



IBM Software Group

IBM® WebSphere® Extended Deployment V6.1

Compute Grid - Example



@business on demand.

© 2007 IBM Corporation
Updated August 28, 2007

This presentation will provide an example of how to use the compute grid component offered in WebSphere Extended Deployment V6.1

Agenda

- Example of configuring and using Compute Grid



This presentation will provide an example of how to configure your WebSphere environment to support Compute Grid and explain how to deploy long-running applications to the environment.

Scheduler configuration

- Use default Derby grid databases and data source, or
- Manually create a new database
 - ▶ DB2®, Oracle, Informix®, or Derby
 - ▶ Use DDL provided with WebSphere Extended Deployment to create scheduler and execution environment tables
 - ▶ Define corresponding data source in WebSphere Application Server
 - The data source must be accessible by all nodes that can run the scheduler and execution endpoints
 - ▶ Configure the scheduler with the JNDI name and authentication information for the data source
- Create servers and clusters for job scheduler and execution environments

3

Compute Grid: Example

© 2007 IBM Corporation

Several steps are required to create an environment that will support long-running applications. First you must create the databases for the scheduler component and for the grid execution environment; DDLs to accomplish this are provided with WebSphere Extended Deployment. If you choose to use Derby, the database instances are created automatically when the runtime components are installed. Once the tables are created, you must define corresponding data sources in WebSphere. The data sources must be accessible to any nodes that will host either the scheduler or run long-running applications. The scheduler must then be configured with the JNDI name for the resource and any security information it needs to use the data source. The deployment of the job scheduler and the grid execution environment, if applicable, is automatic.

Job scheduler configuration panel

Job scheduler

The job scheduler accepts grid jobs and determines where and when to execute them. As part of managing jobs, the job scheduler persists job information in an external job database. This configuration panel allows the deployment target, datasource, database schema name, charge-back accounting and endpoint job log location to be configured for the scheduler.

Configuration

General Properties	Additional Properties
Scheduler hosted by (none) ▾	<input type="checkbox"/> Classification rules
Database schema name LRSSHEMA	<input type="checkbox"/> Custom properties
Data source JNDI name jdbc/lrsched ▾	<input type="checkbox"/> Job classes
Endpoint job log location \${GRID_JOBLOG_ROOT}/jobl	<input type="checkbox"/> Security role to user/group mapping
<input type="checkbox"/> Record usage data in scheduler database	<input type="checkbox"/> WebSphere grid endpoints
<input type="checkbox"/> Record usage data in SMF (z/OS only)	
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	Related Items
	<input type="checkbox"/> JDBC providers
	<input type="checkbox"/> Middleware servers
	<input type="checkbox"/> Service policies

The job scheduler configuration panel is located from the System Administration link in the left frame of the administration console. Under the configuration panel for the scheduler, you can configure where the scheduler will reside, the schema used in the database, and the JNDI name for the data source. You can also specify classification rules and the job classes (resource restrictions) as discussed in the Compute Grid overview presentation.

Develop long-running application

- Develop long-running applications using normal development tools
 - ▶ Java™
 - ▶ J2EE
 - WebSphere Application Developer
 - IBM Rational® Application Developer
 - Eclipse
 - ▶ External programs
 - Compiled (FORTRAN, COBOL, ...)
- A J2EE or Java transactional batch long-running application can be packaged in an ordinary EAR file deployed to a WebSphere Application Server

Application developers create long-running applications based on either the computationally intensive or batch programming model using normal J2EE development tools. Long-running applications are packaged into normal ear files. The EJB jar file for the application contains some specific information for long-running applications, such as the deployment information for the controller bean and the bean implementations of the applications.

Deploy long-running application

- WebSphere
 - ▶ Long-running applications are deployed as regular J2EE applications
 - ▶ When the application is deployed, WebSphere Extended Deployment automatically detects that it is a long-running application
 - ▶ Install process will silently install the grid execution environment
 - ▶ Can mix the transactional and grid applications in a dynamic cluster
 - ▶ The same placement controller is used for both types of workload
- Once the application is deployed, define service policies for the new long-running application

6

Compute Grid: Example

© 2007 IBM Corporation

Long-running applications are deployed like any regular J2EE application. During the deployment process, WebSphere Extended Deployment will detect that it is a long-running application and silently install the grid execution environment if it is not already installed. Once the application has been deployed, an administrator can define service policies for the application in preparation for submitting a job. Long-running applications do not support the same service policies as OLTP applications. The only service policies supported for long-running applications are labeled 'completion time' and 'discretionary'.

Submit job

- Construct xJCL to submit job to application
- Use one of the interfaces provided by the scheduler to submit job
 - ▶ Job Management Console
 - ▶ Command Line Interface,
 - ▶ Web Services,
 - ▶ EJB,
 - ▶ WSGrid utility
- Note job ID assigned by scheduler
- Use job management console

Prior to submitting a job, an administrator must construct an xJCL document to describe the behavior of the application. For WebSphere applications, the xJCL contains a JNDI name to identify which application should be used for a job step. For a non-WebSphere application, the xJCL specifies how to run the application (for example is it a program or a script) and parameters that are passed to the application. Then the administrator has a choice of interfaces to submit the job to the scheduler. Regardless of the interface used, the return value from the submission is the job ID assigned to the job by the scheduler. An administrator can then manage the job using panels in the job management console or any of the other available interfaces.

IBM Software Group 

Job management console



- Web interface
 - Job management
 - Job repository
 - Job schedules

8
Compute Grid: Example © 2007 IBM Corporation

The primary interface to manage grid jobs is the job management console. The job management console resides in the job scheduler system application, and has a browser interface which is independent of the WebSphere administrative console. The job management console has three sections.

The job management section is used to submit a job for execution or manage submitted jobs.

The job repository saves job definition xJCL into the job scheduler database. Saved jobs can be viewed and submitted from the job repository panels.

Finally, the job schedules section allows jobs to be scheduled for running at a single future time or periodically.

Job management console: Job management

- Submit a job
 - ▶ xJCL
 - ▶ Parns
 - ▶ Schedule
- Manage submitted jobs
 - ▶ View job logs
 - ▶ Cancel
 - ▶ Resume
 - ▶ Restart
 - ▶ Stop
 - ▶ Suspend

Select	Job ID	Submitter	Last Update	State
<input type="checkbox"/>	SimpleCIEar:9	.	Wed Oct 11 17:56:40 CDT 2006	Ended

Job submission requires pointing to the xJCL for the job. The xJCL contains information required to run the applications in the job. The xJCL also contains descriptions of parameters (for non-WebSphere applications) or other environmental settings. You can optionally specify when the job will run, similar to the job schedule part of the job management console. Once a job is submitted, it can be viewed and managed in the “View jobs” panel. The state of a job is displayed as submitted, executing, paused, or ended. Also shown is the node and application server where the job ran. Clicking on a job will show details of the running job and provide an opportunity to view the job logs. From this panel you can also manage the job during its life cycle.

Job management console: Job repository

- Save a job definition
 - ▶ xJCL

Save a job

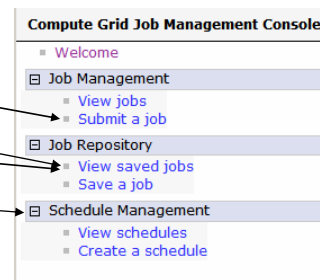
Specify the job name and the file containing the job definition, then click **Save** to store the job in the job repository.

* Job name:

* xJCL path:

Replace the job if the specified job name exists

- Manage saved jobs
 - ▶ Submit
 - ▶ View xJCL
 - ▶ Delete definition
 - ▶ Schedule



Saving a job to the job repository requires a name and a path to the defining xJCL. Once it has been saved, you can view the xJCL by clicking on the job name. You can also replace or remove xJCL job definitions already in the job repository.

You can submit the saved job from the “Submit a job” panel, or schedule it to run periodically on the “Create a schedule” panel.

IBM Software Group IBM

Job management console: Job schedules

- Save a job definition
 - ▶ xJCL
 - ▶ Schedule
 - Date and time
 - Repeating
- Manage schedules
 - ▶ View details
 - ▶ Cancel

→ **Step 1: Create schedule** **Create schedule**

Step 2: Specify job
Step 3: Confirm create schedule

Specify the name of the schedule to create. Specify the start date and time for the job to first run.

* Name:

* Start date (yyyy-MM-dd):
 - -

* Start time (HH:mm:ss):
 : :

* Interval:

11

Compute Grid: Example © 2007 IBM Corporation

Scheduling a job is similar to submitting a job by its xJCL or a job from the repository for delayed submission. You can schedule a job to be submitted once at a given date and time from the “Submit a job” panel or you can schedule the job to run periodically. From the “View schedules” panel, you can view the details of the job schedule or cancel the schedule.

Summary

- 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 (GEE.ear)
- A WebSphere Extended Deployment compute grid can dynamically balance the needs of long-running work against the needs of transactional applications within a cell

In summary, this presentation showed by example how to use the new Compute Grid component provided with WebSphere Extended Deployment V6.1

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 Example.ppt>



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:

DB2 IBM Informix Rational WebSphere

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

EJB, J2EE, Java, 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.

© Copyright International Business Machines Corporation 2007. 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.