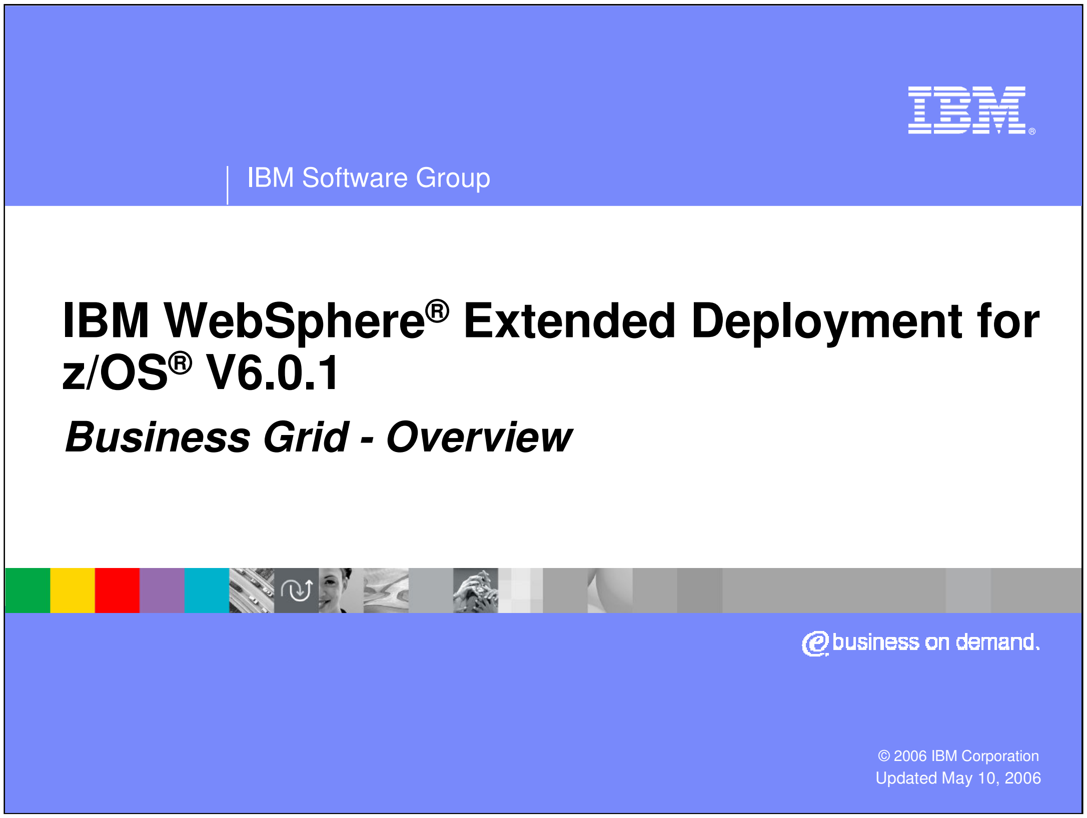


IBM Software Group

IBM WebSphere® Extended Deployment for z/OS® V6.0.1

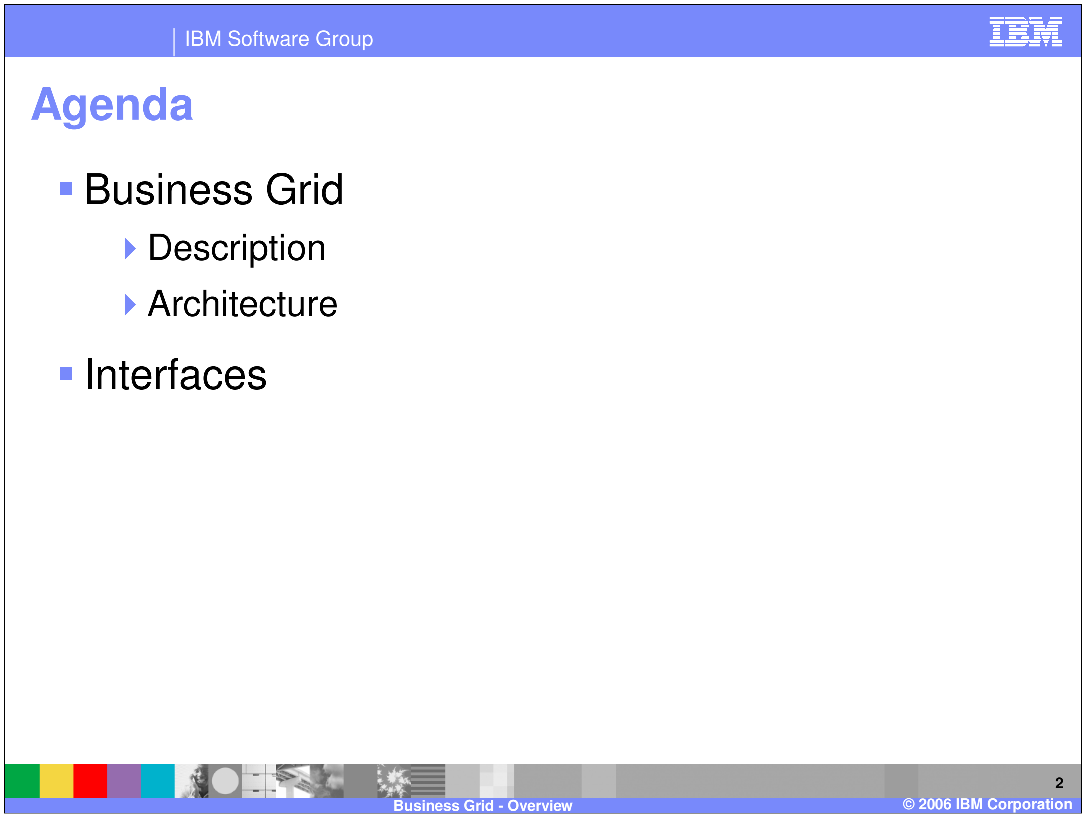
Business Grid - Overview

business on demand.

© 2006 IBM Corporation

Updated May 10, 2006

This presentation will provide an overview of the Business Grid component offered in WebSphere Extended Deployment for z/OS Version 6.0.1.



Agenda

- Business Grid
  - Description
  - Architecture
- Interfaces

© 2006 IBM Corporation

This presentation will explain the basic uses of the Business Grid, and the architecture behind the component. It will also explain the interfaces that are available for working with the Business Grid.

IBM Software Group

Section

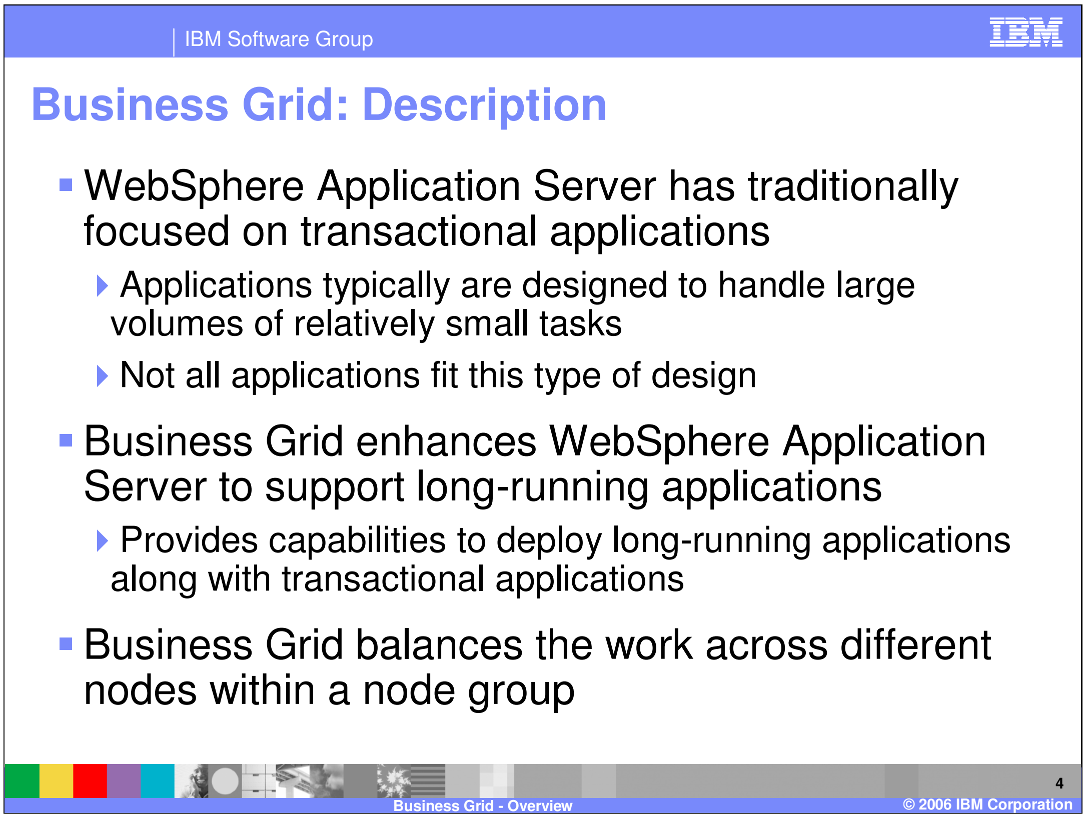
Description

3

Business Grid - Overview

© 2006 IBM Corporation

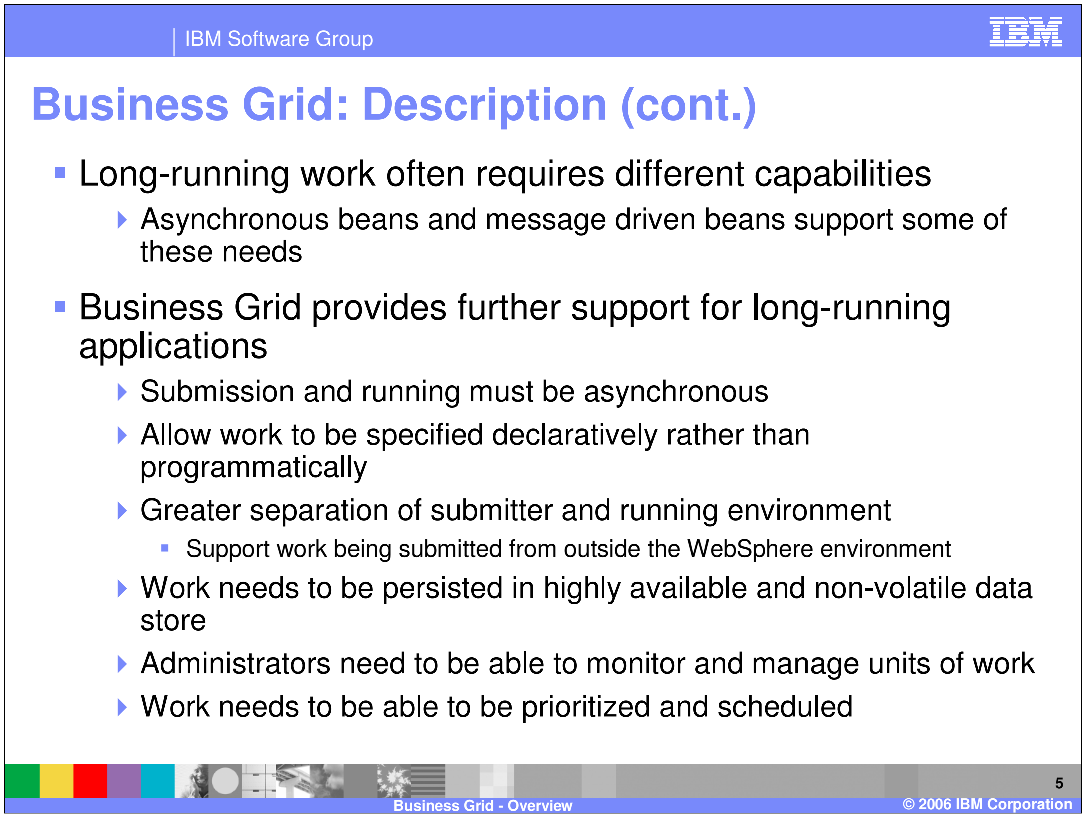
This section will describe the Business Grid component.



Business Grid: Description

- WebSphere Application Server has traditionally focused on transactional applications
  - Applications typically are designed to handle large volumes of relatively small tasks
  - Not all applications fit this type of design
- Business Grid enhances WebSphere Application Server to support long-running applications
  - Provides capabilities to deploy long-running applications along with transactional applications
- Business Grid balances the work across different nodes within a node group

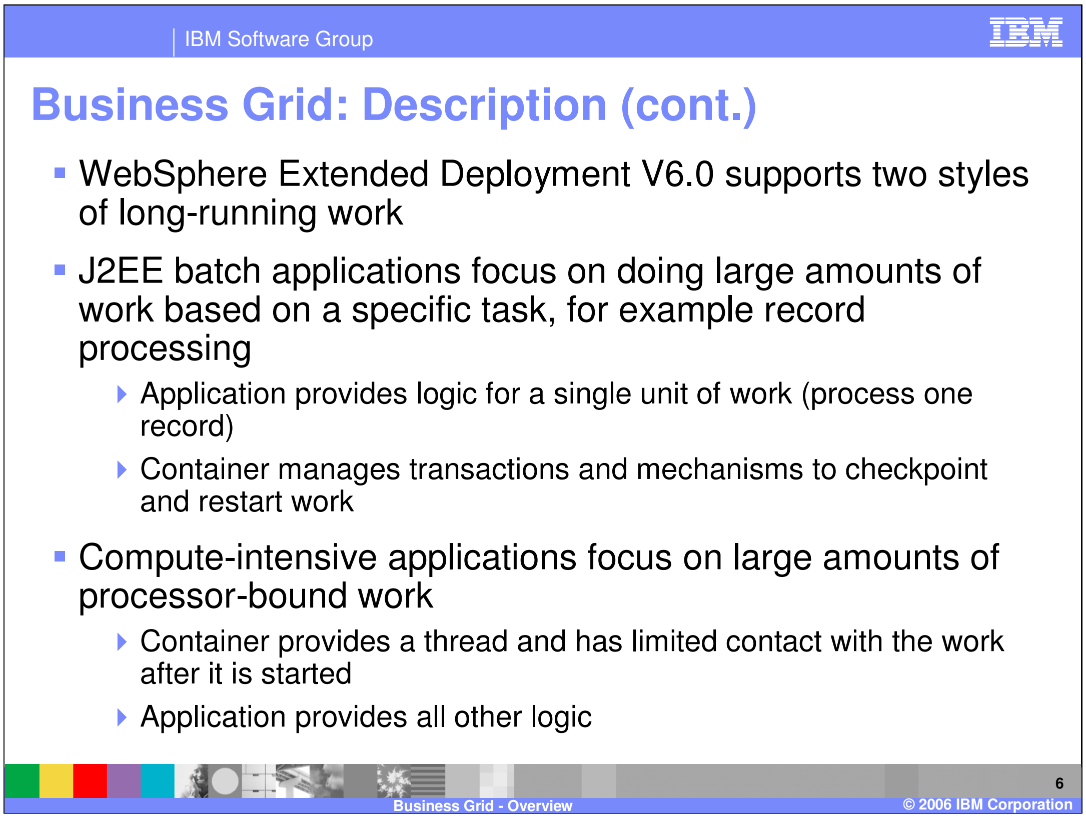
WebSphere Application Server and Java™ 2 Enterprise Edition (J2EE) servers in general have classically focused on lightweight, transactional work. Typically, an individual request can be handled in a few seconds of processor time and relatively small amounts of memory. However, other styles of long-running applications require more resources and different types of support from the runtime environment. Business grid provides WebSphere Application Server support for long-running applications. Long-running work might take hours or even days to complete and consumes large amounts of memory or processing power while it executes. Business grid provides the capability to deploy different types of applications to different nodes within your environment, and balances the work based on policy information.



Business Grid: Description (cont.)

- Long-running work often requires different capabilities
  - Asynchronous beans and message driven beans support some of these needs
- Business Grid provides further support for long-running applications
  - Submission and running must be asynchronous
  - Allow work to be specified declaratively rather than programmatically
  - Greater separation of submitter and running environment
    - Support work being submitted from outside the WebSphere environment
  - Work needs to be persisted in highly available and non-volatile data store
  - Administrators need to be able to monitor and manage units of work
  - Work needs to be able to be prioritized and scheduled

Due to the nature of their work, long-running applications typically require different capabilities from the environment they are running within. Often the submission of a long-running job must be asynchronous from the job being run. This separation of the submission and running environment should also allow the submission of work from outside the WebSphere environment. Once long-running work has begun, it will need to be persisted to highly available data stores. Often administrators will also require the ability to monitor and manage the jobs that have been started. The environment must also be able to schedule and prioritize the work that needs to be done based on policy information users set. A number of these capabilities can be supported with existing J2EE features, such as asynchronous and message driven beans (MDBs). The business grid provides an enhanced environment for long-running applications, making it easier to support this type of work in a WebSphere environment.



Business Grid: Description (cont.)

- WebSphere Extended Deployment V6.0 supports two styles of long-running work
- J2EE batch applications focus on doing large amounts of work based on a specific task, for example record processing
  - Application provides logic for a single unit of work (process one record)
  - Container manages transactions and mechanisms to checkpoint and restart work
- Compute-intensive applications focus on large amounts of processor-bound work
  - Container provides a thread and has limited contact with the work after it is started
  - Application provides all other logic

© 2006 IBM Corporation

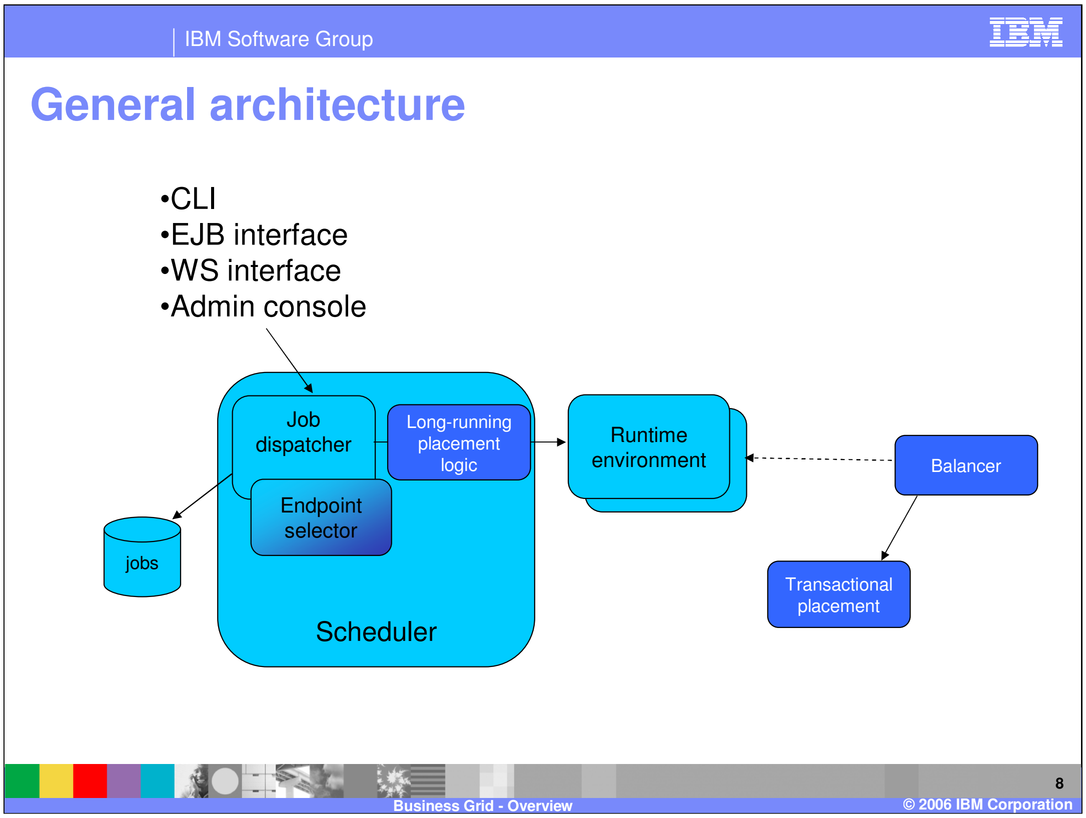
The business grid component supports two types of long-running applications in WebSphere Extended Deployment V6; J2EE batch and compute-intensive. A typical J2EE batch application will do large amounts of work based on repetitive tasks. A J2EE batch application must provide the logic for a single unit of work, and the container provides support to run the job with transactions and the ability to checkpoint and restart the J2EE batch process. For example, a typical J2EE batch application would process records and the application would provide the logic to process a single record. The environment will then manage the process of repeatedly performing the task for a large number of records. Computationally intensive applications perform work that requires large amounts of system resources, in particular processor usage and memory. In this case, the application provides all the logic for the work and the business grid makes sure that the application is appropriately situated within the environment.

Section

Architecture

© 2006 IBM Corporation

This section will explain the architecture of the Business Grid.



IBM Software Group

IBM

General architecture

- CLI
- EJB interface
- WS interface
- Admin console

Job dispatcher

Long-running placement logic

Endpoint selector

Scheduler

jobs

Runtime environment

Balancer

Transactional placement

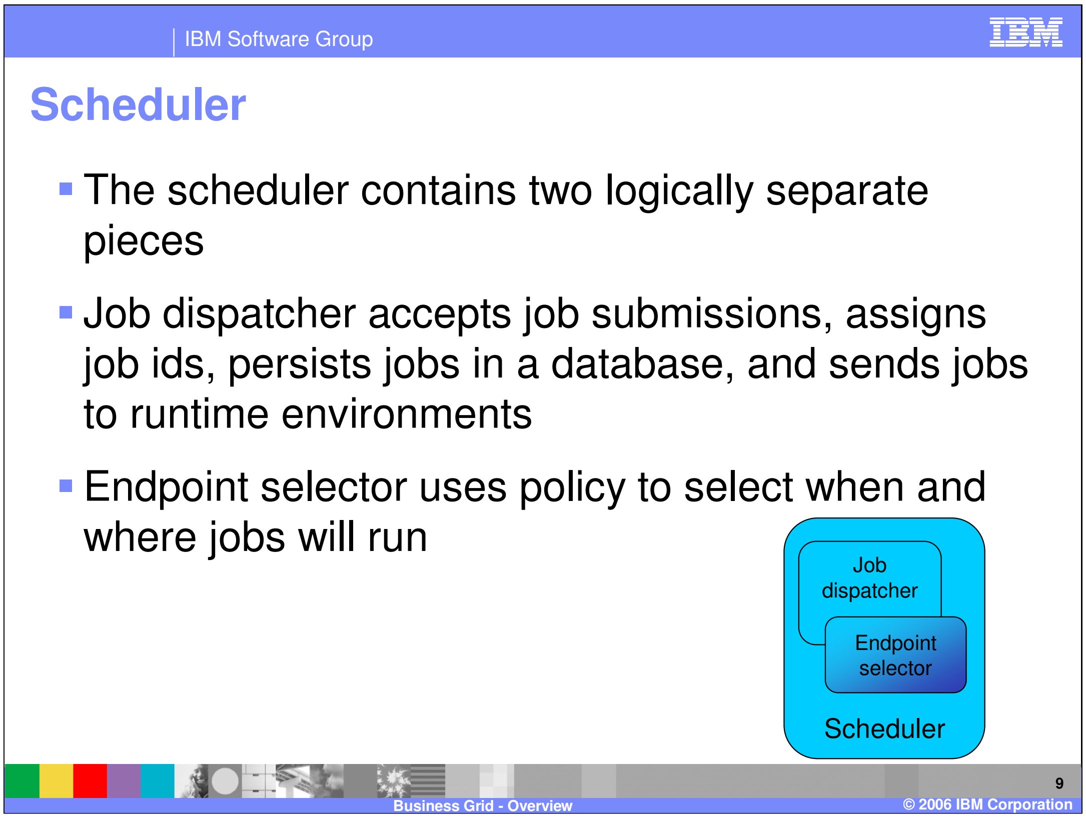
8

Business Grid - Overview

© 2006 IBM Corporation

This graphic illustrates the components involved with the business grid. Each of the individual pieces shown here will be discussed further on the next slides. You primarily interact with the business grid by setting policy information for your environment and by interacting with the J2EE

Job Dispatcher using one or more of the available interfaces. The command line interface allows you to submit and control long-running jobs in the system. The enterprise bean and Web service interfaces provide similar functionality to both J2EE and non-J2EE programs using programmatic interfaces. The administrative console provides a Graphical User Interface (GUI) that allows WebSphere administrators and operators to perform job management functions. All of the functionality from the other interfaces except submitting and storing jobs is also available on the administrative console. In addition, the input stream is converted from EBCDIC to Java objects with jZoS, a very high performance tool.



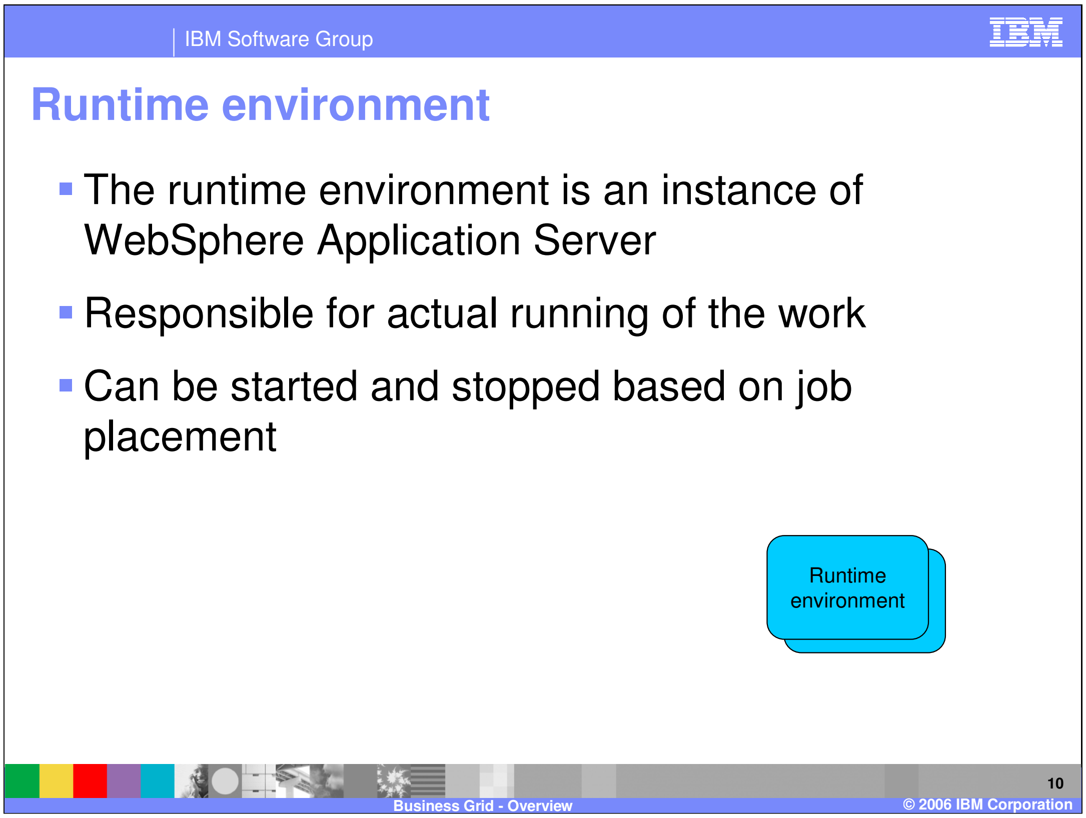
Scheduler

- The scheduler contains two logically separate pieces
- Job dispatcher accepts job submissions, assigns job ids, persists jobs in a database, and sends jobs to runtime environments
- Endpoint selector uses policy to select when and where jobs will run

Job dispatcher
Endpoint selector
Scheduler

© 2006 IBM Corporation

The long-running scheduler is responsible for accepting, persisting and scheduling long-running jobs. It manages the job database, assigns job ids and selects where and when jobs should be run. As part of performing this function, the long-running scheduler is also responsible for starting and stopping instances of long-running dynamic clusters as dictated by the jobs to be run and administrator-defined service policies. This is analogous to the function provided by the Application Placement Controller, or APC, for transactional J2EE applications. The long-running scheduler is a J2EE application that can be made highly available by using APC-provided functionality.



Runtime environment

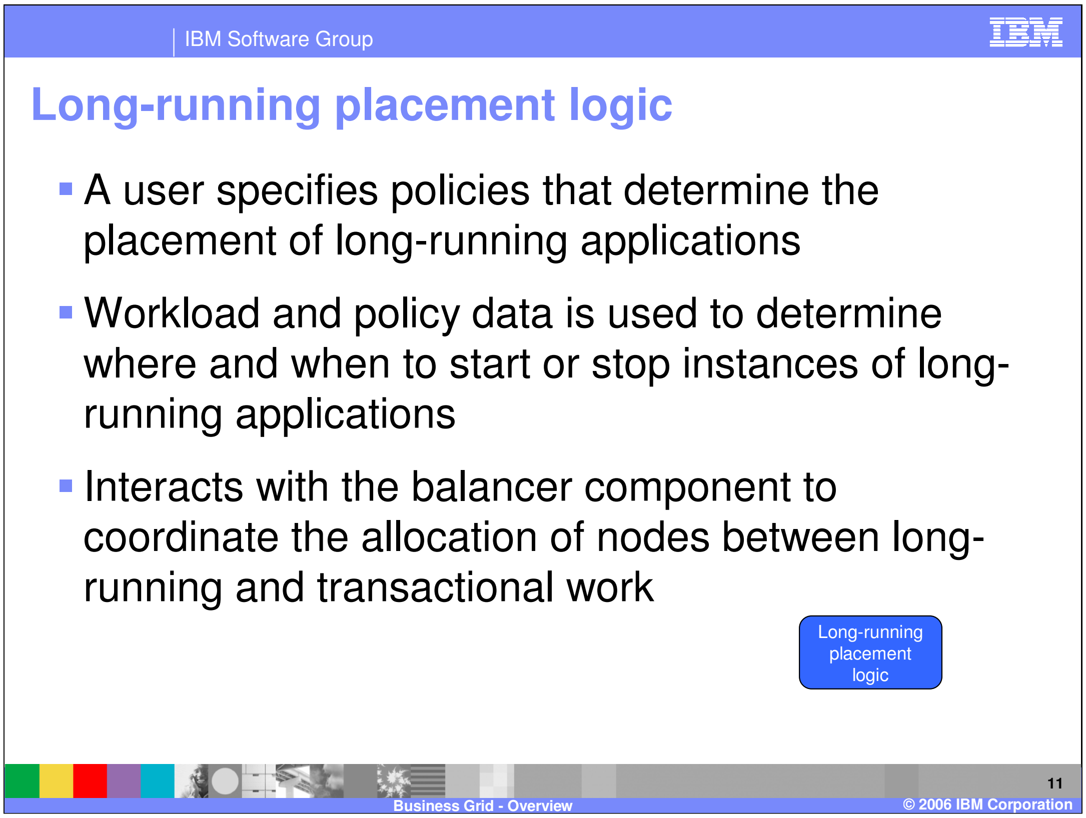
- The runtime environment is an instance of WebSphere Application Server
- Responsible for actual running of the work
- Can be started and stopped based on job placement

Runtime environment

The long running applications need two special runtime environments. WebSphere Extended Deployment V6 provides both environments in a single J2EEapplication, LREE.ear, which is deployed to the dynamic clusters that host long-running applications:

The computationally-intensive runtime environment supports long-running applications that are expected to consume large amounts of processor time. This runtime environment provides a relatively simple programming model based on asynchronous beans.

The J2EE batch runtime environment supports J2EE batch-oriented applications. These applications are expected to perform record processing similar to more traditional J2EE applications, but are driven by J2EE batch inputs rather than interactive users. This environment builds on familiar J2EE entity beans to provide J2EE batch applications a programming model that supports container-managed restartable processing and the ability to pause and cancel running jobs.



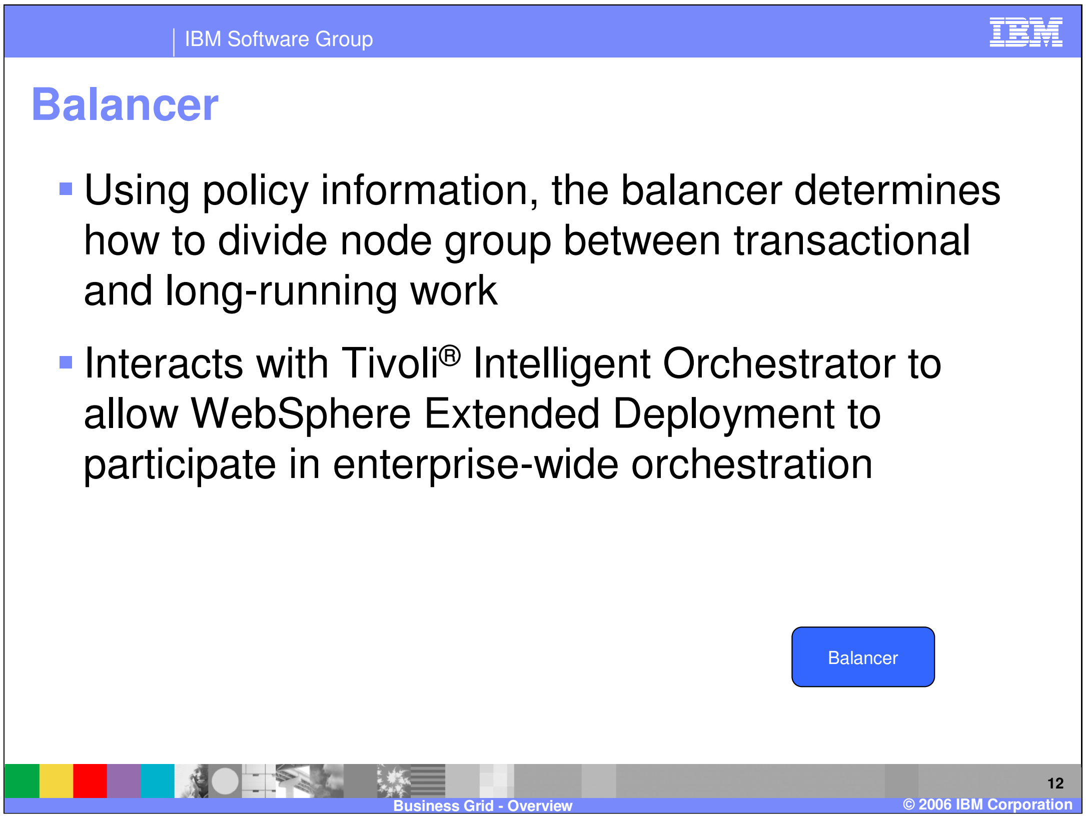
Long-running placement logic

- A user specifies policies that determine the placement of long-running applications
- Workload and policy data is used to determine where and when to start or stop instances of long-running applications
- Interacts with the balancer component to coordinate the allocation of nodes between long-running and transactional work

Long-running placement logic

© 2006 IBM Corporation

Nodes that are to be used for long-running dynamic clusters are assigned to the long-running placement logic, which is part of the long-running scheduler. The long-running placement logic determines which long-running dynamic clusters should be started on these nodes. The balancer may switch a node between long-running and transactional work over time, but WebSphere Extended Deployment will never attempt to automatically start both types of work on the node concurrently. If a node is already running when the balancer component initializes, the balancer component attempts to determine how to assign the node based on the dynamic clusters that are running there. If no dynamic clusters are started on the node, the default assignment is to the application placement controller (transactional work).



IBM Software Group

Balancer

- Using policy information, the balancer determines how to divide node group between transactional and long-running work
- Interacts with Tivoli® Intelligent Orchestrator to allow WebSphere Extended Deployment to participate in enterprise-wide orchestration

Business Grid - Overview

12

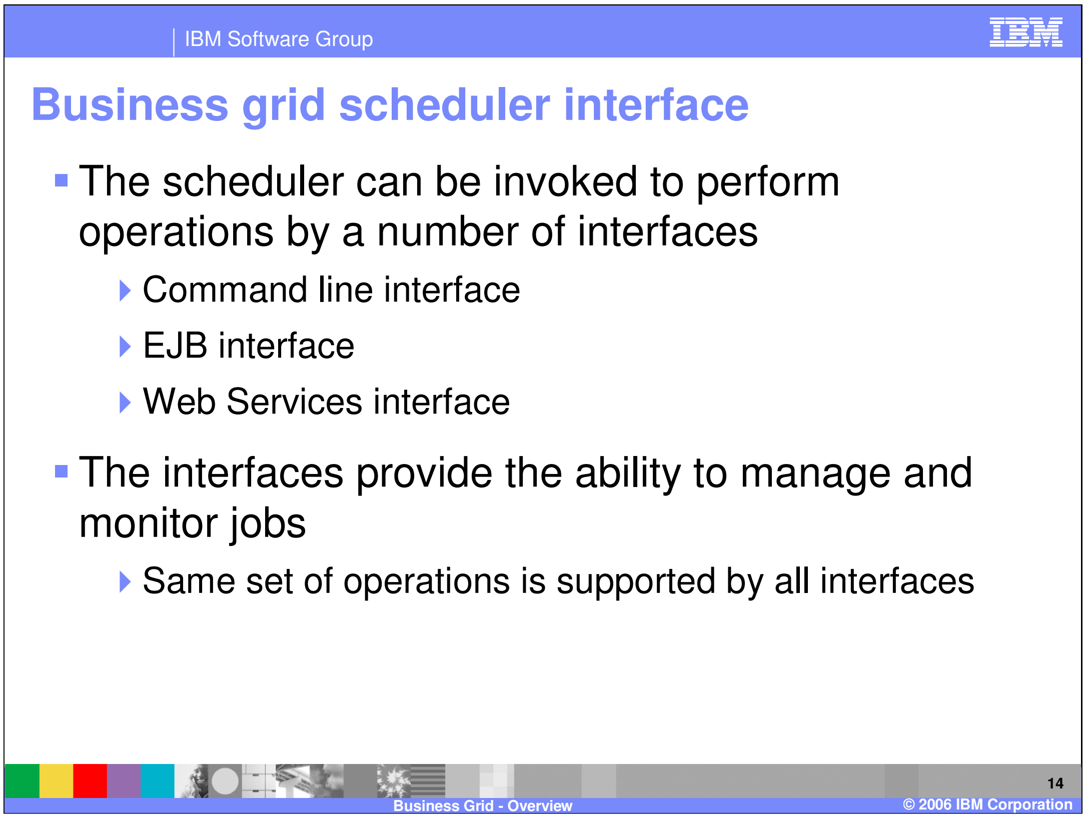
© 2006 IBM Corporation

The balancer component serves as an arbiter of long-running and transactional work within a node group. Due to the nature of long-running work, co-locating it on the same node with transactional work usually has a negative impact on the performance of the transactional work. The balancer makes decisions about which nodes should be performing transactional work and which can be used for long-running work. These decisions are based on a number of factors, including how well the service policies for the two types of work are being met and input from both z/OS and WebSphere workload managers.

Section

Interfaces

This section will discuss the interfaces provided for the Business Grid.

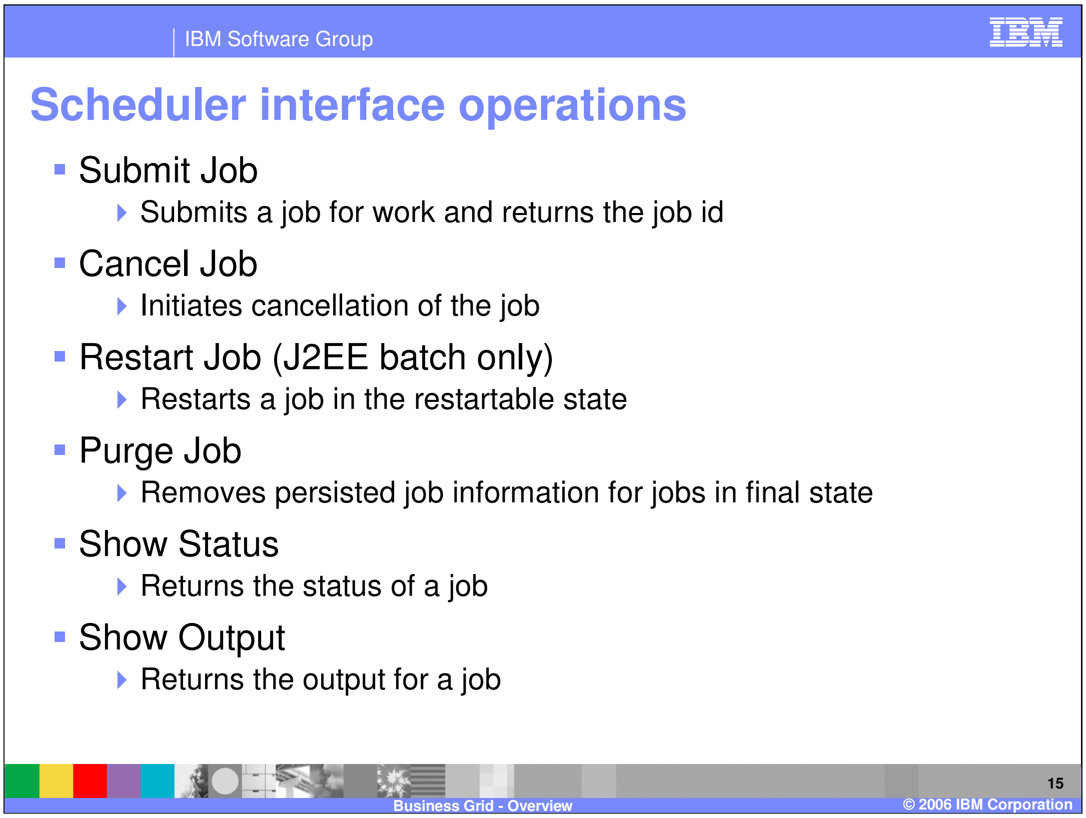


Business grid scheduler interface

- The scheduler can be invoked to perform operations by a number of interfaces
  - Command line interface
  - EJB interface
  - Web Services interface
- The interfaces provide the ability to manage and monitor jobs
  - Same set of operations is supported by all interfaces

© 2006 IBM Corporation

As depicted on the architectural graphic, a number of interfaces are provided to interact with the business grid scheduler component. These interfaces are used to manage and monitor long-running applications within your environment. The same operations, which are detailed on the next slide, are supported by all the interfaces.



IBM Software Group

Scheduler interface operations

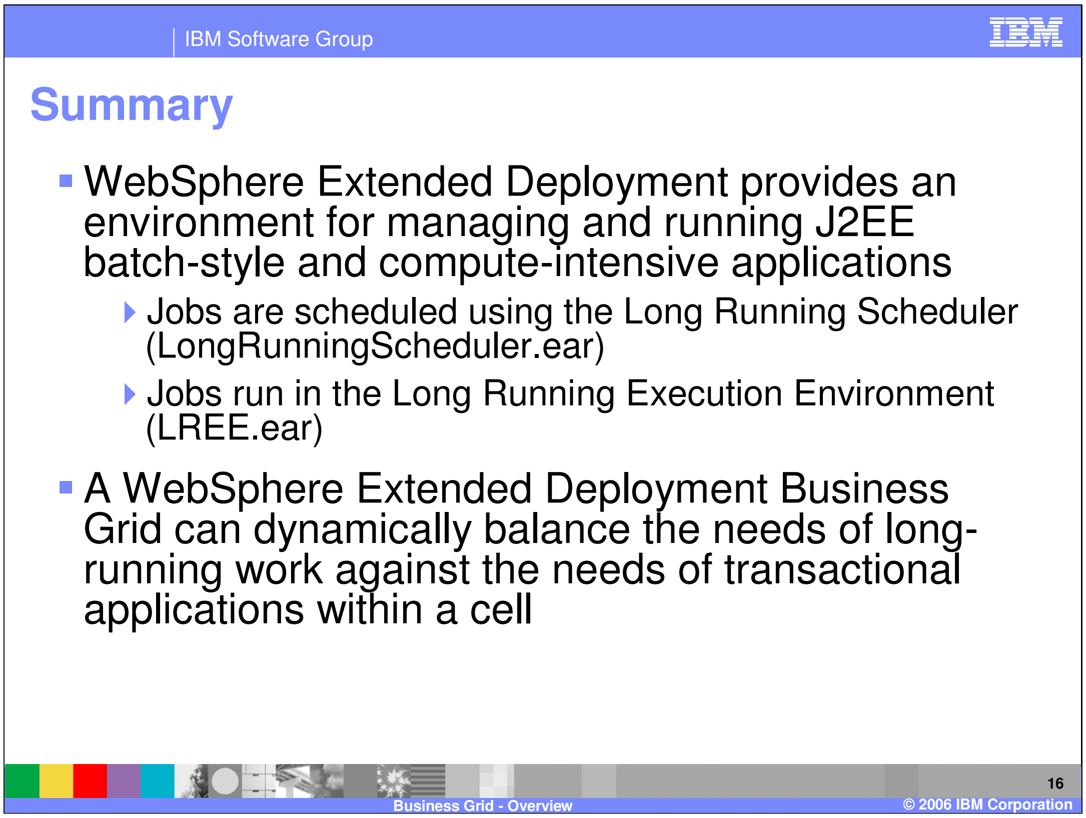
- Submit Job
  - Submits a job for work and returns the job id
- Cancel Job
  - Initiates cancellation of the job
- Restart Job (J2EE batch only)
  - Restarts a job in the restartable state
- Purge Job
  - Removes persisted job information for jobs in final state
- Show Status
  - Returns the status of a job
- Show Output
  - Returns the output for a job

15

Business Grid - Overview

© 2006 IBM Corporation

Using the provided operations, you can submit or cancel a job to the business grid scheduler. When the cancel operation is used, the state for the job is changed to cancel pending, until the business grid can take the appropriate steps to stop the job. You can also restart a J2EE batch job, to continue it's work. There is a purge job command to remove the persisted job information for a finalized job. There are also a number of options for receiving information from a running job, show status and show output.



Summary

- WebSphere Extended Deployment provides an environment for managing and running J2EE batch-style and compute-intensive applications
  - Jobs are scheduled using the Long Running Scheduler (LongRunningScheduler.ear)
  - Jobs run in the Long Running Execution Environment (LREE.ear)
- A WebSphere Extended Deployment Business Grid can dynamically balance the needs of long-running work against the needs of transactional applications within a cell

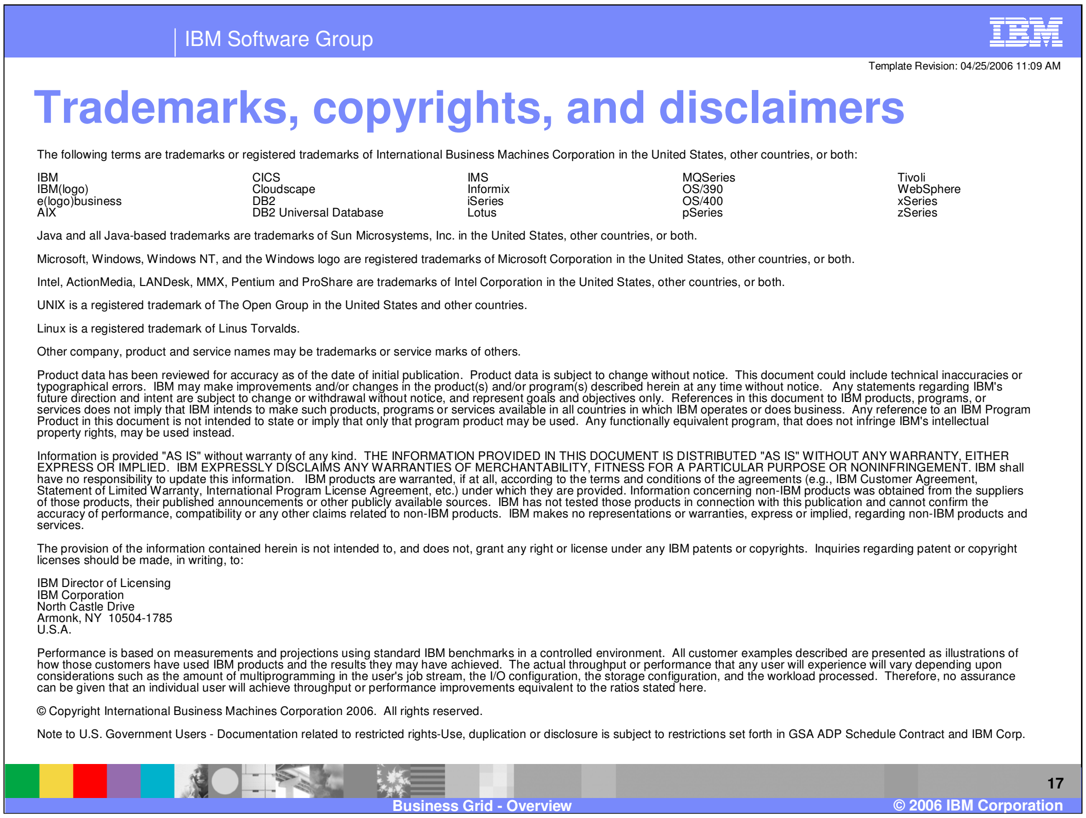
© 2006 IBM Corporation

In summary, this presentation:

•Explained the benefits of the business grid provided by WebSphere Extended Deployment V6.0.1.

•Discussed the differences between computationally intensive and J2EE batch programs.

•Explained the capabilities for the business grid to balance long-running work with transactional work in an environment.



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

© Copyright International Business Machines Corporation 2006. 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.