Running a Workflow on a PowerCenter Grid
Abstract

When an Informatica domain contains multiple nodes, you can configure PowerCenter workflows to run on a grid. When you run a workflow on a grid, the PowerCenter Integration Service runs a service process on each available node of the grid to increase performance and scalability. The service distributes workflow tasks across the nodes in the grid based on node availability and resource availability.

Supported Versions

- PowerCenter 9.x

Table of Contents

Overview ................................................................. 3
High-Level Steps to Run a Workflow on a Grid. ........................................ 4
Configuring the Grid ................................................... 4
  Step 1. Create the Grid ................................................ 4
  Step 2. Configure the PowerCenter Integration Service to Run on the Grid. ..................................... 5
  Step 3. Configure the Service Processes to Be Compatible. ........................................... 5
Configuring Resources (Optional). ....................................... 7
  Configuring the PowerCenter Integration Service to Check Resources. ...................... 8
  Disabling Connection Resources. .................................... 8
  Defining Custom and File/Directory Resources. ........................................ 9
Configuring the Load Balancer (Optional). ................................. 9
  Configuring the Dispatch Mode. .................................... 10
  Creating Service Levels. ....................................... 12
  Defining Resource Provision Thresholds. ........................................ 13
Configuring Workflows to Run on the Grid. .................................... 14
  Step 1. Configure a Workflow to Run on a Grid. ........................................ 15
  Step 2. Assign Service Levels to Workflows (Optional). ........................................ 15
  Step 3. Assign Resources to Tasks (Optional). ........................................ 16
Rules and Guidelines for Configuring a Workflow to Run on a Grid. ................... 16
Other Considerations ....................................................... 17
  Grid Connectivity and Recovery. .................................... 17
  Running a Session on a PowerCenter Grid. ........................................ 17
Optimizing Grid Deployments ................................................ 18
  Storing Files. ..................................................... 18
  Using a Shared File System. ........................................ 19
Overview

When you run a workflow on a node, the PowerCenter Integration Service runs a single service process on a single node. A single node processes all workflow tasks.

When you run a workflow on a grid, you run the workflow on a group of nodes. The PowerCenter Integration Service runs on one node, but multiple service processes run on multiple nodes. The master service process runs the workflow and all tasks except Session, Command, and predefined Event-Wait tasks, which it may distribute to other nodes. The master service process is the PowerCenter Integration Service process that runs the workflow, monitors service processes running on other nodes, and runs the Load Balancer. The Scheduler runs on the master service process node, so it uses the date and time for the master service process node to start scheduled workflows.

The Load Balancer is the component of the Integration Service that dispatches Session, Command, and predefined Event-Wait tasks to the nodes in the grid. The Load Balancer distributes tasks based on node availability. If the Integration Service is configured to check resources, the Load Balancer also distributes tasks based on resource availability.

For example, a workflow contains a Session task, a Decision task, and a Command task. You specify a resource requirement for the Session task. The grid contains four nodes, and Node 4 is unavailable. The master service process runs the Start and Decision tasks. The Load Balancer distributes the Session and Command tasks to nodes on the grid based on resource availability and node availability.

The following figure displays how a PowerCenter Integration Service configured to run on a grid runs multiple service processes on multiple nodes. The PowerCenter Integration Service runs on Node 1. The PowerCenter Integration Service processes run on Node 1, Node 2, and Node 3:

When you run workflows on a grid, all nodes in the grid must be able to communicate. You must ensure that each PowerCenter Integration Service process running on a node is compatible and configured the same.

**Note:** To run workflows on a grid, you must have the Server grid option.
High-Level Steps to Run a Workflow on a Grid

To run a workflow on a PowerCenter grid, complete the following high-level steps:

1. Configure the grid.
   In Informatica Administrator (the Administrator tool), create a grid and configure the PowerCenter Integration Service to run on the grid. You also must ensure that each PowerCenter Integration Service process running on a node is compatible and configured the same.

2. Optionally, configure resources.
   PowerCenter resources are the database connections, files, directories, node names, and operating system types required by a task. As a best practice, configure all nodes in the grid so that they have access to all resources.
   If all nodes cannot access all resources, use the Administrator tool to configure the PowerCenter Integration Service to check resources. When you do this, the Load Balancer matches the resources available to nodes in the grid with the resources required by the workflow. The Load Balancer dispatches tasks in the workflow to nodes where the required resources are available.

3. Optionally, configure the Load Balancer.
   The Load Balancer is a component of the PowerCenter Integration Service that dispatches tasks to PowerCenter Integration Service processes running on nodes in a grid. By default, the Load Balancer uses the round-robin dispatch mode, a single service level, and default values for the resource provision thresholds.
   You can use the default configuration. Or, you can use the Administrator tool to change the dispatch mode, configure additional service levels, or change the resource provision thresholds.

4. Configure workflows to run on the grid.
   In the Workflow Manager, configure each workflow to run on a PowerCenter Integration Service that is configured to run on a grid.
   If you created service levels and resources in the Administrator tool, you also assign service levels to workflows and assign resources to tasks.

   **Note:** You can also use the infacmd command line program to configure the grid, resources, and the Load Balancer. For more information about infacmd commands, see the *Informatica Command Reference*.

Configuring the Grid

A grid is an alias assigned to a group of nodes that run workflows.

To configure the grid, complete the following tasks in the Administrator tool:

1. Create the grid.
2. Configure the PowerCenter Integration Service to run on the grid.
3. Configure the PowerCenter Integration Service processes to be compatible.

**Step 1. Create the Grid**

Create the grid and assign nodes to the grid.

1. In the Administrator tool, click the **Domain** tab.
2. Select Actions > **New** > **Grid**.
   The Create Grid window appears.
3. Enter the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the grid. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters or begin with @. It also cannot contain spaces or the following special characters: `~ % ^ * + = { } \ ; : ' &quot; / ? . , &lt; &gt;</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the grid. The description cannot exceed 765 characters.</td>
</tr>
<tr>
<td>Nodes</td>
<td>Select nodes to assign to the grid.</td>
</tr>
<tr>
<td>Path</td>
<td>Location in the Navigator.</td>
</tr>
</tbody>
</table>

4. Click OK.

**Step 2. Configure the PowerCenter Integration Service to Run on the Grid**

1. In the Administrator tool, click the Domain tab.
2. Select the PowerCenter Integration Service in the Navigator, and click the Properties view.
3. Click Edit in the General Properties area.
4. Select Grid, and then select the grid that you want the PowerCenter Integration Service to run on.
5. Click OK.

**Step 3. Configure the Service Processes to Be Compatible**

When you run a workflow on a grid, a PowerCenter Integration Service process runs on each node in the grid. The service distributes workflow tasks across nodes on the grid. Each service process running on a node must be compatible or configured the same. It must also have access to the directories and input files used by the PowerCenter Integration Service.

To ensure consistent results, complete the following tasks:

- Verify the shared storage location. Verify that the shared storage location configured in $PMRootDir is accessible to each node in the grid. For more information about storing files in optimal locations, see "Storing Files" on page 18.

- Configure the service process. Configure the following service process settings for each node in the grid:
  - Code pages. For accurate data movement and transformation, verify that the code pages are compatible for each service process. Use the same code page for each node where possible.
  - Service process variables. Configure service process variables with identical absolute paths to the shared directories on each node in the grid. If you use a mounted drive or a mapped drive, the absolute path to the shared location must also be identical.
  - Directories for Java components. Point to the same Java directory to ensure that java components are available to objects that access Java, such as Custom transformations that use Java coding.

Use the Administrator tool to configure the service processes.

1. In the Administrator tool, click the Domain tab.
2. Select the PowerCenter Integration Service in the Navigator, and click the Processes view.

The view displays the service process for each node assigned to the grid.
3. Configure the following service process settings for each node in the grid:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codepage</td>
<td>Code page of the PowerCenter Integration Service process node. All code pages for the process nodes must be compatible.</td>
</tr>
</tbody>
</table>
| $PMRootDir         | Root directory accessible by the node. This is the root directory for other service process variables. It cannot include the following special characters: * ? < > " | .  
|                    | Default is <Installation_Directory>\server\infa_shared. The installation directory is based on the service version of the service that you created. When you upgrade the PowerCenter Integration Service, the $PMRootDir is not updated to the upgraded service version installation directory. |
| $PMSessionLogDir   | Default directory for session logs. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/SessLogs.                                                                                                                                                                             |
| $PMBadFileDir      | Default directory for reject files. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/BadFiles.                                                                                                                                                                             |
| $PMCacheDir        | Default directory for index and data cache files. You can increase performance when the cache directory is a drive local to the PowerCenter Integration Service process. Do not use a mapped or mounted drive for cache files. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/Cache.                                                                                                                                                                               |
| $PMTargetFileDir   | Default directory for target files. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/TgtFiles.                                                                                                                                                                           |
| $PMSourceFileDir   | Default directory for source files. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/SrcFiles.                                                                                                                                                                           |
| $PMExtProcDir      | Default directory for external procedures. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/ExtProc.                                                                                                                                                                           |
| $PMTempDir         | Default directory for temporary files. It cannot include the following special characters: * ? < > " | .  
|                    | Default is $PMRootDir/Temp.                                                                                                                                                                               |
| $PMWorkflowLogDir  | Default directory for workflow logs. It cannot include the following special characters: * ? < > " | .  
<p>|                    | Default is $PMRootDir/WorkflowLogs.                                                                                                                                                                          |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PMLookupFileDir</td>
<td>Default directory for lookup files. It cannot include the following special characters: &quot;* ? &lt; &gt; &quot;</td>
</tr>
<tr>
<td></td>
<td>Default is $PMRootDir/LkpFiles.</td>
</tr>
<tr>
<td>$PMStorageDir</td>
<td>Default directory for state of operation files. The PowerCenter Integration Service uses these files for recovery if you have the high availability option or if you enable a workflow for recovery. These files store the state of each workflow and session operation. It cannot include the following special characters: &quot;* ? &lt; &gt; &quot;</td>
</tr>
<tr>
<td></td>
<td>Default is $PMRootDir/Storage.</td>
</tr>
<tr>
<td>Java SDK ClassPath</td>
<td>Java SDK classpath. You can set the classpath to any JAR files you need to run a session that require java components. The PowerCenter Integration Service appends the values you set to the system CLASSPATH.</td>
</tr>
<tr>
<td>Java SDK Minimum Memory</td>
<td>Minimum amount of memory the Java SDK uses during a session.</td>
</tr>
<tr>
<td></td>
<td>If the session fails due to a lack of memory, you may want to increase this value.</td>
</tr>
<tr>
<td></td>
<td>Default is 32 MB.</td>
</tr>
<tr>
<td>Java SDK Maximum Memory</td>
<td>Maximum amount of memory the Java SDK uses during a session.</td>
</tr>
<tr>
<td></td>
<td>If the session fails due to a lack of memory, you may want to increase this value.</td>
</tr>
<tr>
<td></td>
<td>Default is 64 MB.</td>
</tr>
</tbody>
</table>

**Configuring Resources (Optional)**

PowerCenter resources are the database connections, files, directories, node names, and operating system types required by a task. As a best practice, configure all nodes in the grid such that they have access to all resources.

If each service process running on a node in the grid cannot access all resources, use the Administrator tool to configure the PowerCenter Integration Service to check resources. When you do this, the Load Balancer matches the resources available to nodes in the grid with the resources required by the workflow. The Load Balancer dispatches tasks in the workflow to nodes where the required resources are available.

The following table lists the types of resources that you use in PowerCenter:

<table>
<thead>
<tr>
<th>Type</th>
<th>Predefined/ User-Defined</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Predefined</td>
<td>Any resource installed with PowerCenter, such as a plug-in or a connection object. A connection object may be a relational, application, FTP, external loader, or queue connection. When you create a node, all connection resources are available by default. Disable the connection resources that are not available to the node. Any Session task that reads from or writes to a relational database requires one or more connection resources. The Workflow Manager assigns connection resources to the session by default.</td>
</tr>
<tr>
<td>Node Name</td>
<td>Predefined</td>
<td>A resource for the name of the node. A Session, Command, or predefined Event-Wait task requires a node name resource if it must run on a specific node.</td>
</tr>
<tr>
<td>Type</td>
<td>Predefined/ User-Defined</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Operating System Type</td>
<td>Predefined</td>
<td>A resource for the type of operating system on the node. A Session or Command task requires an operating system type resource if it must run a specific operating system.</td>
</tr>
<tr>
<td>Custom</td>
<td>User-defined</td>
<td>Any resource for all other resources available to the node, such as a specific database client version. For example, a Session task requires a custom resource if it accesses a Custom transformation shared library or if it requires a specific database client version.</td>
</tr>
<tr>
<td>File/Directory</td>
<td>User-defined</td>
<td>Any resource for files or directories, such as a parameter file or a file server directory. For example, a Session task requires a file resource if it accesses a session parameter file.</td>
</tr>
</tbody>
</table>

**Note:** When you define a resource for a node, you must verify that the resource is available to the node. If the resource is not available and the PowerCenter Integration Service runs a task that requires the resource, the task fails.

**Configuring the PowerCenter Integration Service to Check Resources**

By default, the PowerCenter Integration Service is configured to ignore task resource requirements when distributing tasks across the grid. Configure the service to check resources before you define resources.

1. In the Administrator tool, click the **Domain** tab.
2. Select the PowerCenter Integration Service in the Navigator, and click the **Properties** view.
3. Click **Edit** in the Advanced Properties area.
4. Clear the **IgnoreResourceRequirements** property.
5. Click **OK**.

**Disabling Connection Resources**

By default, all connection resources are available to all nodes. Use the Administrator tool to disable the connection resources that are not available on the node.

For example, if the node does not have Oracle client libraries, disable the Oracle Application connections.

1. In the Administrator tool, click the **Domain** tab.
2. Select a node in the Navigator.
3. Click the **Resources** view.
4. Select a connection resource that you want to disable.
5. On the Domain tab **Actions** menu, click **Disable Selected Resource**.
6. Repeat steps 2 to 5 for each node in the grid that does not have access to all connection resources.
Defining Custom and File/Directory Resources

Custom and file/directory resources are user-defined resources. Use file/directory resources for parameter files or file server directories. Use custom resources for any other resources available to the node, such as a database client version.

Use the Administrator tool to create custom or file/directory resources when all nodes in the grid do not have access to all files or other custom resources. When you define a custom or file/directory resource, you assign a resource name. The resource name is a logical name that you create to identify the resource.

1. In the Administrator tool, click the Domain tab.
2. Select a node in the Navigator.
3. Click the Resources view.
5. Enter a name for the resource.
   The name cannot have spaces, include carriage returns or tabs, exceed 128 characters, or contain the following characters: \ / * ? < > " | $
   You assign the resource to a task or mapping object instance using this name. To coordinate resource usage, you might want to use a naming convention for file/directory and custom resources.
6. Select a resource type.
7. Click OK.
8. Repeat steps 2 to 7 to define custom and file/directory resources for other nodes in the grid.

Resource Naming Conventions

Using resources with PowerCenter requires coordination and communication between the domain administrator and the workflow developer.

The domain administrator defines resources available to nodes. The workflow developer assigns resources required by Session, Command, and predefined Event-Wait tasks. To coordinate resource usage, you can use a naming convention for file/directory and custom resources.

Use the following naming convention:
resourcetype_description

For example, multiple nodes in a grid contain a session parameter file called sales1.txt. Create a file resource for it named sessionparamfile_sales1 on each node that contains the file. A workflow developer creates a session that uses the parameter file and assigns the sessionparamfile_sales1 file resource to the session.

When the PowerCenter Integration Service runs the workflow on the grid, the Load Balancer distributes the session assigned the sessionparamfile_sales1 resource to nodes that have the resource defined.

Configuring the Load Balancer (Optional)

The Load Balancer is a component of the PowerCenter Integration Service that dispatches tasks to PowerCenter Integration Service processes running on nodes in a grid. It matches task requirements with resource availability to identify the best PowerCenter Integration Service process to run a task.

By default, the Load Balancer uses the round-robin dispatch mode, a single service level, and default values for the resource provision thresholds. You can use the default configuration. After running workflows on the grid, you might want to modify how the Load Balancer is configured to optimize performance.
You configure the following settings to determine how the Load Balancer dispatches tasks:

- **Dispatch mode.** The dispatch mode determines how the Load Balancer dispatches tasks. You can configure the Load Balancer to dispatch tasks in a simple round-robin fashion, in a round-robin fashion using node load metrics, or to the node with the most available computing resources.
- **Service level.** Service levels establish dispatch priority among tasks that are waiting to be dispatched. You can create different service levels that a workflow developer can assign to workflows.
- **Resource provision thresholds.** The Load Balancer checks one or more resource provision thresholds to determine if it can dispatch a task. The Load Balancer checks different thresholds depending on the dispatch mode.

### Configuring the Dispatch Mode

The Load Balancer uses the dispatch mode to select a node to run a task. By default, the Load Balancer uses the round-robin dispatch mode. You can configure the Load Balancer to use the metric-based or adaptive dispatch modes.

The following table compares the differences among dispatch modes:

<table>
<thead>
<tr>
<th>Dispatch Mode</th>
<th>Checks resource provision thresholds?</th>
<th>Uses task statistics?</th>
<th>Uses CPU profile?</th>
<th>Allows bypass in dispatch queue?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round-Robin</td>
<td>Checks maximum processes.</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Metric-Based</td>
<td>Checks all thresholds.</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Checks all thresholds.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Round-Robin Dispatch Mode

In round-robin dispatch mode, the Load Balancer dispatches tasks to nodes in a round-robin fashion. The Load Balancer checks the Maximum Processes resource provision threshold on the first available node. It dispatches the task to this node if dispatching the task does not cause this threshold to be exceeded. If dispatching the task causes this threshold to be exceeded, the Load Balancer evaluates the next node. It continues to evaluate nodes until it finds a node that can accept the task.

The Load Balancer dispatches tasks for execution in the order the Workflow Manager or scheduler submits them. The Load Balancer does not bypass any task in the dispatch queue. Therefore, if a resource-intensive task is first in the dispatch queue, all other tasks with the same service level must wait in the queue until the Load Balancer dispatches the resource-intensive task.

This mode is the least compute-intensive and is useful when the load on the grid is even and the tasks to dispatch have similar computing requirements.

### Metric-Based Dispatch Mode

In metric-based dispatch mode, the Load Balancer evaluates nodes in a round-robin fashion until it finds a node that can accept the task. The Load Balancer checks the resource provision thresholds on the first available node. It dispatches the task to this node if dispatching the task causes none of the thresholds to be exceeded. If dispatching the task causes any threshold to be exceeded, or if the node is out of free swap space, the Load Balancer evaluates the next node. It continues to evaluate nodes until it finds a node that can accept the task.
To determine whether a task can run on a particular node, the Load Balancer collects and stores statistics from the last three runs of the task. It compares these statistics with the resource provision thresholds defined for the node. If no statistics exist in the repository, the Load Balancer uses the following default values:

- 40 MB memory
- 15% CPU

The Load Balancer dispatches tasks for execution in the order the Workflow Manager or scheduler submits them. The Load Balancer does not bypass any tasks in the dispatch queue. Therefore, if a resource intensive task is first in the dispatch queue, all other tasks with the same service level must wait in the queue until the Load Balancer dispatches the resource intensive task.

This mode prevents overloading nodes when tasks have uneven computing requirements.

**Adaptive Dispatch Mode**

In adaptive dispatch mode, the Load Balancer evaluates the computing resources on all available nodes. It identifies the node with the most available CPU and checks the resource provision thresholds on the node. It dispatches the task if doing so does not cause any threshold to be exceeded. The Load Balancer does not dispatch a task to a node that is out of free swap space.

In adaptive dispatch mode, the Load Balancer uses the CPU profile to rank nodes according to the amount of computing resources on the node. This ensures that nodes with higher processing power get precedence for dispatch. The CPU profile is an index of the processing power of a node compared to a baseline system. For example, if the CPU is running 1.5 times as fast as the baseline system, the value of this property is 1.5. By default, the CPU profile is set to 1.0. You can calculate the CPU profile for a node or manually update the value.

To identify the best node to run a task, the Load Balancer also collects and stores statistics from the last three runs of the task and compares them with node load metrics. If no statistics exist in the repository, the Load Balancer uses the following default values:

- 40 MB memory
- 15% CPU

In adaptive dispatch mode, the order in which the Load Balancer dispatches tasks from the dispatch queue depends on the task requirements and dispatch priority. For example, if multiple tasks with the same service level are waiting in the dispatch queue and adequate computing resources are not available to run a resource intensive task, the Load Balancer reserves a node for the resource intensive task and keeps dispatching less intensive tasks to other nodes.

This mode prevents overloading nodes and ensures the best performance on a grid that is not heavily loaded.

**Steps to Configure the Dispatch Mode**

Use the Administrator tool to configure the Load Balancer dispatch mode. You configure a dispatch mode for the domain. Therefore, all PowerCenter Integration Services in a domain use the same dispatch mode.

1. In the Administrator tool, click the **Domain** tab.
2. Select the domain in the Navigator.
3. Click the **Properties** view.
4. Click **Edit** in the General Properties area.
5. Select the dispatch mode and click **OK**.
6. Restart each PowerCenter Integration Service in the domain for the change to take effect.
   - The previous dispatch mode remains in effect until you restart the PowerCenter Integration Service.
7. If you configured the adaptive dispatch mode, calculate the CPU profile for each node in the grid.
By default, the CPU profile is set to 1.0 for each node.

a. Select a node in the Navigator.

b. Click Actions > Recalculate CPU Profile Benchmark.

   The calculation takes approximately five minutes and uses 100% of one CPU on the machine. To get the most accurate value, calculate the CPU profile when the node is idle.

c. Repeat these steps for each node in the grid.

Creating Service Levels

Service levels establish priorities among tasks that are waiting to be dispatched. By default, the Load Balancer uses a single service level with a maximum dispatch wait time of 1,800 seconds. You can create additional service levels to determine the order in which the Load Balancer dispatches tasks.

When the Load Balancer has more tasks to dispatch than the PowerCenter Integration Service can run at the time, the Load Balancer places those tasks in the dispatch queue. When multiple tasks are waiting in the dispatch queue, the Load Balancer uses service levels to determine the order in which to dispatch tasks from the queue.

Each service level that you create has the following properties:

- Name. Name of the service level.
- Dispatch priority. A number that establishes the priority for dispatch. The Load Balancer dispatches high priority tasks before it dispatches low priority tasks. Dispatch priority 1 is the highest priority.
- Maximum dispatch wait time. The amount of time the Load Balancer waits before it changes the dispatch priority for a task to the highest priority. This ensures that no task waits forever in the dispatch queue.

When you create a service level, a workflow developer can assign it to a workflow in the Workflow Manager. All tasks in a workflow have the same service level. The Load Balancer uses service levels to dispatch tasks from the dispatch queue. For example, you create two service levels:

- Service level “Low” has dispatch priority 10 and maximum dispatch wait time 7,200 seconds.
- Service level “High” has dispatch priority 2 and maximum dispatch wait time 1,800 seconds.

When multiple tasks are in the dispatch queue, the Load Balancer dispatches tasks with service level High before tasks with service level Low because service level High has a higher dispatch priority. If a task with service level Low waits in the dispatch queue for two hours, the Load Balancer changes its dispatch priority to the maximum priority so that the task does not remain in the dispatch queue indefinitely.

Steps to Create a Service Level

You create service levels in the domain properties. Therefore, you can use the same service levels for all repositories in a domain.

1. In the Administrator tool, click the Domain tab.
2. Select the domain in the Navigator.
3. Click the Properties view.
4. In the Service Level Management area, click Add.
5. Enter values for the following service level properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Level Name</td>
<td>Name of the service level. The name is not case sensitive and must be unique within the domain. The name cannot have leading or trailing spaces, include carriage returns or tabs, exceed 79 characters, or contain the following characters: / * ? &lt; &gt;</td>
</tr>
<tr>
<td>Dispatch Priority</td>
<td>The initial priority for dispatch. Smaller numbers have higher priority. Priority 1 is the highest priority. Range is 1 to 10. Default is 5.</td>
</tr>
<tr>
<td>Maximum Dispatch Wait Time</td>
<td>The amount of time, in seconds, that can elapse before the Load Balancer escalates the dispatch priority for a task to the highest priority. Range is 1 to 86,400. Default is 1,800.</td>
</tr>
</tbody>
</table>

6. Click OK.

**Defining Resource Provision Thresholds**

The Load Balancer can continue to dispatch tasks to a node as long as the resource provision thresholds defined for the node are not exceeded. The Load Balancer is configured to use default values for the resource provision thresholds. You can use the default values on each node or you can modify the values to optimize performance for your environment.

You define resource provision thresholds for each node in the domain.

1. In the Administrator tool, click the **Domain** tab.
2. Select a node in the Navigator.
3. Click the **Properties** view.
4. Click **Edit** in the Resource Provision Threshold area.
5. Edit the following resource provision thresholds:

<table>
<thead>
<tr>
<th>Resource Provision Threshold</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Processes</td>
<td>Maximum number of running processes allowed for each PowerCenter Integration Service process that runs on the node. This threshold specifies the maximum number of running Session or Command tasks allowed for each Integration Service process running on the node. For example, if you set this threshold to 10 when two PowerCenter Integration Services are running on the node, the maximum number of Session tasks allowed for the node is 20 and the maximum number of Command tasks allowed for the node is 20. Therefore, the maximum number of processes that can run simultaneously is 40. Set this threshold to a high number, such as 200, to cause the Load Balancer to ignore it. To prevent the Load Balancer from dispatching tasks to this node, set this threshold to 0. Default is 10. Minimum is 0. Maximum is 1,000,000,000. Used in all dispatch modes.</td>
</tr>
<tr>
<td>Maximum CPU Run Queue Length</td>
<td>Maximum number of runnable threads waiting for CPU resources on the node. The Load Balancer does not count threads that are waiting on disk or network I/Os. If you set this threshold to 2 on a 4-CPU node that has four threads running and two runnable threads waiting, the Load Balancer does not dispatch new tasks to this node. This threshold limits context switching overhead. Set this threshold to a low number to preserve computing resources for other applications. Set this threshold to a high value, such as 200, to cause the Load Balancer to ignore it. Default is 10. Minimum is 0. Maximum is 1,000,000,000. Used in metric-based and adaptive dispatch modes. Ignored in round-robin dispatch mode.</td>
</tr>
<tr>
<td>Maximum Memory %</td>
<td>Maximum percentage of virtual memory allocated on the node relative to the total physical memory size. If you set this threshold to 120% on a node, and virtual memory usage on the node is above 120%, the Load Balancer does not dispatch new tasks to the node. Set this threshold to a value greater than 100% to allow the allocation of virtual memory to exceed the physical memory size when dispatching tasks. Set this threshold to a high value, such as 1,000, if you want the Load Balancer to ignore it. Default is 150. Minimum is 0. Maximum is 1,000,000,000. Used in metric-based and adaptive dispatch modes. Ignored in round-robin dispatch mode.</td>
</tr>
</tbody>
</table>

6. Click OK.

7. Repeat steps 2 to 6 for each node in the domain for which you want to modify the default resource provision thresholds.

**Configuring Workflows to Run on the Grid**

To configure workflows to run on the grid, complete the following steps in the Workflow Manager:

1. Configure the workflow to run on the grid.
2. If you created service levels for the Load Balancer, assign service levels to workflows.
3. If you configured the PowerCenter Integration Service to check resources, assign resources to tasks.
Step 1. Configure a Workflow to Run on a Grid

Before you can run a workflow, you must assign a PowerCenter Integration Service to run the workflow. To configure a workflow to run on a grid, assign a PowerCenter Integration Service that is configured to run on a grid.

You can choose a PowerCenter Integration Service to run a workflow by editing the workflow properties. You can also assign an Integration Service from the menu. When you assign a service from the menu, you can assign multiple workflows without editing each workflow.

Assigning a Service from the Workflow Properties

When you assign a PowerCenter Integration Service from the workflow properties, you assign the service to a single workflow.

1. Open the workflow in the Workflow Designer.
2. Click Workflows > Edit.
   The Edit Workflow dialog box appears.
3. On the General tab, click the Browse Integration Services button.
   A list of PowerCenter Integration Services appear.
4. Select the PowerCenter Integration Service that you want to run the workflow. Verify that the service is configured to run on a grid.
5. Click OK twice.

Assigning a Service from the Menu

When you assign a PowerCenter Integration Service from the menu, you can assign the service to multiple workflows at the same time. When you assign a service from the menu, you overwrite the service selected in the workflow properties.

1. In the Workflow Manager, close all folders in the repository.
2. Click Service > Assign Integration Service.
   The Assign Integration Service dialog box appears.
3. From the Choose Integration Service list, select the service that you want to assign. Verify that the service is configured to run on a grid.
4. From the Show Folder list, select the folder that you want to view. Or, click All to view workflows in all folders in the repository.
5. Click the Selected check box for each workflow that you want the PowerCenter Integration Service to run.
6. Click Assign.

Step 2. Assign Service Levels to Workflows (Optional)

Each workflow is assigned the default service level. If you created service levels for the Load Balancer, assign the appropriate service level to each workflow. All tasks in the workflow have the same service level.

1. Open the workflow in the Workflow Designer.
2. Click Workflows > Edit.
   The Edit Workflow dialog box appears.
3. On the General tab, select a service level in the Load Balancing area.
4. Click OK.
**Step 3. Assign Resources to Tasks (Optional)**

If you configured the PowerCenter Integration Service to check resources, assign resources to tasks.

The following table lists resource types and the repository objects to which you can assign them:

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Predefined/ User-Defined</th>
<th>Repository Objects that Use Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom</td>
<td>User-defined</td>
<td>Session, Command, and predefined Event-Wait task instances and all mapping objects within a session.</td>
</tr>
<tr>
<td>File/Directory</td>
<td>User-defined</td>
<td>Session, Command, and predefined Event-Wait task instances, and the following mapping objects within a session:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Aggregator transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Custom transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- External Procedure transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- HTTP transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Java transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Joiner transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lookup transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Sorter transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Source qualifiers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQL transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Targets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Union transformation</td>
</tr>
<tr>
<td>Node Name</td>
<td>Predefined</td>
<td>Session, Command, and predefined Event-Wait task instances and all mapping objects within a session.</td>
</tr>
<tr>
<td>Operating System Type</td>
<td>Predefined</td>
<td>Session, Command, and predefined Event-Wait task instances and all mapping objects within a session.</td>
</tr>
</tbody>
</table>

The Workflow Manager assigns connection resources. When you use a relational, FTP, or external loader connection, the Workflow Manager assigns the connection resource to sources, targets, and transformations in a session instance. You cannot manually assign a connection resource in the Workflow Manager.

1. Open the task properties in the Worklet or Workflow Designer.
   - If the task is an Event-Wait task, you can assign resources if the task waits for a predefined event.
2. On the **General** tab, click **Edit**.
3. In the **Edit Resources** dialog box, click the **Add** button to add a resource.
4. In the **Select Resource** dialog box, choose an object you want to assign a resource to. The Resources list shows the resources available to the nodes where the PowerCenter Integration Service runs.
5. Select the resource to assign and click **Select**.
6. In the **Edit Resources** dialog box, click **OK**.

**Rules and Guidelines for Configuring a Workflow to Run on a Grid**

Use the following rules and guidelines when you configure a workflow to run on a grid:

- If you override a service process variable, ensure that the PowerCenter Integration Service can access input files, caches, logs, storage and temporary directories, and source and target file directories.
- When you run a session that creates cache files, configure the root and cache directory to use a shared location to ensure consistency between cache files.
- Configure the workflow to run on a grid when you have multiple concurrent sessions.
To ensure that the Log Viewer can accurately order log events when you run a workflow on a grid, use time synchronization software to ensure that the nodes of a grid use a synchronized date/time.

If the workflow uses an Email task in a Windows environment, configure the same Microsoft Outlook profile on each node to ensure that the Email task can run.

Other Considerations

When you configure a workflow to run on a PowerCenter grid, you might also want to consider the following factors:

- Grid connectivity and recovery.
- Running a session on a PowerCenter grid.

Grid Connectivity and Recovery

When you run a workflow on a grid, service processes run on different nodes. Network failures can cause connectivity loss between processes running on separate nodes. Services may shut down unexpectedly, or you may disable the PowerCenter Integration Service or service processes while a workflow is running. The PowerCenter Integration Service failover and recovery behavior in these situations depends on the service process that is disabled, shuts down, or loses connectivity. Recovery behavior also depends on the following factors:

- High availability option. When you have high availability, workflows fail over to another node if the node or service shuts down. If you do not have high availability, you can manually restart a workflow on another node to recover it.
- Recovery strategy. You can configure a workflow to suspend on error. You configure a recovery strategy for tasks within the workflow. When a workflow suspends, the recovery behavior depends on the recovery strategy you configure for each task in the workflow.
- Shutdown mode. When you disable an Integration Service or service process, you can specify that the service completes, aborts, or stops processes running on the service. Behavior differs when you disable the Integration Service or you disable a service process. Behavior also differs when you disable a master service process or a worker service process. The Integration Service or service process may also shut down unexpectedly. In this case, the failover and recovery behavior depend on which service process shuts down and the configured recovery strategy.
- Running mode. If the workflow runs on a grid, the Integration Service can recover workflows and tasks on another node. If a session runs on a grid, you cannot configure a resume recovery strategy.

For more information about workflow recovery, see the PowerCenter Advanced Workflow Guide.

Running a Session on a PowerCenter Grid

After you have run workflows on a PowerCenter grid, you might want to consider running sessions on the grid. Running sessions on the grid is an advanced capability. To run sessions on a grid, you must have the Session on Grid option.

When you run a workflow on a grid, the PowerCenter Integration Service distributes Session, Command, and predefined Event-Wait tasks across the nodes in a grid.

When you run a session on a grid, the PowerCenter Integration Service distributes session threads across nodes in a grid. The master service process runs the workflow and all tasks except Session, Command, and predefined Event-Wait tasks as it does when you run a workflow on a grid. The Scheduler runs on the master service process node, so it uses the date and time for the master service process node to start scheduled workflows. In addition, the Load Balancer distributes session threads to DTM processes running on different nodes.
When you run a session on a grid, the Load Balancer distributes session threads based on the following factors:

- **Node availability.** The Load Balancer verifies which nodes are currently running, enabled, and available for task dispatch.
- **Resource availability.** If the Integration Service is configured to check resources, it identifies nodes that have resources required by mapping objects in the session.
- **Partitioning configuration.** The Load Balancer dispatches groups of session threads to separate nodes based on the partitioning configuration.

You might want to configure a session to run on a grid when the workflow contains a session that takes a long time to run.

For more information about running sessions on a grid, see the *PowerCenter Advanced Workflow Guide*.

### Optimizing Grid Deployments

When you run PowerCenter workflows on a grid, you can configure the grid, workflows, and sessions to use resources efficiently and maximize scalability.

To improve PowerCenter performance on a grid, complete the following tasks:

- Add nodes to the grid.
- Increase storage capacity and bandwidth.
- Use shared file systems.
- Use a high-throughput network when you access sources and targets over the network.

### Storing Files

When you configure PowerCenter to run on a grid, you specify the storage location for different types of session files, such as source files, log files, and cache files. To improve performance, store files in optimal locations. For example, store persistent cache files on a high-bandwidth shared file system. Different types of files have different storage requirements.

You can store files in the following types of locations:

- **Shared file systems.** Store files on a shared file system to enable all Integration Service processes to access the same files. You can store files on low-bandwidth and high-bandwidth shared file systems.
- **Local.** Store files on the local machine running the Integration Service process when the files do not have to be accessed by other Integration Service processes.

### High Bandwidth Shared File System Files

Because they can be accessed often during a session, place the following files on a high-bandwidth shared file system:

- Source files, including flat files for lookups.
- Target files, including merge files for partitioned sessions.
- Persistent cache files for lookup or incremental aggregation.
- Non-persistent cache files for only grid-enabled sessions on a grid.

This allows the Integration Service to build the cache only once. If these cache files are stored on a local file system, the Integration Service builds a cache for each partition group.
Low Bandwidth Shared File System Files

Because they are accessed less frequently during a session, store the following files on a low-bandwidth shared file system:

- Parameter files or other configuration related files.
- Indirect source or target files.
- Log files.

Local Storage Files

To avoid unnecessary file sharing when you use shared file systems, store the following files locally:

- Non-persistent cache files for sessions that are not enabled for a grid, including Sorter transformation temporary files.
- Individual target files for different partitions when performing a sequential merge for partitioned sessions.
- Other temporary files that are deleted at the end of a session run. In general, to establish this, configure $PmTempFileDir for a local file system.

Avoid storing these files on a shared file system, even when the bandwidth is high.

Using a Shared File System

You can use the following shared file systems for file sharing:

- Network file systems such as CIFS (SMB) on Windows or Network File System (NFS) on UNIX. Although network file systems are not designed for high-performance computing, they can work well for sequential file access.
- Clustered file systems. Clustered file systems provide a group of nodes with high-bandwidth file access, as well as a unified namespace for files and directories. Clustered file system performance is similar to a direct-attached local file system. If you have the High Availability option, use a clustered file system.

Note: For more information about supported shared file systems, see the Informatica Knowledge Base article Statement of Support on Shared File Systems.

Proper configuration and tuning can be critical for small grid performance. You can also configure mappings and sessions to avoid the intrinsic limitations of shared file systems.

Configuring a Shared File System

Use the following general guidelines to configure shared file systems:

- Make sure the network has enough bandwidth.
- Make sure the underlying storage has enough I/O bandwidth.
- Configure the shared file system daemons, particularly the client, to have enough threads to access files quickly. For example, IBM recommends that you estimate the number of files that require simultaneous access and provide at least two biod threads for each file.
  - When you run concurrent sessions on a grid that use flat file sources or targets, provide enough threads so each partition can access the source or target files that they need simultaneously.
- Configure mount points of the shared file system based on access requirements. When running sequential sessions on a grid that use flat file sources or targets, avoid any configuration that might degrade the effectiveness of the default read-ahead or write-behind process. File systems optimize sequential file access with read-ahead and write-behind.
- If necessary, tune the shared file system read-ahead and write-behind settings.
• Review the cache settings of the shared file systems for both the client and server. Increasing the default settings may improve performance.
• Configure the release-behind settings of the file system to free memory pages after data is accessed. Otherwise, system performance might degrade when reading or writing large files.
• Because of the difference in access patterns, you might use different mount points for sources and targets, and persistent caches.

For more information, see the shared file system documentation.

Balancing CPU and Memory Usage

Unlike local file systems, a shared file system server can take extra CPU cycles to access files. If you use one of the computation nodes as the shared file system server for the rest of the nodes, it might become overloaded and become a bottleneck for the entire grid. When the shared file system server is overloaded, CPU cycles can increase, along with repeated transmissions and time-out requests.

To avoid this, use one or more machines as dedicated shared file system servers for your PowerCenter grid nodes. Each machine should have enough storage, CPUs, and network bandwidth for required tasks.

Alternatively, you can cross-mount the shared file system server to distribute the file server load across the nodes of the grid. When PowerCenter mappings and sessions are configured to use an even balance of I/O and CPU usage, cross-mounting shared file system servers can optimize performance. If the number of nodes in the grid is small and you have a balanced mix of I/O and CPU usage, you might not need a dedicated shared file system server.

When you use more than one shared file system server, dedicated or cross-mounted, try to distribute shared files across the servers.

Configuring PowerCenter Mappings and Sessions

One of the most important ways to improve performance is to avoid unnecessary file sharing. When properly configured, shared file systems can provide good performance for the sequential access of source and target files. However, the random access required for persistent cache files, especially large persistent cache files, can be more problematic.

Use the following guidelines for configuring persistent cache files, such as persistent dynamic lookups, for a grid with a shared file system:

• When possible, configure the session cache size to keep smaller persistent cache files in memory.
• Add a Sorter transformation to the mapping to sort the input rows before the persistent lookup. Shifting the work from the persistent lookup to the Sorter transformation can improve performance because the Sorter transformation can use the local file system.
• Group rows that require access to the same page of the lookup cache to minimize the number of times the Integration Service reads each page of the cache.
• When the size of input data is large, use source-based commits to manage input data to allow sorting to be performed in memory.

For example, you have a 4 GB persistent dynamic lookup that cannot be reduced without changing the mapping logic and you have 10 GB of source data. First add a Sorter transformation to sort input data to reduce random access of the lookup cache, then complete the following tasks:

- Configure the session to perform source-based commits with 1 GB commit intervals.
- Set the Sorter transformation transaction scope to Transaction.
- Configure the Sorter transformation for a 1 GB cache size, enough for the source input.
With this configuration, the Integration Service sorts 1 GB of input data at a time and passes rows to the persistent lookup that require access to similar data in the cache.

- If more than one file system is available, configure the cache files for each partition to use different file systems.
- Configure the sessions to distribute the files to different file systems if more than one file system is available.

**Distributing Files Across File Systems**

Distribute files to different file systems to use the combined bandwidth of the file systems assuming each file system uses an independent physical disk sub-system. Distributing files to different file systems can increase performance on a grid that uses either a shared file system or symmetric multiprocessing (SMP).

For optimal I/O bandwidth choose a file system that distributes files across multiple storage devices. If you use a clustered file system, distribute the files between servers. If possible, place the source, target, and cache files on different storage devices.

Use the following guidelines when you distribute files on file systems:

- **Source files.** If you place source files on a file system that enables the Integration Service to read data from a large number of files, tune the file system read-ahead setting before caching large files.
- **Temporary files.** If you place temporary files on a file system that enables the Integration Service to read data from large files and write to temporary files, tune the file system read and write settings for large files.
- **Target files.** If you place target files on a file system that enables the Integration Service to write large files to the disk, tune the file system for simultaneous large block writes. Target files can include merge files for partitioned sessions. Since partitioned sessions on a grid need to write files to the disk, tune the file system for optimal locking performance.

**Configuring Sessions to Distribute Files**

You can manually configure sessions to distribute the file load. You might need to edit sessions when the load changes significantly or when you add new sessions or file systems, including adding new nodes to a grid with a cross-mounted shared file system.

Instead of editing sessions manually, use session variables to distribute files to different directories. This allows you to redirect session files to different file servers when necessary.

Use the following guidelines to use session variables:

- Name variables for session file names and directories to reflect business logic.
- In the parameter file, define each variable so the file load is evenly distributed across all available file systems. You can also define node-specific variables.
- Optionally, automate reconfiguration with a script to process parameter files.

**Note:** When you use a script, use a placeholder in the parameter file so the script can redefine session variables as necessary.

**Author**

Alison Taylor  
Senior Technical Writer

**Acknowledgements**

The author would like to acknowledge Nikunj Bansal and Sean Ma for their contributions to this article.