How to Achieve Greater Availability in Enterprise Data Integration Systems
Abstract

This article describes the benefits of a highly available environment and how PowerCenter’s architecture and high availability features can help companies achieve a fault tolerant and highly available environment.

Supported Versions

- PowerCenter 8.6.x

Table of Contents

Overview ........................................................................................................................................................................... 2
Benefits of High Availability for Data Integration Solutions ............................................................................................... 3
PowerCenter High Availability ........................................................................................................................................... 3
  Domain Configuration Database .................................................................................................................................. 4
  Nodes ........................................................................................................................................................................... 5
  PowerCenter Services .................................................................................................................................................. 5
PowerCenter and the High Availability Option .................................................................................................................. 8
High Availability Requirements and Configuration ............................................................................................................ 8
  Infrastructure Requirements ......................................................................................................................................... 8
  Configuring a Highly Available PowerCenter Environment ........................................................................................... 9
High Availability Center of Excellence ............................................................................................................................ 11
  Guidelines for a Cluster File System .......................................................................................................................... 11
  Best Practices ............................................................................................................................................................. 11
  Example ...................................................................................................................................................................... 12
Conclusion ...................................................................................................................................................................... 13

Overview

Historically, companies relied on single-purpose ETL (extract, transform, and load) applications to integrate data and build data warehouses. Occasional system failures during the nightly ETL operations caused inconvenience, but for the most part could be tolerated. In recent years, however, enterprise data integration technology has evolved beyond ETL applications to support a variety of mission-critical and time-sensitive operational applications across multiple businesses and functional areas. Today’s businesses can lose hundreds of millions of dollars if the underlying data integration technology is not very robust or becomes unavailable due to system failures.

To avoid the financial losses caused by system failures and unplanned downtime, it is critical that a data integration solution be both fault tolerant and highly available. This article describes the benefits of a highly available environment and how PowerCenter’s architecture and high availability features can help companies achieve a fault tolerant and highly available environment.

Note: High availability is not a solution for disaster recovery. You can use high availability features to implement a disaster recovery solution.
Benefits of High Availability for Data Integration Solutions

The term high availability refers to the uninterrupted availability of computer system resources. Fault tolerant systems minimize or prevent interruption due to system failures. Highly available systems maximize the amount of time that they are operational so that end users can continue to access and use the system despite temporary network or hardware failures. If you use PowerCenter to process real-time data, it is recommended that you have a highly available environment since the data integrity of the sources depends on the PowerCenter’s state and ability to read and process messages.

Highly available systems provide the following key capabilities:

- **Resilience.** Highly available systems can tolerate temporary connection failures until a timeout period expires or the failure is resolved. The system tries to reconnect for a specified period of time. If the failure is resolved, there is no interruption in end user activity.

- **Restart and failover.** In highly available systems, when a machine becomes unavailable, processes running on the machine can be restarted on the same machine or on a backup machine. By allowing processes to restart on the same machine or fail over to another machine, the system minimizes or eliminates the downtime due to the failure and maximizes the system operational time.

- **Recovery.** In highly available systems, an interrupted service can complete its operations after it is restarted. A service may be stateful—that is, it records its state of operation in a shared location periodically. When a failure occurs, the system must retrieve the state of the affected service so that it can automatically restart or recover jobs that have terminated abnormally.

Resilience, failover, and recovery capabilities are essential components for a robust data integration solution. When implementing a data integration solution in a multi-node environment, consider the following requirements:

- The solution must utilize an architecture that can tolerate external system failures. End users should notice no interruption in their activities when a machine or network component fails.

- The solution must be resilient to transient failures in network components. When a component comes back on line, the solution must reconnect to it seamlessly.

- The solution must allow services to run without interruption when a component becomes unavailable. If a machine fails, the solution must restart services on a different machine.

- The solution must be able to resume a task when it fails over to a secondary machine. Tasks that have partially completed should not have to be restarted manually or from scratch.

PowerCenter High Availability

PowerCenter high availability can eliminate single points of failure, thereby maximizing system uptime and reducing the costs and risks associated with system failures. PowerCenter resilience, failover, and recovery capabilities ensure that system failures are as transparent as possible to end users. When you implement PowerCenter high availability for a domain, the domain can continue running despite temporary network, hardware, or service failures.

The PowerCenter domain is the fundamental unit of PowerCenter Services administration. The domain configuration database stores domain configuration metadata and usage information. The domain defines the PowerCenter environment. It consists of the following objects and services:

- A set of nodes (physical computers or blades) where PowerCenter Services are installed and run. These nodes are connected by a network.

- **PowerCenter services** that represent different aspects of PowerCenter functionality. For example, the Repository Service manages the PowerCenter metadata repository and the Integration Service runs PowerCenter tasks.
Administrators can configure PowerCenter components to achieve a greater degree of high availability and fault tolerance. The following figure shows an example PowerCenter domain set up for high availability:

**Figure 1: Example PowerCenter Domain with High Availability**

A Service Manager runs on each node in the domain. The Administration Console, PowerCenter Client tools, and the command line programs are clients to the Service Manager. Every client connects to the master gateway to get the current location of the required service. When you have multiple nodes in a PowerCenter domain, the service manager on each node updates the master gateway with its status and the statuses of each service on the node.

In the example above, PowerCenter components are configured as follows:

- Nodes 1, 2, and 3 are assigned to a grid, and Integration Service 2 is configured to run on a grid. A grid is a group of nodes that run sessions and workflows. When an Integration Service runs on a grid, its service processes can run on any of the nodes in the grid. If a node fails, Integration Service 2 can continue operations on other nodes.

- Nodes 1 and 2 are gateway nodes. Node 1 currently acts as the master gateway node, so the Service Manager performs domain functions on this node. Node 3 is a worker node. If the master gateway fails, domain operations continue on another gateway node. If a worker node fails, its service processes fail over to another node.

- The Repository Service is configured to run on node 1, the primary node for the service. If node 1 is unavailable, the Repository Service can run on node 2, the backup nodes for the service.

**Domain Configuration Database**

The domain configuration database stores domain metadata such as the names, host names, and port numbers of nodes in the domain. It also stores CPU and repository usage summary information, the location of workflow and session log binary files, and information about the master gateway node. To prevent the domain from shutting down if the master gateway node fails the administrator should make sure that each gateway node in the domain has access to the domain configuration database. If high availability of the overall data integration system is desired, this database should be made highly available.
**Nodes**

In the PowerCenter domain, a node is the logical representation of a physical machine. A node can be either a gateway or a worker node. Gateway nodes run PowerCenter services, receive service requests from clients, and route requests to the appropriate service and node. Worker nodes run PowerCenter services. Each node, whether it is a gateway or worker, runs a Service Manager process that controls the PowerCenter services running on the node. The Service Manager starts and stops services and service processes on the node and directs requests to PowerCenter services. The Service Manager runs as a process on Windows and a daemon on UNIX. If the Service Manager is not running on a node—for example, because the machine or Service Manager process is shut down—the node is unavailable.

PowerCenter utilizes a system of master and backup gateways to ensure that the domain continues to function if a node becomes unavailable. The PowerCenter administrator can designate multiple gateway nodes. One gateway serves as the master gateway, and the others are backup gateways. PowerCenter stores this information in the domain configuration database.

The Service Manager running on the master gateway node performs domain functions such as user authentication, service address resolution, request authorization, license registration and verification, and event logging. These important functions must be protected from master gateway node failure. To make sure the master gateway node is up and running, the Service Manager on the master gateway node periodically updates the domain configuration database with timestamp information. If the timestamp does not change, Service Manager processes running on the backup gateways elect a new master using an intelligent, distributed election algorithm.

The master gateway election algorithm uses the domain configuration database as the arbitrator in determining which node becomes the master gateway. When a new gateway node is started, it checks the timestamp information in the domain configuration database and tries to contact the master gateway node so that it can join the domain. If the master gateway fails and multiple gateway nodes are started at the same time, the first node to obtain a row lock in the database becomes the new master gateway. This intelligent election process ensures that a domain functions can continue even if the original master gateway node fails.

The PowerCenter high availability option also guards against backup gateway and worker node failure. Every node in the domain sends a heartbeat to the master gateway at a particular interval. The heartbeat includes a list of services running on the node. If a node fails to send a heartbeat, the master gateway marks the node unavailable and reassigns its services to another node. This process ensures that PowerCenter services continue to run despite node failure.

**PowerCenter Services**

The resilience, failover, and recovery capabilities of PowerCenter services minimize or prevent interruption in important data integration operations. The PowerCenter domain can host the following highly available services:

- **PowerCenter Integration Service.** The Integration Service is the ETL service that runs data integration sessions and workflows. It moves data from sources to targets based on workflow and mapping metadata stored in the PowerCenter metadata repository. The Integration Service starts one or more Integration Service processes to run and monitor workflows.

- **PowerCenter Repository Service.** The Repository Service manages the PowerCenter metadata repository. A Repository Service manages a single repository.

The following features make PowerCenter Services highly available:

- **Connection resilience.** PowerCenter components are resilient to the temporary unavailability of other PowerCenter components. PowerCenter components include the Service Manager, application services, the PowerCenter client, and the command line. For example, if the SAP BW Service cannot communicate with an Integration Service it depends on, it keeps trying until it succeeds or times out. With the PowerCenter high availability option, the PowerCenter administrator can configure the Integration Service and the Repository Service to be resilient to external failures such as source and target databases or network connections as well. For example, an Integration Service process keeps trying a failed network connection until it is restored or until the timeout period expires.
• **Failover.** The PowerCenter high availability option allows the administrator to configure the Integration Service and the Repository Service for failover. If a service process fails, the Service Manager can restart the process on the same node or fail it over to another node.

• **Service recovery.** The PowerCenter high availability option allows a service process to resume its unfinished operations after restart or failover. With the PowerCenter high availability option, you can also configure a workflow or task to recover automatically. For more information, see the “Recovery” section.

**Connection Resilience**

Because network component failures are often short-lived, data integration solutions must provide connection resilience. Connection resilience is the ability to reconnect to a component or service after a temporary failure. PowerCenter services establish connections with other PowerCenter services and with external components such as source or target databases and FTP servers. All PowerCenter services are designed to tolerate transient connection failures between the service and other PowerCenter components. The PowerCenter high availability option ensures that the Integration Service and the Repository Service are resilient to external connection failures as well. The PowerCenter administrators configure resilience timeout periods for internal resilience between PowerCenter services and retry periods for external connections.

The PowerCenter administrator configures **resilience timeout period** and the **limit on resilience timeout** for internal PowerCenter resilience between services. The **resilience timeout period** is the time a service attempts to connect to another service. The **limit on resilience timeout** is the amount of time a service waits for a client to connect or reconnect to the service. The **limit on resilience timeout** can override the client resilience timeouts configured for a connecting client. For example, the Integration Service must connect to the Repository Service to obtain the rules for data transformation from the PowerCenter metadata repository. The Repository Service communicates with the repository database to perform metadata transactions for the client application. If a connection between a service and another service terminates due to a network failure, PowerCenter attempts to reestablish the connection. It attempts to connect until the problem is resolved or until the **resilience timeout period** expires.

The PowerCenter administrator configures the **connection retry period** for source and target connections. When a PowerCenter service loses a connection with an external component, it attempts to reestablish the connection until the problem is fixed or until the **connection retry period** for the connection expires. If the connection can be restored within the retry period, PowerCenter restores the connection automatically.

The resilience of PowerCenter services ensures that they run without noticeable interruption and without the need for administrator intervention. The end user never notices a connection failure. Instead, it appears as though the connection paused temporarily. PowerCenter resilience also completely eliminates the need for the administrator to manually restart the ETL process every time the network experiences a transient failure.

**Failover Capability**

To protect its services from external failures, the PowerCenter high availability option provides failover capability for the Repository and Integration Services. Failover capability allows a service to restart on a backup node and continue its operations when the node on which it was originally running is unavailable after the resilience timeout period expires. When a machine becomes unavailable, service processes running on the machine can be restarted on the same node. If the machine becomes available within an acceptable period of time, the system restarts the service process on that machine. If the machine does not become available, the system restarts the service process on a backup machine.

The Repository Service runs on a single node. When the PowerCenter administrator configures the Repository Service for high availability, the administrator chooses a primary node and one or more backup nodes for the service. By default, the service process runs on the primary node for the service. If the primary node is unavailable when the service starts, or becomes unavailable while the service is running, PowerCenter fails over the service to one of the backup nodes. After a service process fails over to a backup node, it does not automatically fail back to the primary node when the primary node becomes available. If the service is running on a backup node and the backup node fails, the service fails over to the primary node if it is available or to another backup node if the primary node is unavailable.

The Integration Service can run multiple service processes at a time—one master service process and one or more worker service processes. The Integration Service can be assigned to primary and backup nodes, or it can be assigned to a grid. If the Integration Service is assigned to a primary node, all of its service processes fail over to a
backup node in the same way the Repository Service process fails over. The Integration Service must have access to the shared storage location for failover to occur successfully.

If the Integration Service is assigned to a grid, its service processes fail over in the following ways:

- If a node is unavailable when the Integration Service starts, or if a node running a worker service process fails or shuts down while the master service process is running, the Service Manager reconfigures the grid to run on one less node.
- If the node running the master service process fails or shuts down while it is running, PowerCenter fails over the master service process to another node on the grid.

After failover, any service that depends on the failed service process can reconnect to it without noticeable interruption or loss of service. This capability ensures that PowerCenter services are robust and fault tolerant.

Recovery

In addition to resiliency and failover capabilities, another key element of high availability is recovery. In situations where PowerCenter restart or failover is required, recovery minimizes the impact on the end user. It allows a service process to automatically resume its unfinished operations.

The Integration Service and Repository Service are stateful, so they record their states of operation in a shared location periodically. The Repository service stores the repository’s state in the repository database. The Integration Service stores the service state on the file system. When a node or service fails, PowerCenter will automatically failover or restart the service and will recover the state of the service from the last saved state. In addition, with PowerCenter high availability option, you can configure PowerCenter to automatically recover any Integration Service workflows and sessions that were interrupted due to the restart or failover.

Integration Service Recovery for Workflows and Tasks

The Integration Service performs ETL operations through a set of instructions known as a session. Sessions tell the Integration Service how and when to move data. A series of sessions and other tasks are grouped into a workflow. Workflow and session execution can be interrupted by node and network failures, as well as by stop or abort requests by a user.

When recovery is enabled for a workflow, the Integration Service stores the state of operation in a shared location that must be configured on a highly available shared file system. For each workflow, the Integration Service stores information about scheduled, running, and completed tasks. For sessions, Integration Service stores source and transformation information in the shared file system. The Integration Service stores relational target information in recovery tables in the target database.

With the PowerCenter high availability option, the workflow developer can configure workflow and task recovery behavior. The workflow developer can configure PowerCenter to recover a workflow automatically if a service process that is running the workflow fails over to another node. The workflow developer can also configure PowerCenter to automatically recover terminated tasks. To configure a workflow to recover automatically, configure the Enable HA recovery option in the workflow properties. To configure a task to recover automatically, configure the automatically recover terminated tasks option in the workflow properties.

When a workflow is recovered, the Integration Service accesses the shared file system to get the state of tasks in the workflow at the point of interruption. The Integration Service recovers the workflow from this point.

For sessions and command tasks within a workflow, the task developer can configure the recovery strategy for the task. Depending on the task, the task developer can configure the recovery strategy as follows:

- **Resume the task from the last checkpoint.** For Session tasks, the Integration Service saves the state of operation in shared file system. It resumes the session from the point of interruption, without reprocessing data. While this strategy slightly increases session overhead, it can greatly decrease recovery time for sessions that process large amounts of data. Checkpoint recovery is only supported for repeatable sources with transactional targets such as relational databases, MQ, and JMS.
- **Restart the task.** The Integration Service reruns the session or command task when it recovers the workflow. When using this option, the task developer must design the task to avoid data duplication—for example, by
enabling relational target truncation. This strategy does not increase processing time for the normal run of a task, so it is especially useful for smaller tasks that run frequently.

- **Fail the task and continue the workflow.** The Integration Service does not recover the session or command task. Instead, it marks the task as failed and continues running the workflow. This strategy prevents a failed task from "holding up" the processing of a workflow.

Repository Service Recovery

The Repository Service stores the repository state of operation in the repository database. This information includes information about repository locks, requests in progress, and connected clients. If the node on which the Repository Service is running fails, PowerCenter restarts the service on the same node or fails over the service to a backup node. The Repository Service restores the repository state of operation from the repository and recovers operations from the point of interruption. With the PowerCenter high availability option, this process is completely transparent—clients maintain access to the repository despite node failure and service failover.

PowerCenter and the High Availability Option

The PowerCenter base product includes some degree of high availability. The high availability option provides greater availability. The following table compares the Standard Edition features with those includes with the high availability option and the Real-Time Edition:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal PowerCenter Resilience</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Repository Database Resilience</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Integration Service External Connection Resilience</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Automatic Failover of the Domain to a Gateway Node</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Automatic Restart of Application Services on the same node</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Failover of the Integration Service and Repository Service on a Backup Node</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

* You must manually configure the workflow or task to recover automatically.

High Availability Requirements and Configuration

The PowerCenter domain and external components can be configured to provide varying degrees of high availability. It is important to realize the software and hardware requirement that must be met for a successful implementation. The requirements and configuration should be implemented based on the requirements of your environment. The software and hardware requirements section describe the requirements that you should review when planning a highly available environment. The configuration section describes how to configure PowerCenter components for high availability.

Infrastructure Requirements

Before setting up an environment to run a highly available PowerCenter domain, review the following requirements:

- **Multiple machines.** When you configure PowerCenter for high availability, you need to install PowerCenter on multiple machines so that you can configure multiple gateway nodes and backup nodes for the Integration Service and the Repository Service. It is recommended to have least three nodes. When you have three nodes in a PowerCenter domain, domain and service failover can still occur when one node is down. To plan that the domain will run with a majority of its nodes at all times, use an odd number of nodes to run the domain. For example, if two nodes are down in a five node domain, the majority of the nodes are still running.
- **Use highly available database system.** The domain configuration database and the PowerCenter repository databases should use a highly available database. The Repository Service stores its state of operations in the repository database. The Integration Service, Metadata Manager Service and Web Services Hub require that the Repository Service is available. If the domain configuration database is on a highly available database, the domain remains available even if one of the machines on which the database is installed fails.

- **Highly available POSIX compliant shared file system that is configured for I/O fencing.** PowerCenter requires a highly available POSIX compliant shared file system that is configured for I/O fencing in order to ensure Integration Service failover and recovery. To be highly available, the shared file system must be configured for I/O fencing. The hardware requirements and configuration of an I/O fencing solution are different for each file system. When possible, it is recommended to use hardware I/O fencing.

PowerCenter nodes need to be on the same shared file system so that they can share resources. For example, the Integration Service on each node needs to be able to access the log and recovery files within the shared file system. Also, all PowerCenter nodes within a cluster must be on the cluster file system's heartbeat network.

The following shared file systems are certified by Informatica for use in Integration Service failover and session recovery:

- Storage Array Network
  - Veritas Cluster Files System (VxFS)
  - IBM General Parallel File System (GPFS)
- Network Attached Storage using NFS v3 protocol
  - EMC UxFS hosted on an EMV Celerra NAS appliance
  - NetApp WAFL hosted on a NetApp NAS appliance

Informatica recommends that customers contact the file system vendors directly to evaluate which file system matches their requirements.

For more information, see the Statement of Support Regarding File System Support for Informatica PowerCenter High Availability Service Failover and Session Recovery on my.informatica.com.

- **Highly available external components.** Use highly available versions of external components used by Integration Services, such as source and target databases, message queues, and FTP servers. If data from an external component is not accessible, the Integration Service will not be able to process data required by a mapping.

- **Redundant network components.** Configure redundant network components, such as routers, cables, and network adapter cards.

Depending on the environment and the degree of availability that is required, the architecture required for a PowerCenter domain with high availability differs.

### Configuring a Highly Available PowerCenter Environment

To configure a highly available PowerCenter environment, the license on the PowerCenter domain must contain the high availability option. You can configure some level of high availability without the high availability option. For more information about which features are included with the high availability option, see the “PowerCenter and the High Availability Option” section.

#### PowerCenter Domain

- **Configure multiple gateway nodes.** Each gateway node in the domain needs access to the domain configuration database. This prevents the domain from shutting down if the master gateway node fails.

- **Configure Informatica Services to restart in the event of node failure.** Informatica Services runs the Service Manager on the node. If the node terminates unexpectedly, Informatica Service should restart when the nodes becomes available.

  On UNIX, configure a Veritas Cluster Server to restart Informatica Services. Or, you can create a script to
automatically start Informatica Service when the node starts. On Windows, go to Control Panel > Administrative Tools > Service and configure Informatica Services to start automatically.

Connection Resilience

- **Configure the resilience timeout for the Repository Service and the Integration Service.** The resilience timeout is the period of time that the service tries to establish or reestablish a connection to another service. If blank, the service uses the domain resilience timeout. The default value is 180 seconds.

- **Configure the domain resilience timeout.** The domain resilience timeout determines how long services attempt to connect as clients to other services. The default value is 30 seconds.

- **Configure the resilience timeout for command line programs.** When you use a command line program to connect to the domain or an application service, you can configure a resilience timeout. The default value is 180 seconds.

- **Configure the database connection timeout for the Repository Service.** The database connection timeout is the period of time that the Repository Service attempts to establish or reestablish a connection to the repository database. The default value is 180 seconds.

- **Configure the connection retry period for the connections to sources and targets.** The connection retry period for sources and targets is configured on the connection object within the PowerCenter Workflow Manager. The default value is 0 seconds.

Service Failover

- **Specify at least one backup node for the Repository Service.** If the node that runs the Repository Service fails, the Repository Service needs a backup node so that it can failover.

- **Specify at least one backup node for the Integration Service or configure it to run on a multi-node grid.** If the node that runs the Integration Service fails, the Integration Service needs a backup node so that the service and its processes can failover.

- **Ensure that any resource required by a task and used by an Integration Service is available on multiple nodes.** Resources required by a task can include a database connection, file, directory, node name, required operating system type, or parameter file. If the Integration Service fails over to a node that does not have the resources required for a task, the task cannot be completed.

Service Restart

- **Configure restart options for services.** If an application service process becomes unavailable while a node is running, the domain tries to restart the process based on the Maximum Restart Attempts option and the Restart Period option in the domain properties.

Workflow, Session and Task Recovery

- **Configure workflows and tasks to recover automatically.** To configure a workflow to recover automatically, configure the Enable HA recovery option in the workflow properties. To configure a task to recover automatically, configure the automatically recover terminated tasks option in the workflow properties.

- **Configure the recovery strategy for sessions and tasks.** To configure a recovery strategy for sessions, configure the Recovery Strategy option in the session properties. To configure a recovery strategy for tasks, configure the Recovery Strategy option in the task properties.

For more information about how to configure PowerCenter for high availability, see the PowerCenter documentation.
High Availability Center of Excellence

The High Availability Center of Excellence (HACOE), developed by Charles W. McDonald Jr., is an example high availability environment that utilizes best practices on a cluster file system.

Guidelines for a Cluster File System

A cluster file system configured for I/O Fencing can require the following components:

- Multi-channel / path private heartbeat network
- Multi-channel / path public network
- IP load balancing solution
- Multi-channel / path storage network connection
- HBA load balancing solution that is compatible with your Storage Array Network
- Storage Array Network with SCSI III Persistent Group Reservation (PGR) disks
- Cluster-ware solution for creating the clustered file system
- I/O Fencing solution

The environment may also require the following components:

- Separate public switches on separate power grids
- Separate VLAN’s / channel or path
- Cluster-ware cluster server solution. Cluster servers can automatically start the Service Manager on a node that has failed. The Veritas Cluster Server Agent increases the availability of PowerCenter Service Manager by restarting the Service Manager on a node that failed. You can also configure Veritas Cluster Server Agent to restart the service manager on each node in a certain order.
- High availability database solution

Best Practices

The following table list general best practices for architecture requirements:

<table>
<thead>
<tr>
<th>Requirement Type</th>
<th>Attribute</th>
<th>Best Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA Cluster File System</td>
<td>File System</td>
<td>Highly Available clustered file system configured for I/O fencing</td>
</tr>
<tr>
<td></td>
<td>Network Interface Cards</td>
<td>At least two multi-interface network cards per host that are bound together at layer 3 of OSI by TCP/IP binding agent</td>
</tr>
<tr>
<td></td>
<td>Disk/Spindles</td>
<td>SCSI III PGR capable</td>
</tr>
<tr>
<td>Heartbeat network</td>
<td>Bandwidth</td>
<td>Gb-Full Duplex, Infiniband or Oracle RAC</td>
</tr>
<tr>
<td></td>
<td>Network</td>
<td>Private</td>
</tr>
<tr>
<td></td>
<td>Connection</td>
<td>Separate VLAN's/path. Full duplex GigE supporting enterprise class switch. 2 switches minimum.</td>
</tr>
<tr>
<td></td>
<td>MTU Frame Size</td>
<td>Jumbo (9194, etc)</td>
</tr>
<tr>
<td></td>
<td>Subnet</td>
<td>Same subnet required</td>
</tr>
<tr>
<td></td>
<td>Paths</td>
<td>3 for application tier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 for database tier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 for Oracle RAC</td>
</tr>
</tbody>
</table>
### Example

The following table lists additional the best practices that were use to configure the High Availability Center of Excellence:

<table>
<thead>
<tr>
<th>Requirement Type</th>
<th>Attribute</th>
<th>Best Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heartbeat network</td>
<td>Paths</td>
<td>3 for application tier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 for database tier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 for Oracle RAC</td>
</tr>
<tr>
<td></td>
<td>TCP/IP</td>
<td>Only 1 heartbeat of the paths listed above for CRS traffic for Oracle RAC</td>
</tr>
<tr>
<td></td>
<td>link-lowpri</td>
<td>For Veritas, use this directive in place of link for public network interfaces.</td>
</tr>
<tr>
<td></td>
<td>CRS-Oracle RAC</td>
<td>For Veritas, use PrivNIC agent to make CRS heartbeat highly available. This</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eliminates the need for IPBind or IPMP use on CRS interconnect.</td>
</tr>
<tr>
<td></td>
<td>Eeprom parameter local-mac-</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td>address</td>
<td></td>
</tr>
</tbody>
</table>

The following configuration was used to run a High Availability Center of Excellence environment:

- **Application/DI Tier HP DL 585 Hosts 8 cores x 32GB RAM (AMD Opteron Dual Core 64-bit)**
  - 15K RPM local drives (2.5") SCSI3 RAID 1+0 configurations
  - RHEL 5 64 – bit, kernel 2.6.18 53
  - Veritas Storage Foundation Suite Cluster File System HA 4.1MP4 64- bit
  - Veritas Netbackup 6.5.1a
  - Using Veritas DMP for multi-pathing
  - PowerCenter 8.6.1 64 – bit
- **Database Tier HP DL 585 Hosts 8 cores x 64GB RAM (AMD Opteron Dual Core 64-bit)**
  - 15K RPM local drives (2.5") SCSI3 RAID 1+0 configurations
  - RHEL 5 64 – bit, kernel 2.6.18 53
  - Veritas Storage Foundation Suite Cluster File System HA 4.1MP4 64 – bit
  - Veritas Netbackup 6.5.1a
  - Using Veritas DMP for multi-pathing
  - 11gR1 Oracle RAC n-node configuration active/active TAF
Conclusion

When you configure a PowerCenter domain for high availability you eliminate single points of failure. Before you configure a highly available PowerCenter domain, configure the underlying architecture to support high availability. A PowerCenter domain is as highly available as the configuration of its infrastructure and PowerCenter components. The requirements, best practices, and configuration options provided in this article include the information that you need to configure a PowerCenter Domain that is highly available.

For more information about PowerCenter high availability, see the PowerCenter documentation.

Authors

Nikunj Bansal  
Development Manager

Padma Heid  
Senior Technical Writer

Sean Ma  
Senior Product Manager

Pinaki Mukerji  
Senior VP Engineering

Lori Troy  
Senior Technical Writer

Acknowledgements

The authors would like to thank Charles W. McDonald Jr. for providing information about the High Availability Center of Excellence, high availability infrastructure requirements, and high availability best practices.