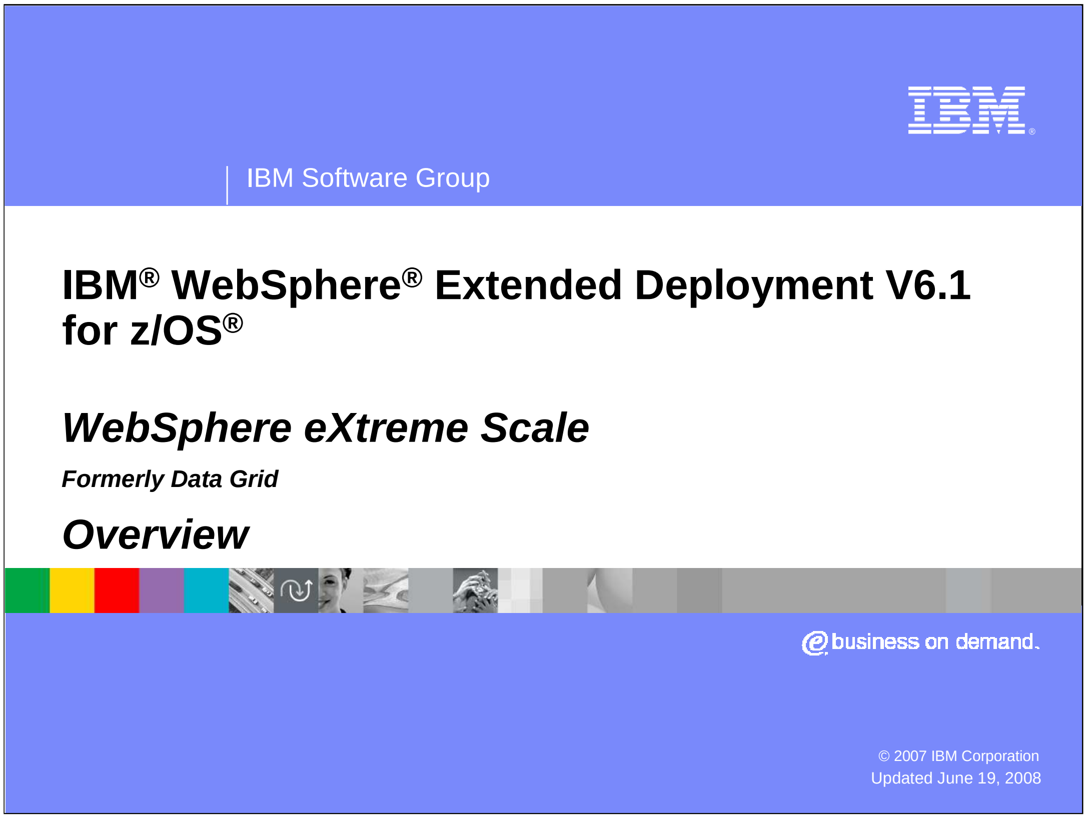


IBM Software Group

IBM® WebSphere® Extended Deployment V6.1 for z/OS®

WebSphere eXtreme Scale

Formerly Data Grid

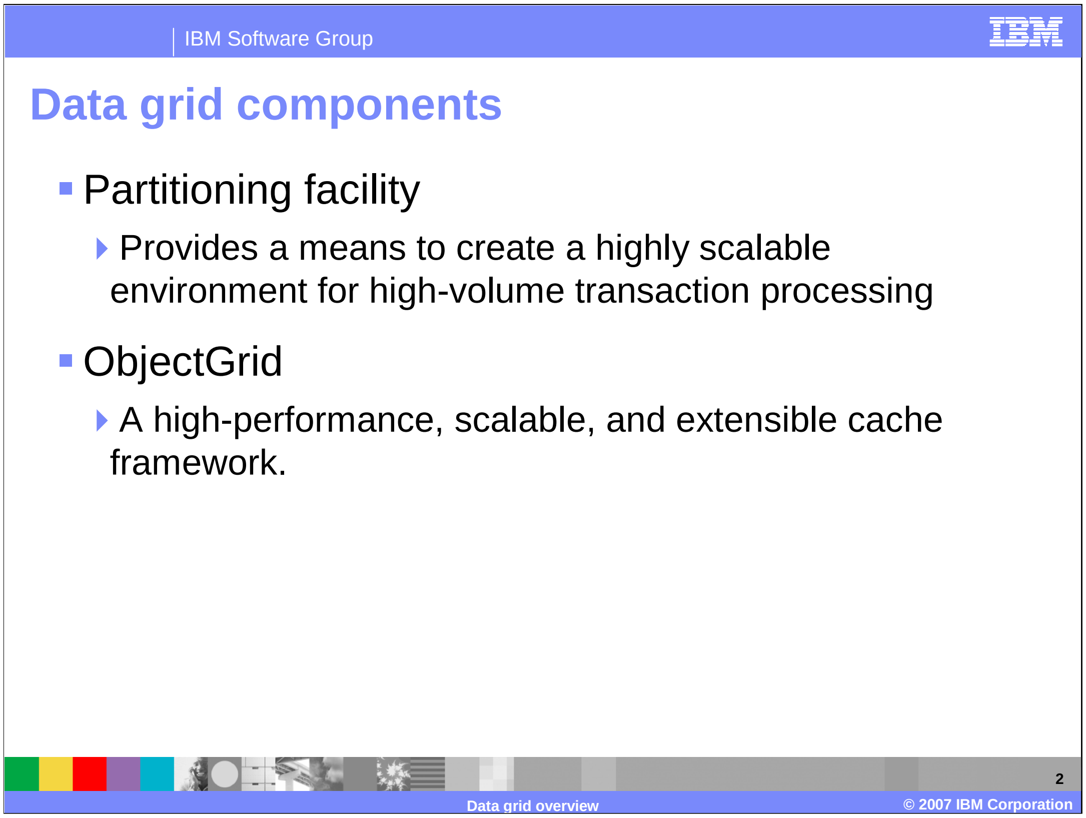
Overview

business on demand.

Updated June 19, 2008

This presentation will give an overview of the data grid components of WebSphere Extended Deployment Version 6.1.

This module was originally recorded for WebSphere Extended Deployment Data Grid, which is now called WebSphere eXtreme Scale. Though the module uses the previous names, the technical material covered is still accurate.

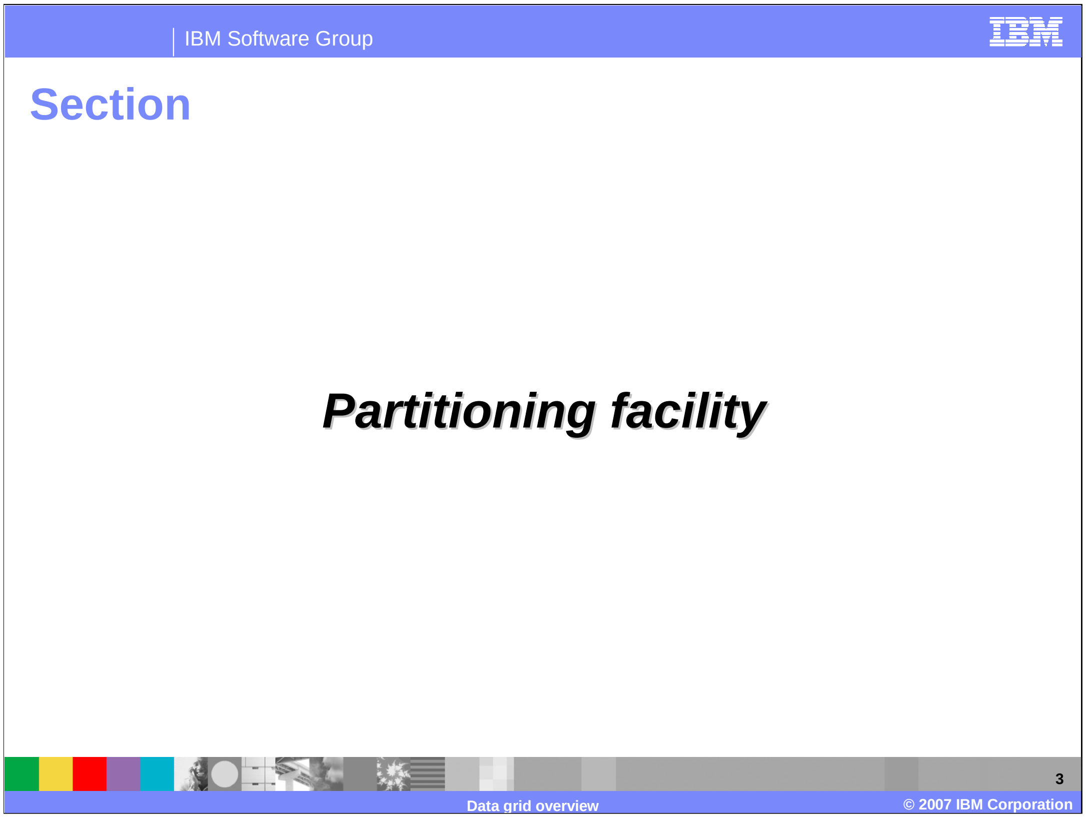


Data grid components

- Partitioning facility
  - Provides a means to create a highly scalable environment for high-volume transaction processing
- ObjectGrid
  - A high-performance, scalable, and extensible cache framework.

The WebSphere Extended Deployment data grid package contains two primary components: the partitioning facility, and ObjectGrid. The partitioning facility provides a programming model and runtime environment for implementing highly scalable solutions for transaction processing. ObjectGrid is a high-performance, transactional, and extensible object cache for Java™ applications.

Each of these components will be discussed briefly in this presentation, and covered in more detail in other presentations.



IBM Software Group

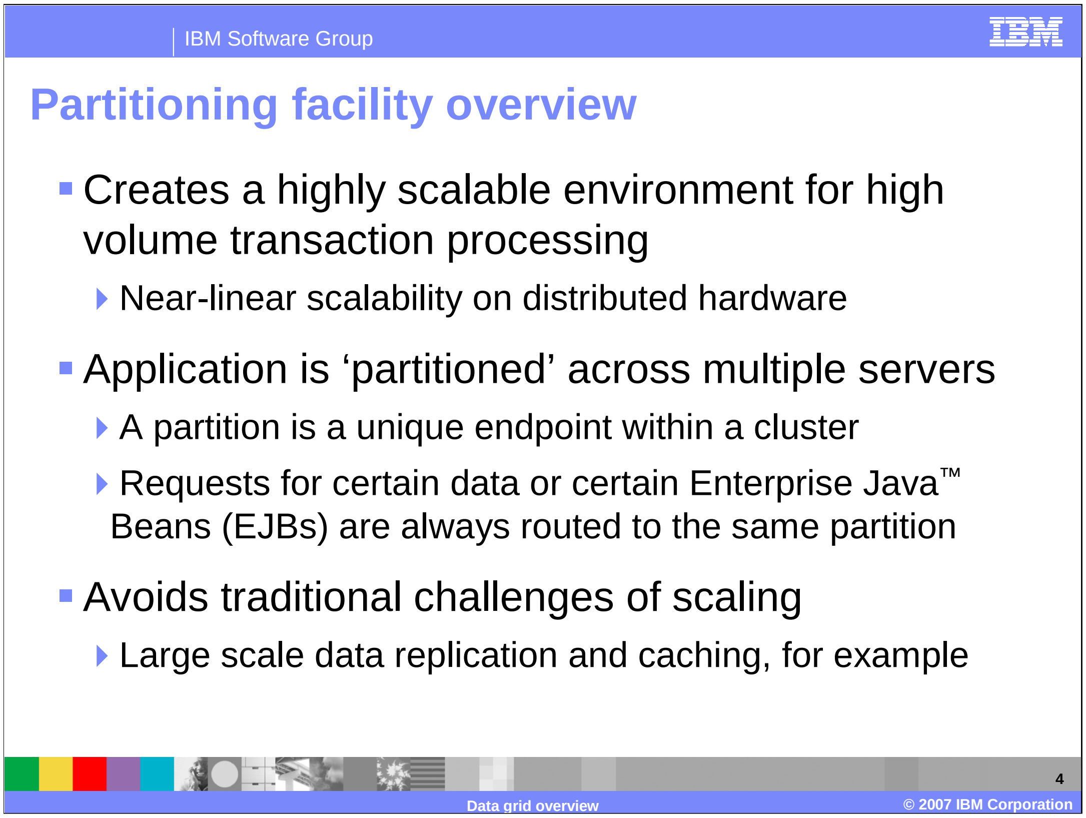
Section

Partitioning facility

3

Data grid overview

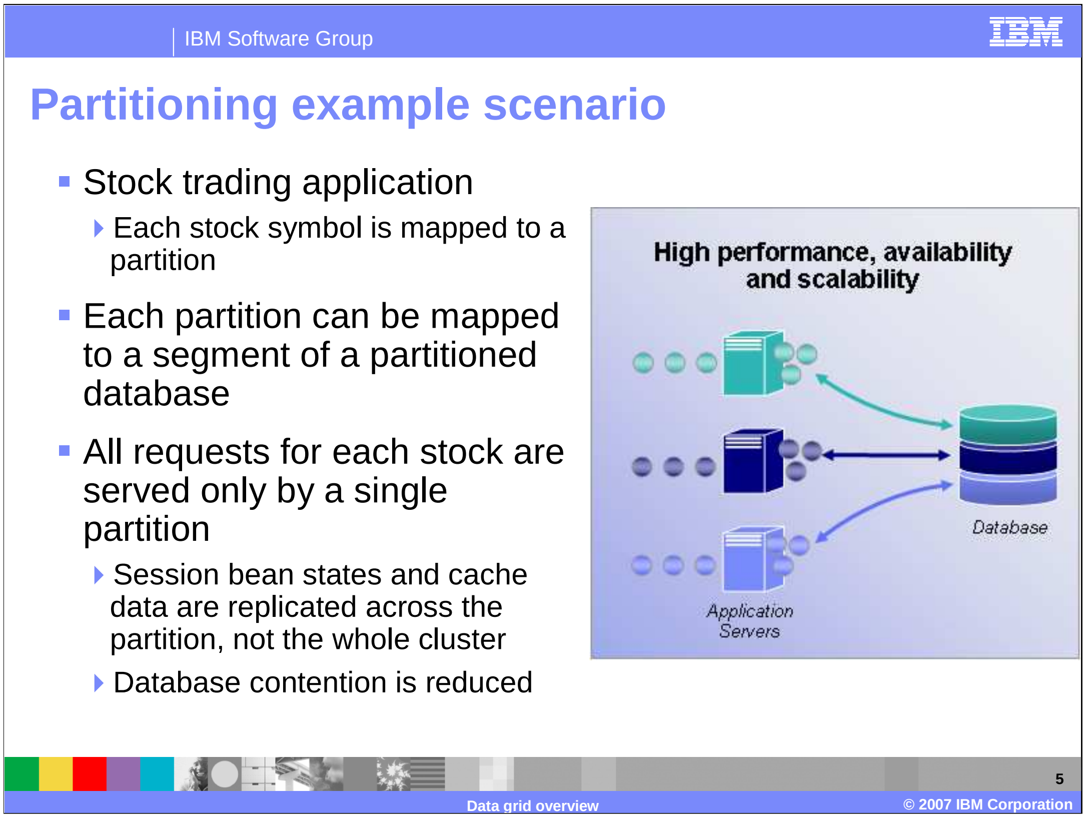
This section covers the partitioning facility feature of WebSphere Extended Deployment data grid.



Partitioning facility overview

- Creates a highly scalable environment for high volume transaction processing
  - Near-linear scalability on distributed hardware
- Application is ‘partitioned’ across multiple servers
  - A partition is a unique endpoint within a cluster
  - Requests for certain data or certain Enterprise Java™ Beans (EJBs) are always routed to the same partition
- Avoids traditional challenges of scaling
  - Large scale data replication and caching, for example

The partitioning facility is a programming framework and runtime environment that makes it possible for high transaction-volume applications to scale linearly by adding hardware capacity. To accomplish this, an application is partitioned across multiple servers in a cluster. Each partition is a uniquely addressable endpoint within the cluster, to which requests for certain data sets are always routed. Partitioning solves some of the traditional challenges of very large clustering, because it can reduce data contention and reduce the overhead of replicating shared data, like caches or state information.



Partitioning example scenario

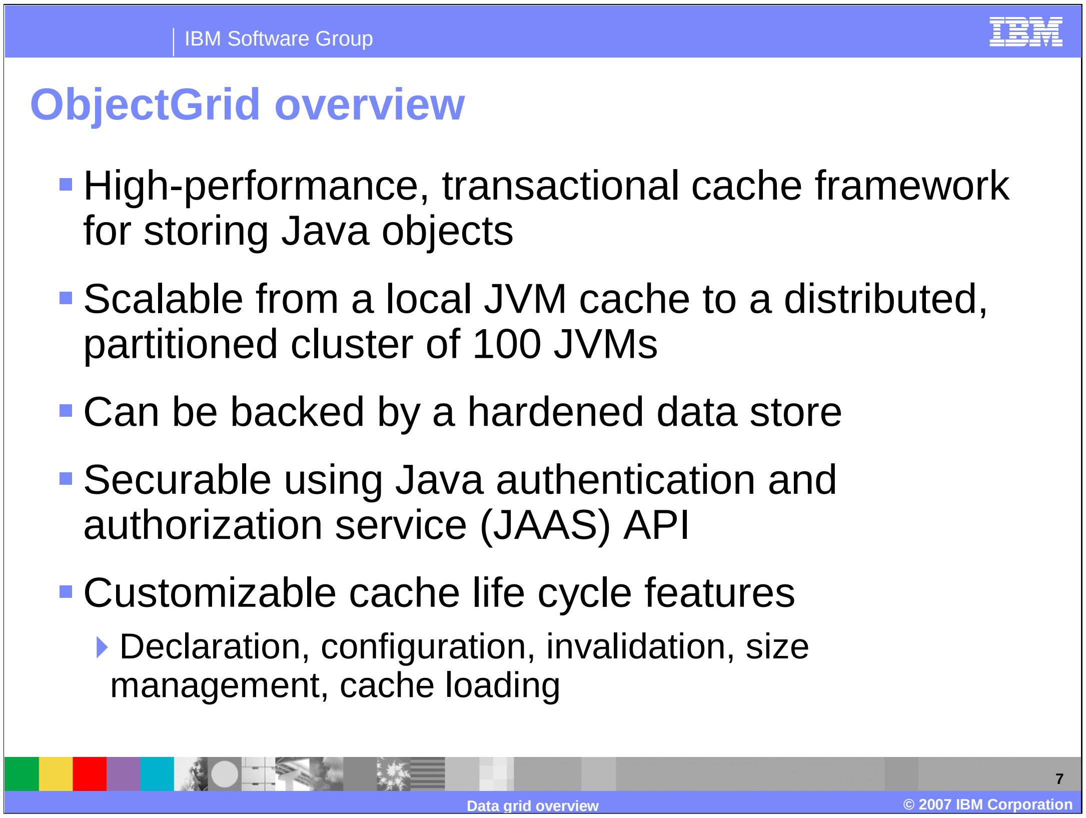
- Stock trading application
  - Each stock symbol is mapped to a partition
- Each partition can be mapped to a segment of a partitioned database
- All requests for each stock are served only by a single partition
  - Session bean states and cache data are replicated across the partition, not the whole cluster
  - Database contention is reduced

In this example scenario, consider a high-volume stock-trading application that has very high year-to-year traffic growth. The application is partitioned across a cluster of servers, such that buy or sell requests for each stock symbol are routed to a partition associated with that symbol. There may be more partitions than servers in the cluster, meaning each server is running multiple partitions. The work is effectively divided into unique data sets across the cluster, simplifying data replication and reducing overhead. Further, if the underlying database is designed such that separate database instances correspond to the data that would be accessed by individual partitions, then contention on the database will be reduced as well. This type of an environment scales extremely well. Adding additional hardware will reduce the number of partitions that must be hosted by each individual server, giving the cluster added capacity without additional overhead normally associated with large-scale clustering. This capability is particularly important for data bases other than DB2 or when data is maintained in memory only.

Section

Object grid

This section covers the object grid feature of WebSphere Extended Deployment Data Grid.



IBM Software Group

ObjectGrid overview

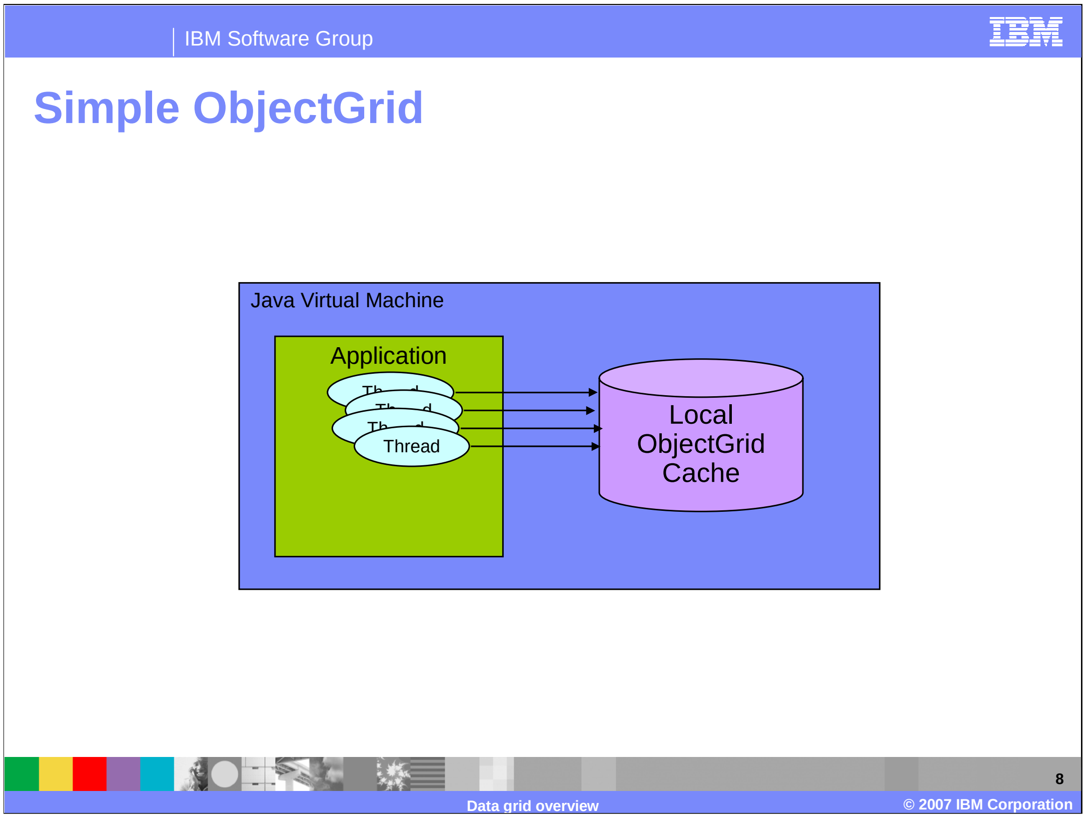
- High-performance, transactional cache framework for storing Java objects
- Scalable from a local JVM cache to a distributed, partitioned cluster of 100 JVMs
- Can be backed by a hardened data store
- Securable using Java authentication and authorization service (JAAS) API
- Customizable cache life cycle features
  - Declaration, configuration, invalidation, size management, cache loading

7

Data grid overview

© 2007 IBM Corporation

ObjectGrid is a WebSphere Extended Deployment technology that provides a high-performance, transactional cache framework for storing Java objects. An ObjectGrid can be used as a generic object cache, and optionally persisted to a hardened store. It can in the same fashion also be used as a local cache for objects stored in a database. ObjectGrid is a highly customizable feature, with interfaces provided for custom data loaders, invalidation and size management schemes, and more.



Simple ObjectGrid

Java Virtual Machine

Application

Thread

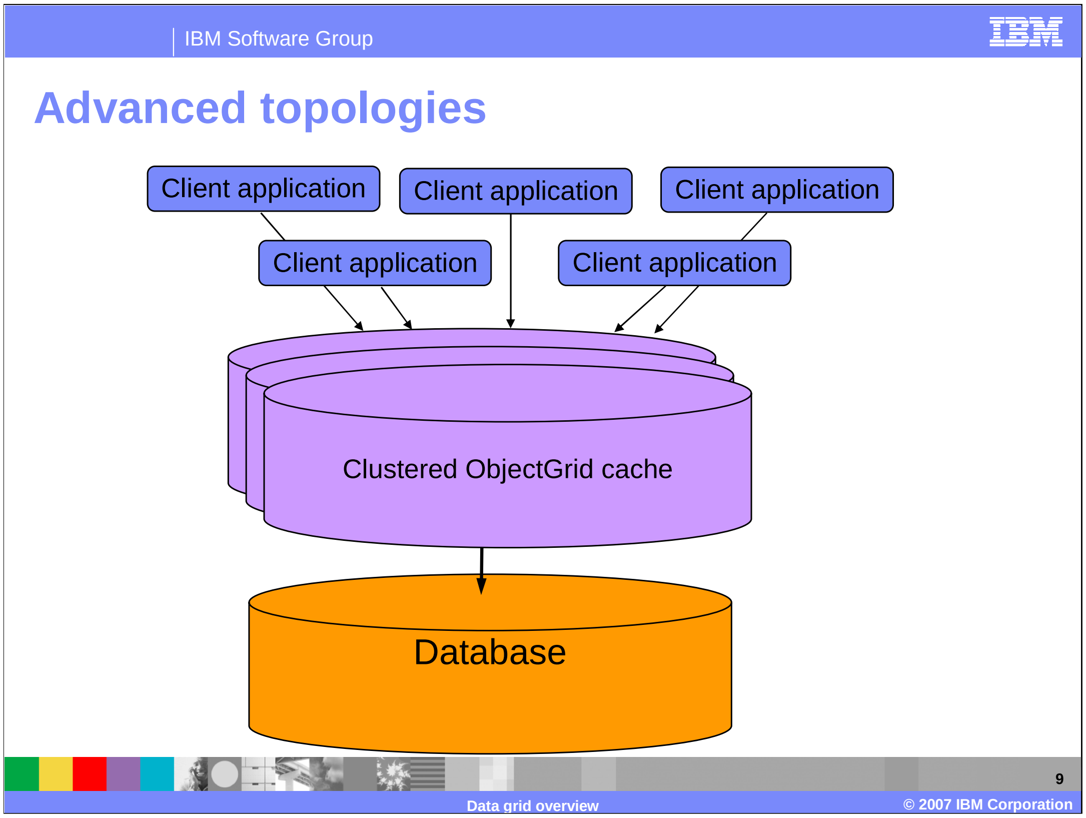
Thread

Thread

Thread

Local ObjectGrid Cache

In the simplest case, the ObjectGrid can be used as an in-memory cache. It can be used to provide consistent, transactional access to temporary data within a single Java Virtual Machine. This can especially benefit high-concurrency applications where multiple threads need to access and modify transient data. The data kept in a local ObjectGrid can be indexed and retrieved using ObjectGrid's query support.



IBM Software Group

Advanced topologies

Client application

Client application

Client application

Client application

Client application

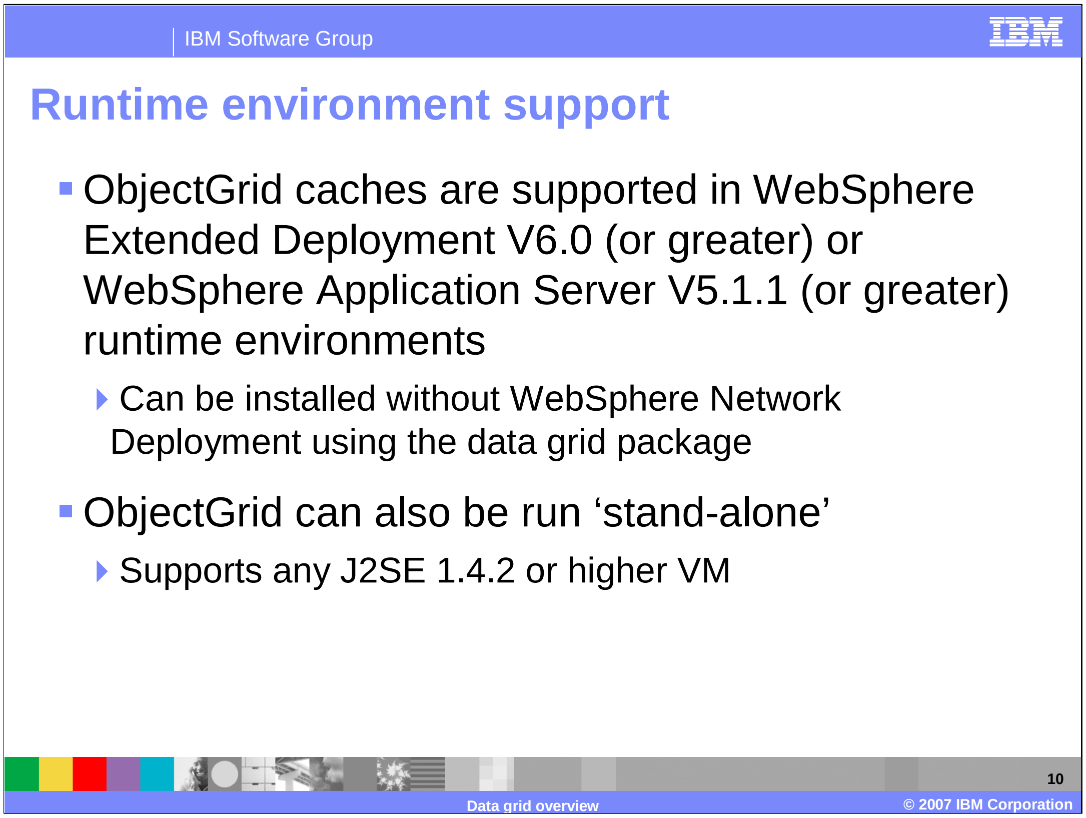
Clustered ObjectGrid cache

Database

9

Data grid overview

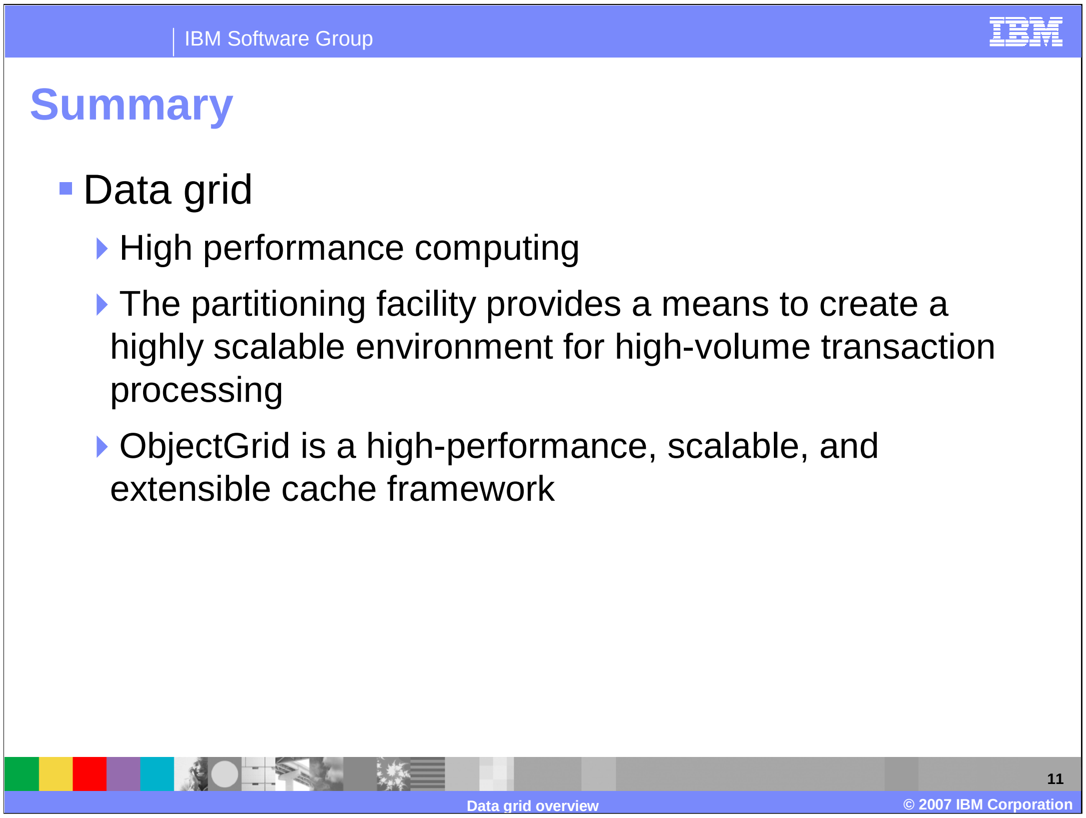
ObjectGrid also supports more complicated topologies, supporting highly available distributed, partitioned and replicated cache which can scale to thousands of containers containing terabytes of data. Distributed ObjectGrid caches offer increased performance, availability and scalability.   Local and distributed ObjectGrid topologies both provide the same application programming model for interacting with the cache.



Runtime environment support

- ObjectGrid caches are supported in WebSphere Extended Deployment V6.0 (or greater) or WebSphere Application Server V5.1.1 (or greater) runtime environments
  - Can be installed without WebSphere Network Deployment using the data grid package
- ObjectGrid can also be run ‘stand-alone’
  - Supports any J2SE 1.4.2 or higher VM

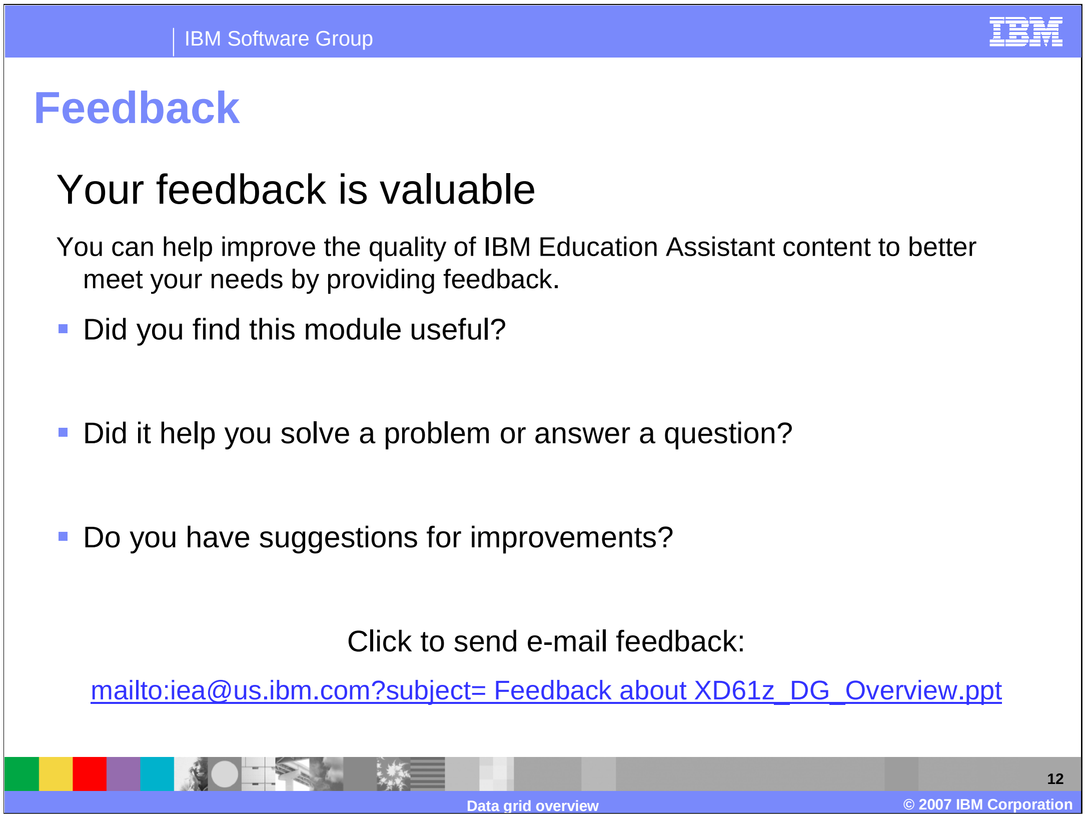
While ObjectGrid technology is provided only with WebSphere Extended Deployment Data Grid, ObjectGrid caching is also supported in a WebSphere Application Server Network Deployment version 5.1 (or higher) server by including the ObjectGrid library in your application classpath. Similarly, ObjectGrid can run in a stand-alone JVM or other application server product. ObjectGrid requires a Java 1.4.2. virtual machine to run, but can take advantage of capabilities of newer Java versions, such as Java 5 annotations.



Summary

- Data grid
  - High performance computing
  - The partitioning facility provides a means to create a highly scalable environment for high-volume transaction processing
  - ObjectGrid is a high-performance, scalable, and extensible cache framework

In summary, WebSphere Extended Deployment Data Grid is an add-on product that provides several advanced features for WebSphere Application Server to enable highly scalable, high transaction applications. The partitioning facility provides a programming model and runtime environment for implementing highly scalable solutions for transaction processing. ObjectGrid is a new high-performance, transactional, and extensible object cache for Java applications.



IBM Software Group

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 e-mail feedback:

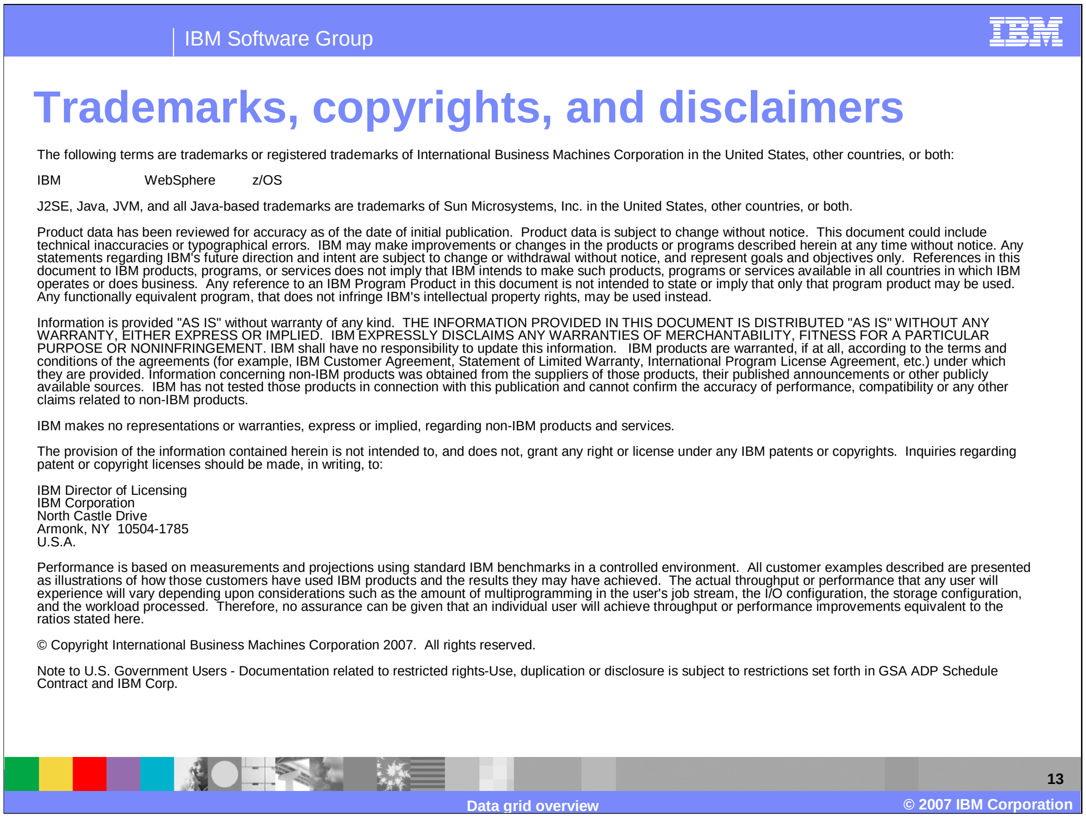
mailto:iea@us.ibm.com?subject= Feedback about XD61z_DG_Overview.ppt

12

Data grid overview

© 2007 IBM Corporation

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



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 WebSphere z/OS

J2SE, Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.

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 (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.

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.