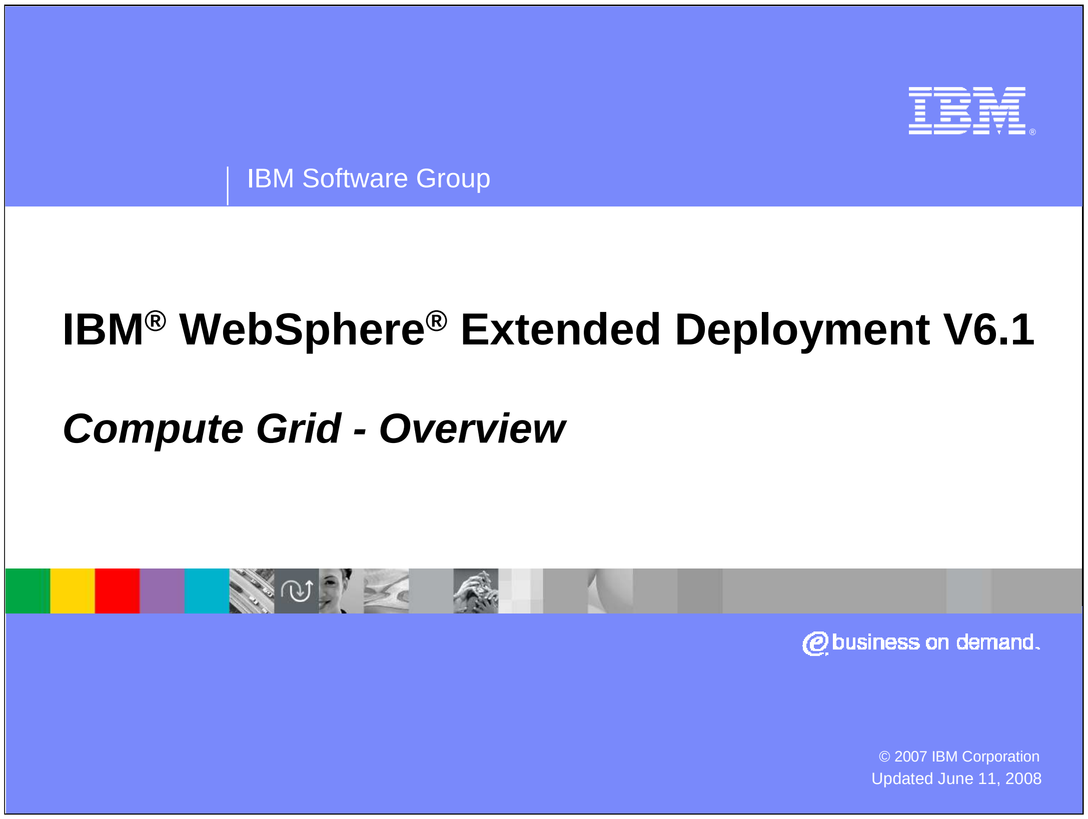


IBM Software Group

IBM® WebSphere® Extended Deployment V6.1

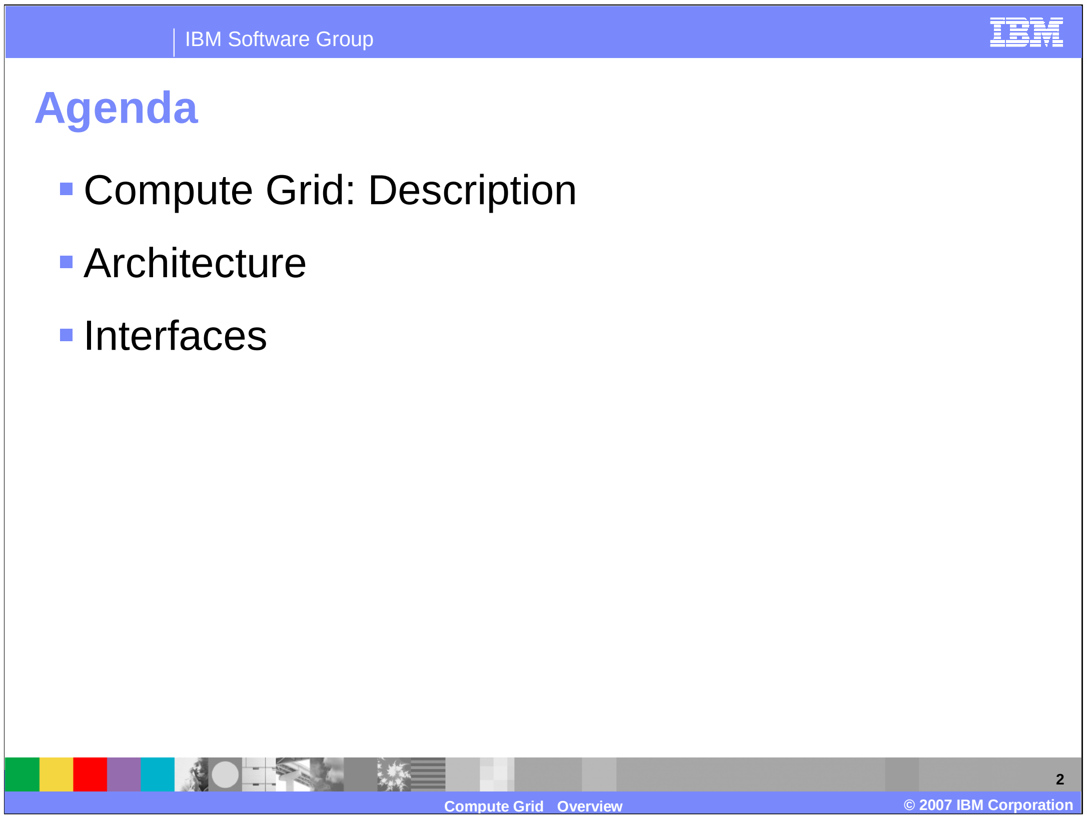
Compute Grid - Overview

business on demand.

This presentation will provide an overview of the Compute Grid component offered in WebSphere Extended Deployment version 6.1. Prior versions of this component were known as Business Grid.

This module references WebSphere Extended Deployment Operations Optimization, which is now called WebSphere Virtual Enterprise.

Though the module uses the previous names, the technical material covered is still accurate.


IBM Software Group

Agenda

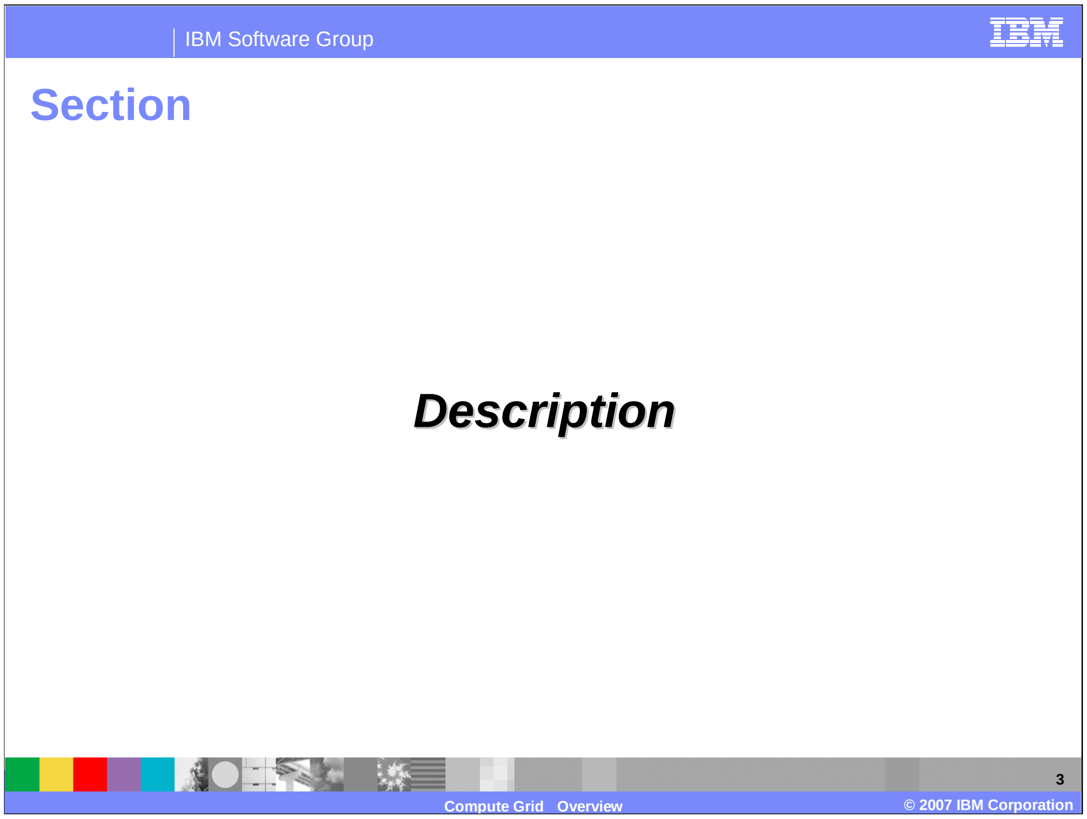
- Compute Grid: Description
- Architecture
- Interfaces

2

Compute Grid Overview

© 2007 IBM Corporation

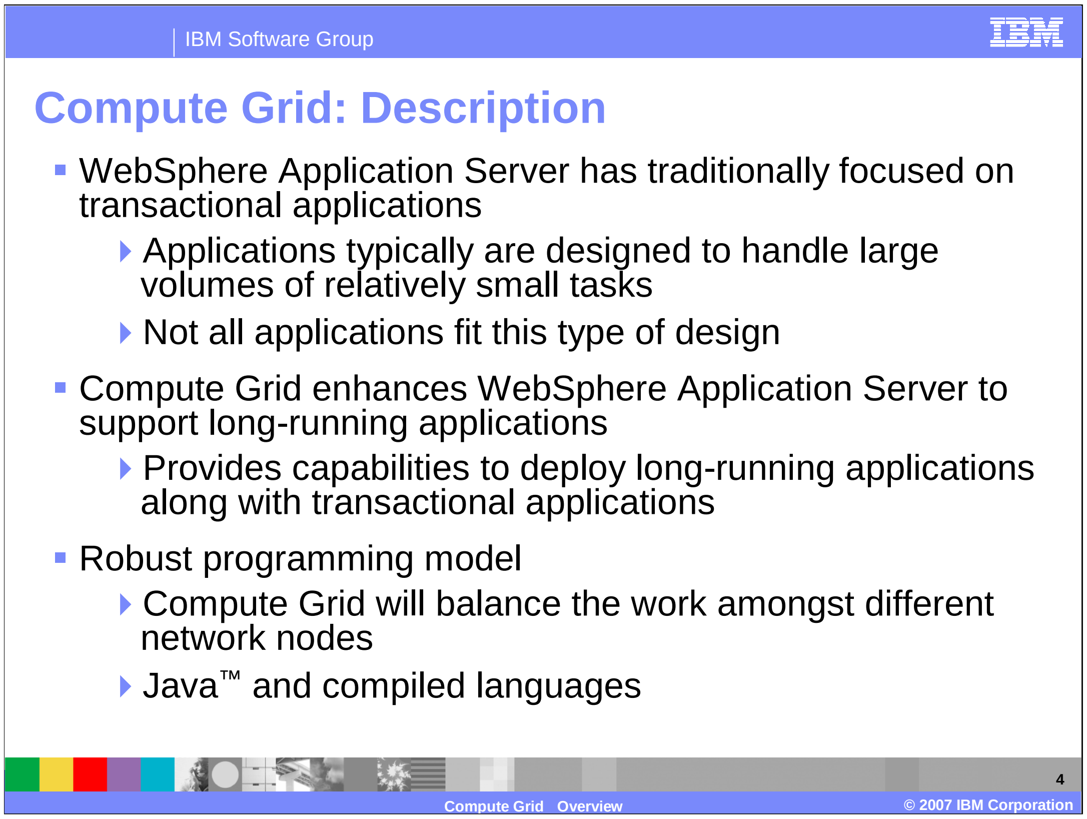
This presentation will explain the basic uses of the Compute Grid and the architecture behind the component. It will also introduce the interfaces that are available for working with the Compute Grid.



Section

Description

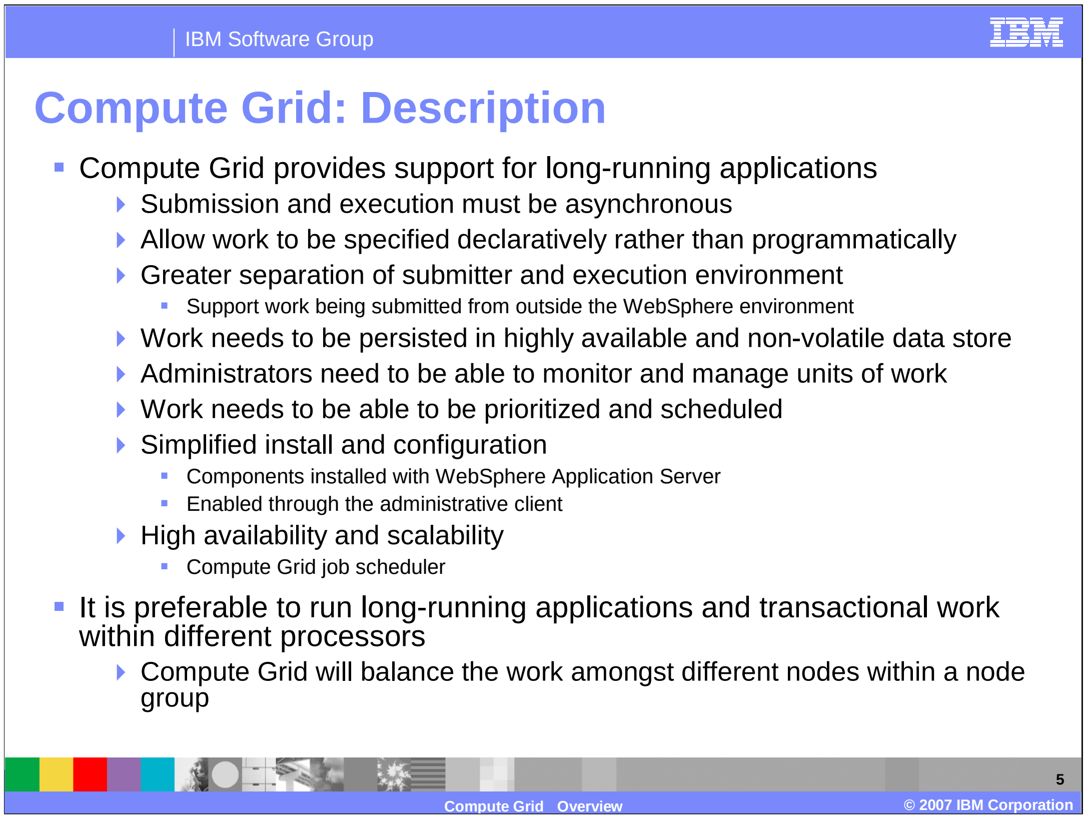
This section will describe the grid computing in WebSphere extended deployment.



Compute 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
- Compute Grid enhances WebSphere Application Server to support long-running applications
  - Provides capabilities to deploy long-running applications along with transactional applications
- Robust programming model
  - Compute Grid will balance the work amongst different network nodes
  - Java™ and compiled languages

WebSphere Application Server and 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. The WebSphere Compute Grid provides scheduling and execution control of background activities in a grid computing environment. A set of managed background activities is called a grid job. Autonomic managers provide resource management and workload balancing to maximize utilization and throughput of grid jobs. These background workloads may be composed of any combination of J2EE components and arbitrary background programs written in Java; compiled languages such as C++, Fortran, or COBOL; or script. WebSphere Compute Grid can balance the work based on policy information.


IBM Software Group

Compute Grid: Description

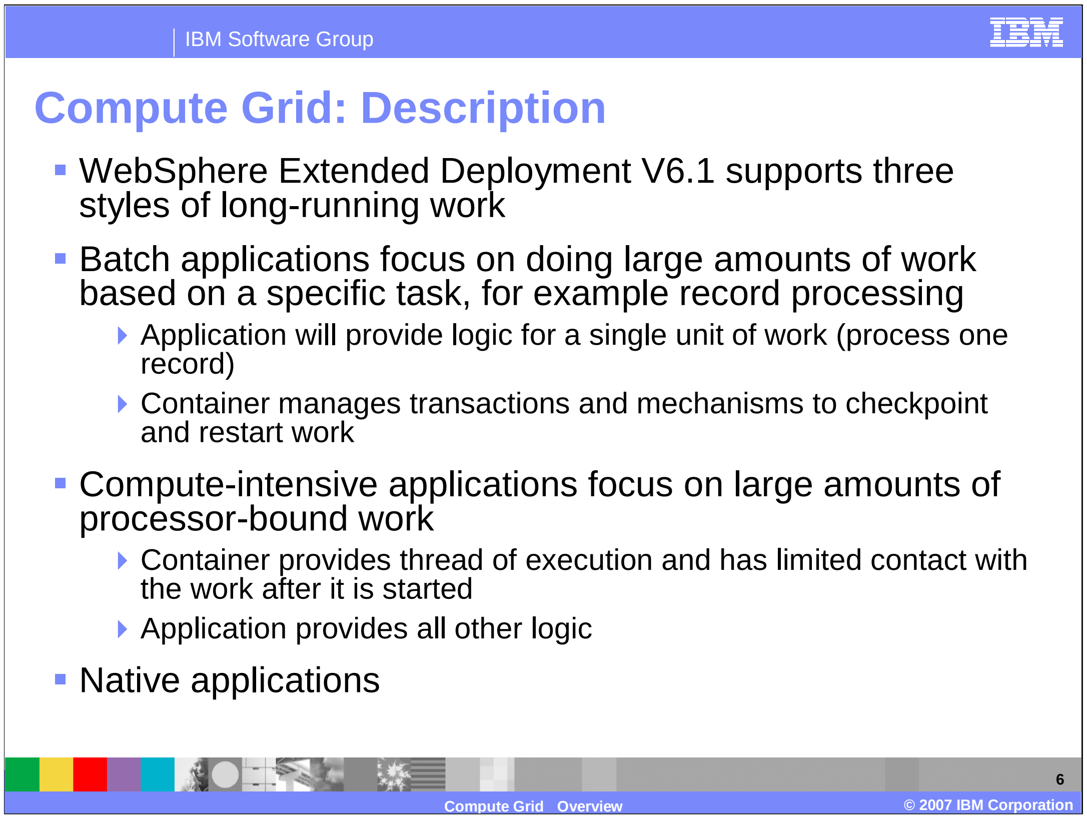
- Compute Grid provides support for long-running applications
  - Submission and execution must be asynchronous
  - Allow work to be specified declaratively rather than programmatically
  - Greater separation of submitter and execution environment
    - Support work being submitted from outside the WebSphere 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
  - Simplified install and configuration
    - Components installed with WebSphere Application Server
    - Enabled through the administrative client
  - High availability and scalability
    - Compute Grid job scheduler
- It is preferable to run long-running applications and transactional work within different processors
  - Compute Grid will balance the work amongst different nodes within a node group

5

Compute Grid Overview © 2007 IBM Corporation

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 execution environment should allow the submission of work from outside the WebSphere environment. Once long-running work has begun, runtime data, such as checkpoints and intermediate results, should 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 specified by users.

Components of the Compute Grid are installed as WebSphere system components and then enabled through administrative functions. The job scan run fully clustered, in either a dynamic or static cluster, to provide high availability and scalability.


IBM Software Group

Compute Grid: Description

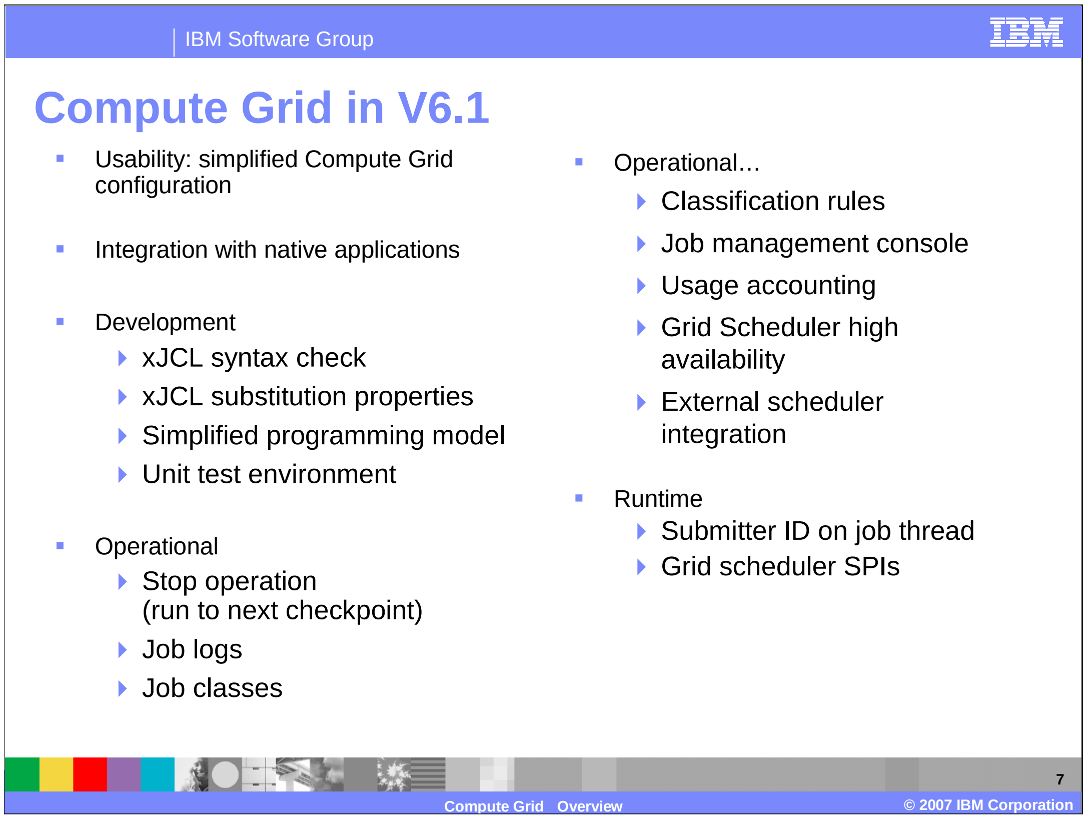
- WebSphere Extended Deployment V6.1 supports three styles of long-running work
- Batch applications focus on doing large amounts of work based on a specific task, for example record processing
  - Application will provide logic for a single unit of work (process one record)
  - Container manages transactions and mechanisms to checkpoint and restart work
- Compute-intensive applications focus on large amounts of processor-bound work
  - Container provides thread of execution and has limited contact with the work after it is started
  - Application provides all other logic
- Native applications

6

Compute Grid Overview

© 2007 IBM Corporation

The Compute Grid component of WebSphere Extended Deployment version 6.1 supports three types of long-running applications: batch, compute-intensive and native. A typical batch application will perform large amounts of work based on repetitive tasks. A batch application must provide the logic for a single unit of work, and the container will provide support to run the job with transactions and the ability to checkpoint and restart the batch process. For example, a typical 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 will provide all the logic for the work and the Compute Grid will make sure that the application is appropriately situated within the environment. Native applications are scheduled from within the WebSphere environment, but run as separate processes.


IBM Software Group

Compute Grid in V6.1

- Usability: simplified Compute Grid configuration
- Integration with native applications
- Development
  - xJCL syntax check
  - xJCL substitution properties
  - Simplified programming model
  - Unit test environment
- Operational
  - Stop operation (run to next checkpoint)
  - Job logs
  - Job classes
- Operational…
  - Classification rules
  - Job management console
  - Usage accounting
  - Grid Scheduler high availability
  - External scheduler integration
- Runtime
  - Submitter ID on job thread
  - Grid scheduler SPIs

7

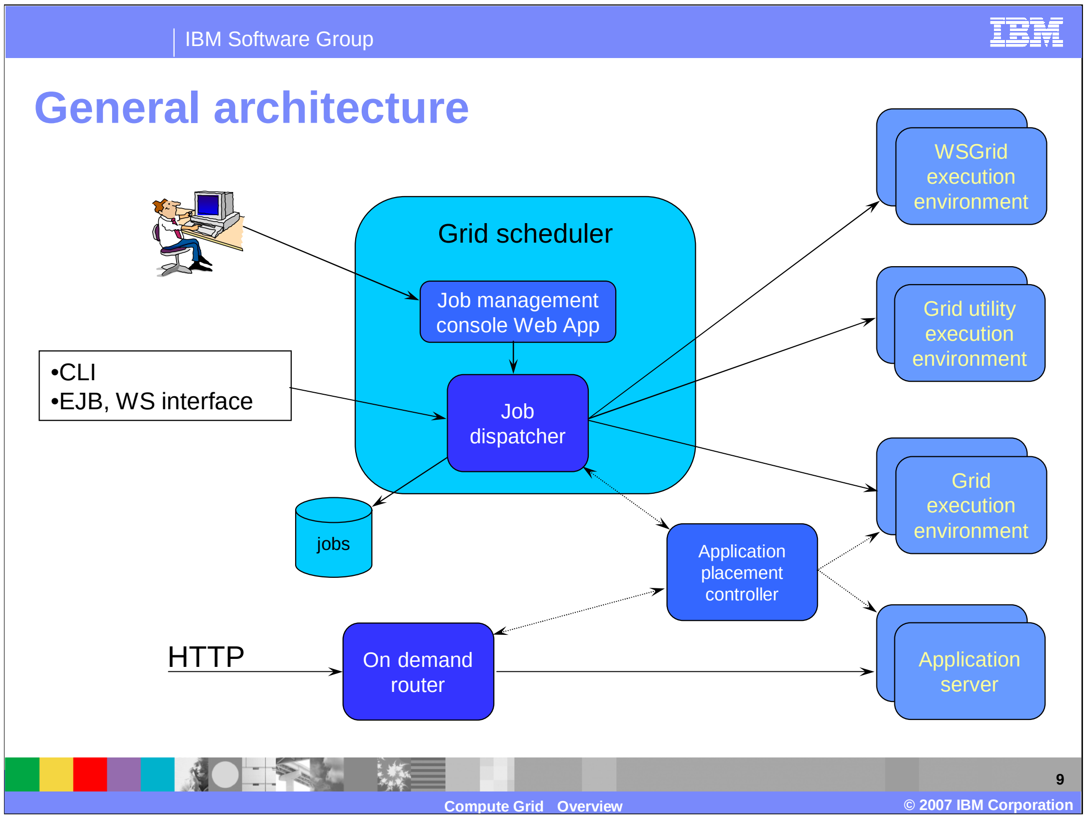
Compute Grid Overview © 2007 IBM Corporation

WebSphere Extended Deployment V6.1 includes many enhancements over previous versions. Some of these improvements include better interoperability with existing applications, easier development, better control of batch jobs, and increased interaction with a wide variety of native platform capabilities.

Section

Architecture

This section will explain the architecture of the Compute Grid.

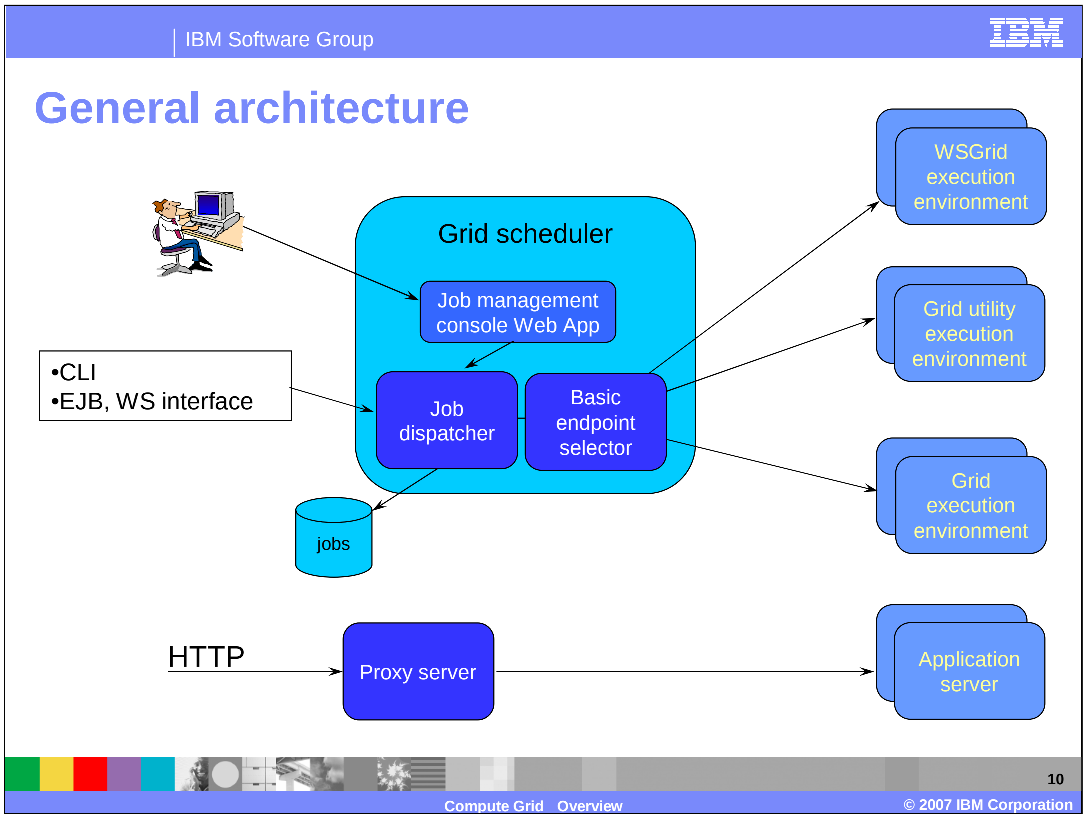


WebSphere Extended Deployment compute grid is basically a group or set of loosely associated WebSphere and non WebSphere applications.

This picture illustrates the components involved with a compute grid. A user interacts with the compute grid by setting policy information for the environment, and by interacting with the job dispatcher using one or more of the available interfaces. The command line interface allows you to submit and control grid jobs in the system. The enterprise Java bean and Web service interfaces provide this functionality to both Java 2 Platform Enterprise Edition (J2EE) and non-J2EE applications. The functionality known as grid job management was formerly located in the administration console but has moved to the job management console in WebSphere Extended Deployment V6.1. The job management console provides a graphical user interface (GUI) that allows WebSphere administrators and operators to perform job management functions. The job management console is part of the Grid Scheduler.

There are three forms of the execution environment. The first form is the grid execution environment and it runs inside a WebSphere application server. The second is an environment provided by the grid utility, through which a native execution grid job can be submitted and controlled. The third is the WSGrid execution environment. The grid utility and WSGrid are peers and represent the primary platform difference between z/OS® and other operating systems. WebSphere Grid, or WSGrid, is found on zSeries platforms and grid utility is found elsewhere. On all platforms, the primary link between the job scheduler and the execution environment is an XML-based JCL file that contains information about how to run the application and how to pass parameters into the application.

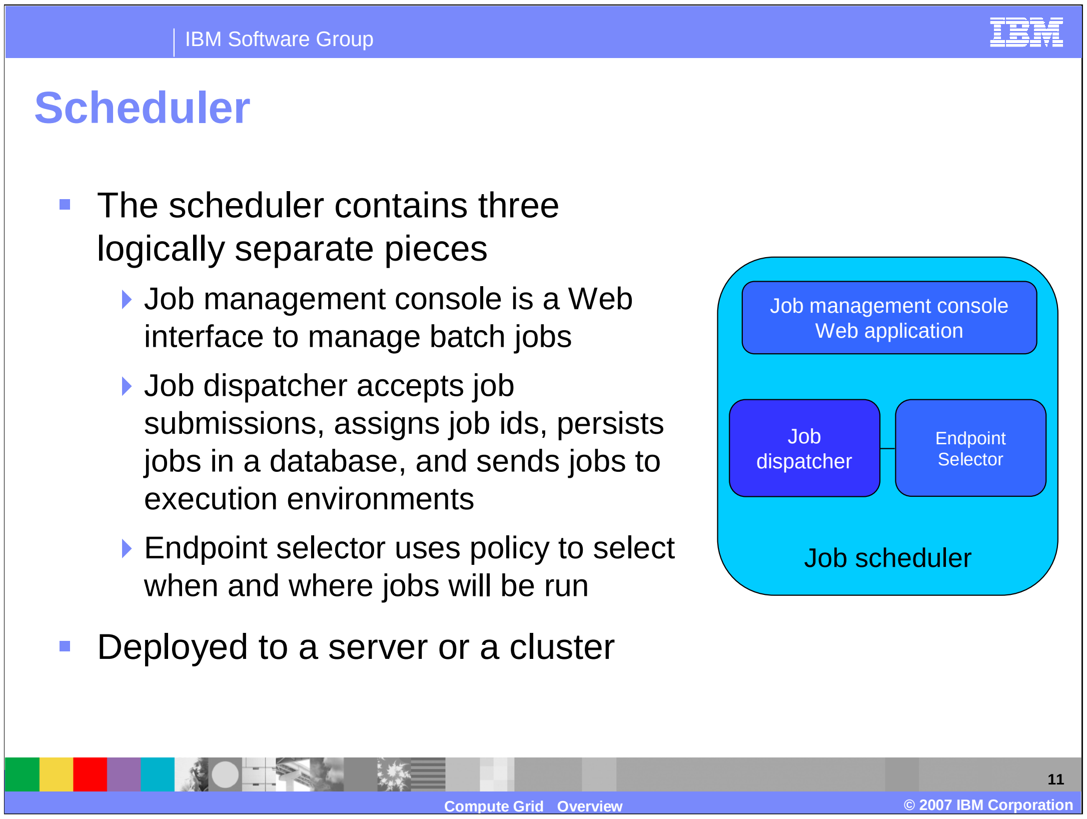
Each of the individual components shown here will be discussed in more detail on the following slides.



General architecture

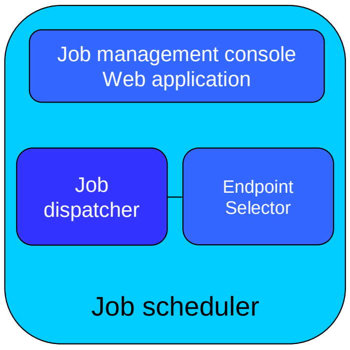
Grid scheduler
Job management console Web App
•CLI
•EJB, WS interface
Job dispatcher
Basic endpoint selector
jobs
WSGrid execution environment
Grid utility execution environment
Grid execution environment
HTTP
Proxy server
Application server

If the operations optimization package is installed, Compute Grid will work with the application placement controller to balance the available resources between long running and OLTP applications. In the absence of dynamic computing capabilities offered by the operations optimization package, the Compute Grid will use a simplified endpoint selector to determine where to dispatch a job. This basic endpoint selector lacks autonomic features found in the application placement controller.



Scheduler

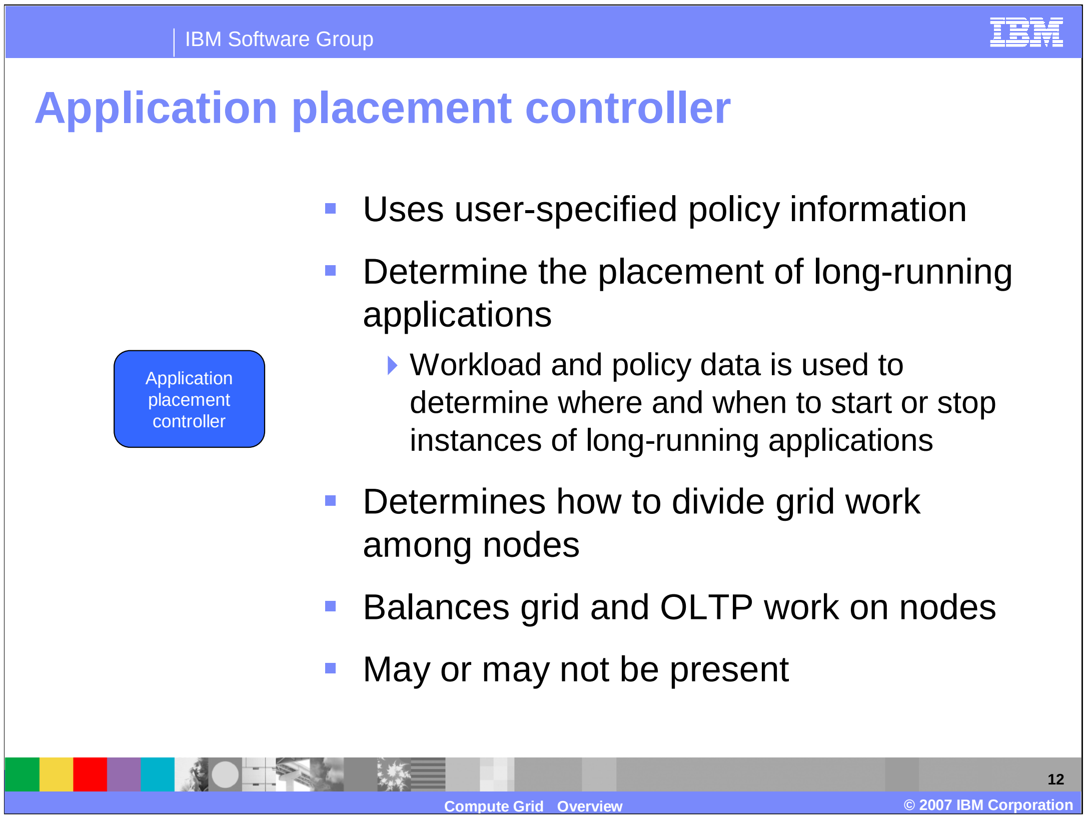
- The scheduler contains three logically separate pieces
  - Job management console is a Web interface to manage batch jobs
  - Job dispatcher accepts job submissions, assigns job ids, persists jobs in a database, and sends jobs to execution environments
  - Endpoint selector uses policy to select when and where jobs will be run
- Deployed to a server or a cluster



The Compute Grid job scheduler is responsible for accepting, persisting and scheduling the execution of long-running jobs. It manages the job database, assigns job identifiers and selects where and when jobs should be run. These endpoints can be J2EE programs; compiled programs such as Java, FORTRAN or COBOL; or even scripts. If the operations optimization package is installed, the Compute Grid job scheduler will use the application placement controller to start and stop instances of long-running dynamic clusters. This decision of when and where to start dynamic cluster instances is based on the jobs to be run and administrator-defined service policies to handle transactional J2EE work.

In Extended Deployment version 6.1, the job management console Web application replaces the administrative console's job management page found in previous versions. The job management console is a Web application packaged with the job scheduler, as part of the long running scheduler enterprise application. This application is installed silently when you configure the long running scheduler.

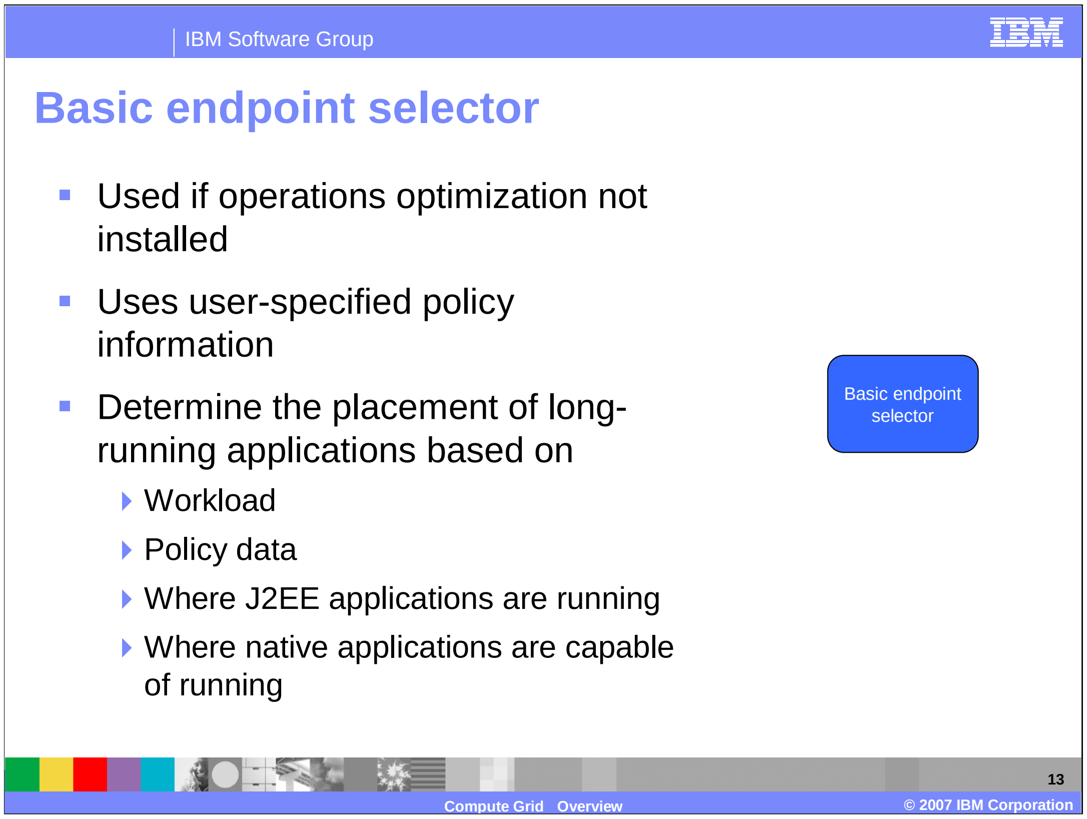
The job scheduler can run in a single server, or fully clustered in either a dynamic or static cluster. This provides both high availability and scalability.



Application placement controller

- Uses user-specified policy information
- 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
- Determines how to divide grid work among nodes
- Balances grid and OLTP work on nodes
- May or may not be present

The application Placement controller is part of the operations optimization package and may or may not be present. If present, the application placement controller serves as the endpoint selector. In version 6.1 the application placement controller has been expanded to act as an arbiter between transactional and grid workloads. The application placement controller uses specified job characteristics to balance the mix of WebSphere and external servers to satisfy the grid computing needs. For grid work scheduled for WebSphere application servers, the application placement controller controls individual servers and dispatches work through the grid execution environment. A more detailed discussion of the Application Placement Controller can be found under the Operations Optimization section of WebSphere Extended Deployment.

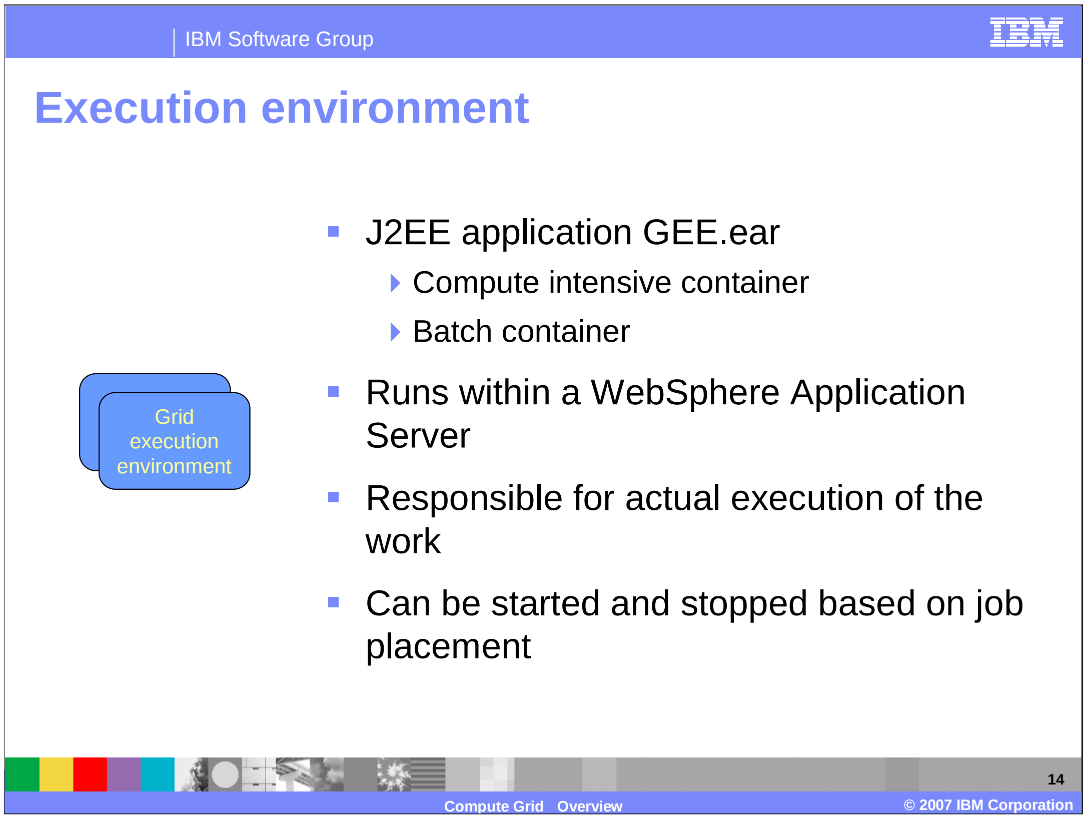


Basic endpoint selector

- Used if operations optimization not installed
- Uses user-specified policy information
- Determine the placement of long-running applications based on
  - Workload
  - Policy data
  - Where J2EE applications are running
  - Where native applications are capable of running

Basic endpoint selector

When operations optimization package is *not* installed, the grid scheduler will use a simplified basic endpoint selector. This component selects an endpoint based on the weight of the endpoint and the number of outstanding jobs for that endpoint.



Execution environment

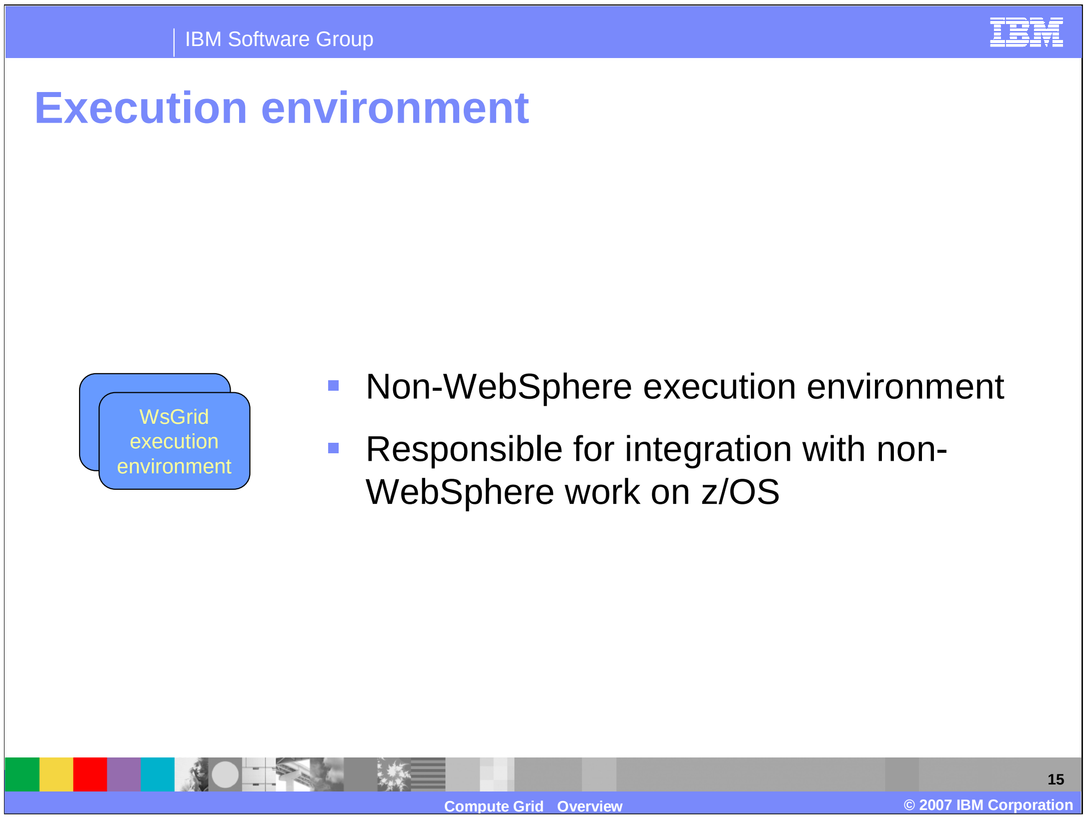
Grid execution environment

- J2EE application GEE.ear
  - Compute intensive container
  - Batch container
- Runs within a WebSphere Application Server
- Responsible for actual execution of the work
- Can be started and stopped based on job placement

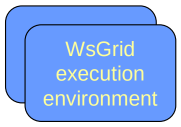
The execution environments provide the runtime environments needed by long-running applications. WebSphere Extended Deployment V6.1 provides two execution environments in a single J2EE application, GEE.ear, which is deployed to the dynamic clusters that host long-running applications:

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

The batch execution environment supports batch-oriented applications. These applications are expected to perform record processing similar to more traditional J2EE applications, but are driven by batch inputs rather than interactive users. This environment provides batch applications with a programming model that supports container-managed restartable processing and the ability to pause and cancel executing jobs. The grid execution environment is a system application which is automatically configured when a grid application is deployed.

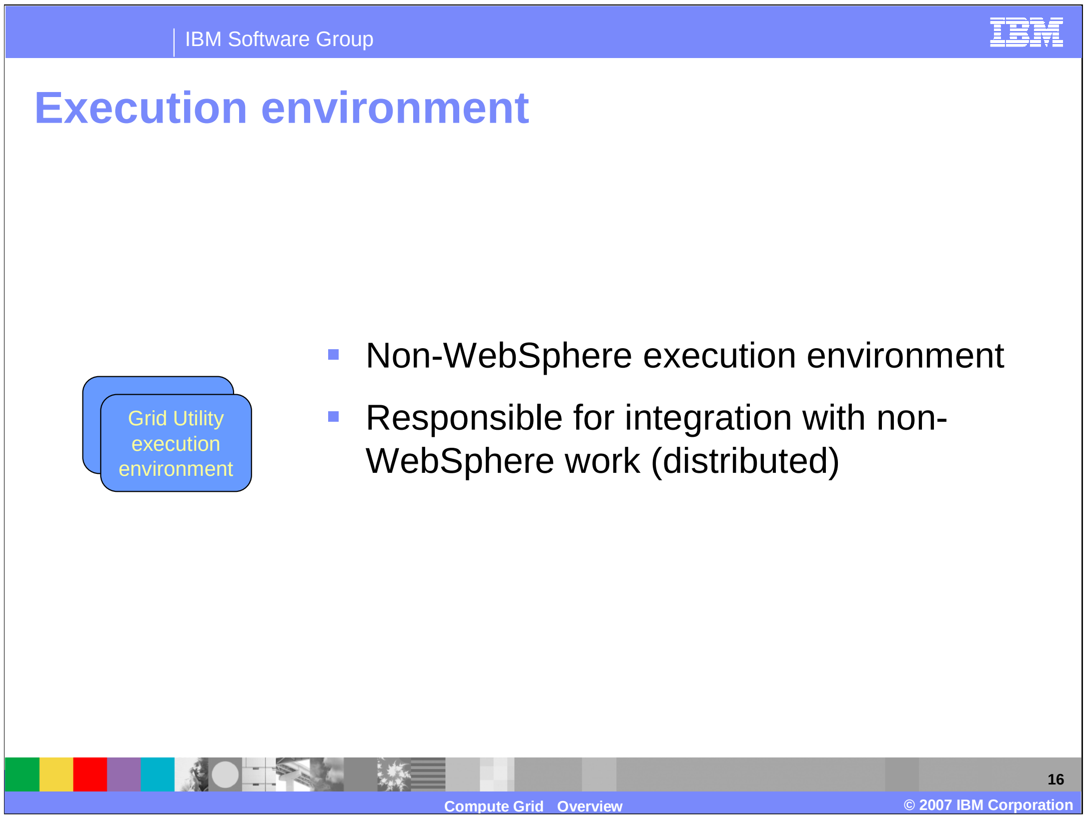


Execution environment

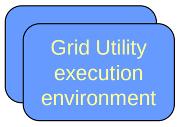


- Non-WebSphere execution environment
- Responsible for integration with non-WebSphere work on z/OS

WsGrid is a natural extension of z/OS batch. WsGrid is discussed in detail in the z/OS Compute Grid programming model presentation.



Execution environment



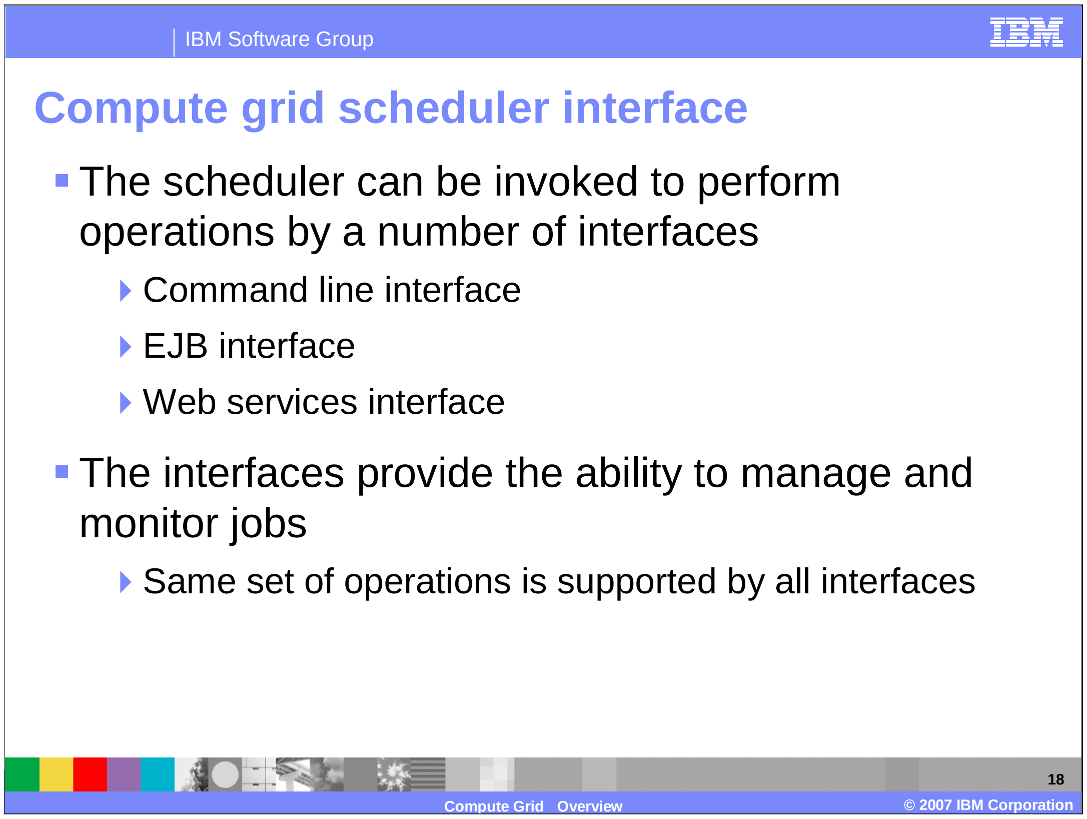
- Non-WebSphere execution environment
- Responsible for integration with non-WebSphere work (distributed)

The grid utility execution environment controls native execution jobs on distributed platforms. This component is added to node agents when they are augmented for compute grid, and is included in the separately installed WebSphere Extended Deployment middleware agent.

Section

Interfaces

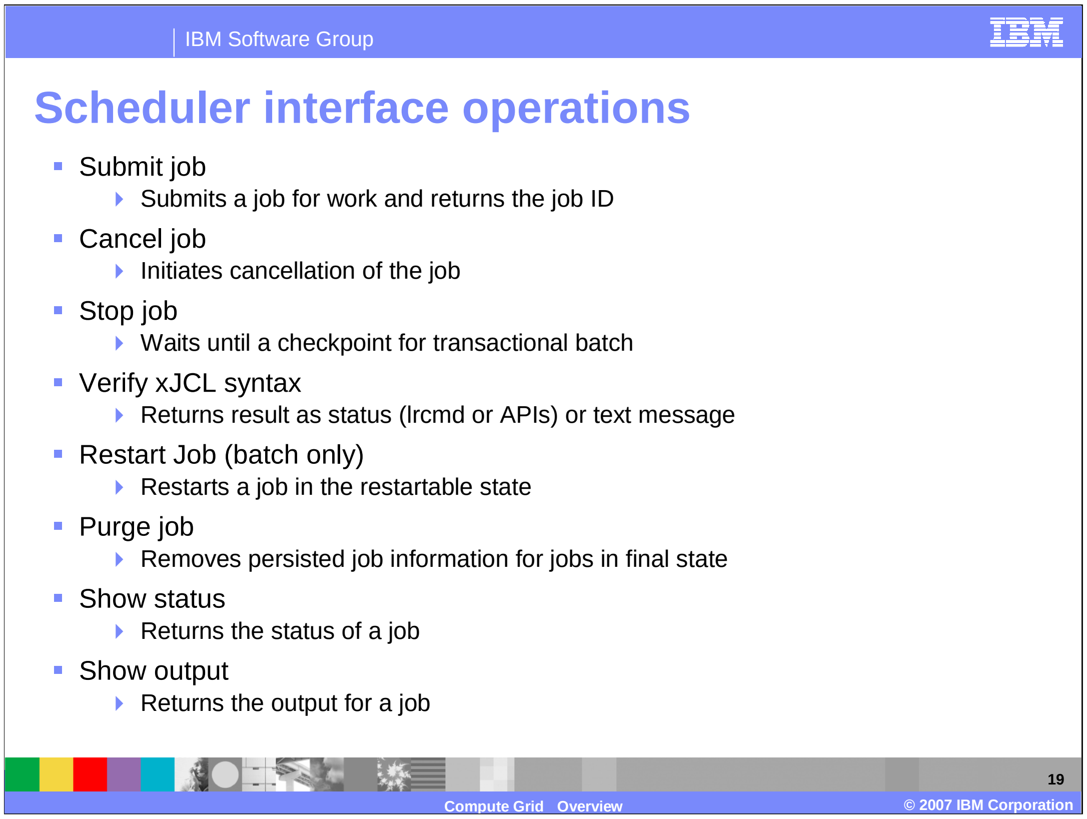
This section will discuss the interfaces provided for the Compute Grid.



Compute 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

As depicted on the general architectural graphic, compute grid scheduler provides several interfaces for user and programmatic interaction. You can use these interfaces 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

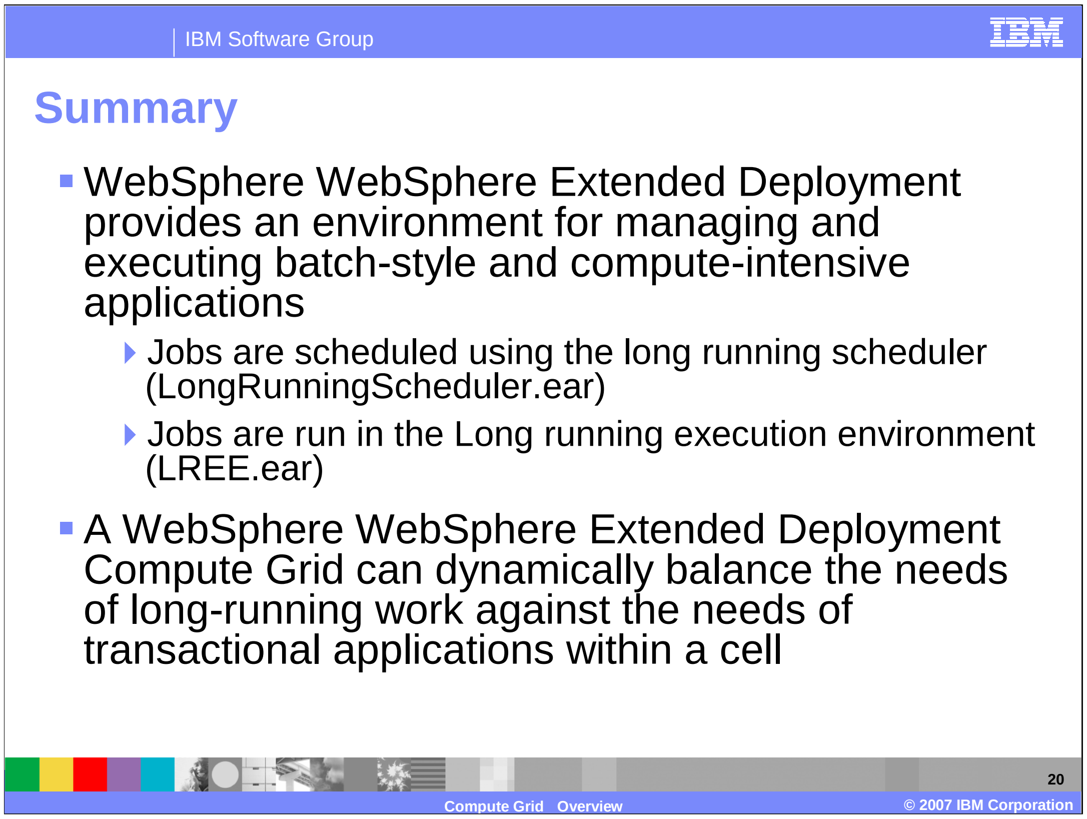
Scheduler interface operations

- Submit job
  - Submits a job for work and returns the job ID
- Cancel job
  - Initiates cancellation of the job
- Stop job
  - Waits until a checkpoint for transactional batch
- Verify xJCL syntax
  - Returns result as status (lrcmd or APIs) or text message
- Restart Job (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

19

Compute Grid Overview © 2007 IBM Corporation

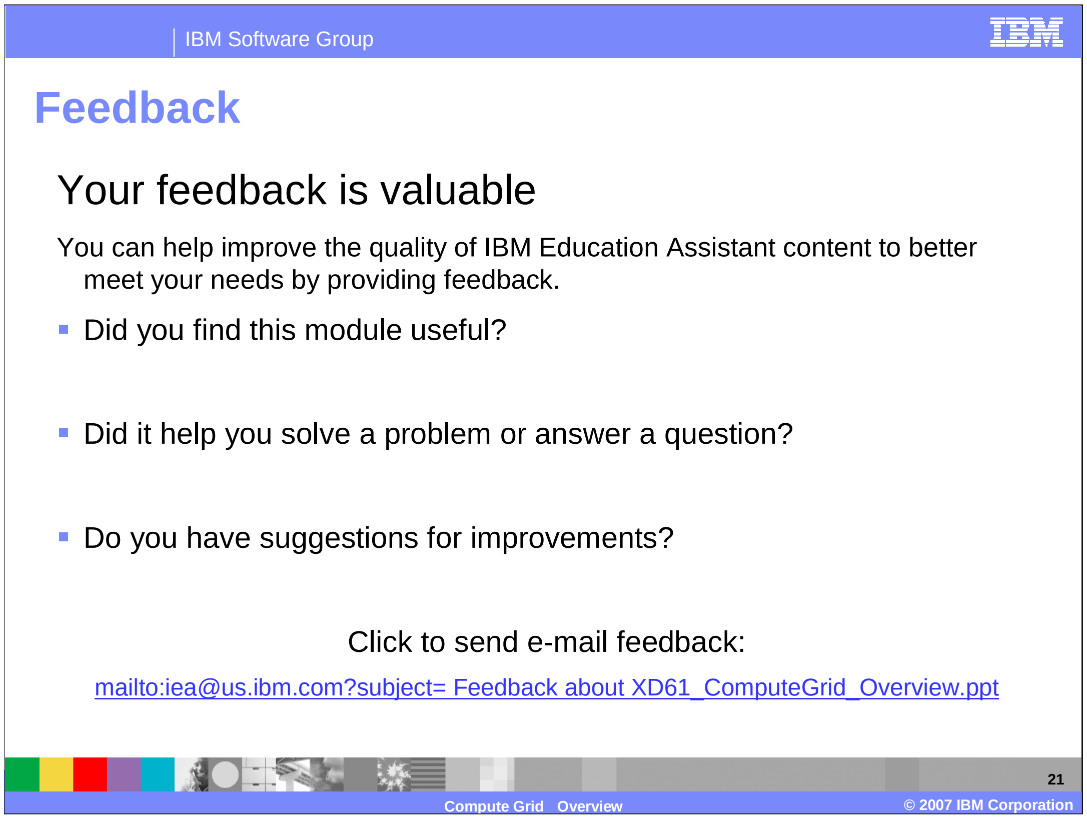
Using the provided operations a user can submit or cancel a job through the compute grid job scheduler. The cancel operation changes the job state to “cancel pending” until the compute grid can take the appropriate steps to stop the job. The stop job operation waits until a checkpoint is taken for batch work; for non-batch jobs it behaves as the cancel operation. Under certain conditions, you can restart a batch job, allowing it to continue it’s work. The purge job command removes the persisted job information for a finalized job. There are also a number of options for scheduling jobs, receiving information from a running job, and showing job status and output.



Summary

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

This presentation explained some of the benefits of the compute grid component of WebSphere Extended Deployment V6.1. It discussed the differences between computationally intensive and batch programs, and explained the capabilities for the compute grid to balance long-running work with transactional work in an environment.


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:

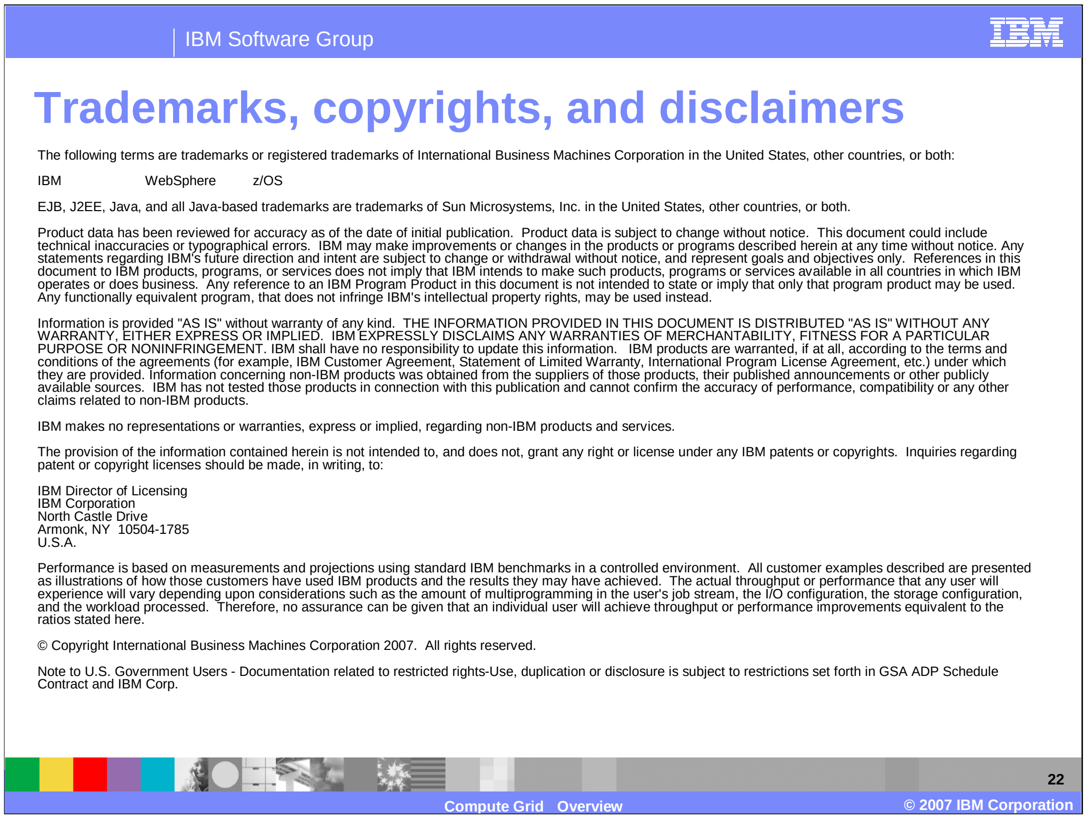
mailto:iea@us.ibm.com?subject= Feedback about XD61_ComputeGrid_Overview.ppt

21

Compute Grid Overview

© 2007 IBM Corporation

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



Trademarks, copyrights, and disclaimers