the cache at base memory size.
- ▶ **Do not try to save and reuse the HttpSession object outside of each servlet or JSP.** The HttpSession object is a function of the HttpServletRequest (you can get it only through `req.getSession()`), and a copy of it is valid only for the life of the `service()` method of the servlet or JSP. You cannot cache the HttpSession object and refer to it outside the scope of a servlet or JSP.
- ▶ **Session clustering requires an affinity mechanism so that all requests for a particular session are directed to the same Java Virtual Machine (JVM) in the cluster.** This conforms to the Servlet 2.2 Specification in that multiple requests for a session cannot be processed in multiple JVMs. One such solution provided by IBM WebSphere Application Server is Session Affinity, which is available as part of the WebSphere plug-ins for Web servers. If one of the servers in the cluster fails, it is possible for the request to be rerouted to another server in the cluster. The new server can access session data from the common SESSIONS table. This is transparent to the service browser, and user. This helps to achieve a greater use of the in-memory cache and reduces hits to the session database.
- ▶ **Use `<% @ page session="false" %>` to turn off the automatic session creation from the JSPs that won't update the session.** Use `<% HttpSession session = javax.servlet.http.HttpServletRequest.getSession(false); %>` to get the already existing session



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

HTTP session vs. stateful Session Bean

http session state in memory are obtained on each http interaction through `getSession` calls. One can store arbitrary information in the http session. The session gets cleaned up after a designated unreferenced time. The reaper thread if implemented well does not usually pose a performance issue. HTTP session objects are an artifact of the HTTP communication channel. As such they bind your application code to an HTTP port of entry (when you decide to move to web services they will be in the way). They should be used for managing presentation only and not business logic

stateful session beans are obtained via explicit create, they are accessed as an EJB is and have a higher overhead to gain access to them. This is because they implement things like security, transactions, etc. The bean attributes are predefined and typed. There is a stateful session bean reaper which does not incur a lot of overhead either. A stateful session bean has similar properties to an HTTP session but binds you closer to I/O (their value is to get you back to the same server instance as well). EJB containers pool stateless session beans and reuses them to service many clients. Stateful session beans can be passivated and reused for other clients. But this involves I/O bottlenecks. Because a stateful session bean caches client conversation in memory, a bean failure may result in losing the entire client conversation

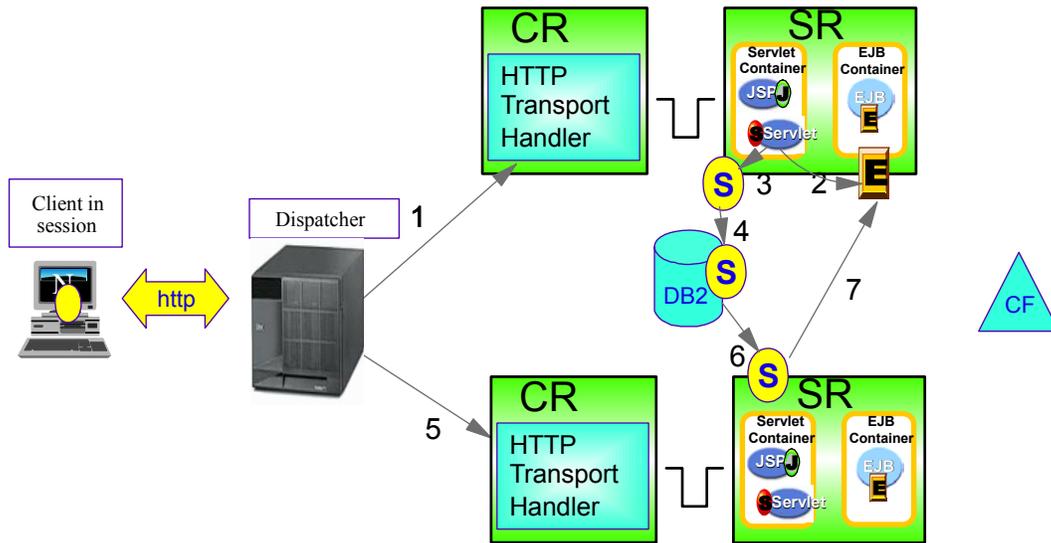
- ▶ HTTP session (in memory) are usually the better performer
 - Servlet HTTP session will be the most suitable choice if the clients talk to the EJB application via HTTP.
 - Serialization of HTTP sessions is as least as costly as the DB2 part of that externalization work
 - use only memory, so software cost stays stable
 - HTTP SOs live in the heap, big HTTP SOs means big heaps and at one point many server regions, watch GC
 - Implementing stateful behavior (HTTP or SFSB) limits WAS ability to do workload balancing. Designs that do not require this are much preferred as they provide a much better scaling experience
- ▶ Stateful Session Beans are performance wise comparable to externalized HTTP SOs (Persistence V1)
 - Stateful session beans should be used when the EJB application is also accessed by non http application client



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

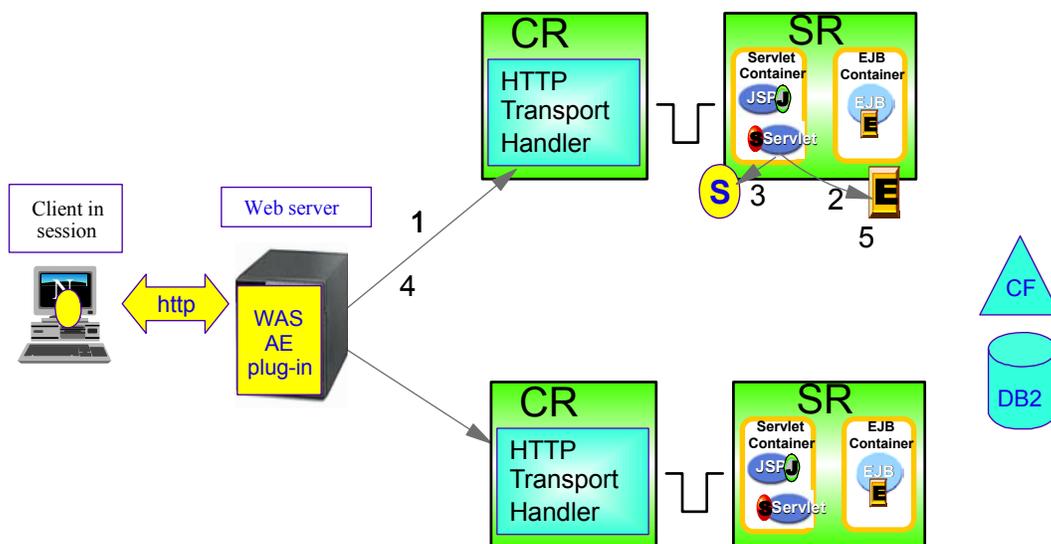
externalized HTTP session and in-memory stateful Session Bean



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

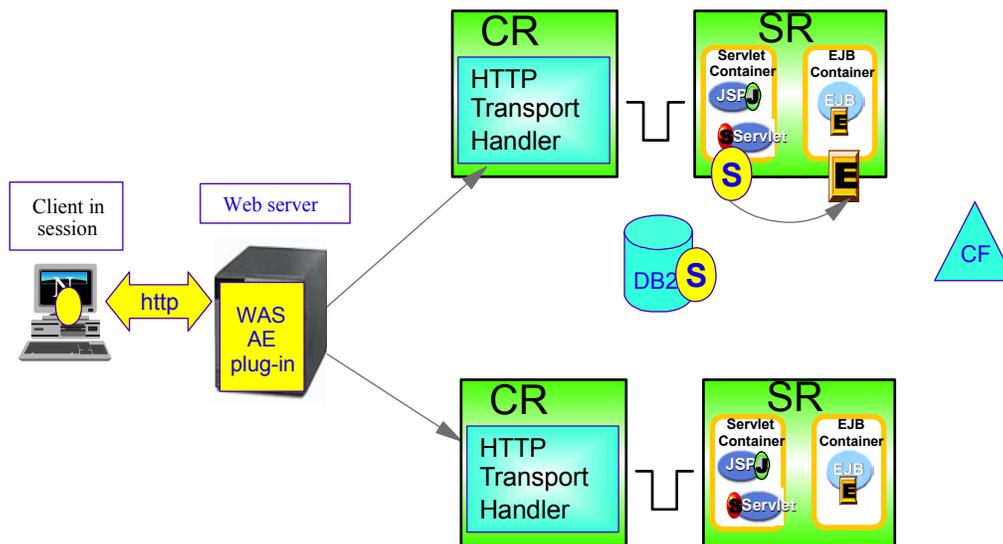
in-memory HTTP session and in-memory stateful Session Bean



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

HTTP session vs. stateful Session Bean



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

Summary

- Keep Sessions in memory for performance
- Keep Session Objects Small
- Session Affinity has to be available at various stages of the distribution mechanism
- WebSphere Edge Server with CBR enabled does the same thing
- For High Availability keep sessions also in DB2, but only with the new performance improvements and session affinity!
- See Redbook 'Enabling High Availability eBusiness on zSeries', SG24-6850
- <http://www-1.ibm.com/support/techdocs/> for a nice description on the whole session management story



ibm.com/redbooks

© Copyright IBM Corp. 2003. All rights reserved.

