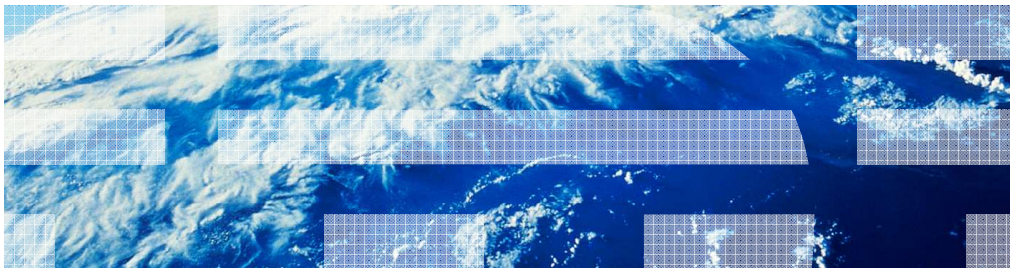


# Memory-to-memory session replication

## IBM WebSphere Application Server V7



This presentation will cover memory-to-memory session replication in WebSphere Application Server V7.

## Agenda

- Memory-to-memory session replication overview
- Topology options and benefits
  - Peer-to-peer
  - Client-server
- Configuration
  - Enabling replication
  - Selecting a replication mode
  - Multiple replicas
- Summary

This presentation will introduce memory-to-memory session replication, the different topology options and their benefits, and how to configure those topologies.

## HTTP sessions

- HTTP session object
  - Can be used by web applications to keep user state information
  - Can be a single point of failure if not available to multiple application servers
- Session distribution options
  - Database session persistence
    - Session objects stored in a database
    - Can be accessed by multiple application servers
  - Memory-to-memory session replication
    - Sessions objects stored in memory by multiple application servers
    - Can be accessed by multiple application servers
    - Does not require use of a database product

HTTP session objects are in-memory objects that can be created and accessed by a web application to keep track of user state information. Applications can fail or work incorrectly when a session object cannot be found. For example, if the contents of a user's shopping cart are stored in an HTTP session object, and that object cannot be found, the application will no longer know the contents of the cart. For this reason, session objects should be distributed, and not kept in memory on the application server where they are in use.

WebSphere Application Server supports distributing session objects by either storing them in a relational database that can be accessed by all members of a cluster, or through memory-to-memory replication, which uses multiple cluster members to store backup copies of session objects in memory.

## Memory-to-memory session replication

- HTTP session data stored in memory across multiple application servers
  - Uses the data replication service
  - Web server plug-in maintains session affinity to a single server
    - Affinity changes to another server upon failure
- Application server configuration options
  - Client
  - Server
  - Both client and server (default)
- Topology options
  - Client-server
    - Some application servers configured as clients
    - Some application servers configured as servers
  - Peer-to-peer
    - All application servers configured as both client and server

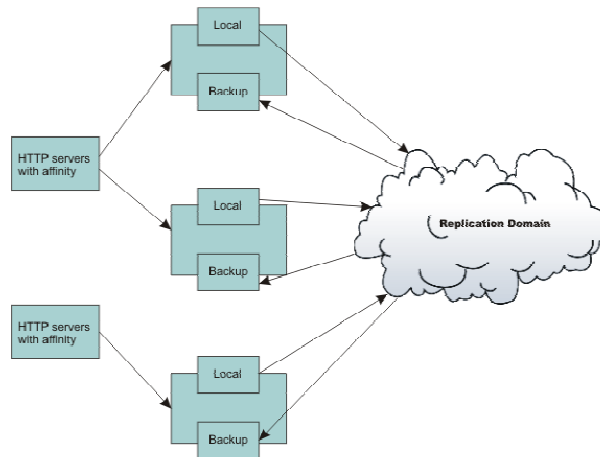
WebSphere Application Server uses the data replication service (DRS for short) to distribute HTTP session data among members of a cluster. web requests from users are always routed to the server that owns the session object by the web server plug-in, unless that server is unavailable. If the server is unavailable, the requests are routed to one of the application servers that contains a backup copy of the session object.

Application servers can be configured as either a client, a server, or both, when configuring memory-to-memory replication. Two common topologies can be created by configuring the servers in one of these ways.

By default, all application servers are configured to be both clients and servers. This creates a “peer-to-peer” topology, in which application servers are both actively serving web requests and hosting backup data for other servers. By setting some servers to client mode and some servers to server mode, you can create a “client-server” topology, in which some servers are dedicated to hosting backup data for other servers. It is important to note that if any application servers are set to “both client and server” mode, all application servers must be set that way.

## Peer-to-peer topology

- All servers configured as both client and server



5

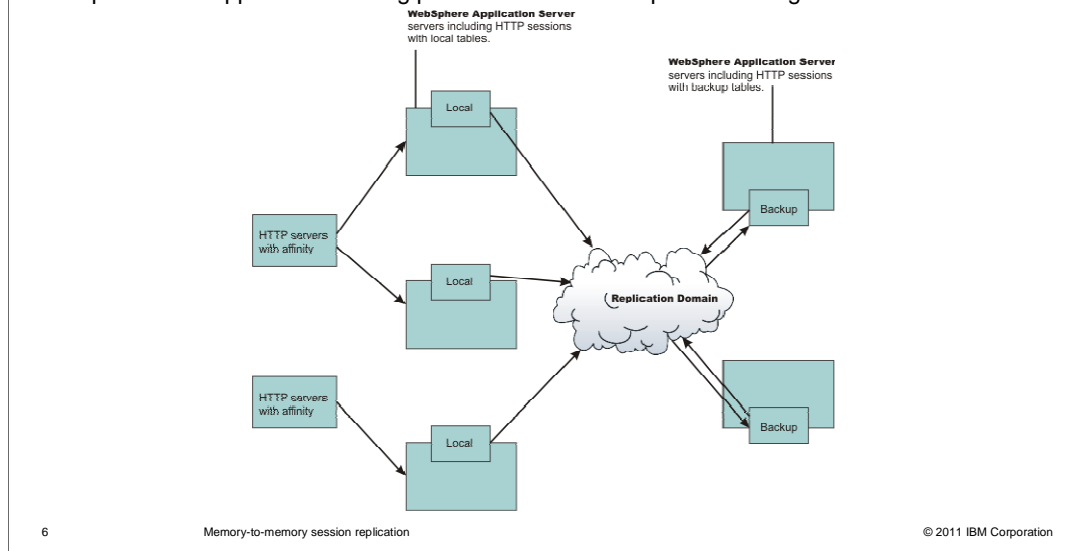
Memory-to-memory session replication

© 2011 IBM Corporation

This diagram shows a basic example of a peer-to-peer topology. Each of the three servers has its own local session objects, which are associated with web requests. Each server also hosts a backup copy of session data from another application server. If one server fails, the web server plug-in will route requests intended for that server to the server that is hosting the backup copy of the session object associated with the incoming request.

## Client-server topology

- Separation of application-serving processes from backup data hosting



This diagram shows a basic example of a client-server topology. In this example, the three application servers in the middle of the diagram have been configured in client mode, which means they are meant to serve application requests, and not backup session data for other servers. The two application servers on the right side of the diagram, alternatively, have been configured in server mode: they only host backup session data, and do not serve application requests.

Should one of the application servers fail, the web server plug-in will route the request to one of the other servers, which can then retrieve a backup copy of the session object from the backup servers.

## Topology comparison

- Peer-to-peer
  - Simpler
  - Requires fewer server processes
  - Supports session hot failover
- Client-server
  - Process isolation
    - Processes serving requests are separate from processes hosting backup data
    - Restarts of backup data servers tolerated
      - Requires more than one backup data server
    - Application-serving process restarts do not risk losing backup data
  - Hardware capacity considerations
    - Session backup can be consolidated to a small number of servers
    - Frees up resources on application serving processes

The peer-to-peer topology is the simpler of the two topologies to configure. It also generally requires fewer server processes, since there are no application servers dedicated to hosting backup data. Additionally, the peer-to-peer topology supports “session hot failover”, meaning in the case of a failure, an incoming request can be routed to a server that already has the session data in memory.

The client-server topology separates processes by task. This isolation enables some flexibility, such as consolidating backup data on a small number of servers to free up resources on application-serving processes. While this topology introduces some practical advantages for production environments, it also generally requires more hardware resources.

## Enabling memory-to-memory replication

- When creating a cluster:

8

Memory-to-memory session replication

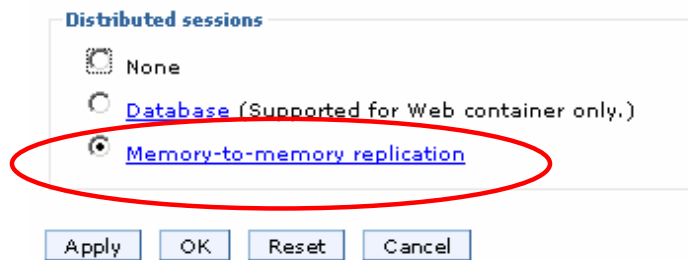
© 2011 IBM Corporation

Memory-to-memory replication can be enabled when creating a cluster. Checking the box shown here will create a replication domain for the cluster.



## Enabling memory-to-memory replication

- For an existing cluster member:
- `servername > web container > Session management > distributed environment settings`



To enable memory-to-memory replication on an existing cluster member, use the radio button shown here, in the “session management” configuration panel. This step must be performed on each cluster member. Clicking the “memory to memory replication” link will take you to the configuration panel shown on the next slide, where you can configure replication options.

## Selecting a replication mode

- *servername* > web container > Session management > distributed environment settings > Memory-to-memory replication
- Replication modes are set individually on each server

\* Replication domain  
cluster1

\* Replication mode  
Both client and server

Apply OK Reset Cancel

This panel allows you to choose which replication domain to use for session replication, and to set the replication mode for the cluster member. Remember that you must change this setting for each cluster member to correctly change the cluster's topology.

## Multiple replicas

- Configured on the replication domain
  - Environment > Replication Domains
- Single replica
  - Each session backed up on one server
- Entire domain
  - Each session backup up on all servers
- Multiple replicas
  - Increase availability
  - Require more resources on each server

### General Properties

Name  
cluster1

\* Request timeout  
5 seconds

#### Encryption

Encryption type  
none

Regenerate encryption key

#### Number of replicas

- Single replica  
 Entire Domain  
 Specify

Number of replicas

Apply OK Reset Cancel

For high availability purposes, you should maintain more than one backup copy of each session object. To do so, modify the “number of replicas” setting on the replication domain that is used to replicate session data. “Single replica” means there is a single backup copy of each session object, and “entire domain” means that each session is backed up on every “server-mode” process in the replication domain. The “specify” option allows you to choose many replicas between one and the number of available servers. This is a useful option for keeping multiple replicas with out the memory overhead required to store copies on all servers.

## Summary

- Memory-to-memory replication
  - Provides a mechanism for backing up HTTP session data on other application servers
  - An alternative to using a database for session persistence
- Topology options
  - Peer-to-peer mode
    - All application servers are used for hosting web applications and hosting backup session data
  - Client-server mode
    - Some application servers are used for hosting web applications
    - Some application servers are used for hosting backup session data
- Multiple replicas
  - Increase redundancy and availability
  - Increase server resource usage

Memory-to-memory replication is a common and practical way to distribute HTTP session objects across multiple application server processes without use of a relational database product. Application servers can be set up in either a “peer-to-peer” topology or a “client-server” topology, and multiple replicas can be used to increase redundancy, and therefore availability.

## References

- WebSphere Application Server V7 Information Center
  - Memory-to-memory replication:  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cprs\\_memory2memory.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cprs_memory2memory.html)

The Information Center article shown here provides even more detail on all of the configuration options that have been discussed in this presentation.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WASv7\\_MemorytoMemoryReplication.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WASv7_MemorytoMemoryReplication.ppt)

This module is also available in PDF format at: [..\\WASv7\\_MemorytoMemoryReplication.pdf](..\\WASv7_MemorytoMemoryReplication.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.