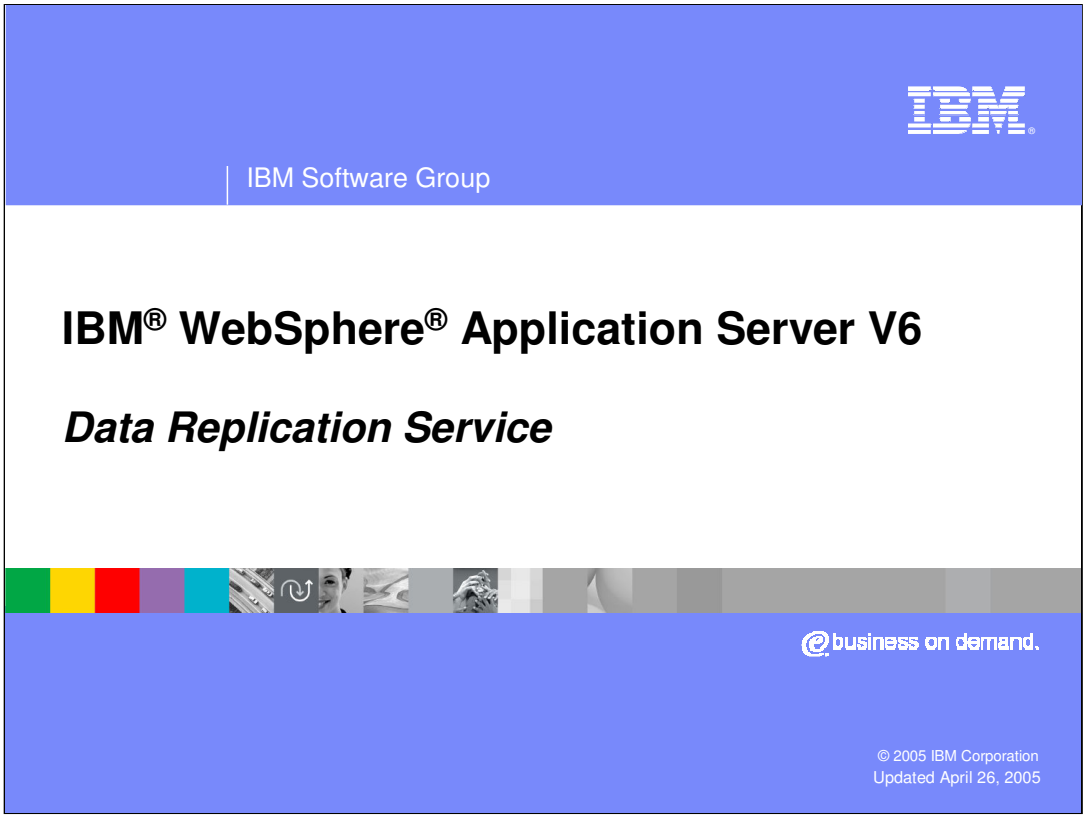
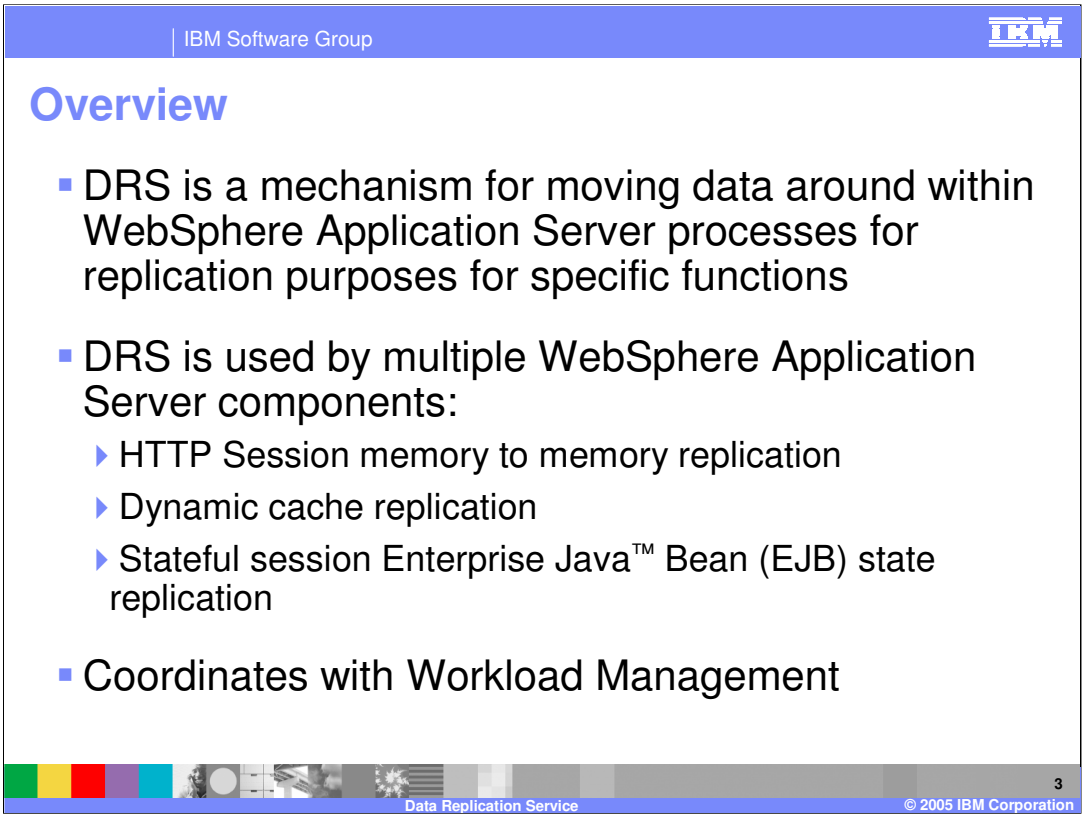


This presentation will focus on the Data Replication Service in WebSphere Application Server V6.

Goals

- Provide a quick overview of Data Replication Service (DRS) and where it is used within WebSphere Application Server
- Discuss new V6 DRS functions

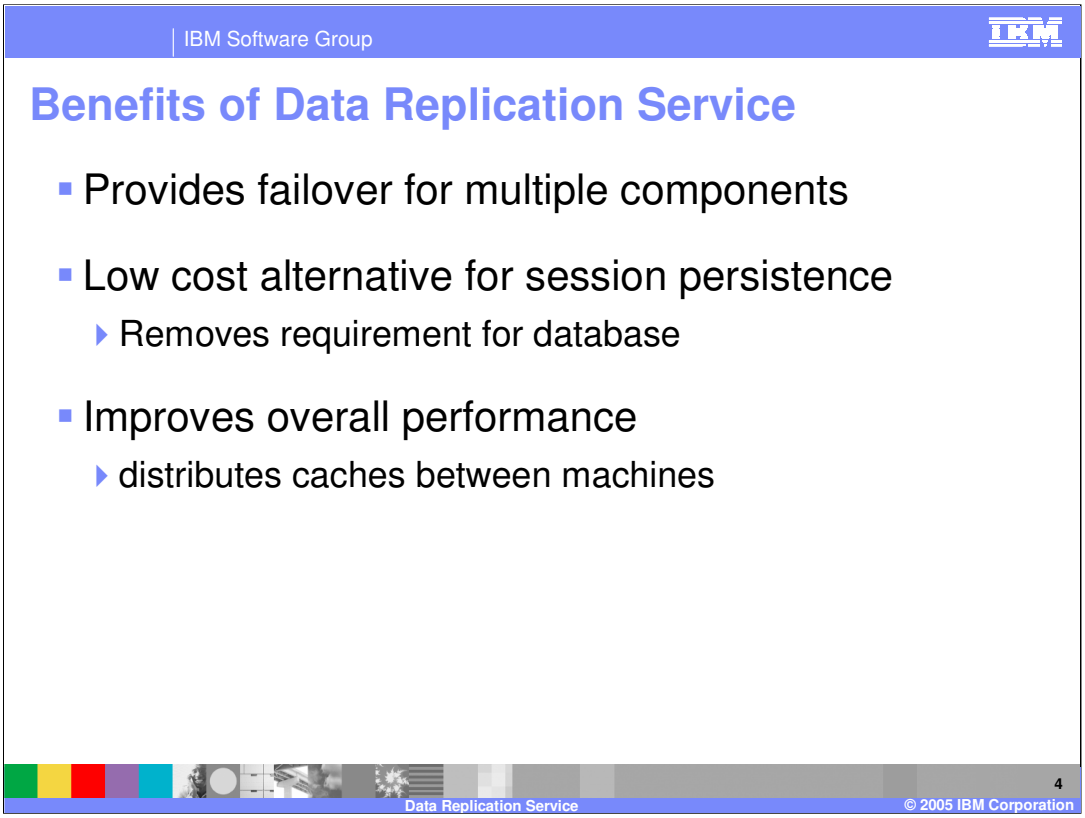
The goal of this presentation is to outline and provide an overview of the Data Replication Service and discuss the new features in V6.



The Data Replication Service is an internal component of the WebSphere Application Server. It is used by other components to move data from place to place. The most visible use of DRS is for replicating persistent HTTP Session data so that if an application server fails, the request can be routed to another application server, and the session data will be available there.

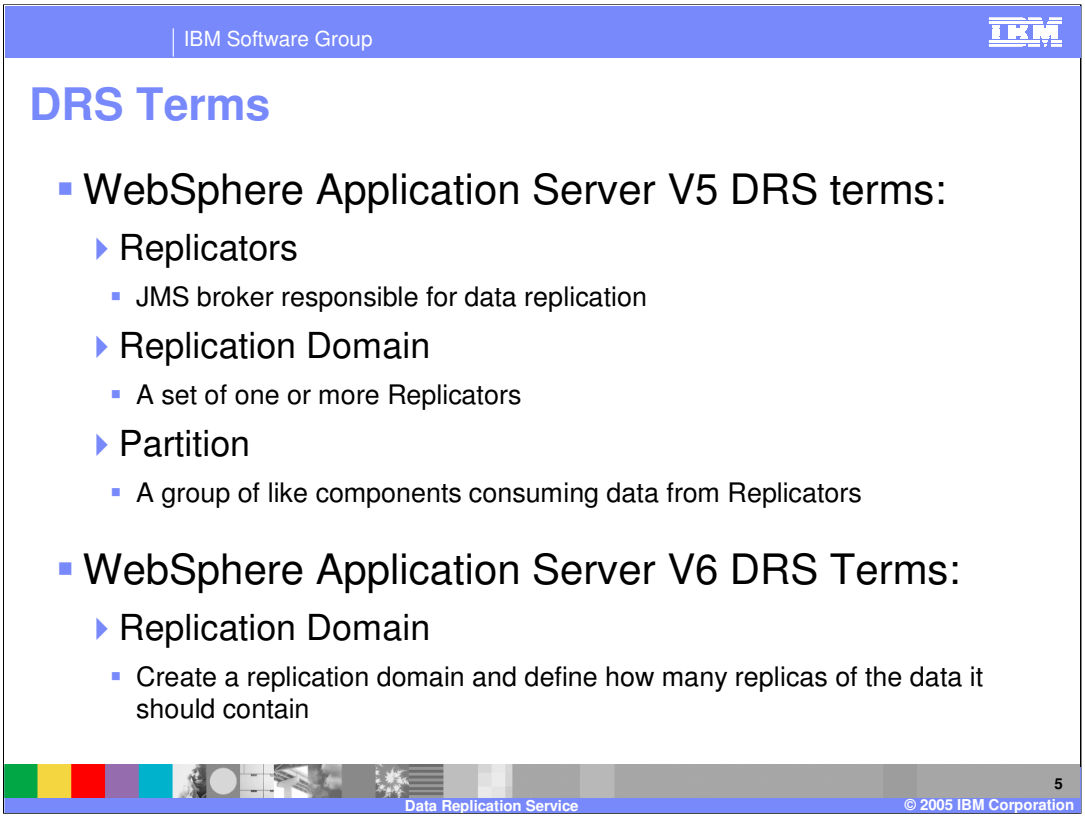
In order to minimize the impact of a failure, DRS coordinates with the Workload Management routing algorithm to assure that requests and data end up in the same place.

A new feature in Version 6 is the capability to capture the state of a stateful session bean and enable failover to another instance of that bean in another application server.



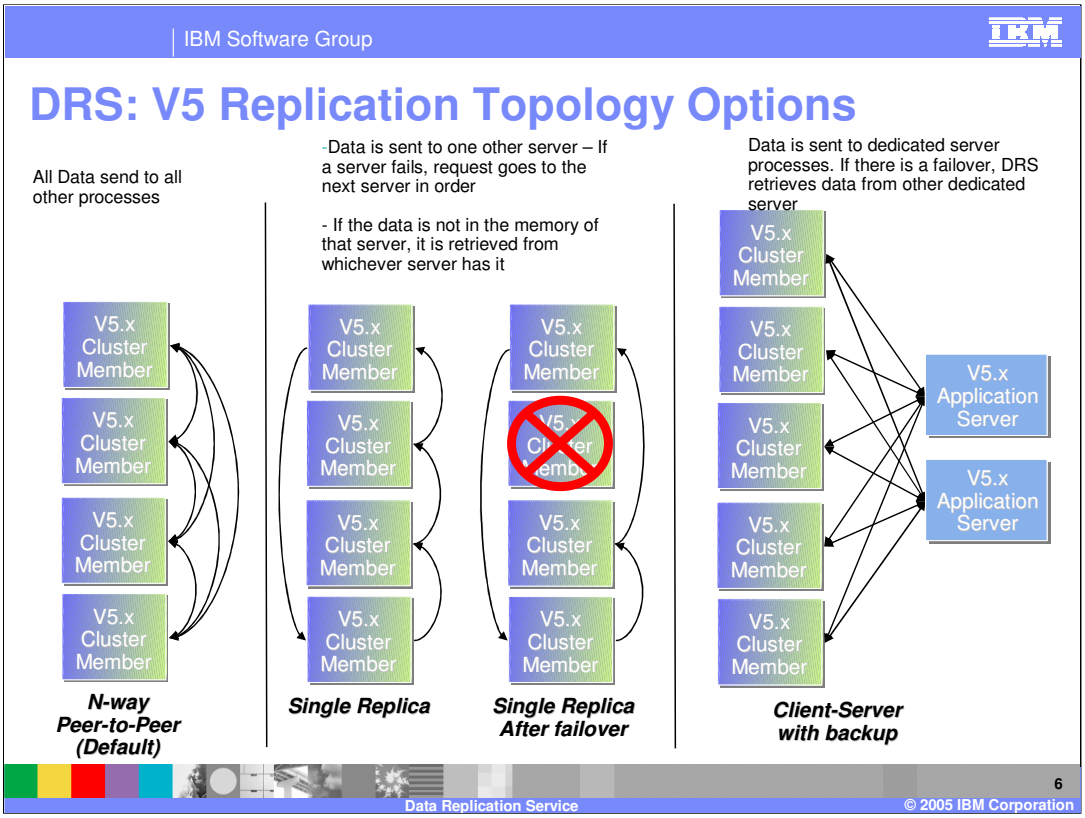
The Data Replication Service provides services in two scenarios: Failover and caching. Failover support assures that HTTP Sessions and EJBs can be moved to another application server. This is transparent to the user.

The second use of Data Replication Service is caching. A servlet or a JSP™ can be configured to have its output cached. Once the data is available in the cache, repeated requests for the same information are handled faster. Since the data in the cache will be the same no matter which application server the request arrived at, it makes sense to cache the same data in of all the application servers. Data Replication Service handles moving cache entries to other servers.



Version 5 of WebSphere Application Server leaves much of the configuration to the Administrator. Replicators are the producer and consumer that are responsible for moving data as Java™ Message Service (JMS) messages. Administrators create Replicators within a Replication Domain; the default configuration is to have all the application servers in a domain talk to all the other application servers. It is possible to reduce the overhead by limiting which application servers talk to which; those that are configured to talk to each other are a partition, as well as being part of a MultiBroker replication Domain.

Version 6 simplifies this. Because the underlying mechanism has changed, it is no longer necessary to manually create and configure Replicators. Also, the concept of Partitioning is masked. Even though it is still possible to limit the number of copies of the data, it is not necessary to expose the details of that configuration.



N-way Peer to Peer:

Looking at the configuration for HTTP Session Persistence, these Cluster Members are application servers that have been created as members of a cluster. Data Replication Service is used to copy session data between the clustered application servers.

The default topology for Data Replication Service in version 5 is for all Replicators to service all channels in the Replicator domain. This means for five application servers, there will be four backups and the original. You can see how this could quickly consume a lot of memory. To reduce the memory overhead without losing the backup, you can set the Single Replica flag to limit the Replicators to one backup copy. In the event of a failover, the Session Manager will find the backup and make it available to the application server where the request lands.

Single Replica:

The single replica configuration scales much better in terms of memory consumption. In the event of a failure of an application server, the request will be routed to the next clustered application server in the rotation. If the session information exists in the new server, it is used; if not, the session manager retrieves it from wherever it was backed up. As soon as the request hits the new server, it's updated session information is sent to another server, so there is always a backup. (Unless the right two servers happen to fail at the same time.)

Client Server

A Client Server topology is one where another application server is configured to store backup session data in the local memory space. This topology reduces the overhead on application servers handling requests, but introduces a single point of failure (SPOF).

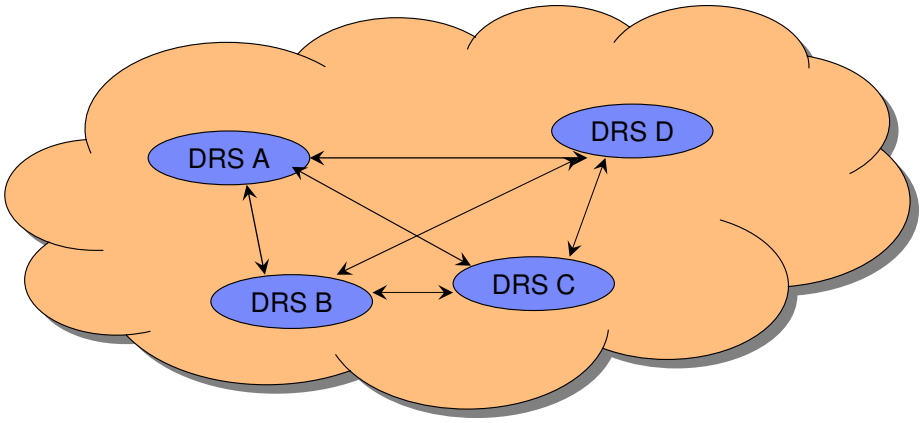
In the Client Server topology, it is useful to configure two independent application servers to store backup session information. By introducing a second server on a second machine, the single point of failure is eliminated.

Section

DRS in V6

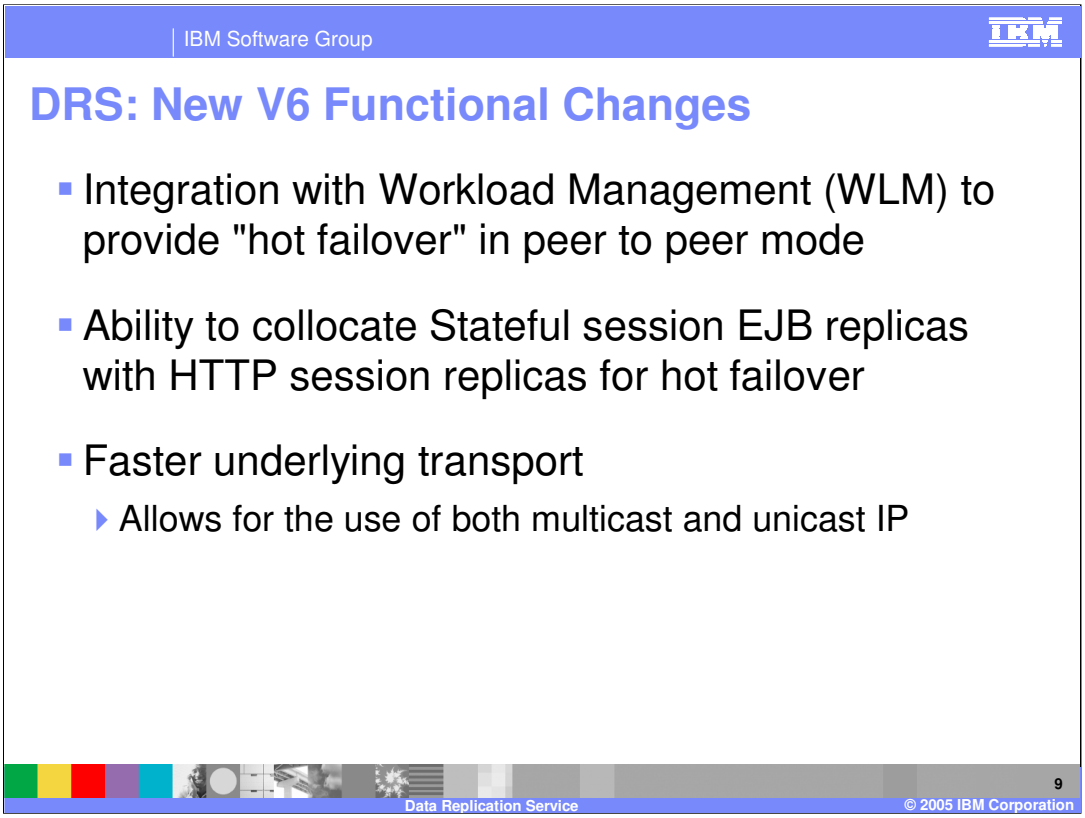
This section address Data Replication Service function in WebSphere Application Server Version 6

DRS in V6



Replication Domain consists of server or cluster members that have the capability of sharing WebSphere Application Server internal data (HTTP Session, Cache) within the domain

A replication domain consists of servers and cluster members that have the capability to replicate information from one cluster member to any other cluster member.



Some changes in Version 6 include cooperation between the Data Replication Service and the Workload Management subsystem to coordinate which cluster members serve as backups for other cluster members. Ideally, session failover data and stateful session bean failover data should end up in the same place – the place that a failed-over session will arrive in the event it needs to be served by a cluster member other than the one that originated the session.

Another improvement is that the underlying mechanism has been rewritten using a proprietary transport to move data. This reduces overhead and improves overall system performance.

DRS: V6 Configuration Simplification

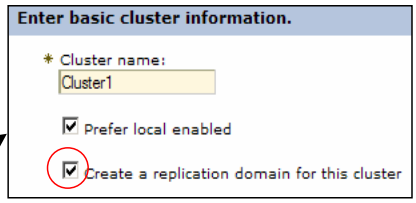
- Replicator configuration is no longer a part of the replication domain configuration
- Configuring a user (consuming component) for DRS involves:
  - Creating a domain and specifying the number of replicas
    - Default number of replicas is 1
    - For cache replication, each server must have a replica
  - Configuring the user to be part of the domain
  - Ensuring that only valid users are part of the domain
- Many fields from V5.x have been removed, making the configuration panels more intuitive

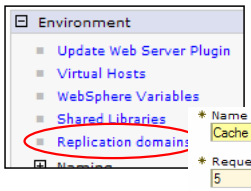
Moving from Version 5 to Version 6 is moving to a more simplified configuration. The change in the underlying communication mechanism removes the need for Replicators. The only configuration decision you can make is how many backup copies of the data will be needed. The default is one.

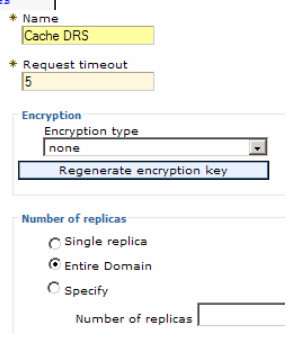
Version 6 benefits from a faster transport mechanism, the channel framework, which eliminates the one-thread-per-queue limitation. Sitting on top of a more robust transport also removes the need for manual partitioning.

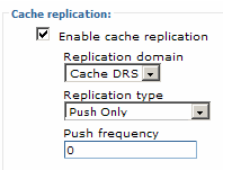
DRS Administration

- DRS domain can be created as follows:
  - Optionally, when creating a cluster - DRS domain name same as cluster name
  - Manually create DRS domain
- Enabling Dynamic Cache to use DRS



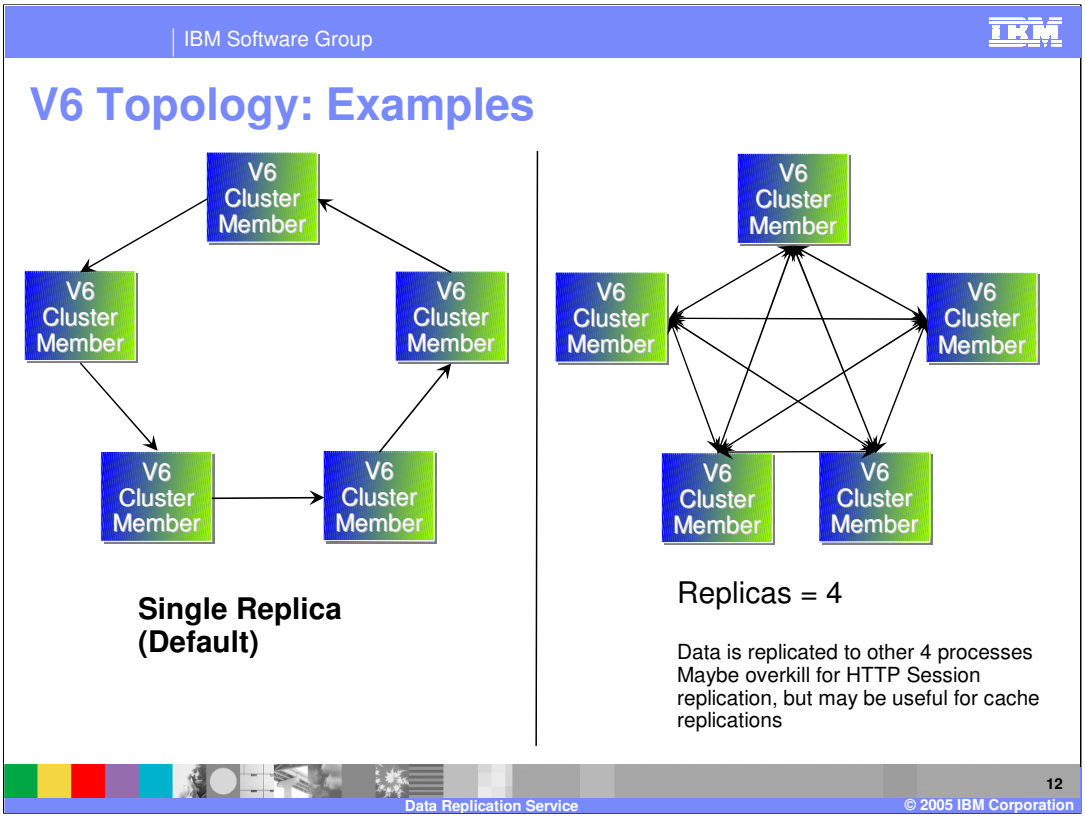






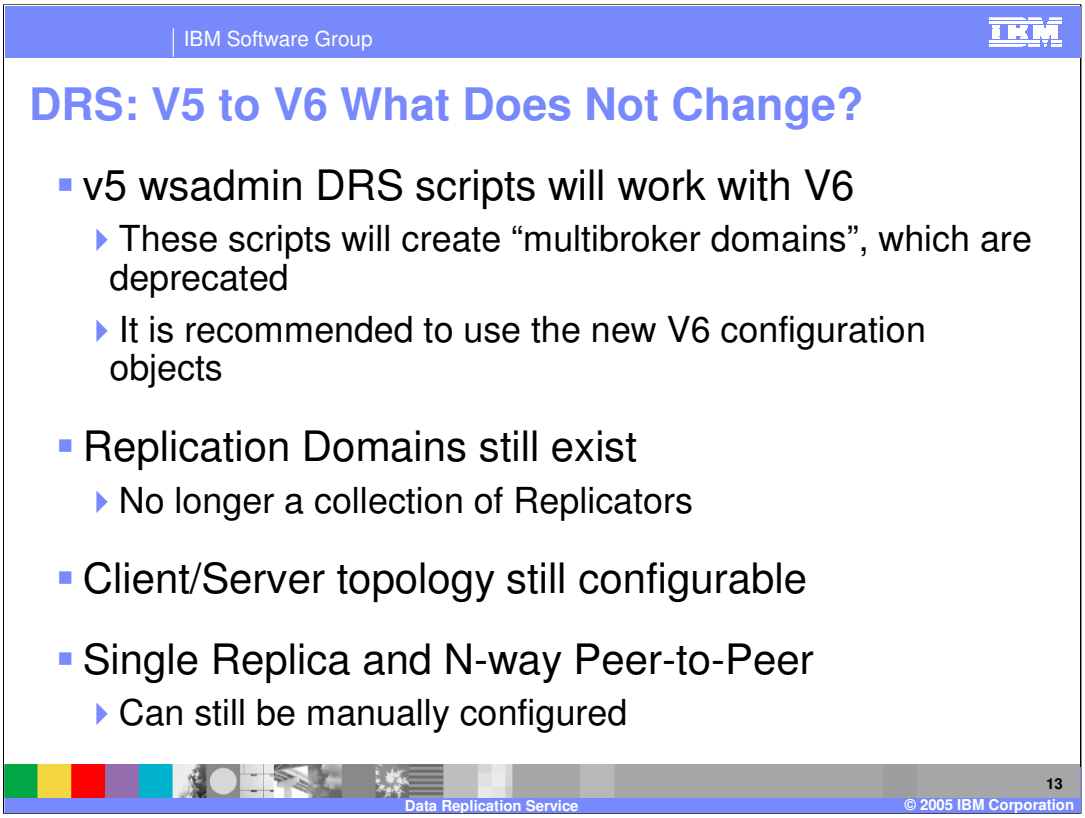
Server -> Container Services -> Dynamic Cache Service

This slide illustrates the locations within the Administrative Console where you can change settings on the Data Replication Service. When you create a cluster, creating a replication domain is as easy as selecting a checkbox, or you can manually create a domain. Because Data Replication Service is used for both cache replication and session data, you can configure cache replication under Server, then Container Service, then select Dynamic Cache Replication.



In the default topology, each server in the domain will hold a replica of the data from one other server.

In the second example, notice that the arrows each have two heads – data flows from each process to every other process, so that for each replicated object, there are four remote copies and the local original. This topology would probably be more than what is needed for HTTP Session replication, but it is the only allowable configuration for cache replication. When caching dynamic content, the cache is only useful if it is available on all the machines where a request could arrive.

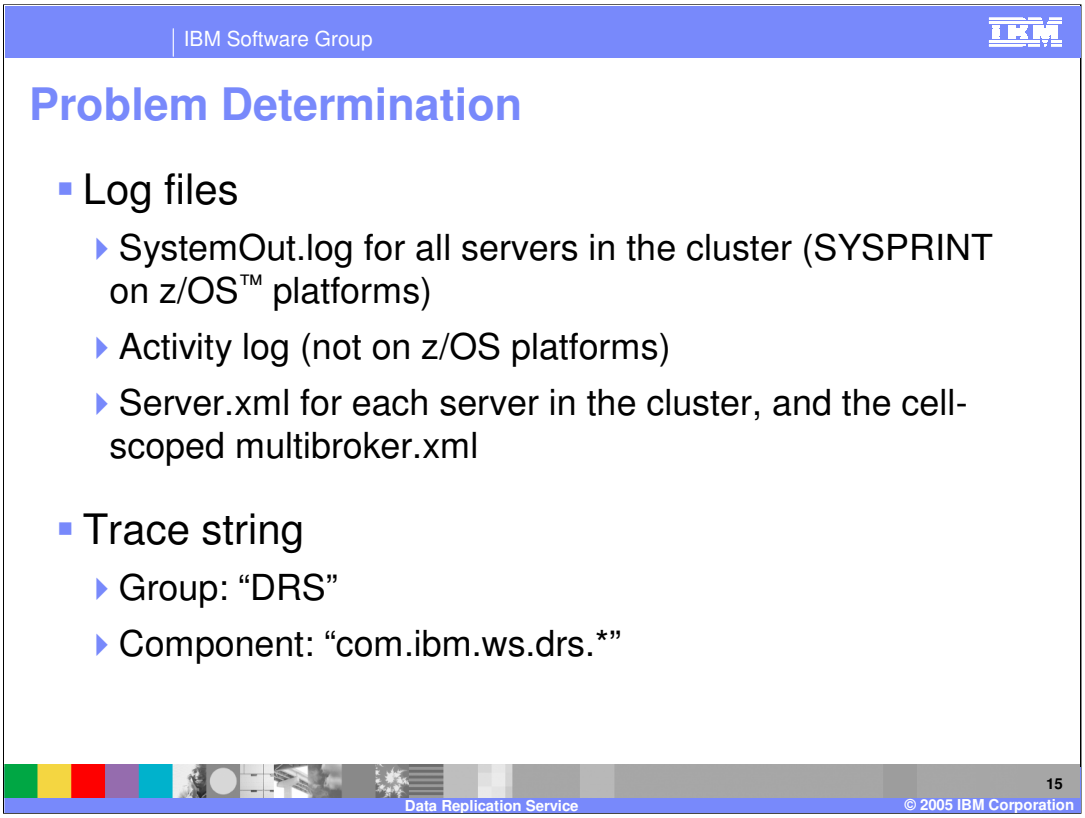


wsadmin scripts that create replication domains for version 5 will still work with version 6, and the common topologies used in version 5 will still be possible. However, using wsadmin scripts that were intended for version 5 is not suggested, because it will create multibroker domains, which do not have the performance benefits of V6 replication domains.

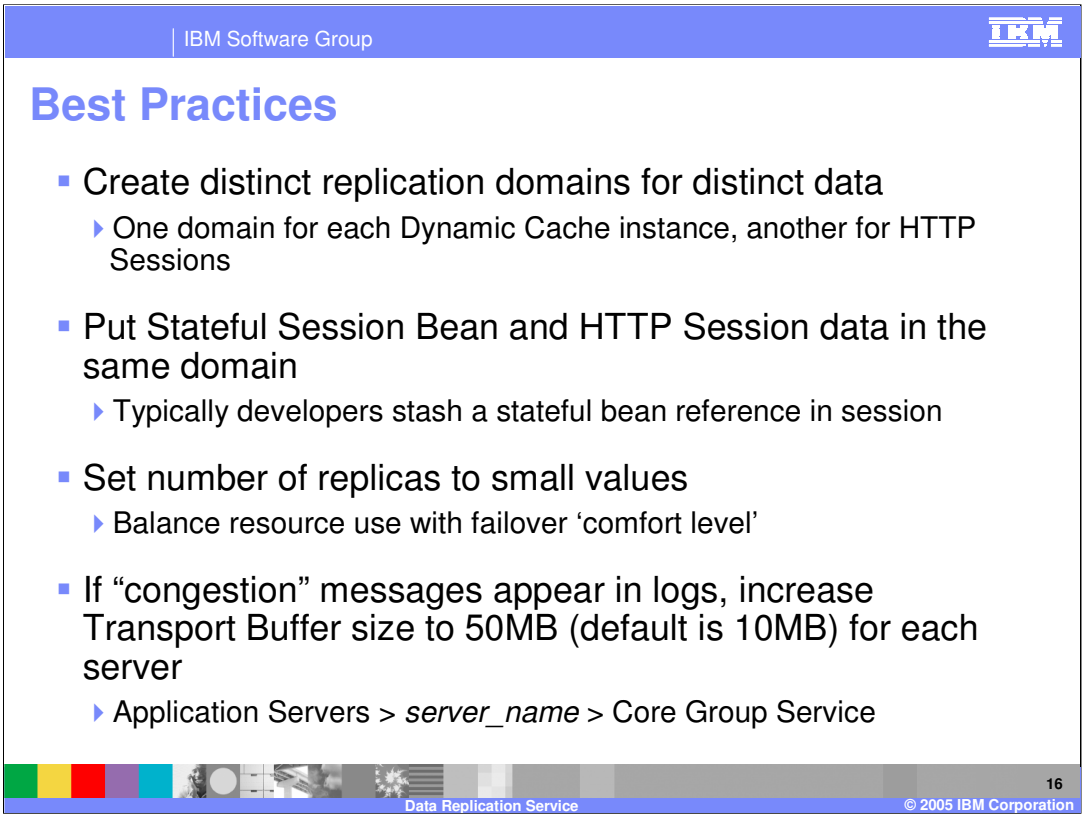
Section

Problem Determination

This section deals with problem determination.



When troubleshooting a DRS problem, you should examine the SystemOut.log (or SYSPRINT) for all servers in the cluster, and their server.xml files, which contain the DRS configuration information. If you need to search deeper, tracing the group named DRS, or the com.ibm.ws.drs.* component will give you more detailed information.



Suggested Best Practices include creating a distinct domain for HTTP and EJB session data, and another domain for Cache replication.

The number of replicas you configure will have an impact on failover and on performance. Using a smaller number improves performance. Increasing the number of replicas may reduce the time it takes a session to move to another server, but it does so at the cost of overall performance. It is suggested that in most cases, one, two, or three replicas should be sufficient.

Summary

- Data Replication Service moves data
- Cache replication improves response times for users
- HTTP Session data replication provides failover support in coordination with WLM component
- Stateful EJB state persistence, failover new in V6
- Configuration simplified, performance enhanced

In summary, this presentation has explained that the Data Replication Service is used for moving data between cluster members or application server processes.

HTTP Session data can be configured to be available in the cluster member where a session will fail over to, and the Data Replication Service coordinates with the Workload Management component.

New in V6 is Stateful Session Bean persistence and failover, and administration of Data Replication Service has been simplified.

Trademarks, Copyrights, and Disclaimers

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

IBM	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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.

Information is provided "AS IS" without warranty of any kind. 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 (e.g., 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.