IBM PowerHA SystemMirror for AIX

Standard Edition

Version 7.2

Planning PowerHA SystemMirror

Note

Before using this information and the product it supports, read the information in “Notices” on page 111.

This edition applies to IBM® PowerHA® SystemMirror® 7.2 Standard Edition for AIX® and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this document

This document introduces the PowerHA SystemMirror for AIX software. This information is also available on the documentation CD that is shipped with the operating system.

Highlighting

The following highlighting conventions are used in this document:

Bold Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.

Italics Identifies parameters whose actual names or values are to be supplied by the user.

`Monospace` Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

Case-sensitivity in AIX

Everything in the AIX operating system is case-sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type LS, the system responds that the command `is not found`. Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related information

- The PowerHA SystemMirror Version 7.2 for AIX PDF documents are available in the PowerHA SystemMirror 7.2 PDFs topic.
- The PowerHA SystemMirror Version 7.2 for AIX release notes are available in the PowerHA SystemMirror 7.2 release notes topic.

Planning PowerHA SystemMirror

Before you configure and install PowerHA SystemMirror, you must plan its implementation for the AIX operating system.

What's new in Planning for PowerHA SystemMirror

Read about new or significantly changed information in the Planning for PowerHA SystemMirror topic collection.

How to see what's new or changed

In this PDF file, you might see revision bars (|) in the left margin that identifies new and changed information.

December 2023

The following plan topics are updated:

- Added a new section about Physical volume encryption in the Physical volumes topic.
- Updated the "Logical volumes" on page 44 topic about logical volume encryption in data volume groups that are configured in PowerHA SystemMirror environment.
- Removed `Cleanup Method` attribute from procedure for adding a user defined resource. For more information see, Configuring user-defined resources. Updated Collocation with Persistent Label and Source attribute in procedure for types of distribution for service IP label aliases. For more information see. "Types of distribution for service IP label aliases" on page 29

June 2022

The following plan topics are updated:

- Added the Enabling ports for cluster communication in PowerHA SystemMirror topic about ports that are used to deploy PowerHA SystemMirror on premises or in the Cloud environment.
- Updated the "Virtual networks in PowerHA SystemMirror" on page 33 topic.

s were made to plUpdated the "Logical volumes" on page 44 topic about logical volume encryption in data volume groups that are configured in PowerHA SystemMirror environment.

December 2021

Updated the "Logical volumes" on page 44 topic about logical volume encryption in data volume groups that are configured in PowerHA SystemMirror environment.

December 2020

Added information about the maximum number of cluster IP address, application monitors, and file system that are supported in a resource groups by PowerHA SystemMirror in the "PowerHA SystemMirror components configuration limits" on page 1 topic.

PowerHA SystemMirror components configuration limits

PowerHA SystemMirror has limits for components and settings. For example, you can have up to 16 nodes in a cluster.

The following table displays the PowerHA SystemMirror components and their corresponding maximum values.

Table 1. PowerHA SystemMirror maximum limits

Component	Maximum limits for the configuration
Nodes in a cluster	16
Resources available in a resource group	512
Backup repository disks in a cluster	6
Resource groups in a cluster	256
Number of networks in a cluster	48
Cluster IP addresses	256
Application servers in a resource group	512
Application monitors in a resource group	512
Service IP labels in a resource group	256
Interfaces per node per network	7
Persistent IP addresses per network per node	1
Volume groups in a resource group	512
File systems in a resource group	512
Cluster name length	63 characters
Node name length	64 characters
Network interface name length	64 characters
Service IP name length	63 characters
Persistent IP name length	64 characters
Resource group name length	64 characters
Application server name length	64 characters
Application monitor name length	64 characters

Notes:

- In PowerHA SystemMirror, you can configure upto 256 cluster IP addresses, which includes boot IP addresses, persistent IP addresses, and service IP addresses.
- You can configure up to 256 service IP labels in a resource group. However, the actual number of service IP labels that you can configure will be less than 256 depending upon number of cluster IP labels which includes boot IP addresses and persistent IP addresses configured to the cluster.

Overview of planning process

You can plan for high availability within a data center with PowerHA SystemMirror Standard Edition for AIX and for multisite high availability and disaster recovery with PowerHA SystemMirror Enterprise Edition for AIX.

Your major goal throughout the planning process is to eliminate single points of failure. A single point of failure exists when a critical cluster function is provided by a single component. If that component fails, the cluster has no other way of providing that function, and the application or service dependent on that component becomes unavailable.

For example, if all the data for a critical application resides on a single disk, and that disk fails, that disk is a single point of failure for the entire cluster. Clients cannot access that application until the data on the disk is restored. Likewise, if dynamic application data is stored on internal disks rather than on external disks, it is not possible to recover an application by having another cluster node take over the disks. Therefore, identifying necessary logical components required by an application, such as file systems and directories (which could contain application data and configuration variables), is an important prerequisite for planning a successful cluster.

Realize that, while your goal is to eliminate all single points of failure, you may have to make some compromises. Usually a cost is associated with eliminating a single point of failure. For example, purchasing an additional hardware device to serve as backup for the primary device increases cost. The cost of eliminating a single point of failure could be compared against the cost of losing services if that component fail. Again, the purpose of the PowerHA SystemMirror is to provide a cost-effective, highly available computing platform that can meet future processing demands.

Note: It is important that failures of cluster components be remedied as soon as possible. Depending on your configuration, PowerHA SystemMirror might not be able to handle a second failure, due to lack of resources.

Planning guidelines

Designing the cluster that provides the best solution for your organization requires careful and thoughtful planning. In fact, adequate planning is the key to building a successful PowerHA SystemMirror cluster. A well-planned cluster is easier to install, provides higher application availability, performs better, and requires less maintenance than a poorly planned cluster.

You might need to plan for additional processes within your environment. For example, patch management and process management processes are critical if you want your environment to handle various types of failures.

For a critical application to be highly available, none of the associated resources should be a single point of failure. As you design a PowerHA SystemMirror cluster, your goal is to identify and address all potential single points of failure. Questions to ask include:

- What application services are required to be highly available? What is the priority of these services?
- What is the cost of a failure compared to the necessary hardware to eliminate the possibility of this failure?
- What is the maximum number of redundant hardware and software components that PowerHA SystemMirror can support?
- What is the required availability of these services? Do they need to be available 24 hours a day, seven days a week, or is eight hours a day, five days a week sufficient?
- What could happen to disrupt the availability of these services?
- What is the allotted time for replacing a failed resource? What is an acceptable degree of performance degradation while operating after a failure?
- Which failures will by automatically detected as cluster events? Which failures need to have custom code written to detect the failure and trigger a cluster event?
- What is the skill level of the group implementing and maintaining the cluster?

To plan, implement, and maintain a successful PowerHA SystemMirror cluster requires continuing communication among many groups within your organization. Ideally, you should assemble the following representatives (as applicable) to aid in PowerHA SystemMirror planning sessions:

- Network administrator
- System administrator
- Database administrator
- Application programming
- Support personnel
- End users

PowerHA SystemMirror supports a variety of configurations, providing you with a great deal of flexibility. For information about designing for the highest level of availability for your cluster, see the IBM white paper *High Availability Cluster Multiprocessing Best Practices*.

Related reference

Eliminating single points of failure: Configuring redundant components supported by PowerHA SystemMirror
The PowerHA SystemMirror software provides numerous options to avoid single points of failure.

Related information

High Availability Cluster Multiprocessing Best Practices

Eliminating single points of failure: Configuring redundant components supported by PowerHA SystemMirror

The PowerHA SystemMirror software provides numerous options to avoid single points of failure.

The following table summarizes potential single points of failure and describes how to eliminate them by configuring redundant hardware and software cluster components.

Cluster components	To eliminate as single point of failure	PowerHA SystemMirror supports
Nodes	Use multiple nodes	Up to 16.
Power sources	Use multiple circuits or uninterruptible power supplies	As many as needed.
Networks	Use multiple networks to connect nodes	Up to 48.
Network interfaces, devices, and labels	Use redundant network adapters	Up to 256.
TCP/IP subsystems	Use networks to connect adjoining nodes and clients	As many as needed.
Disk adapters	Use redundant disk adapters	As many as needed.
Controllers	Use redundant disk controllers	As many as needed.
Disks	Use redundant hardware and disk mirroring, striping, or both	As many as needed.
Applications	Assign a node for application takeover, to configure an application monitor, and to configure clusters with nodes at more than one site.	Flexible configuration policies for high availability within a site and between sites.
Sites	Use more than one site for disaster recovery.	Up to two sites.
Resource groups	Use resource groups to specify how a set of entities should perform.	Up to 64 per cluster.
Cluster resources	Use multiple cluster resources.	Up to 128 for the Clinfo daemon (more can exist in cluster).
Virtual I/O Server (VIOS)	Use redundant VIOS	As many as needed.
HMC	Use redundant HMC	2
Managed System hosting a cluster node	Use separate managed systems for each cluster node	16 nodes

Cluster components	To eliminate as single point of failure	PowerHA SystemMirror supports
Cluster repository disk	Use RAID protection	One active repository disk per site that has the ability to replace the disk after a failure. You must have a spare disk that is available to replace the failed repository disk in the live cluster.

Related reference

Planning guidelines

Designing the cluster that provides the best solution for your organization requires careful and thoughtful planning. In fact, adequate planning is the key to building a successful PowerHA SystemMirror cluster. A well-planned cluster is easier to install, provides higher application availability, performs better, and requires less maintenance than a poorly planned cluster.

Overview of the planning process

This topic describes the steps for planning a PowerHA SystemMirror cluster.

Step 1: Planning for highly available applications

In this step, you plan the core of the cluster: the applications to be made highly available, the types of resources they require, the number of nodes, shared IP addresses, and a mode for sharing disks (nonconcurrent or concurrent access). Your goal is to develop a high-level view of the system that serves as a starting point for the cluster design. After making these initial decisions, start to draw a diagram of the cluster. Initial cluster planning describes this step of the planning process.

Step 2: Planning cluster topology

In this step, you decide on names for the cluster and the nodes. Optionally, you also decide on names for sites and decide which nodes belong to which site. Initial cluster planning describes this step of the planning process.

Step 3: Planning sites

In this step, you decide whether the site will use a stretched cluster or a linked cluster. A stretched cluster contains nodes from sites at the same geographical locations. Stretched clusters must share a repository disk. A linked cluster contains nodes from sites that are located at different geographical locations. Linked clusters use a separate repository disk.

Step 4: Planning cluster network connectivity

In this step, you plan the networks that connect the nodes in your system. You first examine issues relating to TCP/IP and point-to-point networks in a PowerHA SystemMirror environment. Planning cluster network connectivity describes this step of the planning process.

Step 5: Planning shared disk devices

In this step, you plan the shared disk devices for the cluster. You decide which disk storage technologies you will use in your cluster, and examine issues relating to those technologies in the PowerHA SystemMirror environment. Planning shared disk and tape devices describes this step of the planning process.

Step 6: Planning shared LVM components

In this step, you plan the shared volume groups for the cluster. You first examine issues relating to LVM components in a PowerHA SystemMirror environment. Planning shared LVM components describes this step of the planning process.

Step 7: Planning resource groups

Planning resource groups incorporates all of the information you have generated in the previous steps. In addition, you need to decide whether to use dependent resource groups or particular runtime policies for keeping certain related resource groups on the same node or on different nodes. Planning resource groups describes this step of the planning process.

Step 8: Planning cluster event processing

In this step, you plan the event processing for your cluster. Planning for cluster events describes this step of the planning process.

Step 9: Planning PowerHA SystemMirror clients

In this step, you examine issues relating to PowerHA SystemMirror clients. Planning for PowerHA SystemMirror clients describes this step of the planning process.

Related reference

Initial cluster planning
This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

Planning cluster network connectivity
This section describe planning the network support for a PowerHA SystemMirror cluster.

Planning resource groups
These topics describe how to plan resource groups within a PowerHA SystemMirror cluster.

Planning shared LVM components
This section describe planning shared volume groups for a PowerHA SystemMirror cluster.

Planning shared disk and tape devices
This section discusses information to consider before configuring shared external disks in a PowerHA SystemMirror cluster and provides information about planning and configuring tape drives as cluster resources.

Planning for cluster events
These topics describe the PowerHA SystemMirror cluster events.

Planning for PowerHA SystemMirror clients
These topics discuss planning considerations for PowerHA SystemMirror clients. This is the last step before proceeding to the installation of your PowerHA SystemMirror software.

Planning cluster sites
PowerHA SystemMirror clusters can be used within a single site or multiple sites for disaster recovery.

Related information

Resource group behavior during cluster events

Initial cluster planning

This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

Before you start PowerHA SystemMirror planning, make sure that you understand the concepts and terminology relevant to PowerHA SystemMirror.

A PowerHA SystemMirror cluster provides a highly available environment for mission-critical applications. In many organizations, these applications must remain available at all times. For example, a PowerHA SystemMirror cluster could run a database server program that services client applications, keeping it highly available for clients that send queries to the server program.

Related information

PowerHA SystemMirror concepts

Planning cluster nodes

For each critical application, be mindful of the resources required by the application, including its processing and data storage requirements.

For example, when you plan the size of your cluster, include enough nodes to handle the processing requirements of your application after a node fails.

Keep in mind the following considerations when determining the number of cluster nodes:

- A PowerHA SystemMirror cluster can be made up of any combination of IBM Power Systems servers. Ensure that all cluster nodes do not share components that could be a single point of failure (for example, a power supply). Similarly, do not place nodes on a single rack.
- Create small clusters that consist of nodes that perform similar functions or share resources. Smaller, simple clusters are easier to design, implement, and maintain.
- For performance reasons, it may be desirable to use multiple nodes to support the same application. To provide mutual takeover services, the application must be designed in a manner that allows multiple instances of the application to run on the same node.

  For example, if an application requires that the dynamic data resides in a directory called **/data**, the application probably cannot support multiple instances on the same processor. For such an application (running in a nonconcurrent environment), try to partition the data so that multiple instances of the application can run, each accessing a unique database.

  Furthermore, if the application supports configuration files that enable the administrator to specify that the dynamic data for *instance1* of the application resides in the *data1* directory, *instance2* resides in the *data2* directory, and so on, then multiple instances of the application are probably supported.
- Certain configurations, including additional nodes in the cluster design, can increase the level of availability provided by the cluster. Certain configurations also give you more flexibility in planning node fallover and reintegration.

  The most reliable cluster node configuration is to have at least one standby node.
- Choose cluster nodes that have enough I/O slots to support redundant network interface cards and disk adapters.

  Remember, the cluster composed of multiple nodes is still more expensive than a single node, but without planning to support redundant hardware, (such as enough I/O slots for network and disk adapters), the cluster will have no better availability.
- Use nodes with similar processing speed.
- Use nodes with the sufficient CPU cycles and I/O bandwidth to allow the production application to run at peak load. Remember, nodes should have enough capacity to allow PowerHA SystemMirror to operate.

  To plan for this, benchmark or model your production application, and list the parameters of the heaviest expected loads. Then choose nodes for a PowerHA SystemMirror cluster that will not exceed 85% busy when running your production application.

  When you create a cluster, you assign a name to it. PowerHA SystemMirror associates this name with the PowerHA SystemMirror-assigned cluster ID.

Planning for repository disk and cluster multicast IP address

PowerHA SystemMirror clustering can be done through multicast or unicast networking. To choose the multicast mode of clustering, you can plan and provide a multicast IP address for communication within the cluster. By default, if no multicast IP address is provided while deploying the cluster, PowerHA deploys unicast (normal TCP/IP socket communication) based cluster.

Cluster repository disk

You must have one active repository disk per cluster for standard clusters and stretched clusters. You can identify up to six backup repository disks per cluster for standard clusters and stretched clusters. You must have one active repository disk per site for linked clusters. You can identify up to six backup repository disks per site for linked clusters.

PowerHA SystemMirror uses a shared disk to store Cluster Aware AIX (CAA) cluster configuration information. You must have at least 512 MB and no more than 460 GB of disk space allocated for the cluster repository disk. This configuration is automatically kept highly available on the disk that is provided. This feature requires that a dedicated shared disk be available to all nodes that are part of the cluster. This disk cannot be used for application storage or any other purpose.

When planning the disks that you want to use as repository disks, you must plan for a backup or replacement disks, which can be used in case the primary repository disk fails. The backup disk must be the same size and type as the primary disk, but could be in a different physical storage disk. Update your administrative procedures and documentation with the backup disk information. You can also replace a working repository disk with a new one to increase the size or to change to a different storage subsystem. To replace a repository disk, you can use the SMIT interface.

Note: If the shared disk that is used as a repository disk is a mapped virtual SCSI (vSCSI) disk, you must map the disk as an vSCSI disk to all nodes in the cluster. The mapping of the vSCSI disk must be identical across all nodes in the cluster. For example, you cannot map the repository disk using the vSCSI method to one node in the cluster and map the same disk using the N-Port ID Virtualization (NPIV) method to another node in the cluster.

Cluster multicast IP address

You can use a multicast IP address for cluster monitoring and communication. You can specify this address when you create the cluster, or you can have one be generated automatically when you synchronize the initial cluster configuration.

Note: The default mechanism uses unicast communications and requires no extra configuration. However, if you want to use multicast communication, you must continue reading and ensure that your network devices are enabled for multicast communications.

If you decide to use multicast, PowerHA SystemMirror uses multicast-based communication between hosts in the cluster. Your environment's network must allow multicast IP packets to flow between hosts in the cluster. To verify whether nodes in your environment support multicast based communication, use the **`mping`** command. Run the **`mping`** command before you start using PowerHA SystemMirror in your environment.

Note: Some of the network switches allow multicast packets to flow for a while before stopping them. So, it is critical to conduct the **`mping`** test for at least 5 minutes and to make sure that the network fabric allows multicast packet flow without any issues. Also, when switches are cascaded, typically the switches need additional configuration to route multicast packets. To configure multicast packet flow, see the documentation that is provided by the switch vendor to configure multicast packet flow.

A multicast address is also known as a class D address. Every IP datagram whose destination address starts with 1110 is an IP multicast datagram. The remaining 28 bits identify the multicast group on which the datagram is sent. You must configure your kernel to receive packets sent to a specific multicast group, which makes the host join the group in the interface you specified.

Do not use the following multicast groups:

224.0.0.1
This is the all-hosts group. If you ping that group, all multicast-capable hosts on the network should answer, because every multicast-capable host must join that group at startup on all its multicast-capable interfaces.

224.0.0.2
This the all-routers group. All multicast routers must join that group on all its multicast capable interfaces.

224.0.0.4
This is the all DVMRP routers.

224.0.0.5
This is all OSPF routers.

224.0.013
This is all PIM routers.

Note: The range 224.0.0.0-224.0.0.255 is reserved for local purposes, such as administrative and maintenance tasks, and data that they receive is never forwarded by multicast routers. Similarly, the range 239.0.0.0 - 239.255.255.255 is reserved for administrative scoping. These special multicast groups are regularly published in the Assigned Numbers RFC.

PowerHA SystemMirror 7.1.2, or later, supports IP version 6 (IPv6), however, you cannot explicitly specify the IPv6 multicast address. CAA uses an IPv6 multicast address which is derived from the IP version 4 (IPv4) multicast address. To determine the IPv6 multicast address, a standard prefix of *oxFF05* is combined using the logical OR operator with the hexadecimal equivalent of the IPv4 address. For example, the IPv4 multicast address is *228.8.16.129* or *0xE4081081*. The transformation by the logical OR operation with the standard prefix is *0xFF05:: | 0xE4081081*. Thus, the resulting IPv6 multicast address is *0xFF05::E408:1081*.

Related information

Repository disk failure

Replacing a repository disk with SMIT

Troubleshooting multicast

Testing multicast in a network

Planning for disk fencing

Disk fencing is one of the features for quarantine policies that is available in PowerHA SystemMirror.

Disk fencing prerequisites

To use disk fencing in PowerHA SystemMirror, your disks must meet the following requirements:

- All disks that are managed by the storage systems must be enabled for SCSI-3 Persistent Reservation (PR). Some storage systems do not enable SCSI-3 PR capabilities by default.
- All disks must not be in use when disk fencing is enabled (the volume groups must be offline).
- The disks must not be reserved before you start PowerHA SystemMirror. You can use the storage systems software to release any disk reservations.

Critical resource group settings

To use the disk fencing option, you must identify the critical resource group and the storage subsystem must support SCSI-3 PR and ODM reserve_policy of PR_shared. This policy is applied to all disks in the volume groups that are configured for all resource groups.

Reservation type

The disk must have a persistent reservation type of Write Exclusive All Registrant (WEAR). You can run the following **clmgr** commands to check if your disk supports SCSI-3 capability (WEAR - type 7h) to support PowerHA SystemMirror disk fencing:

```
clmgr -a SCSIPR_CAPABLE query physical_volume <disk>
clmgr -a SCSIPR_CAPABLE query volume_group <vg>
```

where *disk* is the name of the disk and *vg* is the name of the volume group.

Note: If the command does not return **Yes**, you cannot use the disk for disk fencing. You must choose some other disk or quarantine policy.

reserve_policy

Defines whether a reservation method is running on the disk. The reserve_policy of PR_shared is the required policy that applies the shared host method for the disk. To view the attributes for the disk, run the **`lsattr -Rl <diskname> -a reserve_policy`** command.

To use the disk fencing feature, you must specify a critical resource group. The critical resource group that you specify must meet the following criteria:

- The critical resource group cannot be added as a child in any parent_child, start_after, or stop_after, dependency relationship.
- The critical resource group must have all nodes in the cluster as participating nodes.
- The critical resource group must have the startup policy as `Online On First Available Node`.

The reserve_policy for a disk of a volume group that has cluster services that are running and disk fencing is enabled, must use the following settings:

Table 2. Disk settings

Option	**Value**
Configured Reserve Policy	PR_shared
Effective Reserve Policy	PR_shared
Reservation Status	SCSI PR reservation (Write Exclusive All Registrant)

EMC storage and SCSI-3 PR

EMC disks do not support SCSI-3 PR capabilities by default. If you attempt to configure disk fencing on EMC storage without enabling SCSI-3 PR, an error occurs.

The following is an example of steps to enable SCSI-3 PR on some EMC storage devices:

Note: Before you complete the following steps, each disk must not be in use on any nodes in the cluster and the volume groups must not be in a varyon state. The following commands are part of the EMC software packages that are installed on an AIX LPAR.

1. Identify the IDs for the storage devices and disks in the EMC storage subsystem by running the following command:

   ```
   powermt display dev=hdiskpowerX
   Pseudo name=hdiskpowerX
   Symmetrix ID=000194900568
   Logical device ID=0036
   Device WWN=6000097000019490056853303030xxxx
   state=alive; policy=SymmOpt; queued-IOs=0
   ```

2. Enable the SCSI-3 PR capabilities in the EMC storage subsystem by running the following command:

   ```
   symconfigure -sid 000194900568 -cmd "set device 0036 attribute=SCSI3_persist_reserv;" commit -v -noprompt
   ```

3. Complete the following steps to rediscover the storage disks in the AIX operating system:
   a. Remove the storage devices or disks by running the `rmdev -R -d -l hdiskpowerX` command, where `hdiskpowerX` is the name of the disk.
   b. Run the **`cfgmgr`** command to discover the storage devices or disks.
4. Verify that the SCSI-3 PR capabilities are enabled for the storage devices or disks by running the following command:

   ```
   /usr/symcli/bin/symdev -sid 000194900568 show 0036 | grep SCSI
       SCSI-3 Persistent Reserve: Enabled
   ```

For more information about configuring EMC for SCSI-3 PR, see the EMC Support website.

Hitachi storage and SCSI-3 PR

Hitachi disks do not support SCSI-3 PR capabilities by default. You must manually enable SCSI-3 PR capabilities for each disk that is part of a volume group that is managed by PowerHA SystemMirror. To enable the SCSI-3 PR capabilities, in the Hitachi Storage Navigator you must enable the Host Mode Options (HMHO) **2** and **72**.

Related information

Configuring a quarantine policy

Troubleshooting disk fencing

Planning cluster sites

PowerHA SystemMirror clusters can be used within a single site or multiple sites for disaster recovery.

If you have multiple sites, you can use the following PowerHA SystemMirror Enterprise Edition features for disaster recovery:

- PowerHA SystemMirror Enterprise Edition for AIX includes options for supporting replication technologies that are available from various storage subsystems.
- PowerHA SystemMirror Enterprise Edition for AIX for GLVM provides host-based replication over TCP/IP networks.

PowerHA SystemMirror 7.1.2, or later, supports different types of definitions for sites and site-specific policies for high availability and disaster recovery (HADR). You can define multiple sites in both PowerHA SystemMirror Standard Edition for AIX and PowerHA SystemMirror Enterprise Edition for AIX.

PowerHA SystemMirror uses Cluster Aware AIX (CAA) for cluster communication and cluster health management.

You can use PowerHA SystemMirror management interfaces to create the following multiple-site solutions:

Stretch clusters
Contains nodes from sites that are located at the same geographical locations. Stretched clusters must share a repository disk across all nodes in the site. Stretched clusters do not support HADR with Storage Replication Management. To use stretch clusters, your network environment must support multicast-based communication.

Linked clusters
Contains nodes from sites that are located at different geographical locations. Linked clusters use a separate repository disk. Linked clusters support cross-site LVM mirroring and HyperSwap®. In linked clusters, CAA uses unicast packets to communicate and manage the sites between the independent CAA clusters.

Planning resources and site policy

PowerHA SystemMirror tries to ensure that the primary instance of a resource group is maintained online at one site, and the secondary instance is maintained online at the other site. Plan which nodes to configure at which site, and where you want the active applications to run, so you can plan the resource group policies accordingly.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

All resources defined to PowerHA SystemMirror must have unique names. The service IP labels, volume groups, and resource group names must be both unique within the cluster and distinct from each other. The name of a resource should relate to the application it serves, as well as to any corresponding device. For example, a service address for a resource group that runs a WebSphere® instance could be named websphere_service_address.

Planning cluster security

PowerHA SystemMirror provides cluster security by controlling user access to PowerHA SystemMirror and providing security for internode communications.

Connection authentication

PowerHA SystemMirror provides connection authentication to protect PowerHA SystemMirror communications between cluster nodes. This connection authentication is also known as standard authentication. Standard authentication includes verified connections by IP address and host name, and limits the commands that can be run with the root privilege. This mode uses the principle of least-privilege for remote command execution, which ensures that no arbitrary command can be run on a remote node with root privilege. A select set of PowerHA SystemMirror commands is considered trusted and is allowed to run as root. All other commands run as user `nobody`. The **/.rhosts** dependency for internode communication is eliminated.

You can also configure a virtual private network (VPN) for internode communications. If you use a VPN, use persistent IP labels for VPN tunnels.

Security Configuration

PowerHA SystemMirror uses the Cluster Aware AIX (CAA) function to create secure communication path for heartbeats and synchronization between nodes in a cluster.

You can use the following CAA methods to create cluster security credentials for nodes in the cluster.

Self signed
PowerHA SystemMirror generates the security credentials.

Security certificate and private key pairs
PowerHA SystemMirror uses existing security certificate and private key pairs that you provide.

Secure Shell (SSH)
PowerHA SystemMirror uses the keys already configured for SSH communication in your environment.

Message authentication and encryption

PowerHA SystemMirror provides security for PowerHA SystemMirror messages sent between cluster nodes as follows:

- Message authentication ensures the origination and integrity of a message.
- Message encryption changes the appearance of the data while it is transmitted and returns it to its original form when received by a node that authenticates the message.
- Messages are either encrypted or hashed depending on a security level of Low, Medium, or High. A Low security level hashes only a few of the messages as compared to a High security level where messages are encrypted.

PowerHA SystemMirror supports the following types of encryption keys for message authentication and encryption:

- Message Digest 5 (MD5) with Data Encryption Standard (DES)
- MD5 with Triple DES
- MD5 with Advanced Encryption Standard (AES).

Select an encryption algorithm that is compatible with the security methodology used by your organization.

PATH variable in /etc/environment file

When the PATH variable is initialized for the cluster executable, the default PATH in the `/etc/environment` file is scanned before adding paths to the cluster executables. If any of the following items are found in the default PATH, these items are skipped and not included in the resulting PATH that is used for the cluster executables:

Note: You should remove any of the following items from the default PATH.

- The `stat()` subroutine fails to return information about the directory
- A directory is world writable

Application planning

Before you start planning for an application, be sure you understand the data resources for your application and the location of these resources within the cluster in order to provide a solution that enables them to be handled correctly if a node fails.

To prevent a failure, you must thoroughly understand how the application behaves in a single-node and multinode environment. Do not make assumptions about the application's performance under adverse conditions.

Use nodes with the sufficient CPU cycles and I/O bandwidth to allow the production application to run at peak load. Remember, nodes should have enough capacity to allow PowerHA SystemMirror to operate.

To plan for this, benchmark or model your production application, and list the parameters of the heaviest expected loads. Then choose nodes for a PowerHA SystemMirror cluster that will not exceed 85% busy, when running your production application.

You can configure multiple application monitors for an application and direct PowerHA SystemMirror to both:

- Monitor the termination of a process or more subtle problems affecting an application
- Automatically attempt to restart the application and take appropriate action (notification or fallover) if restart attempts fail.

This section explains how to record all the key information about your application and begin drawing your cluster diagram.

Keep in mind the following guidelines to ensure that your applications are serviced correctly within a PowerHA SystemMirror cluster environment:

- Lay out the application and its data so that only the data resides on shared external disks. This arrangement not only prevents software license violations, but it also simplifies failure recovery.
- If you are planning to include multitiered applications in parent-child dependent resource groups in your cluster, see the section Planning considerations for multitiered applications. If you are planning to use location dependencies to keep certain applications on the same node, or on different nodes, see the section Resource group dependencies.
- Write robust scripts to start and stop the application on the cluster nodes. The startup script especially must be able to recover the application from an abnormal end, such as a power failure. Ensure that it runs properly in a single-node environment before including the PowerHA SystemMirror software.
- Confirm application licensing requirements. Some vendors require a unique license for each processor that runs an application, which means that you must protect the license for the application by incorporating processor-specific information into the application when it is installed. As a result, even though the PowerHA SystemMirror software processes a node failure correctly, it might be unable to restart the application on the fallover node because of a restriction on the number of licenses for that application available within the cluster. To avoid this problem, be sure that you have a license for each system unit in the cluster that might potentially run an application.
- Ensure that the application runs successfully in a single-node environment. Debugging an application in a cluster is more difficult than debugging it on a single processor.
- Verify that the application uses a proprietary locking mechanism if you need concurrent access.

Related reference

Planning considerations for multitiered applications

Business configurations that use multitiered applications can use parent and child dependent resource groups. For example, the database must be online before the application controller. In this case, if the database goes down and is moved to a different node the resource group containing the application controller would have to be brought down and back up on any node in the cluster.

Resource group dependencies
PowerHA SystemMirror offers a wide variety of configurations where you can specify the relationships between resource groups that you want to maintain at startup, fallover, and fallback.

Planning for capacity on demand

Capacity on Demand (CoD) is one of the facilities of Resource Optimized High Availability (ROHA) function that PowerHA SystemMirror uses to dynamically manage the hardware resources for applications. With CoD, you can activate preinstalled processors that are inactive and not paid for as resource requirements change.

CoD resources are composed of On/Off CoD resources and Enterprise Pool CoD (EPCoD) resources. Both of these resources can dynamically deliver supplementary resources to your environment that are used through normal DLPAR management (allocation or release of resources to an LPAR).

The additional processors and memory, while physically present, are not used until PowerHA SystemMirror decides that the additional capacity that is required is worth the cost. You can use the ROHA function to quickly and easily acquire extra resources to meet peek or unexpected workloads in your environment.

PowerHA SystemMirror integrates with the DLPAR, On/Off CoD, and EPCoD functions. The collection of these functions is called ROHA. The active node is hosted by an LPAR on a frame with sufficient permanent resources. The standby node is hosted by an LPAR on a frame with minimal permanent resources and relies on ROHA to dynamically add extra resources.

When it is necessary to run the application on the standby node, PowerHA SystemMirror use the ROHA function. The ROHA function verifies that the node has sufficient resources to successfully run the application and allocates the necessary resources. The resources can be allocated from the following sources:

On/Off CoD provisioned resources
If the CEC free pool has insufficient resources that can be allocated through the DLPAR to the node, the On/Off CoD function provides more resources to the CEC. These additional resources are added to the CEC free pool and can be used through an LPAR operation. If the application requires more memory or processor, PowerHA SystemMirror, through the ROHA function, can automatically allocate these resources to the standby node.

EPCoD provisioned resources
If the CEC free pool has insufficient resources that can be allocated through the DLPAR to the node, the EPCoD function provides more resources to the CEC. These additional resources are added to the CEC free pool and can be used through a DLPAR operation. If the application requires more memory or processor, PowerHA SystemMirror, through the ROHA function, can automatically allocate these resources to the standby node.

The CEC free pool
The DLPAR function provides the resources to the standby node by allocating the resources, which are available in the free pool. These resources are provided from On/Off CoD pool or from the EPCoD pool.

When you configure PowerHA SystemMirror to use resources through the ROHA function, the LPAR nodes in the cluster do not use more resources until the resources are required by the application.

PowerHA SystemMirror does not activate any CoD resources until the free pool is exhausted. After the free pool is exhausted, more hardware resources are activated by PowerHA SystemMirror. CoD hardware resources are activated and allocated to the LPAR dynamically until the requirements of the application are met. When an application no longer requires the allocated hardware resources, they are released to the free pool, at which point PowerHA SystemMirror deactivates the hardware resources. These hardware resources return to the pool from which they originated (On/Off CoD pool or EPCoD pool).

Related reference

Monitoring in PowerHA SystemMirror
The primary task of PowerHA SystemMirror is to recognize and respond to failures. PowerHA SystemMirror uses the Cluster Aware AIX infrastructure to monitor the activity of its network interfaces, devices, and IP labels.

Related information

Administering PowerHA SystemMirror

Application controllers

To put the application under PowerHA SystemMirror control, you create an application controller resource that associates a user-defined name with the names of specially written scripts to start and stop the application.

By defining an application controller, PowerHA SystemMirror can start another instance of the application on the takeover node when a fallover occurs. This protects your application so that it does not become a single point of failure.

After you define the application controller, you can add it to a resource group. A resource group is a set of resources that you define so that the PowerHA SystemMirror software can treat them as a single unit.

Related reference

Planning resource groups
These topics describe how to plan resource groups within a PowerHA SystemMirror cluster.

Applications integrated with PowerHA SystemMirror

Certain applications, including Workload Manager, can be configured directly as highly available resources, without application controllers or additional scripts. In addition, PowerHA SystemMirror cluster verification ensures the correctness and consistency of certain aspects of your Workload Manager configuration.

PowerHA SystemMirror offers the following PowerHA SystemMirror Smart Assist applications to help you integrate the application into a PowerHA SystemMirror cluster:

Smart Assist for WebSphere
Extends an existing PowerHA SystemMirror configuration to include monitoring and recovery support for various WebSphere components.

Smart Assist for DB2®
Extends an existing PowerHA SystemMirror configuration to include monitoring and recovery support for DB2 Universal Database (UDB) Enterprise Server Edition.

Smart Assist for Oracle
Provides assistance to those involved with the installation of Oracle Application Server 10g (9.0.4) (AS10g) Cold Failover Cluster (CFC) solution on an IBM AIX operating system.

Smart Assist for FileNet® P8
Offers you enterprise-level scalability and flexibility to handle the most demanding content challenges, the most complex business processes, and integration with existing systems in your environment.

Smart Assist for SAP MaxDB
Setup MaxDB and liveCache database instances for high availability.

Smart Assist for Lotus® Domino® Server
Automatically configures PowerHA SystemMirror for environment that already have Lotus Domino configured.

Smart Assist for Tivoli® Storage Manager
Uses the three different areas of Tivoli Storage Manager, server, client, and admin center to provide a highly available solution for your environment.

Smart Assist for SAP
Sets up SAP Netweaver 2004s for high availability by protecting its single point of failures

Smart Assist for Tivoli Directory Server
Automatically configure PowerHA SystemMirror where the Tivoli Directory Server is already installed.

Smart Assist for SAP liveCache Hot Standby
Provides management interfaces that assist you in configuring a PowerHA SystemMirror policy and deploying start methods, stop methods, and monitor methods for the workload stacks in your environment.

Smart Assist for Websphere MQSeries®
Enables programs to communicate with each other across a network of components that are not similar, such as processors, subsystems, operating systems, and communication protocols.

Related information

Smart Assist applications for PowerHA SystemMirror

Application monitoring

PowerHA SystemMirror can monitor applications that are defined to application controllers.

PowerHA SystemMirror monitors applications in one of two ways:

- *Process monitoring* detects the termination of a process, using Reliable Scalable Cluster Technology (RSCT) and Resource Monitoring and Control (RMC) capability.
- *Custom monitoring* monitors the health of an application, using a monitor method that you define.

You can configure multiple application monitors and associate them with one or more application controllers. You can assign each monitor a unique name in SMIT. By supporting multiple monitors per application, PowerHA SystemMirror can support more complex configurations. For example, you can configure one monitor for each instance of an Oracle parallel server in use. Otherwise, you can configure a custom monitor to check the health of the database along with a process stop monitor to instantly detect the end of the database process.

You can use the Application Availability Analysis tool to measure the exact amount of time that any of your PowerHA SystemMirror-defined applications is available. The PowerHA SystemMirror software collects, time stamps, and logs the following information:

- An application monitor is defined, changed, or removed
- An application starts, stops, or fails
- A node fails or is shut down, or startup
- A resource group is taken offline or moved
- Application monitoring via multiple monitors is suspended or resumed.

Related information

Configuring PowerHA SystemMirror cluster topology and resources (extended)

Monitoring a PowerHA SystemMirror cluster

Planning considerations for multitiered applications

Business configurations that use multitiered applications can use parent and child dependent resource groups. For example, the database must be online before the application controller. In this case, if the database goes down and is moved to a different node the resource group containing the application controller would have to be brought down and back up on any node in the cluster.

Environments such as service access points (SAP) require applications to be cycled (stopped and then started again) whenever a database fails. Many application services are provided by an environment like SAP, and the individual application components often need to be controlled in a specific order.

Establishing interdependencies between resource groups is also useful when system services are required to support application environments. Services such as **cron** jobs for pruning log files or for initiating backups need to move from one node to another along with an application, but typically are not initiated until the application is established. These services can be built into application controller start and stop scripts, or they can be controlled through pre-event and post-event processing. However, dependent resource groups simplify the way you configure system services to be dependent upon applications they serve.

Note: To minimize the chance of data loss during the application stop and restart process, customize your application controller scripts to ensure that any uncommitted data is stored to a shared disk temporarily during the application stop process and read back to the application during the application restart process. It is important to use a shared disk because the application might be restarted on a node other than the one on which it was stopped.

You can also configure resource groups with location dependencies so that certain resource groups are kept online on the same node, or on different nodes at startup, fallover, and fallback.

Related reference

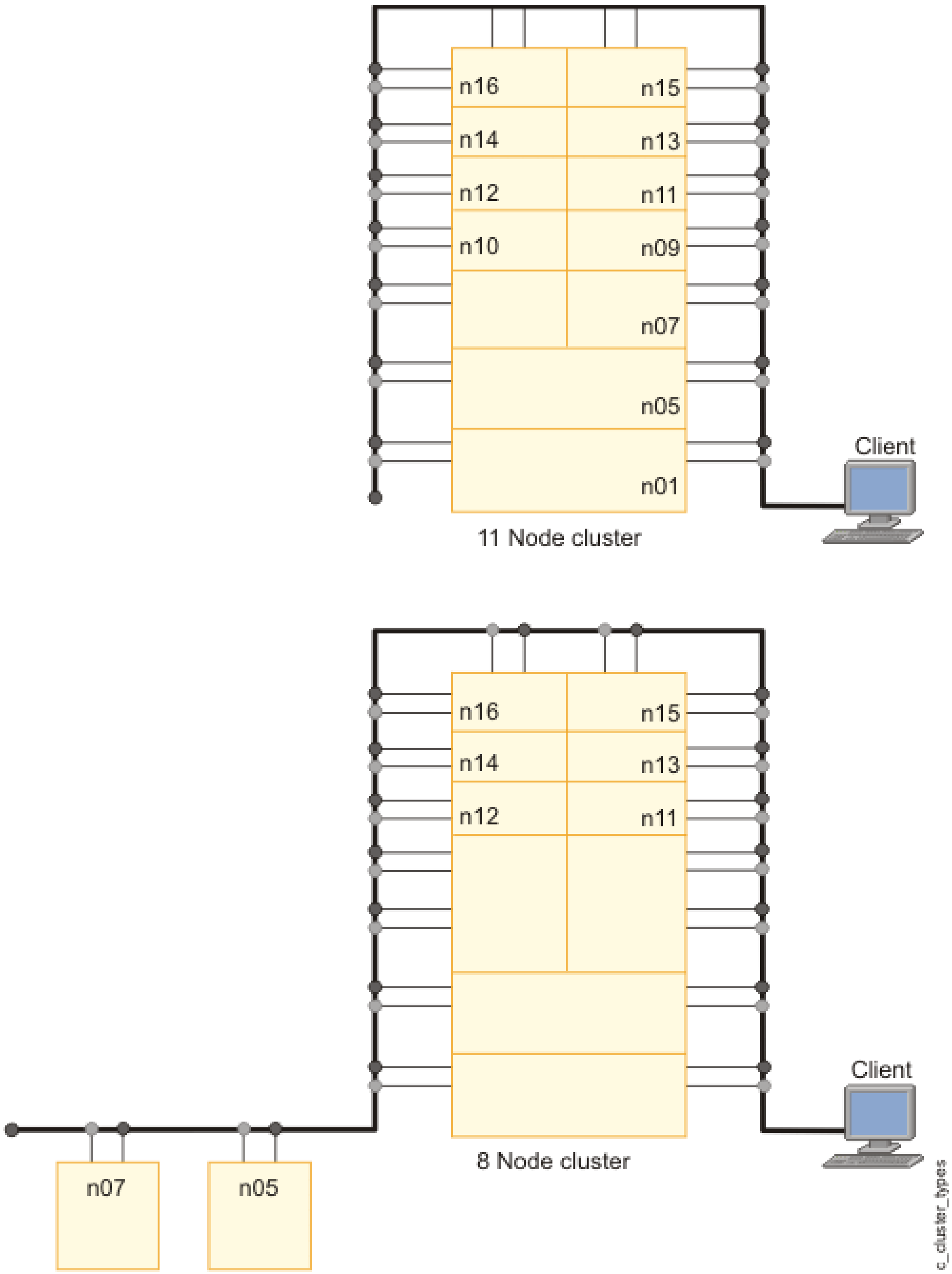
Planning resource groups
These topics describe how to plan resource groups within a PowerHA SystemMirror cluster.

Drawing a cluster diagram

The cluster diagram combines the information from each step in the planning process into one drawing that shows the cluster's function and structure.

The following illustration shows a mixed cluster that includes a rack-mounted system and standalone systems. The diagram uses rectangular boxes to represent the slots supported by the nodes. If your cluster uses thin nodes, darken the outline of the nodes and include two nodes to a drawer. For wide nodes, use the entire drawer. For high nodes, use the equivalent of two wide nodes. Keep in mind that each thin node contains an integrated Ethernet connection.

Begin drawing this diagram by identifying the cluster name and the applications that are being made highly available. Next, darken the outline of the nodes that will make up the cluster. Include the name of each node.



Host name requirements

Starting from PowerHA SystemMirror 7.1.1 there are certain requirements for which interface can be the host name, due to the new Cluster Aware AIX (CAA) layer requirements.

Keep in mind the following requirements when you select the host name:

- The host name cannot be an alias in the **/etc/hosts** file.
- The name resolution for the host name must work both ways. Therefore, only a limited set of characters can be used.
- The IP address that belongs to the host name must be reachable on the server, even when PowerHA is in the **DOWN** state.
- The host name cannot be a service address.

- The host name cannot be an address that is located on a network, which is defined as *private* in PowerHA.
- The host name, the CAA node name, and the **COMMUNICATION_PATH** (that is, the communication path to the node) must be the same.
- By default, the PowerHA node name, the CAA node name, and the **COMMUNICATION_PATH** (that is, the communication path to the node) are set to be the same.
- The host name and the PowerHA node name can be different.
- To change or modify the host name, refer to the following requirements:
  - For PowerHA SystemMirror 7.1.2, or earlier, see Changing the host name for a cluster node in PowerHA SystemMirror 7.1.2, or earlier.
  - For PowerHA SystemMirror 7.1.3, or later, see Changing how PowerHA SystemMirror 7.1.3, or later, responds to host name change.

Note: These requirements leave the base addresses and the persistent address as candidates for the host name. You can use the persistent address as the host name only if you set up the persistent alias manually before you configure the cluster topology.

Planning cluster network connectivity

This section describe planning the network support for a PowerHA SystemMirror cluster.

Prerequisites

In the Initial cluster planning topic, you began planning your cluster, identifying the number of nodes and the key applications you want to make highly available. You started drawing the cluster diagram. This diagram is the starting point for the planning you will do in this section.

Also, by now you should have decided whether or not you will use IP address takeover (IPAT) to maintain specific service IP addresses.

Overview

Your primary goal is to use redundancy to design a cluster topology that eliminates network components as potential single points of failure.

The following table lists these network components with solutions:

Cluster network object	Eliminated as a single point of failure by...
Network	Using multiple networks to connect nodes
Network Interface Card (NIC)	Using redundant NICs on each network

In this section, you complete the following planning tasks:

- Designing the cluster network topology, that is, the combination of IP networks that connect your cluster nodes and the number of connections each node has to each network.
- Adding networking to your cluster diagram.

Related reference

Initial cluster planning
This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

Monitoring in PowerHA SystemMirror

The primary task of PowerHA SystemMirror is to recognize and respond to failures. PowerHA SystemMirror uses the Cluster Aware AIX infrastructure to monitor the activity of its network interfaces, devices, and IP labels.

General network considerations for PowerHA SystemMirror

PowerHA SystemMirror allows internode communication with Ethernet networks.

IP aliases

An IP alias is an IP label or address that is configured onto a network interface controller (NIC) in addition to the normally configured IP label or address on the NIC. The use of IP aliases is an AIX function that PowerHA SystemMirror supports. AIX supports multiple IP aliases on a NIC. Each IP alias on a NIC can be on a separate subnet. AIX also allows IP aliases with different subnet masks to be configured for an interface. PowerHA SystemMirror does not yet support this function.

IP aliases are used in PowerHA SystemMirror as service addresses for IP address takeover.

Network connections

PowerHA SystemMirror requires that each node in the cluster have at least one direct, nonrouted network connection with every other node. The software uses these network connections to pass heartbeat messages among the cluster nodes to determine the state of all cluster nodes, networks, and network interfaces.

PowerHA SystemMirror requires all of the communication interfaces for a given cluster network to be defined on the same physical network and route packets to each other. By default, PowerHA SystemMirror uses unicast communications for heartbeat. If you choose to use multicast heartbeat instead, you need to ensure that your network supports multicast. The communication interfaces must also be able to receive responses from each other without interference by any network equipment. PowerHA SystemMirror also requires that all nodes within a site must have at least one direct network connection with every other node in the same site.

Unicast communications are always used between sites in a linked cluster. Within a site, you can select unicast (the default) or multicast communications.

Between cluster nodes, place only intelligent switches, routers, or other network equipment that transparently passes through multicast and other packets to all cluster nodes. This requirement includes equipment that optimizes protocols.

If such equipment is placed in the paths between cluster nodes and clients, you might need to configure a ping client list in the **clinfo.rc** file to help inform clients of IP address movements. The specific network topology might require other solutions to ensure clients can continue to access the server after IP address takeover.

Bridges, hubs, and other passive devices that do not modify the packet flow can be safely placed between cluster nodes, and between nodes and clients.

Related information

Programming client applications for the Clinfo API

IP labels

In a environment that is not using PowerHA SystemMirror, a host name typically identifies a system, with the host name also being the IP label of one of the network interfaces in the system. Thus, a system can be reached by using its host name as the IP label for a connection.

Host name resolution

In PowerHA SystemMirror, all node host names must be resolved locally with the **/etc/hosts** file. When defining nodes to the cluster, you must specify an IP address or label that resolves locally to the host name, and after you have synchronized the initial cluster configuration, the host name of the node might not be changed.

IP labels in TCP/IP networks

For TCP/IP networks, an IP label and its associated IP address must appear in the **/etc/hosts** file.

The name of the service IP label or address must be unique within the cluster and distinct from the volume group and resource group names. It should relate to the application it serves, as well as to any corresponding device, such as websphere_service_address.

When you assign a service IP label to an interface, use a naming convention that helps identify the interface's role in the cluster. The related entries in the **/etc/hosts** file would be similar to the following:

```
100.100.50.1        net1_en0
100.100.60.1        net2_en1
```

You configure the network interface controller (NIC) by following the instructions in the relevant AIX documentation. AIX assigns an interface name to the NIC when it is configured. The interface name is made up of 2 or 3 characters that indicate the type of NIC, followed by a number that AIX assigns in sequence for each adapter of a certain type. For example, AIX assigns an interface name such as **en0** for the first Ethernet NIC it configures, **en1** for the second, and so on.

Related information

Configuring cluster events

Cluster partitioning

Partitioning, also called node isolation, occurs when a network or network interface controller (NIC) failure isolates cluster nodes from each other.

When a PowerHA SystemMirror node stops receiving network traffic from another node, it assumes that the other node has failed. Depending on your PowerHA SystemMirror configuration, the node might begin acquiring disks from the failed node and making applications and IP labels available. If the failed node is actually still up, data corruption might occur when the disks are taken from it. If the network becomes available again, PowerHA SystemMirror stops one of the nodes to prevent further disk contention and duplicate IP addresses on the network.

PowerHA SystemMirror heartbeat mechanism relies on the IP subsystem and the network infrastructure. Therefore, if the network is congested or a node is congested, the IP subsystem can silently discard the heartbeats. Attempts are made to adjust monitoring characteristics to take network congestion into account and prevent cluster partitioning.

Related reference

Monitoring clusters
The Cluster Aware AIX infrastructure monitors all available and supported network and storage interfaces. The cluster managers on cluster nodes also send messages to each other through the connections between these interfaces.

Example: General network connection

An example of correct PowerHA SystemMirror networking consists of two separate Ethernet networks, each with two network interfaces on each node.

Two routers connect the networks, and they route packets between the cluster and clients, but not between the two networks. A **clinfo.rc** file is installed on each node in the cluster, containing the IP addresses of several client systems.

PowerHA SystemMirror configuration in switched networks

If the networks and the switches are incorrectly defined or configured, unexpected network interface failure events can occur in PowerHA SystemMirror configurations that use switched networks.

Follow these guidelines when configuring switched networks:

- **Virtual local area network (VLAN)**. If VLANs are used, all interfaces on a particular network must be configured on the same VLAN (one network per VLAN). If you are using multicast for heartbeat, all interfaces and network devices must be capable of carrying multicast packets between nodes.
- **Autonegotiation settings**. Some Ethernet network interface cards (NIC) are capable of automatically negotiating their speed and other characteristics, such as half duplex or full duplex. Configure NIC not to use **autonegotiate**, but to run at the desired speed and duplex value. Set the switch port to which the NIC is connected to the same fixed speed and duplex value.

PowerHA SystemMirror and virtual Ethernet

PowerHA SystemMirror supports virtual Ethernet provided by the Virtual I/O Server (VIOS) or integrated virtual Ethernet (IVE) facilities, with the applicable APARs installed. The PowerHA SystemMirror support is identical for VIOS and IVE.

The PCI Hot Plug utility in PowerHA SystemMirror is not applicable to interfaces on a virtual Ethernet. This utility only handles physical interface cards. Because virtual Ethernet uses virtual I/O adapters, you cannot use the utility.

The following list contains additional considerations for PowerHA SystemMirror with virtual Ethernet:

- If VIOS has multiple physical interfaces defined on the same network, or if there are two or more PowerHA SystemMirror nodes using VIOS in the same frame, PowerHA SystemMirror will not be informed of (and therefore will not react to) single physical interface failures. This does not limit the availability of the entire cluster because VIOS routes traffic around the failure. VIOS support is analogous to EtherChannel in this regard. Use methods that are not based on VIOS to provide notification of individual physical interface failures.
- If VIOS has only a single physical interface on a network, then PowerHA SystemMirror detects a failure of that physical interface. However, the failure will isolate the node from the network.

  Note: In VIOS 2.2.0.11, or later, you can use storage area network (SAN) communication between logical partitions by establishing a virtual local area network through a virtual Ethernet adapter on each VIOS client. You can set up SAN communication through VIOS for both NPIV and vSCSI environments.
- In a VIOS environment, failure of the physical network adapter and network components outside the virtualized network might not be detected reliably. To detect external network failures, you must configure the `netmon.cf` file with one or more addresses outside of the virtualized network.

Troubleshooting virtual Ethernet connections

To troubleshoot virtual Ethernet interfaces defined to PowerHA SystemMirror and to detect an interface failure, treat these interfaces as interfaces defined on single adapter networks.

Note: For Ethernet, PowerHA supports any combination of virtual and physical adapters on the same network name.

In particular, list the network interfaces that belong to a VLAN in the **etc/cluster/ping_client_list** or in the PING_CLIENT_LIST variable in the **/usr/es/sbin/cluster/etc/clinfo.rc** script and run the **clinfo** program. This way, whenever a cluster event occurs, the **clinfo** program monitors and detects a failure of the listed network interfaces. Due to the nature of VLAN, other mechanisms to detect the failure of network interfaces are not effective.

Related concepts

Virtual networks in PowerHA SystemMirror

In PowerHA SystemMirror Version 7.1.0, or later, the adapter monitor provided by Cluster Aware AIX (CAA) cannot always identify whether a virtual adapter has lost its corresponding physical adapter.

Enabling ports for cluster communication in PowerHA SystemMirror

You can deploy PowerHA SystemMirror on premises or in the cloud environment by using the standard cluster configuration or linked cluster configuration.

When you deploy PowerHA SystemMirror, all nodes in the cluster must communicate with each other without a failure and exchange heartbeats and cluster messages. The cluster software stack for PowerHA SystemMirror consists of Cluster Aware AIX (CAA), Reliable Scalable Cluster Technology (RSCT), and PowerHA subsystems. Each cluster software uses different ports for cluster communication. If a firewall exists between sites or cluster nodes, ensure that the cluster-related ports are properly opened for cluster communication to avoid network failures.

For PowerHA cluster nodes, the following ports must be enabled for all communication interfaces (boot IP addresses, service IP addresses, and persistent IP addresses) for both in and out network traffic.

Port number	Protocol	Description
22	Transmission Control Protocol (TCP)	Used for Secure Socket Shell (SSH) configuration.
657	TCP	Used for Resource Monitoring and Control (RMC) communication.
657	User Datagram Protocol (UDP)	Used for Resource Monitoring and Control (RMC) communication.
2049	TCP	Used for Network File System (NFS) Tiebreaker (NFSv4) in PowerHA SystemMirror.
4098	UDP	Used for cluster multicast communication.
6174	TCP	The **clinfo_client** daemon uses this port number for the **clstat** utility and for other **clinfo** applications.
6175	TCP	The **clm_smux** daemon uses this port number for the sysinfod SNMP multiplexing (SMUX) peer operations of the Simple Network Management Protocol (SNMP).
6176	TCP	The **clinfo_deadman** daemon uses this port number for **clinfo** monitoring operations.
6181	TCP	The **clcomd** daemon uses this port for cluster configuration. PowerHA SystemMirror verification checks for the **caa_cfg** entry in the `/etc/services` file.
6191	TCP	The **clcomd** daemon uses this port number during the migration process from an older release of PowerHA SystemMirror.
6192	TCP	This port is used for Remote physical volumes (RPV) client-server communication.
6270	TCP	The **clsmuxpd** daemon uses this port number for SNMP operation.
8080	TCP	Used for PowerHA SystemMirror GUI Server.
8081	TCP	Used for PowerHA SystemMirror GUI Agent.

Port number	Protocol	Description
12601	TCP	Reserved by RSCT for the future purpose.
16191	TCP	Used for **clcomd** daemon communication.
42112	TCP	Used for cluster unicast communication.

Note: In addition to the listed ports, you must open other ports that are needed for the respective applications.

Monitoring in PowerHA SystemMirror

The primary task of PowerHA SystemMirror is to recognize and respond to failures. PowerHA SystemMirror uses the Cluster Aware AIX infrastructure to monitor the activity of its network interfaces, devices, and IP labels.

Monitoring connections are necessary because they enable PowerHA SystemMirror to recognize the difference between a network failure and a node failure. For instance, if connectivity on the PowerHA SystemMirror network (this network's IP labels are used in a resource group) is lost, and you have another TCP/IP based network, PowerHA SystemMirror recognizes the failure of its cluster network and takes recovery actions that prevent the cluster from becoming partitioned.

To avoid cluster partitioning, you should configure redundant networks in the PowerHA SystemMirror cluster.

PowerHA SystemMirror automatically monitors interfaces on the following components:

- TCP/IP networks
- Storage area networks
- Repository disks

Designing the network topology

The combination of IP and non-IP (point-to-point) networks that link cluster nodes and clients is called the cluster *network topology*. The PowerHA SystemMirror software supports a large number of IP and point-to-point devices on each node to provide flexibility in designing a network configuration.

When designing your network topology, ensure that clients have highly available network access to their applications. This requires that none of the following network interfaces are a single point of failure:

- The IP subsystem
- A single network
- A single NIC

Eliminating networks as single points of failure

In a single-network setup, each node in the cluster is connected to only one network and has a single service interface available to clients. In this setup, the network is a single point of failure for the entire cluster, and each service interface is a single point of failure.

The following diagram shows a single-network configuration.

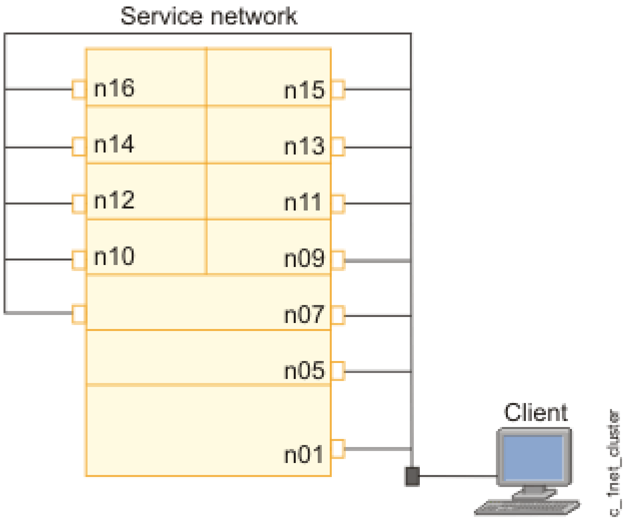


Figure 1. Single-network and single-NIC setup

To eliminate the network as a single point of failure, configure multiple networks so that PowerHA SystemMirror has multiple paths among cluster nodes. Keep in mind that if a client is connected to only one network, that network is a single point of failure for the client. In a multiple-network setup if one network fails, the remaining networks can still function to connect nodes and provide access for clients.

The more networks you can configure to carry heartbeats and other information among cluster nodes, the greater the degree of system availability.

The following diagram illustrates a dual-network setup with more than one path to each cluster node.

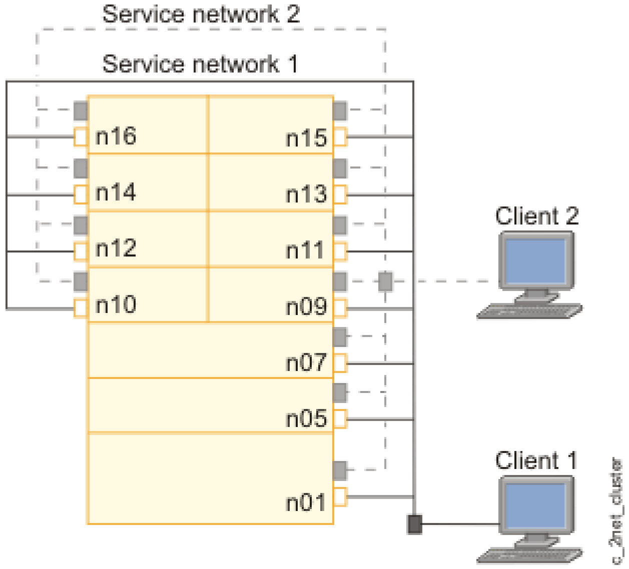


Figure 2. Dual-network setup

Note: Hot replacement of the dual-port Ethernet adapter used to configure two interfaces for one PowerHA SystemMirror IP network is currently not supported.

Eliminating network interface cards as a single point of failure

A network interface card (NIC) physically connects a node to a network.

When configured with a single NIC per network, the NIC becomes a potential single point of failure. To remedy this problem, configure a node with at least two NICs for each network to which it connects. In the following figure, each cluster node has two connections to each network.

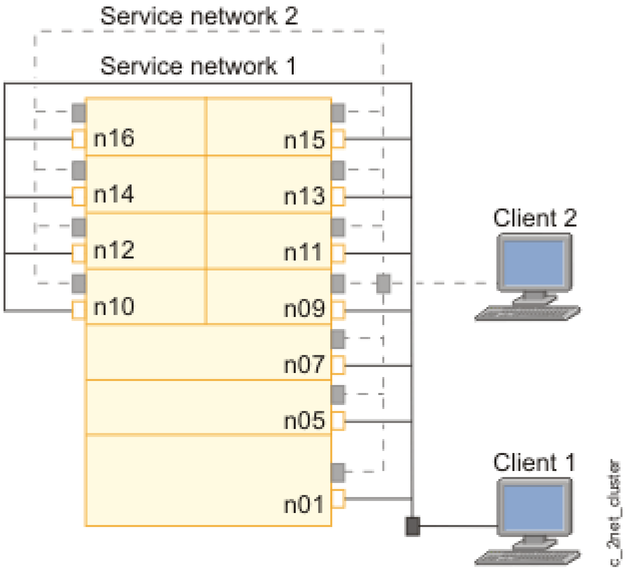


Figure 3. Dual-network and dual-NIC configuration

Note: Hot replacement of the dual-port Ethernet adapter used to configure two interfaces for one PowerHA SystemMirror IP network is currently not supported.

Related information

Resource group behavior during cluster events

Network interface functions

When a node is configured with multiple connections to a single network, the network interfaces serve different functions in PowerHA SystemMirror.

Service interface

A service interface is a network interface configured with a PowerHA SystemMirror service IP label. The service IP label is used by clients to access application programs. The service IP is only available when the corresponding resource group is online.

Persistent node IP label

A persistent node IP label is an IP alias that can be assigned to a specific node on a cluster network. A persistent node IP label always stays on the same node (node-bound), and coexists on an NIC that already has a service or boot IP label defined. A persistent node IP label is does not require installing an additional physical NIC on that node, and is not part of any resource group.

Assigning a persistent node IP label provides a node-bound address that you can use for administrative purposes, because a connection to a persistent node IP label always goes to a specific node in the cluster. You can have one persistent node IP label per network per node.

As one of the best practices for PowerHA SystemMirror, you must configure a persistent IP label for each cluster node. This is useful, for instance, if you must access a particular node in a PowerHA SystemMirror cluster for purposes of running reports or for diagnostics. Having a persistent IP label configured has the advantage that PowerHA SystemMirror can access the persistent IP label on the node despite individual NIC failures, assuming that there are spare NICs on the network.

After a persistent node IP label is configured on a specified network node, it becomes available at boot time and remains configured even if PowerHA SystemMirror is shut down on that node.

You can create persistent node IP labels on Ethernet networks.

The following list describes how PowerHA SystemMirror responds to a failure when a persistent node IP label is configured:

- If a NIC that has a service IP label configured fails and there is also a persistent label defined on this NIC, the persistent label falls over to the same boot interface to which the service IP label falls over.
- If all NICs on the cluster network on a specified node fail, the persistent node IP label becomes unavailable. A persistent node IP label always remains on the same network and on the same node. It does not move between the nodes in the cluster.

Related information

Configuring PowerHA SystemMirror cluster topology and resources (extended)

IP address takeover via IP aliases

PowerHA SystemMirror uses IPAT via IP aliases to keep service IP addresses highly available.

When PowerHA SystemMirror is started on the node, the service IP label is aliased onto one of the boot interfaces that is defined to PowerHA SystemMirror. If that interface fails, the service IP label is aliased onto another interface if one is available on the same network. To use IPAT via IP aliases, the network must support gratuitous ARP.

Subnet considerations for persistent node IP labels

When you configure a persistent node IP label on a cluster network, the IP address associated with a persistent IP label must be on a different subnet than any service address with which it might share an interface. Networks with one interface per node do not require separate subnets.

In some situations you might need to configure a persistent IP label on the same subnet as the service IP label. In this case, to avoid problems with network packets sent from either of the addresses, consider configuring the distribution preference for service IP aliases. This preference lets you configure the type of the distribution preference suitable for the VPN firewall external connectivity requirements.

Note: The subnet considerations are different if you are planning to configure a Network File System (NFS).

Related reference

NFS cross-mounting and IP labels
To enable NFS cross-mounting, each cluster node might act as an NFS client. Each of these nodes must have a valid route to the service IP label of the NFS server node. That is, to enable NFS cross-mounting, an IP label must exist on the client nodes, and this IP label must be configured on the same subnet as the service IP label of the NFS server node.

Types of distribution for service IP label aliases
You can specify in SMIT different distribution preferences for the placement of service IP label aliases.

Planning for IP address takeover via IP aliases
Assigning IP aliases to NICs allows you to create more than one IP label on the same network interface.

Planning for IP address takeover via IP aliases

Assigning IP aliases to NICs allows you to create more than one IP label on the same network interface.

During IP address takeover via IP aliases, when an IP label moves from one NIC to another, the target NIC receives the new IP label as an IP alias and keeps the original IP label and hardware address.

Configuring networks for IPAT via IP aliases simplifies the network configuration in PowerHA SystemMirror. You configure a service address and one or more boot addresses for NICs.

Assigning IP labels for IPAT via IP aliases

PowerHA SystemMirror uses a technology referred to as IP address takeover (IPAT) via IP aliases for keeping IP addresses highly available.

Review the following information when planning for IP address takeover via IP aliases:

- Each network interface must have a boot IP label defined to PowerHA SystemMirror. The interfaces that is defined to PowerHA SystemMirror are used to keep the service IP addresses highly available.

- Hardware Address Takeover (HWAT) cannot be configured for networks that are using IP address takeover via IP aliasing.
- The following subnet requirements apply when there are multiple interfaces on a node attached to the same network:
  - All boot addresses must be defined on different subnets.
  - Service addresses must be on a different subnet from all boot addresses and persistent addresses.

  Note: These subnet requirements avoid the IP route striping function of the AIX operating system, which allows multiple routes to the same subnet and can cause application traffic to be sent to a failed interface. These requirements do not apply to multiple interfaces that are combined into a single logical interface using Ethernet Aggregation or EtherChannel.
- Service address labels configured for IP address takeover via IP aliases can be included in all nonconcurrent resource groups.
- Multiple service labels can coexist as aliases on a given interface.
- The netmask for all IP labels in a PowerHA SystemMirror network must be the same.
- If there are multiple service and persistent labels, PowerHA SystemMirror attempts to distribute them evenly across all available network interfaces. You can specify a location preference such that the persistent and service aliases are always mapped to the same interface. For more information about service labels, see Distribution preference for service IP label aliases.

The boot addresses on a node (the base address assigned by the AIX operating system after a system reboot and before the PowerHA SystemMirror software is started) are defined to PowerHA SystemMirror as boot addresses. PowerHA SystemMirror automatically discovers and configures boot addresses when you first configure your cluster. When the PowerHA SystemMirror software is started on a node, the node's service IP label is added as an alias onto one of the NICs that is defined as a boot address. If the NIC that hosts the service IP fails, PowerHA SystemMirror attempts to move it to another active boot NIC on the same node.

During a node fallover event, the service IP label that is moved is placed as an alias on the target node's NIC in addition to any other service labels that might already be configured on that NIC.

For example, if Node A fails, PowerHA SystemMirror tries to move all resource groups to Node B. If the resource groups contain service IP addresses, PowerHA SystemMirror places the service IP as an alias onto the appropriate NIC on Node B, and any other existing labels remain intact on Node B's NIC. Thus, a NIC on Node B can now receive client traffic that formerly was directed to the service address that was on Node A. Later, when Node A is restarted, it starts on its boot addresses and is available to host the service IP if Node B fail. When Node B releases the requested service IP label, the alias for the service IP labels is deleted on Node B. Node A again puts the service IP label as an alias onto one of its boot address interfaces on the appropriate network.

During IPAT, PowerHA SystemMirror attempts to recover the service IP address on the same node by using a different adapter on the same subnet. If there are no adapters available on the same subnet that is defined to PowerHA SystemMirror, then PowerHA SystemMirror moves the service IP address to the backup node that is defined in the resource group policy. While releasing the service IP address, PowerHA SystemMirror tries to preserve any communication routes for that IP address by moving them to a different adapter on the same subnet, if that adapter is not part of the PowerHA SystemMirror configuration.

If your environment has multiple adapters on the same subnet, all the adapters must have the same network configuration and the adapters must be part of the PowerHA SystemMirror configuration.

When using IPAT via IP aliases, service IP labels are acquired by using all available and appropriate interfaces. If multiple interfaces are available to host the service IP label, the interface is chosen according to the configured distribution policy. By default, an anti-collocation policy is used, and an attempt is made to evenly distribute service IP labels across all the available interfaces.

In PowerHA SystemMirror, there is no impact to any service IP labels aliased on an interface if you remove the boot-time address for the interface with the **ifconfig** command. However, if you use **chdev** command (or the chinet fastpath) to replace a boot address on an interface, this command deletes any service IP aliases and there is no indication that this happened in PowerHA SystemMirror. If you need to change or temporarily unassign the boot address of an interface that is actively hosting service addresses, it is best to move service addresses to another interface first. If no other interfaces are available, use the **ifconfig** command alias to replace a boot address on an interface without disrupting application access.

Related reference

NFS cross-mounting in PowerHA SystemMirror
An NFS cross-mounting is a specific PowerHA SystemMirror NFS configuration where each node in the cluster can act as both the NFS server and the NFS client. While a file system is being exported from one node, the file system is mounted with NFS on all the nodes for the resource group, including the one that is exporting it. Another file system can also be exported from another node, and be mounted with NFS on all nodes.

Planning for service IP label alias placement

If you use IPAT via IP aliases, you can configure a distribution preference for the placement of service IP labels that are configured in PowerHA SystemMirror.

PowerHA SystemMirror lets you specify the distribution preference for the service IP label aliases. These are the service IP labels that are part of PowerHA SystemMirror resource groups and that belong to IPAT via IP aliases networks.

A distribution preference for service IP label aliases is a networkwide attribute used to control the placement of the service IP label aliases on the physical network interface cards on the nodes in the cluster.

Use the distribution preference for IP aliases to address the following cluster requirements:

- Firewall considerations
- Cluster configurations that use VLANs (when applications are expecting to receive packets from a specific network interface)
- Specific requirements for the placement of IP labels in the cluster

Distribution preference for service IP label aliases

Configuring a distribution preference for service IP label aliases does the following:

- Lets you customize the load balancing for service IP labels in the cluster, taking into account the persistent IP labels previously assigned on the nodes.
- Enables PowerHA SystemMirror to redistribute the alias service IP labels according to the preference you specify.
- Allows you to configure the type of the distribution preference suitable for the VPN firewall external connectivity requirements.

Although the service IP labels might move to another network interface, PowerHA SystemMirror ensures that the labels continue to be allocated according to the specified distribution preference. That is, the distribution preference is maintained during startup and the subsequent cluster events, such as a fallover, fallback, or a change of the interface on the same node. For instance, if you specified the labels to be mapped to the same interface, the labels will remain mapped on the same interface, even if the initially configured service IP label moves to another node.

The distribution preference is exercised in the cluster as long as acceptable network interfaces are available. PowerHA SystemMirror always keeps service IP labels active, even if the preference cannot be satisfied.

Types of distribution for service IP label aliases

You can specify in SMIT different distribution preferences for the placement of service IP label aliases.

The types of distribution preferences follow.

Type of distribution preference	Description
Anti-collocation	This is the default. PowerHA SystemMirror distributes all service IP label aliases across all boot IP labels by using a least-loaded selection process.
Collocation	PowerHA SystemMirror allocates all service IP label aliases on the same network interface card (NIC).
Anti-collocation with persistent labels	PowerHA SystemMirror distributes all service IP label aliases across all active physical interfaces that are not hosting the persistent IP label. PowerHA SystemMirror places the service IP label alias on the interface that is hosting the persistent label only if no other network interface is available. **Note:** If you did not configure persistent IP labels, PowerHA SystemMirror lets you select the anti-collocation with persistent distribution preference, but it issues a warning and uses the regular anti-collocation preference by default.
Collocation with persistent labels	All service IP label aliases are allocated on the same NIC that is hosting the persistent IP label. This option might be useful in VPN firewall configurations where only one interface is granted external connectivity and all IP labels (persistent and service) must be allocated on the same interface card. **Note:** If you did not configure persistent IP labels, you can use PowerHA SystemMirror to select the collocation with persistent distribution preference, but it issues a warning and uses the regular collocation preference by default.
Collocation with persistent label and source	Service labels are mapped to the same physical interface that currently has the persistent IP label for this network. This option allows you to choose one service label as the source for outgoing communication. The service label chosen in the next field will be used as the source address.
Anti-collocation with source	Service labels are mapped by using the anti-collocation preference. If there are not enough adapters, more than one service label can be placed on one adapter. With this choice one label is chosen as the source address for outgoing communication. The interface label chosen in the **Source IP Label for outgoing packets** field is the source address.
Collocation with source	Service labels are mapped by using Collocation preference. With this choice one service label is chosen as the source address for outgoing communication. The interface label chosen in the **Source IP Label for outgoing packets** field is the source address.
Anti-Collocation with persistent label and source	Service labels are mapped by using the anti-collocation with persistent preference. One service address can be chosen as a source address for the case when there are more service addresses than boot adapters.

The following rules apply to the distribution preference:

- If there are insufficient interfaces available to satisfy the preference, PowerHA SystemMirror allocates service IP label aliases and persistent IP labels to an existing active network interface card.
- If you did not configure persistent labels, PowerHA SystemMirror lets you select the collocation with persistent and anti-collocation with persistent distribution preferences, but it issues a warning and uses the regular collocation or anti-collocation preferences by default.
- You can change the IP labels distribution preference dynamically: the new selection becomes active during subsequent cluster events. PowerHA SystemMirror does not interrupt the processing by relocating the currently active service IP labels at the time the preference is changed.

When a service IP label fails and another one is available on the same node, PowerHA SystemMirror recovers the service IP label aliases by moving them to another NIC on the same node. During this event, the distribution preference that you specified remains in effect.

Related information

Administering PowerHA SystemMirror

Planning for site-specific service IP labels

You can have a service IP label that is configurable on multiple nodes and is associated with a resource group that can move between nodes or sites.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

An IP address that is valid at one site might not be valid at the other site because of subnet issues, you can associate a service IP label that is configurable on multiple nodes with a specific site. Site-specific service IP labels are configured in PowerHA SystemMirror and can be used with or without PowerHA SystemMirror Enterprise Edition for AIX networks. This label is associated with a resource group and is active only when the resource group is in an Online Primary state at the associated site.

Planning for other network conditions

The most typical network planning consideration involve name serving in a PowerHA SystemMirror environment, and setting up cluster monitoring and failure detection.

Using PowerHA SystemMirror with NIS and DNS

Some of the commands used to troubleshoot network and interface problems after a failure require IP lookup to determine the IP address associated with a specified IP label.

If network information service (NIS) or domain name server (DNS) is in operation, IP lookup defaults to a name server system for name and address resolution. However, if the name server was accessed through an interface that has failed, the request does not complete, and eventually times out. This time out can significantly slow down PowerHA SystemMirror event processing.

To ensure that cluster event completes successfully and quickly, PowerHA SystemMirror disables NIS or DNS host name resolution by setting the following AIX environment variable during service IP label swapping:

```
NSORDER = local
```

As a result, the **/etc/hosts** file of each cluster node must contain all PowerHA SystemMirror defined IP labels for all cluster nodes.

DNS requests sent from processes other than PowerHA SystemMirror

Disabling NIS or DNS host name resolution is specific to the PowerHA SystemMirror event script environment. PowerHA SystemMirror sets the NSORDER variable to *local* when it attaches a service IP label and when it swaps IP labels on an interface.

Other processes continue to use the default system name resolution settings (for example, applications outside of PowerHA SystemMirror that require DNS IP address resolution). If these processes request IP lookup, then during the network interface reconfiguration events in PowerHA SystemMirror the processes still may not be able to contact an external name server. The request to the DNS will succeed after PowerHA SystemMirror completes the network interface reconfiguration event.

Monitoring clusters

The Cluster Aware AIX infrastructure monitors all available and supported network and storage interfaces. The cluster managers on cluster nodes also send messages to each other through the connections between these interfaces.

All available and supported network and storage interfaces in the cluster are used to, monitor interfaces, ensure connectivity to cluster peers, and report when a connection fails.

Contact your IBM representative for the current list of supported adapters, disks and multipath drivers. PowerHA SystemMirror supports monitoring and communication over the following types of interfaces:

- Ethernet
- 4 GB and 8 GB Emulex Fibre Channel adapters
- The repository disk (SAN and SAS disks are supported)
- FCoE (Fibre Channel over Ethernet)

Related information

PowerHA Hardware Support Matrix

Planning for VPN firewall network configurations in PowerHA SystemMirror

PowerHA SystemMirror allows you to specify the distribution preference for the service IP label aliases. These are the service IP labels that are part of PowerHA SystemMirror resource groups and that belong to IPAT via IP aliasing networks.

Certain VPN firewall configurations allow external connectivity to only one NIC at a time. If your firewall is configured this way, allocate all PowerHA SystemMirror service and persistent IP labels on the same interface.

To have PowerHA SystemMirror manage the IP labels to satisfy the requirements of such a VPN firewall:

- Specify the persistent IP label for each node in the cluster. The persistent IP label is mapped to an available interface on the selected network.
- Specify the collocation with persistent distribution preference for the network containing the service IP labels. This ensures that all service IP label aliases are allocated on the same physical interface that is hosting the persistent IP label.

Related reference

Types of distribution for service IP label aliases
You can specify in SMIT different distribution preferences for the placement of service IP label aliases.

Related information

Administering PowerHA SystemMirror

Planning for IP version 6 address with PowerHA SystemMirror

Internet Protocol version 6 (IPv6) is supported in PowerHA SystemMirror 7.1.2, or later.

Before you implement IPv6 in your environment, you must consider the following areas of PowerHA SystemMirror:

- Cluster Aware AIX (CAA) automatically uses heartbeats for any IP address configured in a node. To prevent CAA from heartbeating any IPv6 address on a particular interface, your network that uses PowerHA SystemMirror must be identified as a private network.
- IPv6 uses dynamic configuration of adapter addresses and other network attributes. By default, IPv6 addresses do not persist across a system reboot operation, however, you can configure your system to run the **autoconf6** command during startup. You need to plan for how and where the **autoconf6** command is run in your system environment.
- PowerHA SystemMirror uses link local addresses as boot addresses. You can configure a second alias address to be used as the PowerHA SystemMirror boot address. Link local addresses are convenient to use for boot addresses, because they are always configured for any interface that uses IPv6. You must plan for the IPv6 boot addresses that you can use in your environment.
- When IPv6 is used in a cluster, the **autoconf6** command is run for each interface that is configured with an IPv6 address. By default, the **autoconf6** command creates LL addresses starting with FE80. However, if you configured the **ndpd-host** daemon to listen for a router advertised prefix in the network, then a global IPv6 address might be assigned which can have a network address that starts with 2xxx. PowerHA SystemMirror does not recognize these addresses as being LL. This particular combination of IPv6 addresses are not supported.

Related information

Heartbeating over Internet Protocol networks

Persisting IPv6 addresses across system reboot

Planning networks for internode communication with Oracle

Oracel uses the **private** network attribute setting to select networks for Oracle internode communications. This attribute is not used by PowerHA SystemMirror and will not affect PowerHA SystemMirror in any way. The default attribute is **public**.

Changing the network attribute to **private** makes the network Oracle-compatible by changing all interfaces to service.

After creating your cluster networks (either manually or using discovery), you can change the network attribute by following this SMIT path:

Cluster Nodes and Networks > **Manage Networks and Network Interfaces** > **Networks** > **Change/ Show a Network**

Select the network to be changed, and then change the Network Attribute setting to **private**. Synchronize the cluster after making this change.

Rules for configuring private networks

Follow these steps to configure private networks for use by Oracle:

1. Configure the network and add all interfaces. You cannot change the attribute if the network has no interfaces.
2. Change the network attribute to **private**.
3. Verify that private networks have either all boot or all service interfaces. If the network has all boot interfaces (the default when using discovery) PowerHA SystemMirror converts these interfaces to service (Oracle only looks at service interfaces).
4. Synchronize the cluster after changing the attribute.

Related information

Verifying and synchronizing a PowerHA SystemMirror cluster

Virtual networks in PowerHA SystemMirror

In PowerHA SystemMirror Version 7.1.0, or later, the adapter monitor provided by Cluster Aware AIX (CAA) cannot always identify whether a virtual adapter has lost its corresponding physical adapter.

For example, if a network cable is unplugged from a Virtual I/O Server (VIOS), it cannot communicate with the external network. Thus, the VIOS partitions might report their individual virtual interfaces as available, when they cannot reach any external LAN beyond the virtual network.

If you are setting up a virtual network or an Integrated Virtual Ethernet (IVE) network in PowerHA SystemMirror for the first time, you must create a `netmon.cf` file in the `/usr/es/sbin/cluster` directory. In the `netmon.cf` file, you must have at least one line for each virtual interface by using the following format:

```
!REQD owner target
```

The following list describes the variables that are used in the `netmon.cf` file.

!REQD

An explicit string that must be at the beginning of the line without any leading spaces.

owner

The interface whose online or offline status is determined by whether it can ping any of the specified targets. The owner can be specified as a hostname, IP address, or interface name. If you use a hostname, it must resolve to an IP address or the line is ignored. You can specify the `!ALL` string to indicate that all adapters use the specified target.

target

The IP address or hostname you want the owner to try to ping. To use a hostname, the target must be resolvable to an IP address.

When you are creating or changing the `netmon.cf` file, consider the following information:

- When you change the `netmon.cf` file in PowerHA SystemMirror Version 7.1, or later, you do not have to restart cluster services to apply the changes. The **cthags** subsystem automatically re-reads the `netmon.cf` file approximately every minute.
- You must select targets that are outside the virtual network environment.
- Targets that you identify must be maintained through changes in your network environment.
- You can provide only one target per line. However, in IBM AIX 7.1 with Technology Level 4, or earlier, you specify the same owner entry up to 32 different lines in the `netmon.cf` file. In IBM AIX 7.1 with Technology Level 4, or later, and AIX Version 7.2, or later, only the last five entries for an owner entry are considered. For an owning adapter listed on more than one line, the adapter is considered available if it can ping any of the provided targets.
- Do not use multiple targets that are on the same physical system or device because a single system or device failure causes the adapter state to be marked as down. Also, targets on other PowerHA SystemMirror nodes must not be used because a node failure can be interpreted as a network failure, and might prevent PowerHA SystemMirror from responding to the node failure.
- Each virtual adapter must have at least one line inside the `netmon.cf` file that specifies a target that can be pinged from the boot IP address on that interface, or a persistent IP alias if one is configured.
- Network hardware that can be pinged, such as gateways and routers, are useful as target addresses because PowerHA SystemMirror nodes already use them.

If some adapters on the same network are virtual and others are not, it is perfectly acceptable to use the **`!REQD`** format. For both the virtual adapter `en0` and the physical adapter `en1`, you must use the **`!REQD`** format in the `netmon.cf` file.

In PowerHA SystemMirror, all entries in the `netmon.cf` file must begin with the **`!REQD`** format. Any line that does not begin with the **`!REQD`** format is ignored as a comment by CAA. At least one target for an adapter must respond to a ping for the adapter to be considered active.

Note: This format also applies to IVE networks. However, you cannot use a target that is a member of the IVE network in the same physical system.

Examples

The following examples explain the content in the `netmon.cf` file:

1. In this example, the adapter that owns `host1.ibm` is only available if it can ping 100.12.7.9 or whatever `host4.ibm` resolves to. The adapter that owns 100.12.7.20 is only available if it can ping 100.12.7.10 or whatever `host5.ibm` resolves to. If 100.12.7.20 is the IP address that `host1.ibm` resolves to, then all four targets belong to that same adapter.

```
!REQD host1.ibm 100.12.7.9
!REQD host1.ibm host4.ibm
!REQD 100.12.7.20 100.12.7.10
!REQD 100.12.7.20 host5.ibm
```

2. In this example, all adapters are available only if they can ping the 100.12.7.9, 110.12.7.9, or 111.100.1.10 IP addresses. The `en1` owner entry has an additional target of 9.12.11.10.

```
!REQD !ALL 100.12.7.9
!REQD !ALL 110.12.7.9
!REQD !ALL 111.100.1.10
!REQD en1 9.12.11.10
```

Related reference

PowerHA SystemMirror and virtual Ethernet

PowerHA SystemMirror supports virtual Ethernet provided by the Virtual I/O Server (VIOS) or integrated virtual Ethernet (IVE) facilities, with the applicable APARs installed. The PowerHA SystemMirror support is identical for VIOS and IVE.

Related information

netmon.cf File

Avoiding network conflicts

You can avoid network conflicts on IP addresses. A verification will notify you of duplicate IP addresses. Correct the duplicate address and resynchronize the cluster.

Adding the network topology to the cluster diagram

Sketch the networks to include all TCP/IP networks. Identify each network by name and attribute. In the boxes in each node that represents slots, record the interface label.

You can now add networking to the sample cluster diagram started in Overview of the planning process.

Related reference

Overview of the planning process
This topic describes the steps for planning a PowerHA SystemMirror cluster.

Enabling ports for cluster communication in PowerHA SystemMirror

You can deploy PowerHA SystemMirror on premises or in the cloud environment by using the standard cluster configuration or linked cluster configuration.

When you deploy PowerHA SystemMirror, all nodes in the cluster must communicate with each other without a failure and exchange heartbeats and cluster messages. The cluster software stack for PowerHA SystemMirror consists of Cluster Aware AIX (CAA), Reliable Scalable Cluster Technology (RSCT), and PowerHA subsystems. Each cluster software uses different ports for cluster communication. If a firewall exists between sites or cluster nodes, ensure that the cluster-related ports are properly opened for cluster communication to avoid network failures.

For PowerHA cluster nodes, the following ports must be enabled for all communication interfaces (boot IP addresses, service IP addresses, and persistent IP addresses) for both in and out network traffic.

Port number	Protocol	Description
22	Transmission Control Protocol (TCP)	Used for Secure Socket Shell (SSH) configuration.
657	TCP	Used for Resource Monitoring and Control (RMC) communication.
657	User Datagram Protocol (UDP)	Used for Resource Monitoring and Control (RMC) communication.
2049	TCP	Used for Network File System (NFS) Tiebreaker (NFSv4) in PowerHA SystemMirror.
4098	UDP	Used for cluster multicast communication.
6174	TCP	The **clinfo_client** daemon uses this port number for the **clstat** utility and for other **clinfo** applications.
6175	TCP	The **clm_smux** daemon uses this port number for the sysinfod SNMP multiplexing (SMUX) peer operations of the Simple Network Management Protocol (SNMP).

Port number	Protocol	Description
6176	TCP	The **clinfo_deadman** daemon uses this port number for **clinfo** monitoring operations.
6181	TCP	The **clcomd** daemon uses this port for cluster configuration. PowerHA SystemMirror verification checks for the **caa_cfg** entry in the /etc/services file.
6191	TCP	The **clcomd** daemon uses this port number during the migration process from an older release of PowerHA SystemMirror.
6192	TCP	This port is used for Remote physical volumes (RPV) client-server communication.
6270	TCP	The **clsmuxpd** daemon uses this port number for SNMP operation.
8080	TCP	Used for PowerHA SystemMirror GUI Server.
8081	TCP	Used for PowerHA SystemMirror GUI Agent.
12601	TCP	Reserved by RSCT for the future purpose.
16191	TCP	Used for **clcomd** daemon communication.
42112	TCP	Used for cluster unicast communication.

Note: In addition to the listed ports, you must open other ports that are needed for the respective applications.

Planning shared disk and tape devices

This section discusses information to consider before configuring shared external disks in a PowerHA SystemMirror cluster and provides information about planning and configuring tape drives as cluster resources.

Prerequisites

You have completed the planning steps in the planning cluster network connectivity and planning application and application controllers sections.

Refer to AIX documentation for the general hardware and software setup for your disk and tape devices.

Overview of shared disk and tape devices

In a PowerHA SystemMirror cluster, shared disks are external disks connected to more than one cluster node that are used for application shared storage.

In a nonconcurrent configuration, only one node at a time owns the disks. If the owner node fails, the cluster node with the next highest priority in the resource group node list acquires ownership of the shared disks and restarts applications to restore critical services to clients. This ensures that the data stored on the disks remains accessible to client applications.

Typically, takeover occurs within 30 - 300 seconds. This range depends on the number and types of disks used, the number of volume groups, the file systems (whether shared or Network File System (NFS) cross-mounted), and the number of critical applications in the cluster configuration.

When planning the shared external disk for your cluster, the objective is to eliminate single points of failure in the disk storage subsystem. The following table lists the disk storage subsystem components, with suggested ways to eliminate them as single points of failure.

Cluster object	**Eliminated as single point of failure by...**
Disk adapter	Using redundant disk adapters
Controller	Using redundant disk controllers
Disk	Using redundant hardware and LVM disk mirroring or RAID mirroring

In this section, you perform the following planning tasks:

- Choosing a shared disk technology.
- Planning the installation of the shared disk storage. This includes:
  - Determining the number of disks required to handle the projected storage capacity. You need multiple physical disks on which to put the mirrored logical volumes. Putting copies of a mirrored logical volume on the same physical device defeats the purpose of making copies. For more information about creating mirrored logical volumes, see Planning shared LVM components.
  - Determining the number of disk adapters that each node will contain to connect to the disks or disk subsystem.

    Physical disks containing logical volume copies should be on separate adapters. If all logical volume copies are connected to a single adapter, the adapter is potentially a single point of failure. If the single adapter fails, PowerHA SystemMirror moves the volume group to an alternate node. Separate adapters prevent the need for this move.
  - Understand the cabling requirements for each type of disk technology.
- Adding the selected disk configuration to the cluster diagram.
- Planning for configuring a direct fibre channel tape unit attachment as a cluster resource.

Related reference

Planning shared LVM components
This section describe planning shared volume groups for a PowerHA SystemMirror cluster.

Choosing a shared disk technology

The PowerHA SystemMirror software supports disk technologies as application shared external disks in a highly available cluster.

For specific information on what disk technologies are supported on specific version of PowerHA SystemMirror and the AIX operating system, see PowerHA hardware support matrix.

Related information

OEM disk, volume group, and file systems accommodation

Disk power supply considerations

Reliable power sources are critical for a highly available cluster. Each mirrored disk chain in the cluster should have a separate power source. As you plan the cluster, make sure that the failure of any one power source (PDU, power supply, or building circuit) does not disable more than one node or mirrored chain.

The IBM DS4000® series are less prone to power supply problems because they have redundant power supplies.

Planning for nonshared disk storage

Keep some considerations in mind regarding nonshared disk storage.

These considerations include:

- Internal disks. The internal disks on each node in a cluster must provide sufficient space for:

  - AIX software (approximately 500 MB)
  - PowerHA SystemMirror software (approximately 50 MB for a server node)
  - Executable modules of highly available applications
- Root volume group. The root volume group for each node must not reside on the shared SCSI bus.
- AIX Error Notification Facility. Use the AIX Error Notification Facility to monitor the disks and adapters on each node. You can enable the automatic error notification in PowerHA SystemMirror to monitor both shared and nonshared disks by entering `smit sysmirror` from the command line and selecting **Problem Determination Tools** > **PowerHA SystemMirror Error Notification** > **Configure Automatic Error Notification** > **Add Error Notify Methods for Cluster Resources**.
- Disk adapter use. Because shared disks require their own adapters, you cannot use the same adapter for both a shared and a nonshared disk. The internal disks on each node require one SCSI adapter apart from any other adapters within the cluster.
- Volume group use. Internal disks must be in a different volume group from the external shared disks.

The executable modules of the highly available applications should be on the internal disks and not on the shared external disks, for the following reasons:

- Licensing
- Application startup

Related information
Configuring AIX for PowerHA SystemMirror

Licensing

Vendors might require that you purchase a separate copy of each application for each processor or multiprocessor that might run it, and protect the application by incorporating processor-specific information into the application when it is installed.

Thus, if you are running your application executable from a shared disk, it is possible that after a fallover, PowerHA SystemMirror will be unable to restart the application on another node, because, for example, the processor ID on the new node does not match the ID of the node on which the application was installed.

The application might also require that you purchase what is called a node-bound license, that is, a license file on each node that contains information specific to the node.

There might also be a restriction on the number of floating licenses (available to any cluster node) available within the cluster for that application. To avoid this problem, be sure that there are enough licenses for all processors in the cluster that might potentially run an application at the same time.

Starting applications

Applications might contain configuration files that you can customize during installation and store with the application files. These configuration files usually store information, such as path names and log files, that are used when the application starts.

You might need to customize your configuration files if your configuration requires both of the following:

- You plan to store these configuration files on a shared file system.
- The application cannot use the same configuration on every fallover node.

For example, in a two-node mutual takeover configuration, both nodes might be running different instances of the same application, and standing by for one another. Each node must be aware of the location of configuration files for both instances of the application, and must be able to access them after a fallover. Otherwise, the fallover will fail, leaving critical applications unavailable to clients.

To decrease how much you will need to customize your configuration files, place slightly different startup files for critical applications on local file systems on either node. This allows the initial application parameters to remain static. The application will not need to recalculate the parameters each time it is called.

Planning a shared disk installation

This section summarize the basic hardware components required to set up a PowerHA SystemMirror cluster.

Your cluster requirements depend on the configuration you specify. To ensure that you account for all required components, complete a diagram for your system. In addition, consult the hardware information for detailed information about cabling and attachment for the particular devices you are configuring.

PowerHA SystemMirror and virtual SCSI

PowerHA SystemMirror supports virtual SCSI (VSCSI) with the applicable APARs installed.

The following restrictions apply to using VSCSI in a cluster configuration:

- If file systems are used on the standby nodes, they are not mounted until the point of fallover so that accidental use of data on standby nodes is impossible.
- If shared volumes are accessed directly (without file systems) in enhanced concurrent mode, these volumes are accessible from multiple nodes, so access must be controlled at a higher layer such as databases.
- From the point of view of the virtual I/O server (VIOS), physical disks are shared, not logical volumes or volume groups.
- All volume group construction and maintenance on these shared disks is done from the PowerHA SystemMirror nodes, not from VIOS.

Disk adapters

Remove any SAS terminators on the adapter. Use external terminators in a PowerHA SystemMirror cluster. If you terminate the shared SAS bus on the adapter, you lose termination when the cluster node that contains the adapter fails.

Cables

The cables required to connect nodes in your cluster depend on the type of SCSI bus you are configuring. Select cables that are compatible with your disk adapters and controllers. For information on the type and length SCSI cable required, see the hardware documentation that accompanies each device you want to include on the SCSI bus.

Example: DS4000 Storage Server configuration

This example shows a configuration for high availability when using an IBM DS4000 Storage Server in a PowerHA SystemMirror environment.

See the following figure.

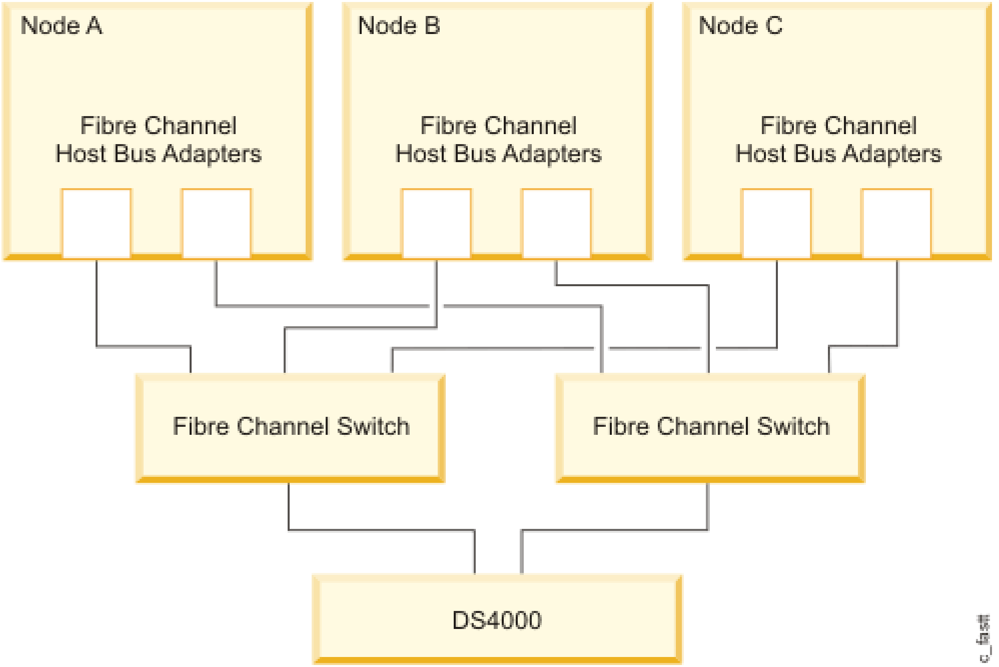

Node A
Fibre Channel Host Bus Adapters
Node B
Fibre Channel Host Bus Adapters
Node C
Fibre Channel Host Bus Adapters
Fibre Channel Switch
Fibre Channel Switch
DS4000
c_fastt

Figure 4. DS4000 Storage Server environment

Adding the disk configuration to the cluster diagram

After you have chosen a disk technology, add your disk configuration to the cluster diagram you started in Initial cluster planning.

For the cluster diagram, draw a box representing each shared disk. Then label each box with a shared disk name.

Related reference

Initial cluster planning
This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

Planning for tape drives as cluster resources

You can configure a tape drive as a cluster resource, making it highly available to multiple nodes in a cluster.

Direct fibre channel tape unit attachments are supported. Management of shared tape drives is simplified by the following PowerHA SystemMirror functions:

- Configuration of tape drives using SMIT
- Verification of proper configuration of tape drives
- Automatic management of tape drives during resource group start and stop operations
- Reallocation of tape drives on node failure and node recovery
- Controlled reallocation of tape drives on cluster shutdown
- Controlled reallocation of tape drives during dynamic reconfiguration

Limitations

When you plan to include tape drives as cluster resources remember the following:

- A tape loader or stacker is treated like a simple tape drive by PowerHA SystemMirror.
- No more than two cluster nodes can share the tape resource.

- Tape resources cannot be part of concurrent resource groups.
- The tape drive must have the same name (for example, **/dev/rmt0**) on both nodes that shares the tape device.
- When a tape special file is closed, the default action is to release the tape drive. PowerHA SystemMirror is not responsible for the state of the tape drive after an application has opened the tape.
- No means of synchronizing tape operations and application controllers is provided. If you decide that a tape reserve and release operation should be done asynchronously, provide a way to notify the application controller to wait until the reserve and release operation is complete.

Reserving and releasing shared tape drives

When a resource group with tape resources is activated, the tape drive is reserved to allow its exclusive use.

This reservation is held until an application releases it, or the node is removed from the cluster:

- When the special file for the tape is closed, the default action is to release the tape drive. An application can open a tape drive with a do-not-release-on-close flag. PowerHA SystemMirror will not be responsible for maintaining the reservation after an application is started.
- Upon stopping cluster services on a node and bringing resource groups offline, the tape drive is released, allowing access from other nodes.
- Upon unexpected node failure, a forced release is done on the takeover node. The tape drive is then reserved as part of resource group activation.

Setting tape drives to operate synchronously or asynchronously

If a tape operation is in progress when a tape reserve or release is initiated, it may take might minutes before the reserve or release operation completes. PowerHA SystemMirror allows synchronous or asynchronous reserve and release operations. Synchronous and asynchronous operation is specified separately for reserve and release.

Synchronous operation

With synchronous operation, (the default value), PowerHA SystemMirror waits for the reserve or release operation, including the execution of a user-defined recovery procedure, to complete before continuing.

Asynchronous operation

With asynchronous operation, PowerHA SystemMirror creates a child process to perform the reserve or release operation, including the execution of a user-defined recovery procedure, and immediately continues.

Recovery procedures

Recovery procedures are highly dependent on the application accessing the tape drive.

Rather than trying to predict likely scenarios and develop recovery procedures, PowerHA SystemMirror provides for the execution of user defined recovery scripts for the following operations:

- Tape start
- Tape stop

Tape start scripts and stop scripts

Tape start and stop operations occur during node start and stop, node fallover and reintegration, and dynamic reconfiguration. These scripts are called when a resource group is activated (tape start) or when a resource group is deactivated (tape stop). Sample start and stop scripts can be found in the **/usr/es/sbin/cluster/samples/tape** directory:

```
tape_resource_stop_example
```

- During tape start, PowerHA SystemMirror reserves the tape drive, forcing a release if necessary, and then calls the user-provided tape start script.
- During tape stop, PowerHA SystemMirror calls the user-provided tape stop script, and then releases the tape drive.

Note: You are responsible for correctly positioning the tape, terminating processes or applications, writing to the tape drive, and writing end-of-tape marks within these scripts.

Other application-specific procedures should be included as part of the start server and stop server scripts.

Adapter fallover and recovery

Tape drives with more than one SCSI interface are not supported. Therefore, only one connection exists between a node and a tape drive. The usual notion of adapter fallover does not apply.

Node fallover and recovery

If a node that has tape resources that are part of a PowerHA SystemMirror resource group fails, the takeover node reserves the tape drive, forcing a release if necessary, and then calls the user-provided tape start script.

On reintegration of a node, the takeover node runs the tape stop script and then releases the tape drive. The node being reintegrated reserves the tape drive and calls the user-provided tape start script.

Network fallover and recovery

PowerHA SystemMirror does not provide tape fallover and recovery procedures for network failure.

Planning shared LVM components

This section describe planning shared volume groups for a PowerHA SystemMirror cluster.

Prerequisites

You should also be familiar with how to use the Logical Volume Manager (LVM).

Overview

Planning shared logical volume manager (LVM) components for a PowerHA SystemMirror cluster depends on the type of shared disk device and the method of shared disk access.

To avoid a single point of failure for data storage, use data redundancy as supported by LVM or your storage system.

Related information

OEM disk, volume group, and file systems accommodation

Operating system and device management

Planning for LVM components

The logical volume manager (LVM) controls disk resources by mapping data between physical and logical storage.

Physical storage refers to the actual location of data on a disk. *Logical storage* controls how data is made available to the user. Logical storage can be discontiguous, expanded, and replicated, and can span multiple physical disks. These facilities provide improved availability of data.

Physical volumes

A physical volume is a single physical disk or a logicalunit that is presented by a storage array.

The physical volume is partitioned to provide the AIX operating system with a way of managing how data is mapped to the volume. The following figure shows a conventional use of physical partitions within a physical volume.

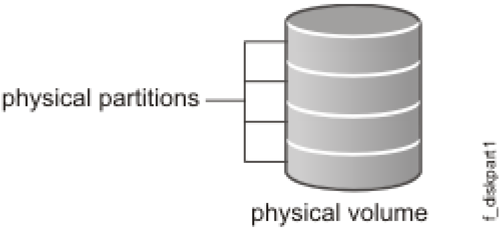


Figure 5. Physical partitions on a physical volume

When planning shared physical volumes, ensure that:

- The list of PVIDs for a volume group is identical on all cluster nodes that have access to the shared physical volume.
- The setting for the concurrent attribute of the volume group is consistent across all related cluster nodes.

Physical volume encryption

Starting with PowerHA SystemMirror Version 7.2.8 and IBM AIX 7.3 with Technology Level 1, you can encrypt the physical volume by using platform keystore (PKS) authentication method. During an I/O operation, data that is written to the physical volume is encrypted and data that is read from the physical volume is decrypted. For more information about encrypting physical volumes, see Encrypting physical volumes.

PowerHA SystemMirror Version 7.2.8, or later, supports the following type of authentication (key-protection) method to enable physical volume encryption.

platform keystore (PKS)

Starting with HMC Version 9.2.950 with system firmware level FW950, the PKS feature in PowerVM® creates an encrypted nonvolatile keystore. The PKS feature provides logical partitions with more capabilities to protect the sensitive information. The total partition size that is allocated for a logical partition (LPAR) for the PKS feature is in the range of 4 KB to 64 KB. Before you configure PKS as an authentication method in PowerHA SystemMirror, you must enable the Hardware Management Console (HMC) keystore.

Volume groups

A volume group is a set of physical volumes that the AIX operating system treats as a contiguous, addressable disk region. You can place multiple physical volumes in the same volume group. The actual number depends on how the volume group is created.

The following figure shows a volume group of three physical volumes:

f_diskpart2

Figure 6. Volume group of three physical volumes

In the PowerHA SystemMirror environment, a *shared volume group* is a volume group that resides entirely on the external disks that are shared by the cluster nodes. A nonconcurrent shared volume group can be varied on by only one node at a time.

When working with a shared volume group:

- Do not include an internal disk in a shared volume group, because it cannot be accessed by other nodes. If you include an internal disk in a shared volume group, the **varyonvg** command fails.
- Do not activate the shared volume groups in a PowerHA SystemMirror cluster manually at system boot. Use cluster event scripts to do this.
- Ensure that the automatic varyon attribute in the AIX ODM is set to **No** for shared volume groups listed within a resource group. The PowerHA SystemMirror cluster verification utility automatically corrects this attribute for you upon verification of cluster resources and sets the automatic varyon attribute to **No**.
- If you define a volume group to PowerHA SystemMirror, do not manage it manually on any node outside of PowerHA SystemMirror while PowerHA SystemMirror is running on other nodes. This can lead to unpredictable results. If you want to perform actions on a volume group independent of PowerHA SystemMirror, stop the cluster services, perform a manual volume group management task, leave the volume group varied off, and restart PowerHA SystemMirror. To ease the planning of PowerHA SystemMirror's use of physical volumes, the verification utility checks for:
  - Volume group consistency
  - Disk availability

Related information

mkvg command

varyonvg command

Logical volumes

A logical volume is a set of logical partitions that the AIX operating system makes available as a single storage unit, that is, the logical view of a disk.

A logical partition is the logical view of a physical partition. Logical partitions might be mapped to one, two, or three physical partitions to implement mirroring.

In the PowerHA SystemMirror environment, logical volumes can be used to support a journaled file system or a raw device.

Logical volume encryption

Starting with PowerHA SystemMirror Version 7.2.6 and IBM AIX Version 7.3, the Logical Volume Manager (LVM) enables data encryption for data volume groups that are configured in PowerHA SystemMirror environment. For more information about encrypting logical volumes, see Encrypting logical volumes.

PowerHA SystemMirror Version 7.2.6, or later, supports the following types of authentication (key-protection) methods to enable the logical volume encryption.

platform keystore (PKS)

Starting with HMC Version 9.2.950 with system firmware level FW950, the PKS feature in PowerVM creates an encrypted non-volatile keystore. The PKS feature provides logical partitions with more capabilities to protect the sensitive information. The total partition size allocated for an LPAR for the PKS feature is in the range of 4 KB to 64 KB. You must enable the Hardware Management Console (HMC) keystore before you configure PKS as an authentication method in PowerHA SystemMirror.

key server

You must configure the key server before you enable the logical volume encryption in PowerHA SystemMirror. You can use any key management server, which supports the is Key Management Interoperability Protocol (KMIP) as a key server.

File systems

A file system is written to a single logical volume.

Ordinarily, you organize a set of files as a file system for convenience and speed in managing data.

In the PowerHA SystemMirror system, a shared file system is a journaled file system that resides entirely in a shared logical volume.

You want to plan shared file systems to be placed on external disks that are shared by cluster nodes. Data resides in file systems on these external shared disks in order to be made highly available.

The order in which file systems are mounted is usually not important. However, if this is important to your cluster, you need to plan for some things:

- File systems that exist within a single resource group are mounted in alphanumeric order when the resource group comes online. They are also unmounted in reverse alphanumeric order when the resource group is taken offline.
- If you have shared, nested file systems, additional care is needed. If you have shared, nested file systems within a single resource group, then you must set the filesystems recovery method for the resource group to sequential to guarantee the correct mount order.
- If you have nested file systems that reside in different resource groups, you must additionally plan a parent-child relationship for those resource groups to guarantee the correct mount order.

Planning LVM mirroring

Logical volume manager (LVM) mirroring provides the ability to allocate more than one copy of a physical partition to increase the availability of the data. When a disk fails and its physical partitions become unavailable, you still have access to mirrored data on an available disk. The LVM performs mirroring within the logical volume.

Within a PowerHA SystemMirror cluster, you can mirror the following:

- Logical volume data in a shared volume group
- The log logical volume for each shared volume group with file systems

Mirroring physical partitions

To improve the availability of the logical volume, you allocate one, two, or three copies of a physical partition to mirror data that is contained in the partition.

If a copy is lost due to an error, the other undamaged copies are accessed, and the AIX operating system continues processing with an accurate copy. After access is restored to the failed physical partition, AIX resynchronizes the contents (data) of the physical partition with the contents (data) of a consistent mirror copy.

The following figure shows a logical volume composed of two logical partitions with three mirrored copies. In the diagram, each logical partition maps to three physical partitions. Each physical partition should be designated to reside on a separate physical volume within a single volume group. This configuration provides the maximum number of alternative paths to the mirror copies and, therefore, the greatest availability.

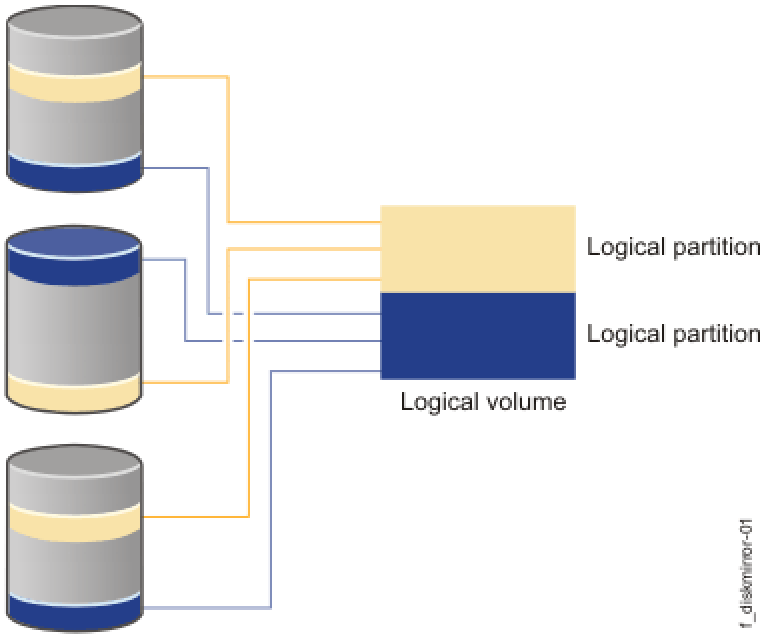


Figure 7. Logical volume of two logical partitions with three mirrored copies

The mirrored copies are transparent, meaning that you cannot isolate one of these copies. For example, if you delete a file from a logical volume with multiple copies, the deleted file is removed from all copies of the logical volume.

The following configurations increase data availability:

- Allocating three copies of a logical partition rather than allocating one or two copies
- Allocating the copies of a logical partition on different physical volumes rather than allocating the copies on the same physical volume
- Allocating the copies of a logical partition across different physical disk enclosures instead of the same enclosure, if possible
- Allocating the copies of a logical partition across different disk adapters. rather than using a single disk adapter

Although using mirrored copies spanning multiple disks (on separate power supplies) together with multiple disk adapters ensures that no disk is a single point of failure for your cluster, these configurations might increase the time for write operations.

Specify the **superstrict** disk allocation policy for the logical volumes in volume groups for which forced varyon is specified. This configuration:

- Guarantees that copies of a logical volume always reside on separate disks
- Increases the chances that forced varyon will be successful after a failure of one or more disks.

If you plan to use forced varyon for the logical volume, apply the **superstrict** disk allocation policy for disk enclosures in the cluster.

For more information about forced varyon, see the section Using quorum and varyon to increase data availability.

Related reference

Using quorum and varyon to increase data availability

How you configure quorum and varyon for volume groups can increase the availability of mirrored data.

Mirroring journal logs

Nonconcurrent access configurations support journaled file systems and enhanced journaled file systems.

The AIX operating system uses journaling for its file systems. In general, this means that the internal state of a file system at startup (in terms of the block list and free list) is the same state as at shutdown. In practical terms, this means that when AIX starts up, the extent of any file corruption can be no worse than at shutdown.

Each volume group contains a **jfslog** or **jfs2log** log, which is itself a logical volume. This log typically resides on a different physical disk in the volume group than the journaled file system. However, if access to that disk is lost, changes to file systems after that point are in jeopardy.

To avoid the possibility of that physical disk being a single point of failure, you can specify mirrored copies of each **jfslog** or **jfs2log** log. Place these copies on separate physical volumes.

Planning for LVM split-site mirroring

You can set up disks located at two or three different locations for remote Logical Volume Manager (LVM) mirroring by using a storage area network (SAN). For example, split-site mirroring uses LVM to replicate data between the disk subsystem at each different location for disaster recovery.

Note: PowerHA SystemMirror only supports two site configurations.

A SAN is a high-speed network that allows your environment to establish direct connections between storage devices and systems (nodes). Thus, two or more systems located at different locations can access the same physical disks by using a SAN network connection. You can combine remote disks into a volume group using LVM. You can import this volume group to the nodes located at different locations.

The logical volumes in the volume group that contain the remote disks can have up to three remote mirrors. You can set up at least one remote mirror at each location. The data stored in the logical volume is highly available. Therefore, in case of a failure in your environment, for example all nodes are not available at one location, the remote mirror at another location has the latest data.

PowerHA SystemMirror automatically synchronizes all remote mirrors after a disk or node failure occurs, and the nodes are brought back online. Automatic synchronization happens even if one of the disks is in the PVREMOVED state or the PVMISSING state. Automatic synchronization is not available for all cases of LVM split-site mirroring. If it is not available, you can use C-SPOC to synchronize the data.

When planning for an LVM split-site mirroring configuration, you must also plan for the repository disk used in the cluster. You must verify that there is a second disk ready to be used as a repository disk when the primary disk fails.

Note: Cluster Aware AIX supports live repository replacement without impacting critical cluster functions.

Example

The following figure is a configuration example of LVM split-site mirroring using a SAN.

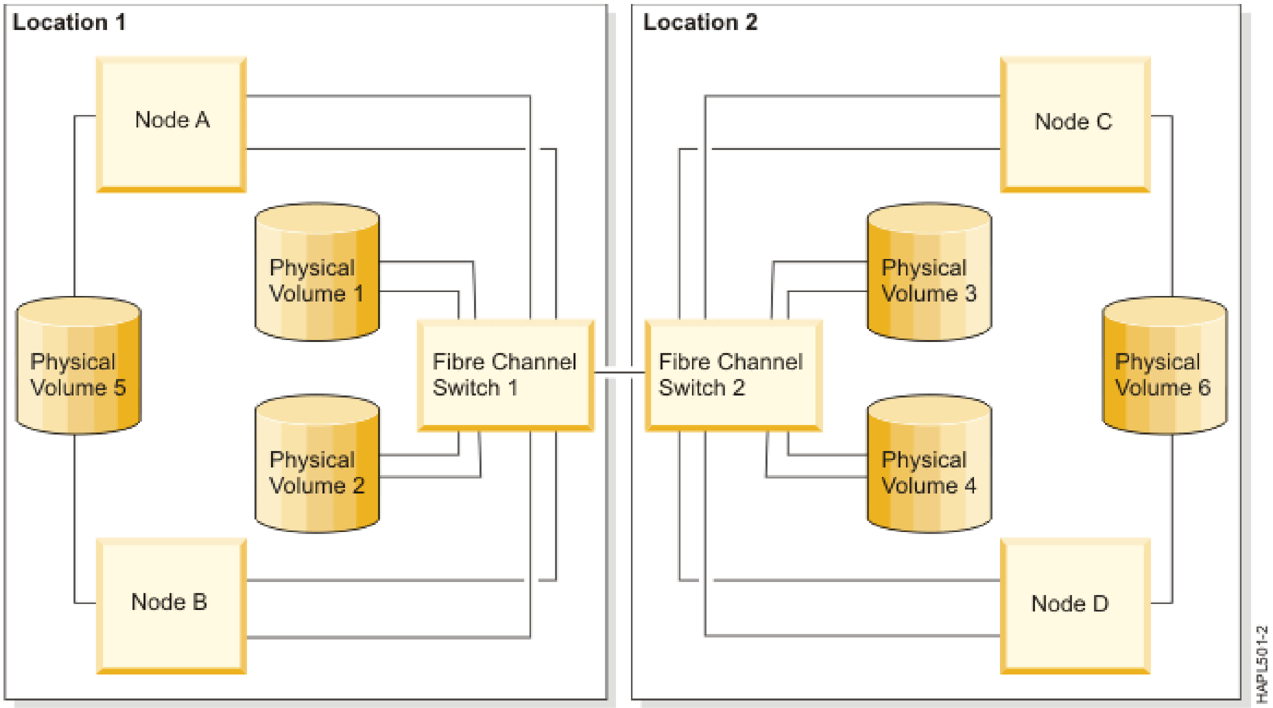


Figure 8. LVM split-site mirroring configuration using a SAN

You can mirror the disks that are connected to at least one node at each of the two locations. In this example, PV4 is available for Node A and Node B on Location 1 and Node C on Loaction 2 using the Fibre Channel Switch 1 and Fibre Channel Switch 2 connection. You can have a mirror of PV4 on Location 1. The disks that are connected to the nodes on only one location (PV5 and PV6) cannot be mirrored across locations.

You can use the AIX LVM mirrored pools function to ensure that data is correctly and completely mirrored between the two locations. If PV1 and PV2 are in one mirrored pool, and PV3 and PV4 are in a separate mirrored pool, then LVM allows one complete copy of the data to be present at each location. You must use superstrict mirrored pools to guarantee that a complete copy of the data is at each location.

In this example, you can use mirrored pools to help maintain mirroring even as disks are added, removed, or replaced in a volume group. You can also use the C-SPOC function to define mirrored pools and associate them with the disks at each location.

Related information

Managing LVM split site mirroring

Mirroring across sites

You can setup disks that are located at two different sites to use remote Logical Volume Manager (LVM) mirroring, by using a storage area network (SAN). Data is replicated between the disk subsystem at each site for disaster recovery.

A *SAN* is a high-speed network that allows the establishment of direct connections between storage devices and processors . Thus, two or more nodes located at different sites can access the same physical disks, which can be separated by some distance, through the common SAN. These remote disks can be combined into a volume group by using LVM, and this volume group can be imported to the nodes that are located at different sites. The logical volumes in this volume group can have up to three mirrors. Thus, you can set up at least one mirror at each site. The information stored on this logical volume is kept highly available, and in case of certain failures, the remote mirror at another site will still have the latest information so that the operations can be continued on the other site.

PowerHA SystemMirror automatically synchronizes mirrors after a disk or node failure and subsequent reintegration. PowerHA SystemMirror handles the automatic mirror synchronization even if one of the disks is in the PVREMOVED or PVMISSING state. The automatic synchronization is not possible for all cases, but you can use C-SPOC to synchronize the data from the surviving mirrors to stale mirrors after a disk or site failure, and subsequent reintegration.

Note: In PowerHA SystemMirror Enterprise Edition, you can also use mirroring in a cluster that spans two sites by using the Geographic Logical Volume Manager (GLVM) mirroring function.

Planning for disk access

You can configure disks to have either enhanced concurrent access or nonconcurrent access.

- **Enhanced concurrent access**. The data on the disks is available to all connected nodes concurrently and all the nodes have access to the metadata on the disks. This access mode allows for fast disk takeover, because the volume group can be brought online before the metadata is read.

  All shared volume groups must be configured as enhanced concurrent mode volume groups, whether or not they will be concurrently accessed. You can migrate existing volume groups to enhanced concurrent mode.

  Using journaled file system (JFS) and the enhanced journaled file system (JFS2) is not supported if you configure the corresponding enhanced volume group (in the Resource Group policy) to be concurrently accessed from two or more nodes.

  Concurrent access configurations that use IBM TotalStorage™ DS Series or IBM 2105 Enterprise Storage Servers do not use logical volume manager (LVM) mirroring. Instead, these systems provide their own data redundancy.

- **Nonconcurrent access**. Only one node at a time can access information on the disks.

  If the resource group containing those disks moves to another node, the new node can then access the disks, read the metadata (information about the current state of the volume groups and other components), run the **varyon** command on the volume groups, and mount any associated file systems.

  Nonconcurrent access configurations typically use journaled file systems. In some cases, a database application running in a nonconcurrent environment might bypass the journaled file system and access the raw logical volume directly.

Related reference

Enhanced concurrent access
Any disk supported by PowerHA SystemMirror for attachment to multiple nodes must be placed in an enhanced concurrent mode volume group, and can be used in either concurrent or nonconcurrent environments (as specified by the type of resource group)

Enhanced concurrent access

Any disk supported by PowerHA SystemMirror for attachment to multiple nodes must be placed in an enhanced concurrent mode volume group, and can be used in either concurrent or nonconcurrent environments (as specified by the type of resource group)

- Concurrent. An application runs on all active cluster nodes at the same time.

  To allow such applications to access their data, concurrent volume groups are varied on for all active cluster nodes. The application has the responsibility to ensure consistent data access.

- Nonconcurrent. An application runs on one node at a time.

  The volume groups are not concurrently accessed. They are still accessed by only one node at any given time.

When you varyon the volume group in enhanced concurrent mode on all nodes that own the resource group in a cluster, the LVM allows access to the volume group on all nodes. However, it restricts the higher-level connections, such as NFS mounts and JFS mounts, on all nodes, and allows them only on the node that currently owns the volume group in PowerHA SystemMirror.

You can use the AIX MPIO function to access disk subsystems through multiple paths. Multiple paths provide more throughput and higher availability than the use of a single path. In particular, when multiple paths are used, failure of a single path due to an adapter, or a cable or switch failure will not cause applications to lose access to data. While PowerHA SystemMirror will attempt to recover from complete loss of access to a volume group, that loss itself is going to be temporarily disruptive. The AIX MPIO function can prevent a single component failure from causing an application outage.

When fast disk takeover is used, the disk reservation function is not used. If the cluster becomes partitioned, nodes in each partition could accidentally varyon the volume group in active state. Because active state varyon of the volume group allows mounting of file systems and changing physical volumes, this situation can result in different copies of the same volume group. For more information about fast disk takeover and using multiple networks, see the section Using fast disk takeover.

Concurrent access requirements for MPIO accessed disks

Enhanced concurrent mode is the only option for creating concurrent volume groups. In PowerHA SystemMirror, enhanced concurrent mode volume groups do not use disk reserves. The concurrent access that is required for MPIO accessed disks is automatically provided in PowerHA SystemMirror.

About enhanced concurrent mode

All concurrent volume groups are created as enhanced concurrent mode volume groups by default. For enhanced concurrent volume groups, the Concurrent Logical Volume Manager (CLVM) coordinates changes between nodes through the Group Services component of the Reliable Scalable Cluster Technology (RSCT) function in the AIX operating system. Group Services protocols flow over the communications links between the cluster nodes.

Related tasks

Converting volume groups to enhanced concurrent mode

Related reference

Using fast disk takeover
PowerHA SystemMirror automatically detects failed volume groups and initiates a fast disk takeover for enhanced concurrent mode volume groups that are included as resources in nonconcurrent resource groups.

Using fast disk takeover

PowerHA SystemMirror automatically detects failed volume groups and initiates a fast disk takeover for enhanced concurrent mode volume groups that are included as resources in nonconcurrent resource groups.

Fast disk takeover is especially useful for fallover of enhanced concurrent mode volume groups made up of a large number of disks. This disk takeover mechanism is faster than disk takeover used for standard volume groups included in nonconcurrent resource groups. During fast disk takeover, PowerHA SystemMirror skips the extra processing needed to break the disk reserves, or update and synchronize the logical volume manager (LVM) information by running lazy update.

Fast disk takeover has been observed to take no more than 10 seconds for a volume group with two disks. This time is expected to increase very slowly for larger numbers of disks and volume groups. The actual time observed in any configuration depends on factors outside of PowerHA SystemMirror control, such as the processing power of the nodes and the amount of unrelated activity at the time of the fallover. The actual time observed for completion of fallover processing depends on additional factors, such as whether or not a file system check is required, and the amount of time needed to restart the application.

Note: Enhanced concurrent mode volume groups are not concurrently accessed. They are only accessed by one node at any given time. The fast disk takeover mechanism works at the volume group level, and is thus independent of the number of disks used.

Fast disk takeover and active and passive varyon

An enhanced concurrent volume group can be made active on a node, or varied on as either active or passive.

To enable fast disk takeover, PowerHA SystemMirror activates enhanced concurrent volume groups in the active and passive states.

Active varyon

Active varyon behaves the same as ordinary varyon, and makes the logical volumes available. When an enhanced concurrent volume group is varied on in active state on a node:

- Operations on file systems, such as file system mounts
- Operations on applications
- Operations on logical volumes, such as creating logical volumes
- Synchronizing volume groups

Passive varyon

When an enhanced concurrent volume group is varied on in passive state, the LVM provides the equivalent of disk fencing for the volume group at the LVM level.

Passive state varyon allows only a limited number of read-only operations on the volume group:

- LVM read-only access to the volume group's special file
- LVM read-only access to the first 4 Kb of all logical volumes that are owned by the volume group.

The following operations are not allowed when a volume group is varied on in passive state:

- Operations on file systems, such as file systems mounting
- Any operations on logical volumes, such as having logical volumes open
- Synchronizing volume groups

PowerHA SystemMirror and active and passive varyon

PowerHA SystemMirror correctly varies on the volume group in active state on the node that owns the resource group, and changes active and passive states appropriately as the state and location of the resource group changes.

- Upon cluster startup:
  - On the node that owns the resource group, PowerHA SystemMirror activates the volume group in active state. PowerHA SystemMirror activates a volume group in active state only on one node at a time.
  - PowerHA SystemMirror activates the volume group in passive state on all other nodes in the cluster.
- Upon fallover:
  - If a node releases a resource group or if the resource group is being moved to another node for any other reason, PowerHA SystemMirror switches the varyon state for the volume group from active to passive on the node that releases the resource group. PowerHA SystemMirror then activates the volume group in active state on the node that acquires the resource group.
  - The volume group remains in passive state on all other nodes in the cluster.
- PowerHA SystemMirror does the following processes when node reintegration occurs:
  - Changes the varyon state of the volume group from active to passive on the node that releases the resource group.
  - Varies on the volume group in active state on the joining node .
  - Activates this volume group in passive state on all other nodes in the cluster.

Note: The switch between active and passive states is necessary to prevent mounting file systems on more than one node at a time.

Using quorum and varyon to increase data availability

How you configure quorum and varyon for volume groups can increase the availability of mirrored data.

Using quorum

Quorum ensures that more than half of the physical disks in a volume group are available.

Quorum does not keep track of logical volume mirrors, and is therefore not a useful way to ensure data availability. You can lose quorum when you still have all your data. Conversely, you can lose access to some of your data, and not lose quorum.

Quorum is beneficial for volume groups on RAID arrays, such as the ESS and IBM TotalStorage DS Series. The RAID device provides data availability and recovery from loss of a single disk. Mirroring is typically not used for volume groups contained entirely within a single RAID device. If a volume group is mirrored between RAID devices, forced varyon can bring a volume group online despite loss of one of the RAID devices.

Decide whether to enable or disable quorum for each volume group. The following table shows how quorum affects when volume groups varyon and off:

	Condition for volume group to varyon	**Condition for volume group to vary off**
Quorum enabled	More than 50% of the disks in the volume group are available	Access is lost to 50% or more of the disks
Quorum disabled	All of the disks in the volume group are available	Access is lost to all of the disks

Quorum checking is enabled by default. You can disable quorum by using the **`chvg -Qn vgname`** command, or by using the **smit chvg** fastpath.

Related information

chvg command

Quorum in concurrent access configurations

Quorum must be enabled for a PowerHA SystemMirror concurrent access configuration. Disabling quorum could result in data corruption. Any concurrent access configuration where multiple failures could result in no common shared disk between cluster nodes has the potential for data corruption or inconsistency.

The following figure shows a cluster with two sets of IBM disk subsystems configured for no single point of failure. The logical volumes are mirrored across subsystems and each disk subsystem is connected to each node with separate NICs.

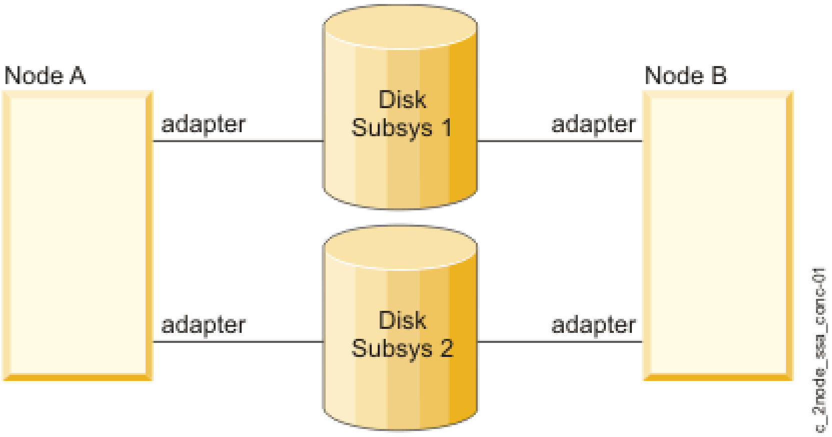


Figure 9. IBM disk subsystem concurrent access configuration

If multiple failures result in a communications loss between each node and one set of disks in such a way that node A can access subsystem 1 but not subsystem 2, and node B can access subsystem 2 but not subsystem 1. Both nodes continue to operate on the same baseline of data from the mirrored copy they can access. However, each node does not see modifications made by the other node to data on disk. As a result, the data becomes inconsistent between nodes.

With quorum protection enabled, the communications failure results in one or both nodes varying off the volume group. Although an application does not have access to data on the volume group that is varied off, data consistency is preserved.

Selective fallover triggered by loss of quorum

PowerHA SystemMirror selectively provides recovery for nonconcurrent resource groups (with the startup policy not Online on All Available Nodes) that are affected by failures of specific resources. PowerHA SystemMirror automatically reacts to an LVM_SA_QUORCLOSE loss-of-quorum error associated with a volume group going offline on a cluster node. In response to this error, a nonconcurrent resource group goes offline on the node where the error occurred.

If the AIX Logical Volume Manager takes a volume group in the resource group offline due to a loss of quorum for the volume group on the node, PowerHA SystemMirror selectively moves the resource group to another node. You can change this default behavior by customizing resource recovery to use a notify method instead of fallover.

PowerHA SystemMirror launches selective failover and recovers the affected resource groups in response to an LVM_SA_QUORCLOSE error. This error is generated by AIX LVM for specific error conditions, even if the volume group is not defined as quorum enabled. AIX LVM might also generate other types of error notifications; however, PowerHA SystemMirror doe not react to these by default. In these cases, you still need to configure customized error notification methods, or use AIX automatic error notification methods to react to volume group failures.

You can use the rootvg system event to monitor the loss of access to rootvg. If the system loses access, PowerHA SystemMirror logs an event in the system error log and reboots the system by default. You can change this setting using SMIT to log an event but not reboot the system.

You can monitor the rootvg events only when the rootvg disk is using the native AIX Multipath I/O (MPIO) driver and the rootvg disk is not an internal parallel SCSI disk. To verify whether the rootvg disk is using the MPIO driver, on the command line, type `lspath -l` *hdiskname*, where *hdiskname* is the name of the rootvg disk. If the rootvg disk is not using the MPIO driver, the following error message is displayed:

```
lspath: 0514-538 Cannot perform the requested function because the
        specified device does not support multiple paths.
```

Related information

Error notification method used for volume group loss

PowerHA SystemMirror monitoring system events

Selective fallover for handling resource groups

Using forced varyon

PowerHA SystemMirror provides a forced varyon function to use in conjunction with AIX automatic error notification methods. The forced varyon function enables you to have the highest possible data availability.

Forcing a varyon of a volume group lets you keep a volume group online as long as there is one valid copy of the data is available. Use a forced varyon only for volume groups that have mirrored logical volumes.

Note: Use caution when using this function to avoid creating a partitioned cluster.

You can use SMIT to force a varyon of a volume group on a node if the normal **varyon** command fails on that volume group due to a lack of quorum but with one valid copy of the data available. Using SMIT to force a varyon is useful for local disaster recovery, when data is mirrored between two disk enclosures and one of the disk enclosures becomes unavailable.

Note: You can specify a forced varyon attribute for volume groups on SCSI disks that use logical volume manager (LVM) mirroring, and for volume groups that are mirrored between separate RAID or ESS devices.

If you want to force the volume group to varyon when disks are unavailable, use **varyonvg -f**, which forces the volume group to varyon, whether or not there are copies of your data. You can specify forced varyon in SMIT for volume groups in a resource group.

Forced varyon and cluster partitioning

If you are using forced varyon, it is important that multiple network connections between nodes that shares the storage exists. Multiple network connections help to ensure that each node always has a communication path to the other nodes, even if one network fails. Having multiple network connections prevents your cluster from becoming partitioned. Otherwise, a network failure might cause nodes to attempt to take over resource groups that are still active on other nodes. In this situation, if you have set a forced varyon setting, you might experience data loss or divergence.

Using NFS with PowerHA SystemMirror

The PowerHA SystemMirror software provides availability enhancements to Network File System (NFS) handling.

These enhancements include:

- Reliable NFS server capability that allows a backup processor to recover current NFS activity should the primary NFS server fail, preserving the locks on NFS file systems and the duplicate request cache. This functionality is restricted to two-node Resource Groups if it contains NFS version 2 or version 3 exports. Resource groups with only NFS version 4, or later, can support up to 16-node configurations.
- NFS Configuration Assist to ease the setup and configuration.
- Preconfigured application controller and application monitor (clam_nfsv4) to monitor NFS version 4 exports and the NFS daemons.
- Ability to specify a network for NFS mounting.
- Ability to define NFS exports and mounts at the directory level.
- Ability to specify export options for NFS-exported directories and file systems.

For NFS to work as expected on a PowerHA SystemMirror cluster, there are specific configuration requirements. Therefore, you must plan for the following tasks:

- Creating shared volume groups
- Exporting NFS file systems
- NFS mounting and fallover

The PowerHA SystemMirror scripts handles default NFS behavior. You might need to modify the scripts to handle your particular configuration.

You can configure NFS in all resource groups that behave as nonconcurrent; that is, they do not have an Online on All Available Nodes startup policy.

Relinquishing control over NFS file systems in a PowerHA SystemMirror cluster

After you configure resource groups that contain NFS file systems, you relinquish control over NFS file systems to PowerHA SystemMirror.

After NFS file systems become part of resource groups that belong to an active PowerHA SystemMirror cluster, PowerHA SystemMirror takes care of cross-mounting and unmounting the file systems during cluster events (such as fallover of a resource group that contains the file system to another node in the cluster).

If for some reason you stop the cluster services and must manage the NFS file systems manually, the file systems must be unmounted before you restart the cluster services. This enables management of NFS file systems by PowerHA SystemMirror after the nodes join the cluster.

Reliable NFS server capability

A PowerHA SystemMirror cluster can take advantage of AIX extensions to the standard NFS functions that enable it to handle duplicate requests correctly and restore lock state during NFS server fallover and reintegration.

When NFS clients use NFS locking to arbitrate access to the shared NFS file system, there is a limit of two nodes per resource group. Each resource group that uses reliable NFS contains one pair of PowerHA SystemMirror nodes.

Independent pairs of nodes in the cluster can provide Reliable NFS services. For example, in a four-node cluster, you can set up two NFS client and server pairs (for example, Node A and Node B provides one set of Reliable NFS services, and Node C and NodeD can provide another set of Reliable NFS services.) Pair 1 can provide reliable NFS services for one set of NFS file systems, and pair 2 can provide reliable NFS services to another set of NFS file systems. This is true whether or not NFS cross-mounting is configured. PowerHA SystemMirror does not impose a limit to the number of resource groups or NFS file systems as long as the nodes participating in the resource groups follow the constraints outlined in this example..

Specifying an IP address for NFS

The host name must be able to be resolved to an IP address that is always present on the node and always active on an interface. The host name cannot be a service IP address that might move to another node.

To ensure that the IP address that is going to be used by NFS always resides on the node, you can:

- Use an IP address that is associated with a persistent label
- For an IPAT via aliases configuration, use the IP address used at boot time
- Use an IP address that resides on an interface that is not controlled by PowerHA SystemMirror

Shared volume groups

When creating shared volume groups, typically you can leave the **Major Number** field blank and let the system provide a default. However, NFS uses volume group major numbers to help uniquely identify exported file systems. Therefore, all nodes to be included in a resource group containing an NFS-exported file system must have the same major number for the volume group on which the file system resides.

In the event of node failure, NFS clients attached to a PowerHA SystemMirror cluster operate the same way as when a standard NFS server fails and reboots. Accesses to the file systems hang and then recover when the file systems become available again. However, if the major numbers are not the same, when another cluster node takes over the file system and re-exports the file system, the client application will not recover. The client application will not recover because the file system exported by the node will appear to be different from the one exported by the failed node.

Exporting NFS file systems and directories

The process of exporting NFS file systems and directories in PowerHA SystemMirror is different from that in the AIX operating system.

Keep in mind the following points when planning for exporting NFS file systems and directories in PowerHA SystemMirror:

- NFS file systems and directories to export:

  In AIX, you specify the NFS file systems and directories to export by using the **smit mknfsexp** command (which creates the **/etc/exports** file). In PowerHA SystemMirror, you specify NFS file systems and directories to export by including them in a resource group in PowerHA SystemMirror.
- Export options for NFS exported file systems and directories:

  If you want to specify special options for exporting NFS in PowerHA SystemMirror, you can create a **/usr/es/sbin/cluster/etc/exports** file. This file has the same format as the regular AIX **/etc/exports** file.

  Note: Using this alternative exports file is optional. PowerHA SystemMirror checks the **/usr/es/sbin/cluster/etc/exports** file when exporting an NFS file system or directory. If there is an entry for the file system or directory in this file, PowerHA SystemMirror uses the options listed. If the NFS file system or directory for export is not listed in the file or if the alternative file does not exist, the file system or directory is exported from NFS with the default option of root access for all cluster nodes.
- A resource group that specifies file systems to export:

  In SMIT, set the **Filesystems Mounted before IP Configured** field for the resource group to **true**. Setting the **Filesystems Mounted before IP Configured** field to **true** ensures that IP address takeover is performed after exporting the file systems. If the IP addresses were managed first, the NFS server would reject client requests until the file systems had been exported.
- Stable storage for NFS version 4 exports:

  Use stable storage for NFS version 4 exports and have it accessible to all the participating nodes of the resource group. NFS version 4 uses this file system space to store the state information related to the NFS client transactions. The state information in the stable storage is crucial for the smooth fallover, fallback, and move options of the resource group from one node to other node, while keeping the NFS version 4 client's state unaffected.

  While the resource group is online, the location of the stable storage cannot be changed.

Related reference

NFS cross-mounting in PowerHA SystemMirror
An NFS cross-mounting is a specific PowerHA SystemMirror NFS configuration where each node in the cluster can act as both the NFS server and the NFS client. While a file system is being exported from one node, the file system is mounted with NFS on all the nodes for the resource group, including the one that is exporting it. Another file system can also be exported from another node, and be mounted with NFS on all nodes.

NFS and fallover

For PowerHA SystemMirror and NFS to work properly together, the IP address for the NFS server must be configured as a resource in a resource group for high availability.

To ensure the best NFS performance, NFS file systems used by PowerHA SystemMirror should include the entry `vers = <version number>` in the **options** field in the **/etc/filesystems** file.

Related reference

NFS cross-mounting and IP labels

To enable NFS cross-mounting, each cluster node might act as an NFS client. Each of these nodes must have a valid route to the service IP label of the NFS server node. That is, to enable NFS cross-mounting, an IP label must exist on the client nodes, and this IP label must be configured on the same subnet as the service IP label of the NFS server node.

NFS cross-mounting in PowerHA SystemMirror

An NFS cross-mounting is a specific PowerHA SystemMirror NFS configuration where each node in the cluster can act as both the NFS server and the NFS client. While a file system is being exported from one node, the file system is mounted with NFS on all the nodes for the resource group, including the one that is exporting it. Another file system can also be exported from another node, and be mounted with NFS on all nodes.

When your environment uses NFS version 2 and version 3 to export in resource groups, the cross-mount capability is restricted to only two node resource groups. If the resource group contains only NFS version 4 (or later) exports, the cross-mount capability is extended up to any number of nodes that are supporting a resource group.

Each node in the resource group is part of a mutual takeover (or active-active) cluster configuration, providing and mounting an NFS file system.

By default, resource groups that contain exported NFS file systems automatically cross-mount these file systems (if both export and import are configured) as follows:

- On the node currently hosting the resource group, all NFS file systems in the group are NFS exported.
- Each node that might host this resource group NFS mounts all the NFS file systems in the resource group.

Applications access the NFS file systems on any node that is part of the resource group.

When a fallover occurs for a resource group that is configured with IP address takeover, the NFS file system is locally mounted by the takeover node and re-exported. All other nodes in the resource group maintain their NFS file system mount.

In an NFS cross-mount configuration, any NFS mount that is defined in a resource group must have a corresponding NFS export in a resource group. If your NFS cross-mounts do not follow this configuration, the following message is displayed:

```
claddres: WARNING: NFS mounts were specified for the resource group
'<RG name>';however no NFS exports have been specified.
```

If an NFS cross-mount field contains a value, the corresponding NFS exported file system must also contain a value.

NFS cross-mounting and IP labels

To enable NFS cross-mounting, each cluster node might act as an NFS client. Each of these nodes must have a valid route to the service IP label of the NFS server node. That is, to enable NFS cross-mounting, an IP label must exist on the client nodes, and this IP label must be configured on the same subnet as the service IP label of the NFS server node.

If the NFS client nodes have service IP labels on the same network, this is not an issue. However, in certain cluster configurations, you need to create a valid route.

Ways to create a route to the NFS server

The easiest way to ensure access to the NFS server is to have an IP label on the client node that is on the same subnet as the service IP label of the NFS server node.

To create a valid route between the NFS client node and the node that is exporting the file system, you can configure your environment in either of the following ways:

- A separate NIC with an IP label configured on the service IP network and subnet
- A persistent node IP label on the service IP network and subnet.

Be aware that these solutions do not provide automatic root permissions to the file systems because of the export options for NFS file systems that are set in PowerHA SystemMirror by default.

To enable root level access to NFS-mounted file systems on the client node, add all of the node's IP labels or addresses to the `root =` option in the cluster exports file: **/usr/es/sbin/cluster/etc/exports**. You can do this on one node, because synchronizing the cluster resources propagates this information to the other cluster nodes.

Related reference

Exporting NFS file systems and directories
The process of exporting NFS file systems and directories in PowerHA SystemMirror is different from that in the AIX operating system.

Creating and configuring stable storage for NFS version 4 exports
Stable storage is a file system space that is used to save the state information by the NFS version 4 server. This is very crucial for maintaining NFS version 4 client's state information to facilitate smooth and transparent fallover/fallback/move of the Resource group from one node to another.

Requirements of stable storage:

- The suggested size is 512 MB.
- It is suggested that you have a dedicated file system for the stable storage, but a subdirectory of an existing file system is acceptable.
- The file system and volume group where the Stable Storage resides should be part of the resource group.
- PowerHA SystemMirror makes a best-effort attempt to verify that the path specified for stable storage belongs to a file system in the resource group; however, these checks are not able to take symbolic links into account since the file systems might not be mounted locally when the verification occurs. Avoid using symbolic links that can interfere with verification in PowerHA SystemMirror.
- Ensure that the stable storage directory is empty (free of any prior data) prior to adding it to the resource group.
- While stable storage must be stored in file systems managed by the resource group, it should not be stored in an NSF directory that is exported by the resource group.
- Configuration assistant offers an AUTO_SELECT option. If you select this option, PowerHA SystemMirror uses a volume group from the list of volume groups that are part of the given resource group. PowerHA SystemMirror then creates a logical volume and a file system to use as the stable storage location.

Example: Two-node NFS cross-mounting configuration

The this example, Node A currently hosts a nonconcurrent resource group, RG1, which includes `/fs1` as an exported NFS file system and `service1` as a service IP label.

In this example, Node B currently hosts a nonconcurrent resource group, RG2, which includes `/fs2` as an exported NFS file system and `service2` as a service IP label. On reintegration, `/fs1` is passed back to Node A, locally mounted, and exported. Node B mounts it over NFS again.

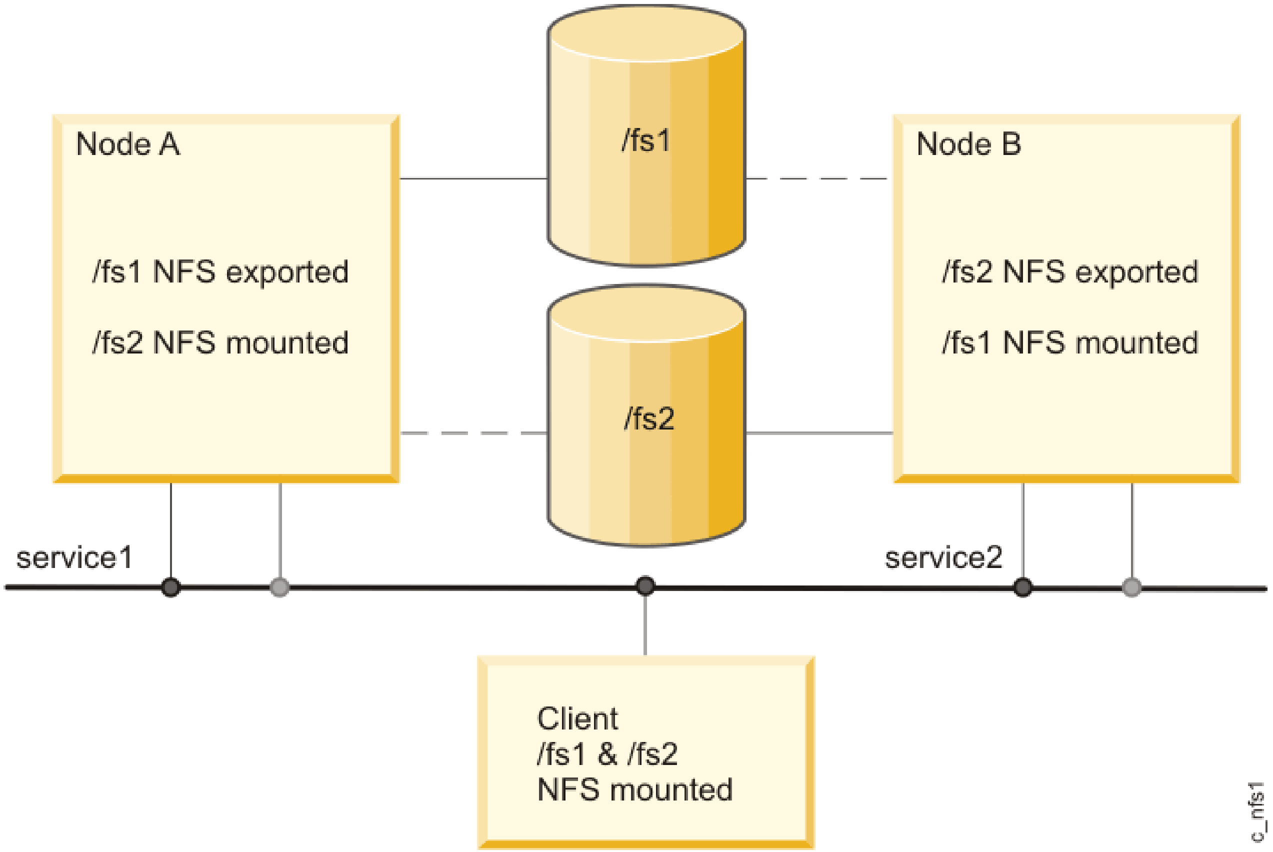


Figure 10. Two node NFS cross-mounting

The two resource groups would be defined in SMIT as follows.

Resource group	**RG1**	**RG2**
Participating node names	`Node A Node B`	`Node B Node A`
File systems The file systems to be locally mounted by the node currently owning the resource group.	`/fs1`	`/fs2`
File systems to export The file system to NFS export by the node currently owning the resource group. The file system is a subset of the file system listed previously.	`/fs1`	`/fs2`
File systems to NFS mount The file systems and directories to be NFS mounted by all nodes in the resource group. The first value is NFS mount point. The second value is the local mount point.	`/mnt1;/fs1`	`/mnt2;/fs2`
File systems mounted before IP configured	`true`	`true`

In this scenario:

- Node A locally mounts and exports `/fs1`, then over-mounts on `/mnt1`.
- Node B NFS-mounts `/fs1,` on `/mnt1` from Node A.

Setting up a resource group like this ensures the expected default node-to-node NFS behavior.

When Node A fails, Node B closes any open files in Node A: `/fs1`, unmounts it, mounts it locally, and re-exports it to waiting clients.

After takeover, Node B has:

- `/fs2` locally mounted
- `/fs2` NFS-exported
- `/fs1` locally mounted
- `/fs1` NFS-exported
- `service1:/fs1` NFS mounted over `/mnt1`
- `service2:/fs2` NFS mounted over `/mnt2b`

Both resource groups contain both nodes as possible owners of the resource groups.

Resource group takeover with cross-mounted NFS file systems

This section describes how to set up nonconcurrent resource groups with cross-mounted NFS file systems so that NFS file systems are handled correctly during takeover and reintegration. In addition, nonconcurrent resource groups support automatic NFS mounting across servers during fallover.

Setting up NFS mount point different from local mount point

PowerHA SystemMirror handles NFS mounting in nonconcurrent resource groups in a couple of ways.

These include:

- The node that currently owns the resource group mounts the file system over the file system's local mount point, and this node exports the NFS file system.
- All the nodes in the resource group (including the current owner of the group) mount the NFS file system over a different mount point.

Therefore, the owner of the group has the file system mounted twice. One file system is mounted as a local mount and the other is an NFS mount.

Note: The NFS mount point must be outside the directory tree of the local mount point.

Because IPAT is used in resource groups that have NFS-mounted file systems, the nodes will not unmount and remount NFS file systems during a fallover. When the resource group falls over to a new node, the acquiring node locally mounts the file system and NFS exports it. (The NFS-mounted file system is temporarily unavailable to cluster nodes during fallover.) As soon as the new node acquires the IPAT label, access to the NFS file system is restored.

All applications must refers to the file system through the NFS-mounted file system. If the applications used must always reference the file system by the same mount point name, you can change the mount point for the local file system mount (for example, change it to mount point_local and use the previous local mount point as the new NFS mount point).

Default NFS mount options for PowerHA SystemMirror

The default options used by PowerHA SystemMirror when performing NFS mounts are `hard`, `intr`.

To set soft mounts or any other options on the NFS mounts:

1. Enter `smit mknfsmnt`.
2. In the **MOUNT now, add entry to /etc/filesystems or both?** field, select the **file systems** option.
3. In the **/etc/filesystems entry will mount the directory on system RESTART** field, accept the default value of **no**.

   This procedure adds the options you have chosen to the **/etc/filesystems**entry created. The PowerHA SystemMirror scripts then read this entry to use any options you might have selected.

Creating and configuring NFS mount points on clients

An NFS mount point is required to mount a file system using NFS. In a nonconcurrent resource group all the nodes in the resource group mount the NFS file system. You create an NFS mount point on each node in the resource group. The NFS mount point must be outside the directory tree of the local mount point.

After the NFS mount point is created on all nodes in the resource group, configure the **NFS Filesystem to NFS Mount** attribute for the resource group.

To create NFS mount points and to configure the resource group for the NFS mount:

1. On each node in the resource group, create an NFS mount point by executing the following command:

   ```
   mkdir /mount point
   ```

   where *mount point* is the name of the local NFS mount point over which the remote file system is mounted.
2. In the **Change/Show Resources and Attributes for a Resource Group** SMIT panel, the **Filesystem to NFS Mount** field must specify both mount points.

   Specify the nfs mount point and the local mount point, separating the two with a semicolon. For example:

   ```
   /nfspoint;/localpoint
   ```

   If there are more entries, separate them with a space:

   ```
   /nfspoint1;/local1 /nfspoint2;/local2
   ```
3. Optional: If there are nested mount points, nest the NFS mount points in the same manner as the local mount points so that they match correctly.
4. Optional: When cross-mounting NFS file systems, set the **Filesystems Mounted before IP Configured** field in SMIT for the resource group to **true**.

Planning resource groups

These topics describe how to plan resource groups within a PowerHA SystemMirror cluster.

Overview for resource groups

PowerHA SystemMirror organizes resources into resource groups. Each resource group is handled as a unit that contains shared resources such as IP labels, applications, file systems, and volume groups. You define the policies for each resource group that define when and how it will be acquired or released.

In Initial cluster planning, you made preliminary choices about the resource group policies and the takeover priority for each node in the resource group node lists. In this section you do the following:

- Identify the individual resources that constitute each resource group.
- For each resource group, identify which type of group it is: concurrent or nonconcurrent.
- Define the participating node list for the resource groups. The node list consists of the nodes assigned to participate in the takeover of a given resource group.
- Identify the resource group startup, fallover, and fallback policy.
- Identify applications and their resource groups for which you want to set up location dependencies, parent-child dependencies, or both.
- Identify the intersite management policies of the resource groups. Are there replicated resources to consider?
- Identify other attributes and runtime policies to refine resource group behavior.

The following terms are used in this section:

- *Participating node list*. A list of nodes that can host a particular resource group, as defined in the participating node names for a resource group in SMIT.

  Be aware that the combination of the different resource group policies and the current cluster conditions also affect the resource group placement on the nodes in the cluster.
- *Home node (or the highest priority node for this resource group)*. The first node that is listed in the participating node list for any nonconcurrent resource groups.

PowerHA SystemMirror resource groups support NFS file systems.

Related reference

Initial cluster planning
This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

NFS cross-mounting and IP labels
To enable NFS cross-mounting, each cluster node might act as an NFS client. Each of these nodes must have a valid route to the service IP label of the NFS server node. That is, to enable NFS cross-mounting, an IP label must exist on the client nodes, and this IP label must be configured on the same subnet as the service IP label of the NFS server node.

General rules for resources and resource groups

There are some general rules and restrictions for resources and resource groups.

The following rules and restrictions apply to resources and resource groups:

- In order for PowerHA SystemMirror to keep a cluster resource highly available, it must be part of a resource group. If you want a resource to be kept separate, define a group for that resource alone. A resource group can have one or more resources defined.
- A resource cannot be included in more than one resource group.
- The components of a resource group must be unique. Put the application along with the resources it requires in the same resource group.
- The service IP labels, volume groups, and resource group names must be both unique within the cluster and distinct from each other. The name of a resource should relate to the application it serves, as well as to any corresponding device, such as websphere_service_address.
- If you include the same node in participating node lists for more than one resource group, make sure that the node has the memory, and network interfaces, necessary to manage all resource groups simultaneously.

Types of resource groups: Concurrent and nonconcurrent

To categorize and describe resource group behavior, you must first divide the resource groups into two types: concurrent and nonconcurrent.

Concurrent resource groups

A concurrent resource group can be online on multiple nodes. All nodes in the node list of the resource group acquire that resource group when they join the cluster. There are no priorities among nodes. Concurrent resource groups can be configured to run on all nodes in the cluster.

The only resources included in a concurrent resource group are volume groups with raw logical volumes, raw disks, and application controllers that use the disks. The device on which these logical storage entities are defined must support concurrent access.

Concurrent resource groups have the Online on All Available Nodes startup policy and do not fallover or fallback from one node to another.

Nonconcurrent resource groups

Nonconcurrent resource groups cannot be online on multiple nodes. You can define a variety of startup, fallover, and fallback policies for these resource groups.

You can fine-tune the nonconcurrent resource group behavior for node preferences during a node startup, resource group fallover to another node in the case of a node failure, or when the resource group falls back to the reintegrating node.

Related reference

Resource group attributes for startup, fallover, and fallback
Each attribute affects resource group startup, resource group fallover to another node in the case of a node failure, or resource group fallback to the reintegrating node.

Resource group policies for startup, fallover, and fallback

Resource group behaviors are separated into three kinds of node policies.

These policies are:

- *Startup* policy defines on which node the resource group will be activated when a node joins the cluster and the resource group is not active on any node.
- *Fallover* policy defines to which node the resource group will fall over when the resource group must leave the node where it is currently online due to a failure condition (or if you stop the cluster services on a node using the fallover option).
- *Fallback* policy defines to which node the resource group will fall back when a node joins and the resource group is already active on another node.

PowerHA SystemMirror allows you to configure only valid combinations of startup, fallover, and fallback behaviors for resource groups. The following table summarizes the basic startup, fallover, and fallback behaviors you can configure for resource groups in PowerHA SystemMirror.

Startup behavior	Fallover behavior	Fallback behavior
Online only on home node (first node in the node list)	Any of these: • Fallover to next priority node in the list • Fallover using dynamic node priority	Any of these: • Never fall back • Fall back to higher priority node in the list
Online using node distribution policy	Any of these: • Fallover to next priority node in the list • Fallover using dynamic node priority	Never fall back
Online on first available node	Any of these: • Fallover to next priority node in the list • Fallover using dynamic node priority	Any of these: • Never fall back • Fall back to higher priority node in the list
Online on all available nodes	Bring offline (on error node only)	Never fall back

In addition to the node policies described in the previous table, other issues might determine the resource groups that a node acquires.

Related reference

Planning for cluster events
These topics describe the PowerHA SystemMirror cluster events.

Resource group attributes

This section provides an overview of the resource group attributes that you can use to fine-tune the startup, fallover, and fallback policies of resource groups.

Resource group attributes for startup, fallover, and fallback

Each attribute affects resource group startup, resource group fallover to another node in the case of a node failure, or resource group fallback to the reintegrating node.

The following table summarizes which resource group startup, fallover, or fallback policies are affected by a specific attribute or runtime policy. Not every resource group that is available is not listed in the following table.

Attribute	Startup policy	Fallover policy	Fallback policy
Settling time	X		
Node distribution policy	X		
Dynamic node priority		X	
Delayed fallback timer			X
Resource groups parent and child dependency	X	X	X
Resource Groups Location Dependencies	X	X	X

Related reference

Parent and child dependent resource groups
Related applications in different resource groups are configured to be processed in logical order.

Resource-group location dependencies
Certain applications in different resource groups stay online together on a node or stay online on different nodes.

Related information

PowerHA SystemMirror resources and resource groups

Settling time for startup

You can modify a resource group's startup behavior by specifying a settling time for a resource group that is currently offline and has a startup policy of Online on First Available Node.

With a settling time is specified, you can avoid having a resource group activated on the first available node when multiple nodes are starting cluster services at the same time. Also, a higher priority node for the resource group might join the cluster during this time period.

Settling time lets the cluster manager wait for a specified amount of time before activating a resource group. Use this attribute to ensure that a resource group does not bounce among nodes, while nodes with increasing priority for the resource group are brought online.

If the node that is starting is the first node in the node list for this resource group, the settling time period is skipped and PowerHA SystemMirror immediately attempts to acquire the resource group on this node.

The settling time has the following characteristics:

- Affects only those resource groups that are currently offline, and for which you have specified the startup policy to be Online on First Available Node. You configure one settling time for all such resource groups.
- Activates when the first node that can acquire the resource group joins the cluster, unless this is the first node in the node list (then the settling time is ignored and the group is acquired).

  If the first node that joins the cluster and can potentially acquire the resource group fails, this condition can either cancel the settling time, or reset it.
- Delays the activation of the group during **node_up** events, in case higher priority nodes join the cluster.

Note: If a settling time period is specified for a resource group and a resource group is currently in the ERROR state, the cluster manager waits for the settling time period before attempting to bring the resource group online during a **node_up** event.

Node reintegration with settling time configured

In general, when a node joins the cluster, it can acquire resource groups. The following list describes the role of the settling time in this process:

- If the node is the highest priority node for a specific resource group, the node immediately acquires that resource group and the settling time is ignored. This is the only one circumstance under which PowerHA SystemMirror ignores the setting.
- If the node is able to acquire some resource groups, but is not the highest priority node for those groups, the resource groups are not acquired on that node. Instead, they wait during the settling time interval to see whether a higher priority node joins the cluster.

When the settling time interval ends, PowerHA SystemMirror moves the resource group to the highest priority node that is currently available and that can take the resource group. If PowerHA SystemMirror does not find appropriate nodes, the resource group remains offline.

Node distribution policy

You can configure a startup behavior of a resource group to use the node distribution policy during startup. This policy ensures that only one resource group that has this policy enabled is acquired on a node during startup.

You can use *node distribution policy* for cluster startup to ensure that PowerHA SystemMirror activates only one resource group that has this policy enabled on each node. This policy helps you distribute your CPU-intensive applications on different nodes.

The facts about node distribution policy are:

- If two resource groups with this policy enabled are offline at the time when a particular node joins, only one of the two resource groups is acquired on a node. PowerHA SystemMirror gives preference to the resource group that has fewer nodes in the node list and then sorts the list of resource groups based on Resource group ID’s.
- If one of the resource groups is a parent resource group (has a child resource group), PowerHA SystemMirror gives preference to the parent resource group, and it is activated on a node.
- To ensure that your resource groups are distributed not only at startup but also for recovery events (fallover and fallback), use location dependencies.

Related reference

Resource group dependencies
PowerHA SystemMirror offers a wide variety of configurations where you can specify the relationships between resource groups that you want to maintain at startup, fallover, and fallback.

Dynamic node priority policy

You can configure a resource group's fallover behavior to use dynamic node priority. It allows you to use a predefined Reliable Scalable Cluster Technology (RSCT) resource variable such as **lowest**

CPU load, to select the takeover node, or a user defined dynamic node priority variable, such as **cl_highest_udscript_rc** to select the takeover node.

Setting a dynamic node priority policy allows you to use a predefined Resource Monitoring and Control (RMC) resource variable, such as **lowest CPU load**, to select the takeover node. With a dynamic priority policy enabled, the order of the takeover node list is determined by the state of the cluster at the time of the event, as measured by the selected RMC resource variable. You can set different policies for different groups or the same policy for several groups.

Another option is to use a user-defined dynamic node priority variable such as **cl_highest_udscript_rc**. If you use this option, you must provide a script and execution timeout value, which is invoked on all candidate failover nodes at the time of the event. Return values are collected from each and the takeover node is selected based on the return values of the script and the selected dynamic node priority variable.

If you decide to define dynamic node priority policies using RMC resource variables to determine the fallover node for a resource group, consider the following points:

- Dynamic node priority policy is most useful in a cluster where all the nodes have equal processing power and memory.
- Dynamic node priority policy is irrelevant for clusters of fewer than three nodes.
- Dynamic node priority policy is irrelevant for concurrent resource groups.
- Dynamic node priority policy is not supported when sites are configured.

Remember that selecting a takeover node also depends on such conditions as the availability of a network interface on that node.

Delayed fallback timer

You can configure a resource group's fallback behavior to occur at one of the predefined recurring times (daily, weekly, monthly and yearly, or on a specific date and time) by specifying and assigning a delayed fallback timer.

You can use a *delayed fallback timer* to set the time for a resource group to fall back to a higher priority node. You can configure the fallback behavior for a resource group to occur at a predefined recurring time (daily, weekly, monthly, or a specific date).

The delayed fallback timer has the following characteristics:

- Specifies the time when a resource group that is online and residing on a nonhome or low priority node falls back its home node or a higher priority node.
- Affects the movement of the resource group to another node. For example, if you move the nonconcurrent resource group (that has a fallback timer attribute) to another node by using the resource group management (RGM) utility (**clRGmove**), the group stays on the destination node (unless you reboot the cluster, which is rarely done). If the destination node goes down and then is reintegrated, the resource group also falls back to this node at the specified time.

Node reintegration with a delayed fallback timer set

The resource group does not immediately fall back to its higher priority node under the following condition.

- You have configured a delayed fallback timer for a resource group.
- A higher priority node joins the cluster.

  At the time specified in the Delayed Fallback Timer attribute, one of two scenarios takes place:
- *A higher priority node is found*. If a higher priority node is available for the resource group, PowerHA SystemMirror attempts to move the resource group to this node when the fallback timer expires. If the acquisition is successful, the resource group is acquired on that node.

  However, if the acquisition of the resource group on the node fails, PowerHA SystemMirror attempts to move the resource group to the next higher priority node in the group node list, and so on. If the acquisition of the resource group on the last node that is available fails, the resource group goes into an error state. You must take action to fix the error and bring such a resource group back online.

- *A higher priority node is not found*. If there are no higher priority nodes available for a resource group, the resource group remains online on the same node until the fallback timer expires again. for example, if a daily fallback timer expires at 11:00 p.m. and there are no higher priority nodes available for the resource group to fallback on, the fallback timer recurs the next night at 11:00 p.m.

  A fallback timer that is set to a specific date does not recur.

Resource group dependencies

PowerHA SystemMirror offers a wide variety of configurations where you can specify the relationships between resource groups that you want to maintain at startup, fallover, and fallback.

You can configure the following resource groups dependencies:

- Parent and child dependent resource groups. Related applications and other resources in different resource groups are configured to be processed in the correct order.
- Start after dependencies. Specify that a resource group can only start after another resource group or groups is active in the cluster.
- Stop after dependencies. Specify that a resource group can only be stopped after another resource group or groups is brought offline.
- Resource group location dependencies. Certain applications in different resource groups stay online together on a node or stay online on different nodes.

Keep the following points in mind when planning how to configure these dependencies:

- Although by default all resource groups are processed in parallel, PowerHA SystemMirror processes dependent resource groups according to the order dictated by the dependency, and not necessarily in parallel. Resource group dependencies are honored cluster wide and override any customization for serial order of processing of any resource groups included in the dependency.
- Dependencies between resource groups offer a predictable and reliable way of building clusters with multitiered applications.

The following limitations apply to configurations that combine dependencies. Verification will fail if you do not have only one resource group belonging to an Online on Same Node dependency set and an Online On Different Nodes dependency set at the same time.

Parent and child dependent resource groups

Related applications in different resource groups are configured to be processed in logical order.

Configuring a resource group dependency allows for better control for clusters with multitiered applications where one application depends on the successful startup of another application, and both applications are required to be kept highly available with PowerHA SystemMirror.

The following example illustrates the parent-child dependency behavior:

- If resource group A depends on resource group B, resource group B must be brought online before resource group A is acquired on any node in the cluster. Note that resource group A is defined as a *child resource group*, and resource group B as a *parent resource group*.
- If child resource group A depends on parent resource group B, during a node startup or node reintegration, child resource group A cannot come online before parent resource group B gets online. If parent resource group B is taken offline, the child resource group A is taken offline first, since it depends on resource group B.

  Business configurations that use multitiered applications can utilize parent-child dependent resource groups. For example, the database must be online before the application controller. In this case, if the database is moved to a different node, the resource group containing the application controller would have to be brought down and back up on any node in the cluster.

If a child resource group contains an application that depends on resources in the parent resource group and the parent resource group falls over to another node, the child resource group is temporarily stopped and automatically restarted. Similarly, if the child resource group is concurrent, PowerHA SystemMirror takes it offline temporarily on all nodes, and brings it back online on all available nodes. If the fallover of the parent resource group is not successful, both the parent and the child resource groups go into an ERROR state.

Consider the following when planning for parent-child dependent resource groups:

- Plan applications you need to keep highly available and consider whether your business environment requires one application to be running before another application can be started.
- Ensure that those applications that require sequencing are included in different resource groups. This way, you can establish dependencies between these resource groups.
- Plan for application monitors for each application that you are planning to include in a child or parent resource group. For an application in a parent resource group, configure a monitor in the **monitoring startup** mode.

  To minimize the chance of data loss during the application stop and restart process, customize your application controller scripts to ensure that any uncommitted data is stored to a shared disk temporarily during the application stop process and read back to the application during the application restart process. It is important to use a shared disk as the application may be restarted on a node other than the one on which it was stopped.

Related reference

Planning considerations for multitiered applications
Business configurations that use multitiered applications can use parent and child dependent resource groups. For example, the database must be online before the application controller. In this case, if the database goes down and is moved to a different node the resource group containing the application controller would have to be brought down and back up on any node in the cluster.

Resource-group location dependencies

Certain applications in different resource groups stay online together on a node or stay online on different nodes.

If failures do occur over the course of time, PowerHA SystemMirror distributes resource groups so that they remain available, but not necessarily on the nodes you originally specified, unless they have the same home node and the same fallover and fallback policies.

Resource-group location dependency offers you an explicit way to specify that certain resource groups will always be online on the same node, or that certain resource groups will always be online on different nodes. You can combine these location policies with parent-child dependencies or start after dependencies and stop after dependencies to have all child or source resource groups online on the same node while the parent or target is online on a different node. You can also have all child or source resource groups online on different nodes for better performance.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

If you have replicated resources, you can combine resource groups into a site dependency to keep them online at the same site.

PowerHA SystemMirror supports the following types of resource-group location dependencies between resource groups:

- **Online on same node**

  The following rules and restrictions apply to the Online on same node dependency set of resource groups. Verification will fail if you do not follow these guidelines:

  - All resource groups configured as part of a given same node dependency set must have the same node list (the same nodes in the same order).

  - All nonconcurrent resource groups in the same node dependency set must have the same startup, fallover, and fallback policies.
  - Online Using Node Distribution Policy is not allowed for startup.
  - If a Dynamic Node Priority Policy is configured as a Fallover Policy, all resource groups in the set must have the same policy.
  - If one resource group has a fallback timer configured, it applies to the set of resource groups. All resource groups in the set must have the same fallback time setting.
  - Both concurrent and nonconcurrent resource groups can be included.
  - You can have more than one same node dependency set in the cluster.
  - PowerHA SystemMirror enforces the condition that all resource groups in the same node dependency set that are active (ONLINE) are required to be on the same node. Some resource groups in the set can be offline or in the error state.
  - If one or more resource groups in the same node dependency set fail, PowerHA SystemMirror tries to place all resource groups in the set on the node that can host all resource groups that are currently ONLINE (the ones that are still active) plus one or more failed resource groups.
- **Online on same site**

  The following rules and restrictions are applicable to the Online on same site dependency set of resource group. Verification will fail if you do not follow these guidelines:

  - All resource groups in a same site dependency set must have the same Intersite Management policy but might have different startup policies, fallover policies, and fallback policies. If fallback timers are used, these must be identical for all resource groups in the set.
  - The fallback timer does not apply to moving a resource group across site boundaries.
  - All resource groups in the same site dependency set must be configured, so that the nodes that can own the resource groups are assigned to the same primary and secondary sites.
  - Both concurrent and nonconcurrent resource groups can be included.
  - You can have more than one same site dependency set in the cluster.
  - All resource groups in the same site dependency set that are active (ONLINE) are required to be ONLINE on the same site, even though some resource groups in the set might be OFFLINE or in the ERROR state.
  - If you add a resource group that is included in a same node dependency set to a same site dependency set, all the other resource groups in the same node dependency set must be added to the same site dependency set.
- **Online on different nodes**

  The following rules and restrictions apply to the online on different nodes dependency set of resource groups. Verification will fail if you do not follow these guidelines:

  - Only one online on different nodes dependency set is allowed per cluster.
  - Plan startup policies so that each resource group in the set will start up on a different node.
  - If a parent dependency or child dependency is specified, the child resource group cannot have a higher priority than its parent resource group.

    After the cluster is running with these groups configured, be aware that:

  - If a resource group with high priority is online on a node, then no other less priority resource group in the different nodes dependency set can come online on that node.
  - If a resource group with a higher priority falls over or falls back to a given node, the resource group with the higher priority will come online and the cluster manager takes the less priority resource group offline and moves it to another node if this is possible.
  - Resource groups with the same priority cannot come online (startup) on the same node. Priority of a resource group for a node within the same priority level is determined by the group's alphabetical order in the set.

– Resource groups with the same priority do not cause one another to be moved from the node after a fallover or fallback.

Related reference

Special considerations for using sites
The following information is about special considerations if a resource group has dependencies specified.

Start after dependencies

In start after dependencies, the target resource group must be online on any node in the cluster before a source (dependent) resource group can be activated on a node. There is no dependency when releasing resource groups and the groups are released in parallel.

The following are guidelines and limitations for start after dependencies.

- A resource group can serve as both a target and a source resource group, depending on which end of a given dependency link it is placed.
- You can specify three levels of dependencies for resource groups.
- You cannot specify circular dependencies between resource groups.
- This dependency applies only at the time of resource group acquisition. There is no dependency between these resource groups during resource group release.
- A source resource group cannot be acquired on a node until its target resource group is fully functional. If the target resource group does not become fully functional, the source resource group goes into an offline due to target offline state. If you notice that a resource group is in this state, you might need to troubleshoot which resources might need to be brought online manually to resolve the resource group dependency.
- When a resource group in a target role falls over from one node to another, there will be no effect on the resource groups that depend on it.
- After the source resource group is online, any operation (bring offline, move resource group) on the target resource group does not effect the source resource group.
- If the target resource group is offline, you cannot manually move a resource group or bring a resource group online on the source resource group.

Note: You should configure several application monitors, especially a monitor that checks the application startup for the application that is included in the target resource groups. This process verifies that the application in the target resource group starts successfully.

Stop after dependencies

In stop after dependencies, the target resource group must be offline on any node in the cluster before a source (dependent) resource group can be brought offline on a node. There is no dependency when acquiring resource groups and the groups are acquired in parallel.

The following are limitations and guidelines for stop after dependencies.

- A resource group can serve as both a target and a source resource group, depending on which end of a given dependency link it is placed.
- You can specify three levels of dependencies for resource groups.
- You cannot specify circular dependencies between resource groups.
- This dependency applies only at the time a resource group is released. There is no dependency between these resource groups during resource group acquisition.
- A source resource group cannot be released on a node until its target resource group is offline.
- When a resource group in a source role falls over from one node to another, first the target resource group is released and then the source resource group is released. After that, both resource groups are acquired in parallel, assuming that there is no start after or parent-child dependency between these resource groups.
- If the target resource group is offline, you cannot manually move a resource group or bring a resource group offline on the source resource group.

Moving resource groups to another node

When PowerHA SystemMirror moves the resource group to another node, you have some options.

These options include:

- Resource groups remain on the nodes to which they are moved.

  You can move resource groups that have the Never Fallback policy to another node. When you do so, you can tell PowerHA SystemMirror to leave the resource group on the destination node until you decide to move the group again.
- When you move a resource group with the **RG_move** command, it remains on the node to which it was moved either indefinitely (until you tell PowerHA SystemMirror to move it to another node) or until you reboot the cluster.

  If you do stop the cluster services, which rarely has to be done, and you want to permanently change the resource group's node list and highest priority node, change the resource group's attributes and restart the cluster.
- If you take the resource group online or offline on any of the nodes, it remains online or offline either until the next cluster reboot, or until you manually bring the group online elsewhere in the cluster.
- If your resource group has the fallback to highest priority node policy, the group falls back to its destination node, after you move it.

  For instance, if the group has node A configured as its highest priority node and you move it to node B, then this group will remain on node B and will treat this node now as its highest priority node. You can always choose to move the group again to node A. When you use SMIT to do so, PowerHA SystemMirror informs you as to whether the original highest priority node (node A) is now available to host the group.

  You can keep track of all resource groups that were manually moved by using the **clRGinfo** command.

Moving resource groups by using the clRGmove command

You can use the **clRGmove** command to move a resource group to another node, or to take a resource group online or offline. You can run the **clRGmove** command by using SMIT or from the command line.

If you use **clRGmove** with resource groups that have the Never Fallback fallback policy, the resource group remains on that node until you move it elsewhere.

The following paragraphs describe the rules which apply when using **clRGmove** to manage resource groups with different policies.

Moving parent-child dependent resource groups

- If the parent resource groups are offline due to your request made through the **clRGmove** command, PowerHA SystemMirror rejects manual attempts to bring the child resource groups that depend on these resource groups online. The error message lists the parent resource groups that must be brought online first.
- If you have a parent and a child resource group online, and would like to move the parent resource group to another node or take it offline, PowerHA SystemMirror prevents you from doing so before a child resource group is taken offline.

Moving start after dependent resource groups

In this type of dependency, the target resource group must be online on any node in the cluster before a source (dependent) resource group can be activated on a node. The following rules apply to resource groups with a start after dependency:

- If the target resource groups are offline due to your request made through the **`clRGmove`** command, PowerHA SystemMirror rejects manual attempts to bring the source resource groups that depend on these resource groups online. The error message lists the target resource groups that must be brought online first.

Moving stop after dependent resource groups

In this type of dependency, the target resource group must be offline on any node in the cluster before a source (dependent) resource group can be brought offline on a node. The following rules apply to resource groups with a stop after dependency:

- If you have a target and a source resource group online, and would like to move the source resource group to another node or take it offline, PowerHA SystemMirror prevents you from doing so before a target resource group is taken offline.

Moving location-dependent resource groups

- If you move a same-site dependent resource group to the other site, the entire set of resource groups in the dependency is moved to the other site.
- If you move a same-node dependent resource group to another node, the entire set of resource groups in the dependency is moved.
- You cannot move a resource group to any node that hosts an online resource group that is part of a different-node dependency. You must first take offline the resource group that is included in the different node dependency on the selected node.

Planning cluster networks and resource groups

Most nonconcurrent resource groups use service IP labels to provide access to the application to clients over the network. PowerHA SystemMirror uses IP address takeover (IPAT) via IP aliasing to keep these service addresses available in the cluster.

IPAT does not apply to concurrent resource groups or online on all available nodes resource groups.

It is important to consider how clients reach application addresses over the cluster network, when there are firewalls or VPNs that are configured in the environment.

Related reference

Planning for IP address takeover via IP aliases
Assigning IP aliases to NICs allows you to create more than one IP label on the same network interface.

Planning parallel or serial order for processing resource groups

By default, PowerHA SystemMirror acquires and releases all individual resources configured in your cluster in parallel. However, you can specify a specific serial order according to which some or all of the individual resource groups are acquired or released.

The following processes are completed during acquisition:

1. PowerHA SystemMirror acquires the resource groups serially in the order that you specified in the list.
2. PowerHA SystemMirror acquires the remaining resource groups in parallel.

During the release of resource groups, the process is reversed:

1. PowerHA SystemMirror releases the resource groups for which you did not define a specific serial order in parallel.
2. The remaining resource groups in the cluster are processed in the order that you specified for these resource groups in the list.
3. If you upgraded your cluster from a previous version of PowerHA SystemMirror, for more information about which processing order is used in this case, see Upgrading a PowerHA SystemMirror cluster.

   Note: Even if you specify the order of resource group processing on a single node, the actual fallover of the resource groups may be triggered by different policies. Therefore, it is not guaranteed that resource groups are processed cluster-wide in the order specified because the serial customized processing order of resource groups applies to their processing on a particular node only.

4. When resource groups are processed in parallel, fewer cluster events occur in the cluster. In particular, events such as **node_up_local** or **get_disk_vg_fs** do not occur if resource groups are processed in parallel.
5. As a result, using parallel processing reduces the number of particular cluster events for which you can create customized pre-event or post-event scripts. If you start using parallel processing for some of the resource groups in your configuration, be aware that your existing pre-event or post-event scripts might not work for these resource groups.
6. Parallel and serial processing of resource groups is reflected in the event summaries in the **hacmp.out** file.

   For more information about how to configure customized serial acquisition and release order of resource groups, see Configuring processing order for resource groups.

Dependent resource groups and parallel or serial order

Although by default PowerHA SystemMirror processes resource groups in parallel, if you establish dependencies between some of the resource groups in the cluster, processing may take longer than it does for clusters without dependent resource groups as there may be more processing to do to handle one or more **rg_move** events

Upon acquisition, first the parent or higher priority resource groups are acquired, then the child resource groups are acquired. Upon release, the order is reversed. The remaining resource groups in the cluster (those that do not have dependencies themselves) are processed in parallel.

Also, if you specify serial order or processing and have dependent resource groups configured, make sure that the serial order does not contradict the dependency specified. The resource groups dependency overrides any serial order in the cluster.

Related reference

Planning for cluster events
These topics describe the PowerHA SystemMirror cluster events.

Related information

Configuring processing order for resource groups

Upgrading a PowerHA SystemMirror cluster

Planning resource groups in clusters that have sites

The combination of intersite management policy and the node startup, fallover policies, and fallback policies that you select determines the resource group startup, fallover, and fallback behavior.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

Site support in PowerHA SystemMirror allows a variety of resource group configurations.

Concurrent resource groups and sites

You can use the following policies for concurrent resource groups:

Policy type	Description
Intersite management policy	Online on Both Sites Online on either site Prefer primary site Ignore
Startup Policy	Online on all available nodes

Policy type	Description
Fallover Policy	Bring offline (on Error node only)
Fallback Policy	Never fallback

Nonconcurrent resource groups and sites

For nonconcurrent resource groups, you can use the following policies:

Policy type	Description
Intersite management policy	Online on either site Prefer primary site Ignore
Startup Policy	Online on home node Online on first available node Online using node distribution policy
Fallover Policy	Fallover to next priority node (in the node list)
Fallback Policy	Fallback to higher priority node (in the node list) Never fallback

General resource group behavior in clusters with sites

Nonconcurrent resource groups defined with an intersite management policy of prefer primary site or online on either site have two instances when the cluster is running.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

The following instance are running in nonconcurrent resource groups:

- A primary instance on a node at the primary site
- A secondary instance on a node at the secondary site

The **clRGinfo** command shows these instances as:

- ONLINE
- ONLINE SECONDARY

Concurrent resource groups (online on all nodes) with an intersite management policy of online on both sites have multiple ONLINE instances and no ONLINE SECONDARY instances when the cluster is running at both locations.

Concurrent resource groups with an intersite management policy of prefer primary site or online on either site have primary instances on each node at the primary site and secondary instances on nodes at the secondary site.

Resource groups with replicated resources are processed in parallel on a **node_up** event, according to their site management and node startup policies, taking any dependencies into account. When resource groups fall over between sites, the secondary instances are acquired and brought ONLINE SECONDARY on the highest priority node that is available at the new backup site. More than one secondary instance can be on the same node. The primary instance of the resource group is acquired and brought ONLINE on the highest priority node that is available to host this resource group on the new active site. This order of processing ensures that the backup site is ready to receive backup data when the primary instance starts.

If the secondary instance cannot be brought to the ONLINE SECONDARY state, the primary instance will still be brought ONLINE, if possible.

Special considerations for using sites

The following information is about special considerations if a resource group has dependencies specified.

Dependent resource groups and sites

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

You can specify a dependency between two or more resource groups that reside on nodes, which are located at different sites. In this case, if either the parent or child moves to the other site, the dependent group moves also. If the parent group cannot be activated on the fallover site, the child resource group will also remain inactive.

The dependency applies only to the state of the primary instance of the resource group. If the parent group's primary instance is OFFLINE, and the secondary instance is ONLINE SECONDARY on a node, the child group's primary instance will be OFFLINE.

During resource group recovery, resource groups can fall over to nodes on either site. The sequence for acquiring dependent resource groups is the same as that of clusters without sites, in which the parent resource group is acquired first, and then the child resource group is acquired. The release logic is reversed when the child resource group is released before a parent resource group is released.

If you have sites that are defined in clusters without sites, you must configure application monitoring for applications included in resource groups with dependencies.

Related reference

Resource-group location dependencies
Certain applications in different resource groups stay online together on a node or stay online on different nodes.

Examples: Resource group behavior in clusters that have sites

The fallback policy applies to the ONLINE and ONLINE SECONDARY instances of the resource group.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

The intersite management policy for a resource group determines the fallback behavior of the ONLINE instances of the resource groups between sites, which governs the location of the secondary instance.

The ONLINE SECONDARY instance is located at the site that does not have the ONLINE instance. The following table shows the expected behavior of resource groups during site events according to the startup and intersite management policies.

Note: Most of the restrictions in the following table apply only when multiple interfaces per node are on the same network. When only one interface per node is on the network, which can be common when using ether the Fibre Channel or virtual Ethernet, most of these restrictions do not apply.

Nodestartup policy (applies within site)	**Intersite management policy**	**Startup, fallover, or fallback behavior**
Online on home node only	Prefer primary site	**Cluster Startup** **Primary site** The home node acquires the resource group in the ONLINE state. Nonhome nodes leave the resource group. **Secondary site** The first node that joins this site acquires the resource group in ONLINE SECONDARY state. **Intersite fallover** The ONLINE instance falls between sites when no nodes at the local site can acquire the resource group. The secondary instance moves to the other site and is brought to the ONLINE SECONDARY state on the highest priority node that is available, if possible. **Intersite fallback** The ONLINE instance falls back to the primary site when a node from the primary site joins. The secondary instance moves to the other site and is brought to the ONLINE SECONDARY state on the highest priority node that is available, if possible.
Online on first available node or Online using node distribution policy	Prefer primary site	**Cluster Startup** **Primary site** The node that joins first from the primary site, and meets the criteria, acquires the resource group in the ONLINE state. The resource group is OFFLINE on all other nodes at the primary site. The node distribution policy applies only to the primary instance of the resource group. **Secondary site** The first node to join the cluster in this site acquires all secondary instances of resource groups with this startup policy in the ONLINE_SECONDARY state (no distribution). **Intersite fallover** The ONLINE instance falls between sites when no nodes at the local site can acquire the resource group. The secondary instance moves to the other site and is brought to the ONLINE SECONDARY state on the highest priority node that is available, if possible. **Intersite fallback** The ONLINE instance falls back to the primary site when a node from the primary site joins. The secondary instance moves to the other site and is brought to the ONLINE SECONDARY state on the highest priority node that is available, if possible.

Nodestartup policy (applies within site)	Intersite management policy	Startup, fallover, or fallback behavior
Online on all available nodes	Prefer primary site	**Cluster Startup** **Primary site** All nodes acquire the resource group in the ONLINE state. **Secondary site** All nodes acquire the resource group in the ONLINE_SECONDARY state. **Intersite fallover** The ONLINE instances fall between sites when all nodes at the local site go OFFLINE or fail to start the resource group. The secondary instances move to the other site and are brought to the ONLINE SECONDARY state where possible. **Intersite fallback** ONLINE instances fall back to the primary site when a node on the primary site rejoins. Nodes at the secondary site acquire the resource group in the ONLINE_SECONDARY state.
Online on home node only	Online on either site	**Cluster Startup** **Primary site** The home node that joins the cluster (from either site) acquires the resource group in the ONLINE state. Nonhome nodes leave the resource group OFFLINE. **Secondary site** The first node to join from the other site acquires the resource group in the ONLINE_SECONDARY state. **Intersite fallover** The ONLINE instance falls between sites when no nodes at the local site can acquire the resource group. The secondary instance moves to the other site and is brought to the ONLINE_SECONDARY state on the highest priority node available, if possible. **Intersite fallback** The ONLINE instance does not fall back to the primary site when a node on the primary site rejoins. The highest priority rejoining node acquires the resource group in the ONLINE_SECONDARY state.

Nodestartup policy (applies within site)	Intersite management policy	Startup, fallover, or fallback behavior
Online on first available node or Online using node distribution policy	Online on either site	**Cluster Startup** **Primary site** The node that joins first from either site, that meets the distribution criteria, acquires the resource group in the ONLINE state. **Secondary site** After the resource group is ONLINE, the first joining node from the other site acquires the resource group in the ONLINE_SECONDARY state. **Intersite fallover** The ONLINE instance falls between sites when no nodes at the local site can acquire the resource group. **Intersite fallback** The ONLINE instance does not fall back to the primary site when the primary site joins. A rejoining node acquires the resource group in the ONLINE_SECONDARY state.
Online on all available nodes	Online on either site	**Cluster Startup** **Primary site** The node that joins first from either site acquires the resource group in the ONLINE state. After an instance of the group is active, the remaining nodes in the same site also activate the group in the ONLINE state. **Secondary site** All nodes acquire the resource group in the ONLINE_SECONDARY state. **Intersite fallover** The ONLINE instance falls between sites when all nodes at the local site go OFFLINE or fail to start the resource group. **Intersite fallback** The ONLINE instance does not fall back to the primary site when the primary site joins. Rejoining nodes acquire the resource group in the ONLINE_SECONDARY state.
Online on all available nodes	Online at both sites	**Cluster Startup** All nodes at both sites activate the resource group in the ONLINE state. **Intersite fallover** No fallover occurs. Resource group is either in the OFFLINE state or in the ERROR state. **Intersite fallback** No fallback occurs.

Customizing intersite resource group recovery

Intersite resource group recovery can be configured for automatic processing or failure notification only.

A particular instance of a resource group can fall over within one site, but it cannot move between sites. If no nodes are available on the site where the affected instance resides, that instance goes into the ERROR or ERROR_SECONDARY state. It does not stay on the node where it failed. This behavior applies to both primary and secondary instances.

The Cluster Manager moves the resource group if a **node_down** or **node_up** event occurs, even if fallover between sites is disabled. You can also manually move a resource group between sites.

Enabling or disabling fallover between sites

If you migrated from a previous release of PowerHA SystemMirror, you can change the resource group recovery policy to allow the Cluster Manager to move a resource group to another site to avoid having the resource group go into the ERROR state.

Recovery of primary instances of replicated resource groups across sites

When fallover across sites is enabled, PowerHA SystemMirror tries to recover the primary instance of a resource group in situations where an interface that is connected to an intersite network fails or becomes available.

Recovery of secondary instances of replicated resource groups across sites

When fallover across sites is enabled, PowerHA SystemMirror tries to recover the secondary instance and the primary instance of a resource group in these situations:

- If an acquisition failure occurs while the secondary instance of a resource group is being acquired, the Cluster Manager tries to recover the resource group's secondary instance, as it does for the primary instance. If no nodes are available for the acquisition, the resource group's secondary instance goes into the global ERROR_SECONDARY state.
- If the quorum loss is triggered, and the resource group has its secondary instance in the ONLINE state on the affected node, PowerHA SystemMirror tries to recover the secondary instance on another available node.
- If all XD_data networks fail, PowerHA SystemMirror will move all ONLINE resource groups that have GLVM resources to another available node on that site. This function of the primary instance is mirrored to the secondary instance so that secondary instances can be recovered through selective fallover.

Using SMIT to enable or disable inter-site resource group recovery

To enable or disable intersite resource group recovery, complete the following steps:

1. From the command line, enter `smit sysmirror`.
2. From the SMIT interface, select **Custom Cluster Configurations** > **Resources** > **PowerHA SystemMirror Extended Resources Configuration** > **Customize Inter-site Resource Group Recovery**, and press Enter.

Planning for replicated resources

PowerHA SystemMirror offers integrated support for replicated resources.

With PowerHA SystemMirror replicated resources, you can use the following functions:

- Enables you to dynamically reconfigure resource groups that contain replicated resources with the PowerHA SystemMirror Enterprise Edition for AIX and the PowerHA SystemMirror site configurations.
- Consolidates PowerHA SystemMirror Enterprise Edition for AIX verification into standard cluster verification by automatically detecting and calling the installed PowerHA SystemMirror Enterprise Edition for AIX product's verification utilities.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

Configuring replicated resources

If you installed a PowerHA SystemMirror Enterprise Edition for AIX product, you can use the integrated support for the configuring the different replication technologies.

The following replicated resources are supported PowerHA SystemMirror Enterprise Edition for AIX configurations:

- Resource groups with concurrent node policy can have nonconcurrent site management policy.
- Intersite recovery of resource groups containing PowerHA SystemMirror Enterprise Edition for AIX replicated resources is allowed by default for new installations of PowerHA SystemMirror. Configurations that are updated and migrated from previous releases maintain the preexisting behavior. You can configure this behavior to be a **fallover** or **notify** option on cluster-initiated resource group movement. If you select the **notify** option, you need to configure a pre-event or post-event script, or a remote notification method.
- Parent, child, and location dependency configurations for replicated resource groups.
- Node-based resource group distribution startup policy for resource groups with PowerHA SystemMirror sites.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

You cannot configure a resource group to use a nonconcurrent node policy and a concurrent inter-site management policy.

Processing replicated resources

PowerHA SystemMirror event processing automatically supports the configured replicated resources.

Replicated resources include the following functions:

- Whenever possible, PowerHA SystemMirror processes events in parallel. Events are dynamically phased so that PowerHA SystemMirror processes the primary and secondary instances of a resource group in proper order (release_primary, release_secondary, acquire_secondary, acquire_primary).
- PowerHA SystemMirror recovers the secondary instances as well as the primary instances of the replicated resource groups during volume group losses, acquisition failures, and **local_network_down** events if another node or network is available.
- PowerHA SystemMirror is more likely to acquire the secondary instance of a resource group upon site fallover than any other instance.PowerHA SystemMirror can consider all nodes at the secondary site as targets, not just the node that previously hosted the primary instance, which was the case in the previous version of PowerHA SystemMirror.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

Moving resource groups that have replicated resources

You can move the primary instance of a resource group that has replicated resources to another site, and PowerHA SystemMirror will automatically process the secondary instances of the group in the same operation.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

The Cluster Manager uses dynamic event phasing in that it moves the secondary instance from that site to the other site if a node is available to host it. Every attempt is made to maintain the secondary instance in the SECONDARY_ONLINE state. Even if a node at a given site is configured so that it cannot host more than one primary instance, it might host more than one secondary instance to keep them all in the SECONDARY_ONLINE state.

Planning for Workload Manager

IBM offers AIX Workload Manager (WLM) as a system administration resource included with the AIX operating system.

WLM allows you to set targets for and to set limits on CPU, physical memory use, and disk I/O bandwidth for different processes and applications. This provides better control over the use of critical system resources at peak loads. PowerHA SystemMirror allows you to configure WLM classes into PowerHA SystemMirror resource groups so that the starting and stopping of WLM and the active WLM configuration can be under cluster control.

PowerHA SystemMirror does not verify every aspect of your WLM configuration; therefore, it is your responsibility to ensure the integrity of the WLM configuration files. After you add the WLM classes to a PowerHA SystemMirror resource group, the verification utility checks only whether the required WLM classes exist. Therefore, you must understand how WLM works and configure it carefully. Incorrect but acceptable configuration parameters can impede the productivity and availability of your system.

For complete information on how to set up and use Workload Manager, see the IBM *AIX Workload Manager (WLM)* Redbooks® publication.

Workload Manager distributes system resources among processes that request them according to the class they are in. Processes are assigned to specific classes according to class assignment rules. Planning for WLM integration with PowerHA SystemMirror includes two basic steps:

1. Using AIX SMIT panels to define the WLM classes and class assignment rules related to highly available applications.
2. Using PowerHA SystemMirror SMIT panels to establish the association between the WLM configuration and the PowerHA SystemMirror resource groups.

Related information

AIX Workload Manager (WLM) Redbooks

Workload Manager classes

Workload Manager distributes system resources among processes that request them according to the class to which the processes are assigned.

The properties of a class include:

- Name of the class. A unique alphanumeric string no more than 16 characters long.
- Class tier. A number 0 - 9. This number determines the relative importance of a class from most important (tier 0) to least important (tier 9).
- Number of the CPU and physical memory shares. The actual amount of resources allotted to each class depends on the total number of shares in all classes (thus, if two classes are defined on a system, one with two shares of target CPU use and the other with three shares, the first class will receive 2/5 and the second class will receive 3/5 of the CPU time).
- Configuration limits. Minimum and maximum percentage limits of CPU time, physical memory, and disk I/O bandwidth accessible to the process.

You set up class assignment rules that tell WLM how to classify all new processes (as well as those already running at the time of WLM startup) according to their group ID (GID), user ID (UID), and the full path name.

Workload Manager reconfiguration, startup, and shutdown

This section describes the way WLM is reconfigured or started or stopped after you place WLM under the control of PowerHA SystemMirror.

Workload Manager reconfiguration

After WLM classes are added to a PowerHA SystemMirror resource group, then at the time of cluster synchronization on the node, PowerHA SystemMirror reconfigures WLM to use the rules required by the classes associated with the node. In the event of dynamic resource reconfiguration on the node, WLM is reconfigured in accordance with any changes made to WLM classes associated with a resource group.

Workload Manager startup

WLM startup occurs either when the node joins the cluster or when a dynamic reconfiguration of the WLM configuration takes place.

The configuration is node-specific and depends the resource groups in which the node participates. If the node cannot acquire any resource groups associated with WLM classes, WLM is not started.

For all nonconcurrent resource groups that do not have the Online Using Node Distribution startup policy, the startup script determines whether the resource group is running on a primary or on a secondary node and adds the corresponding WLM class assignment rules to the WLM configuration. For all other nonconcurrent resource groups, and for concurrent access resource groups that the node can acquire, the primary WLM class associated with each resource group is placed in the WLM configuration. The corresponding rules are added to the rules table.

Finally, if WLM is currently running and was not started by PowerHA SystemMirror, the startup script restarts WLM from the user-specified configuration, saving the previous configuration. When PowerHA SystemMirror is stopped, it returns WLM back to its previous configuration.

Failure to start up WLM generates an error message logged in the **hacmp.out** log file, but node startup and the resource reconfiguration proceeds.

Workload Manager shutdown

WLM shutdown occurs either when the node leaves the cluster or on dynamic cluster reconfiguration. If WLM is currently running, the shutdown script determines whether the WLM was running before being started by the PowerHA SystemMirror and what configuration it was using. It then either does nothing (if WLM is not currently running), or stops WLM (if it was not running before PowerHA SystemMirror startup), or stops it and restarts it in the previous configuration (if WLM was previously running).

Limitations and considerations

Keep in mind some limitations and considerations when planning your Workload Manager configuration

These limitations and considerations include:

- Some WLM configurations might impede PowerHA SystemMirror performance. Be careful when designing your classes and rules, and be alert as to how they might affect PowerHA SystemMirror.
- You can have no more than 27 nondefault WLM classes across the cluster, because one configuration is shared across the cluster nodes.
- A PowerHA SystemMirror Workload Manager configuration does not support subclasses, even though WLM allows them. If you configure subclasses for any WLM classes that are placed in a resource group, a warning will be issued upon cluster verification, and the subclasses will not be propagated to other nodes during synchronization.
- On any given node, only the rules for classes associated with resource groups that can be acquired by a node are active on that node.

Planning for cluster events

These topics describe the PowerHA SystemMirror cluster events.

Overview

PowerHA SystemMirror provides the following ways to manage event processing:

- Customize predefined events
- Define new events

In PowerHA SystemMirror, resource groups are processed in parallel by default, if possible, unless you specify a customized serial processing order for all or some of the resource groups in the cluster.

The logic and sequence of events as described in examples might not list all the events.

See Starting and stopping cluster services for information about:

- Steps you take to start and stop cluster services
- Interaction with the AIX **shutdown** command and interaction of PowerHA SystemMirror cluster services with Reliable Scalable Cluster Technology (RSCT)

Related reference

Planning parallel or serial order for processing resource groups
By default, PowerHA SystemMirror acquires and releases all individual resources configured in your cluster in parallel. However, you can specify a specific serial order according to which some or all of the individual resource groups are acquired or released.

Planning site and node events

You can define sites in PowerHA SystemMirror Standard Edition for AIX or PowerHA SystemMirror Enterprise Edition. You must define sites to enable PowerHA SystemMirror Enterprise Edition storage replication support, including Geographic Logical Volume Manager (GLVM) and Metro Mirror. When you associate nodes and storage devices with a site, you can use PowerHA SystemMirror to implement a split-site Logical Volume Manager (LVM) mirroring configuration. PowerHA SystemMirror identifies appropriate selections that are based on site information and verifies the consistency of the mirroring configuration at a site level.

Note: Sites are supported only in PowerHA SystemMirror 7.1.2, or later, in both the Enterprise Edition and the Standard Edition. Replication management is supported only in PowerHA SystemMirror Enterprise Edition.

Site event scripts are included in the PowerHA SystemMirror software. If sites are not defined, no site events are generated. The PowerHA SystemMirror **site_event** scripts run as follows if sites are defined:

- The first node in a site runs the **site_up** event before it completes **node_up** event processing. The **site_up_complete** event runs after the **node_up_complete** event.
- When the last node in a site goes down, the **site_down** event runs before the **node_down** event, and the **site_down_complete** event runs after the **node_down_complete** event.

Without installing PowerHA SystemMirror Enterprise Edition, you can define pre-event and post-events to run when a site changes state. In this case, you can define all site-related processes.

Site events (including the **check_for_site_up** event and the **check_for_site_down** event) are logged in the **hacmp.out** log file.

If sites are defined, the **site_up** event runs when the first node in the site comes up and the **site_down** event runs when the last node in the site goes down. The event script sequence for handling resource groups in general is:

site_up
    site_up_remote

node_up
rg_move events to process resource group actions

node_up_complete

site_up_complete
site_up_remote_complete

site_down
site_down_remote

node_down
rg_move events to process resource group actions

node_down_complete

site_down
site_down_remote_complete

Planning node_up and node_down events

A **node_up** event is initiated by a node joining the cluster at cluster startup, or by rejoining the cluster at a later time.

Establishing initial cluster membership

This topic describes the steps taken by the cluster manager on each node when the cluster starts and the initial membership of the cluster is established. It shows how the cluster managers establish communication among the member nodes, and how the cluster resources are distributed as the cluster membership grows.

First node joins the cluster

1. PowerHA SystemMirror cluster services are started on node A. The Reliable Scalable Cluster Technology (RSCT) subsystem examines the state of the network interfaces and begins communicating with the RSCT subsystem on the other cluster nodes. The cluster manager on node A accumulates the initial state information and then broadcasts a message indicating that it is ready to join the cluster on all configured networks to which it is attached (**node_up**).
2. Node A interprets the lack of a response to mean that it is the first node in the cluster.
3. Node A initiates a **process_resources** script, which processes the node's resource configuration information.
4. When the event processing has completed, node A becomes a member of the cluster. PowerHA SystemMirror runs **node_up_complete**.

   All resource groups defined for node A are available for clients at this point.

   If the **Online on First Available Node** startup policy is specified as a startup behavior for resource groups, node A will take control of all these resource groups.

   If node A is defined as part of a nonconcurrent resource group that has the **Online Using Node Distribution** startup policy, this node takes control of the first resource group listed in the node environment.

   If node A is defined as part of a concurrent access resource configuration, it makes those concurrent resources available.

   For resource groups that have **Online on First Available Node** startup policy and the settling time configured, node A waits for the settling time interval before acquiring such resource groups. The settling time facilitates waiting for a higher priority node to join the cluster.

Second node joins the cluster

1. PowerHA SystemMirror Cluster Services are started on node B. Node B broadcasts a message indicating that it is ready to join the cluster on all configured networks to which it is attached (**node_up**).
2. Node A receives the message and sends an acknowledgment.
3. Node A adds node B to a list of active nodes, and starts keepalive communications with node B.
4. Node B receives the acknowledgment from node A. The message includes information identifying node A as the only other member of the cluster. (If there were other members, node B would receive the list of members.)
5. Node B processes the **process_resources** script and sends a message to let other nodes know when it is finished.

   Processing the **process_resources** script might include node A releasing resources it currently holds. If both node A and node B are in the resource group node list for one or more resource groups and node B has a higher priority for one or more of those resources. This is true only for resource groups that support fallback.

   Note that if the delayed fallback timer is configured, any resource group that is online on node A and for which node B is a higher priority node will fall back to node B at the time specified by the delayed fallback timer.
6. Meanwhile, node B has been monitoring and sending keepalive, and waiting to receive messages about changes in the cluster membership. When node B finishes its own **process_resources** script, it notifies node A.

   During its **node_up** processing, node B claims all resource groups configured for it (see step 3). Note that if the delayed fallback timer is configured, the resource group will fall back to a higher priority node at the time specified by the timer.
7. Both nodes process a **node_up_complete** event simultaneously.

   At this point, node B includes node A in its list of member nodes and its list of keepalive.
8. Node B sends a “new member” message to all possible nodes in the cluster.
9. When node A gets the message, it moves node B from its list of active nodes to its list of member nodes.

   At this point, all resource groups configured for node A and node B are available to cluster clients.

   Remaining nodes join the cluster
10. As PowerHA SystemMirror cluster services start on each remaining cluster node, steps 4 through 9 repeat, with each member node sending and receiving control messages, and processing events in the order outlined. Note especially that all nodes must confirm the **node_up_complete** event before completing the processing of the event and moving the new node to the cluster member list.

    As new nodes join, the RSCT subsystem on each node establishes communications and begins sending heartbeats. Nodes and adapters join RSCT heartbeat rings that are formed according to the definition in the PowerHA SystemMirror configuration. When the status of a network interface card (NIC) or node changes, the cluster manager receives the state change and generates the appropriate event.

Rejoining the cluster

When a node rejoins the cluster, the cluster managers running on the existing nodes initiate a **node_up** event to acknowledge that the returning node is up. When these nodes have completed processing their **process_resources** script, the new node then processes a **node_up** event so that it can resume providing cluster services.

This processing is necessary to ensure the proper balance of cluster resources. As long as the existing cluster managers first acknowledge a node rejoining the cluster, they can release any resource groups belonging to that node if necessary. Whether or not the resource groups are actually released in this situation depends on how the resource groups are configured for takeover (or dependencies). The new node can then start its operations.

Sequence of node_up events

The following list describes the sequence of **node_up** events:

node_up
This event occurs when a node joins or rejoins the cluster.

process_resources
This script calls the sub events needed for the node to acquire the service address (or shared address), gets all its owned (or shared) resources, and take the resources. This includes making disks available, varying on volume groups, mounting file systems, exporting file systems, mounting NFS file systems, and varying on concurrent access volume groups.

process_resources_complete
Each node runs this script when resources have been processed.

node_up_complete
This event occurs after the resources are processed and after the **node_up** event successfully completed. Depending on whether the node is local or remote, this event calls the **start_server** script to start application controllers on the local node, or allows the local node to mount an NFS file system only after the remote node is completely up.

node_up events with dependent resource groups

If dependencies between any resource groups are configured in the cluster, PowerHA SystemMirror processes all events related to resource groups in the cluster with the use of **rg_move** events that are launched for all resource groups when **node_up** events occur.

The cluster manager then takes into account all node policies, especially the configuration of dependencies for resource groups, and the current distribution and state of resource groups on all nodes in order to properly handle any acquiring, releasing, bringing online or taking offline of resource groups before a **node_up_complete** event can run.

Parent and child or location dependencies between resource groups offer a predictable and reliable way of building clusters with multitiered applications. However, **node_up** processing in clusters with dependencies could take more time than the parallel processing in clusters without resource groups' dependencies. You might need to adjust the **config_too_long** warning timer for **node_up** events.

node_down events

When all network interfaces are down, or a node does not respond to heartbeats, the cluster managers then run a **node_down** event. Depending on the cluster configuration, the peer nodes then take the necessary actions to get critical applications up and running and to ensure that data remains available.

A **node_down** event can be initiated by a node:

- Stopping cluster services and bringing resource groups offline
- Stopping cluster services and moving resource groups to another node
- Stopping cluster services and placing resource groups in an unmanaged state.
- Failing.

Stopping cluster services and bringing resource groups offline

When you stop cluster services and bring resource groups offline, PowerHA SystemMirror stops on the local node after the **node_down_complete** event releases the stopped node's resources. The other nodes run the **node_down_complete** event and do not take over the resources of the stopped node.

Stopping cluster services and moving resource groups

When you stop cluster services and move the resource groups to another node, PowerHA SystemMirror stops after the **node_down_complete** event on the local node releases its resource groups. The surviving nodes in the resource group node list take over these resource groups.

Stopping cluster services and placing resource groups in an unmanaged state

When you stop cluster services and place resource groups in an unmanaged state, PowerHA SystemMirror software stops immediately on the local node. The **node_down_complete** event is run on the stopped node. The cluster managers on remote nodes process **node_down** events, but do not take over any resource groups. The stopped node does not release its resource groups.

Node failure

When a node fails, the cluster manager on that node does not have time to generate a **node_down** event. In this case, the cluster managers on the surviving nodes recognize that a **node_down** event has occurred (when they realize the failed node is no longer communicating), and they trigger **node_down** events.

This initiates a series of sub events that reconfigure the cluster to deal with that failed node. Based on the cluster configuration, surviving nodes in the resource group node list take over the resource groups.

Sequence of node_down events

The following list describes the default parallel sequence of **node_down** events:

1. **node_down**
2. This event occurs when a node intentionally leaves the cluster or fails.
3. In some cases, the **node_down** event receives the **forced** parameter.
4. All nodes run the **node_down** event.
5. All nodes run the **node_down** event.
6. All nodes run the **process_resources** script. After the cluster manager evaluates the status of affected resource groups and the configuration, it initiates a series of sub events to redistribute resources as configured for fallover or fallback.
7. All nodes run the **process_resources_complete** script.
8. **node_down_complete**

Network events

PowerHA SystemMirror distinguishes between two types of network failure, *local* and *global*, and uses different network failure events for each type of failure. The network failure event script is often customized to send mail.

Sequence of network events

The following table shows the network events.

Table 3. Sequence of network events

Event name	Description
network_down (local)	This event occurs when only a particular node has lost contact with a network. The event has the following format: `network_down node_name network_name` The cluster manager takes selective recovery action to move affected resource groups to other nodes if Service IP is configured as part of the resource group. The results of the recovery actions are logged to **hacmp.out**.
network_down (global)	This event occurs when all of the nodes connected to a network have lost contact with a network. It is assumed in this case that a network-related failure has occurred rather than a node-related failure. This event has the following format: `network_down -1 network_name` **Note:** The **-1** argument is **-** *one*. This argument indicates that the **network_down** event is global. The global network failure event mails a notification to the system administrator, but takes no further action since appropriate actions depend on the local network configuration.
network_down_complete (local)	This event occurs after a local network failure event has completed. It has the following format: `network_down_complete node_name network_name` When a local network failure event occurs, the cluster manager takes selective recovery actions for resource groups containing a service network interface card (NIC) connected to that network.
network_down_complete (global)	This event occurs after a global network failure event has completed. It has the following format: `network_down_complete -1 network_name` The default processing for this event takes no actions because appropriate actions depend on the network configuration.
network_unstable	This event occurs when PowerHA is receiving continuous state changes for the network. The event has the following format: `network_unstable network_name` This event only runs after a network_down_complete for the corresponding network. No additional network events will run for this network until the stability is restored. When PowerHA is no longer receiving multiple, continuous state changes for the network, and the network is up, a **“network_up”** will run.

Table 3. Sequence of network events (continued)

Event name	Description
network_stable	This event occurs when PowerHA is no longer receiving continuous state changes for the network. The event has the following format: `network_stable network_name` This event only ever runs after a **`network_unstable`** event has been run for the corresponding network. Once a network has stabilized, any future state changes will result in normal event processing for those changes. If the network is currently "up" when stability is restored, a **"network_up"** event will be run.
network_up	This event occurs when the cluster manager determines a network has become available for use. Whenever a network becomes available again, PowerHA SystemMirror attempts to bring resource groups containing service IP labels on that network back online.
network_up_complete	This event occurs only after a **network_up** event has successfully completed. This event is often customized to notify the system administrator that an event demands manual attention. Whenever a network becomes available again, PowerHA SystemMirror attempts to bring resource groups containing service IP labels on that network back online.

Network interface events

The cluster manager reacts to the failure, unavailability, or joining of network interfaces by initiating an event.

The following table shows the network interface events.

Table 4. Network interface events

Network interface events	Event description
swap_adapter	This event occurs when the interface that hosts a service IP label on a node fails. The **swap_adapter** event moves the service IP label onto a boot interface on the same PowerHA SystemMirror network and then reconstructs the routing table. If the service IP label is an IP alias, it is put onto the boot interface as an additional IP label. Otherwise, the boot IP label is removed from the interface and placed on the failed interface. If the interface now holding the service IP label later fails, **swap_adapter** can switch to another boot interface if one exists. If a persistent node IP label was assigned to the failed interface, it moves with the service label to the boot interface. **Note:** PowerHA SystemMirror removes IP aliases from interfaces at shutdown. It creates the aliases again when the network becomes operational. The **hacmp.out** file records these changes.
swap_adapter_complete	This event occurs only after a **swap_adapter** event has been successfully completed. The **swap_adapter_complete** event ensures that the local address resolution protocol (ARP) cache is updated by deleting entries and pinging cluster IP addresses.

Table 4. Network interface events (continued)

Network interface events	Event description
fail_interface	This event occurs if an interface fails and there is no boot interface available to recover the service address. Takeover service addresses are monitored. It is possible to have an interface failure and no available interface for recovery and another interface up on the same network. This event applies to all networks, including those using IP aliasing for recovery. When a boot NIC fails on a network that is configured for IPAT via IP aliases, the **fail_interface** event is run. If the interface that failed was a service label, an **rg_move** event is then triggered.
join_interface	This event occurs if a boot interface becomes available or recovers. This event applies to all networks, including those using IPAT via IP aliases for recovery. Networks using IP aliases by definition do not have boot interfaces defined, so the **join_interface** event that is run in this case simply indicates that a boot interface joins the cluster.

Failure of a single network interface does not generate events

If only one network interface is active on a network, the cluster manager cannot generate a failure event for that network interface, because it has no peers with which to communicate for determining the health of the interface. Situations that have a single network interface follow:

- One-node clusters
- Multinode clusters with only one node active
- Multinode clusters with virtual Ethernet interfaces
- Failure of all but one interface on a network, one at a time

For example, starting a cluster with all service or boot interfaces disconnected produces the following results:

- First node active: No failure events are generated.
- Second node active: One failure event is generated.
- Third node active: One failure event is generated.

Clusterwide status events

By default, the cluster manager recognizes a time limit for reconfiguring a cluster and processing topology changes. If the time limit is reached, the cluster manager initiates a **config_too_long** event.

Whole cluster status events include:

Table 5. Clusterwide status events

Cluster wide status event names	Event description
config_too_long	This system warning occurs each time a cluster event takes more time to complete than a specified time-out period. This message is logged in the **hacmp.out** file. The time-out period for all events is set to 360 seconds by default. You can use SMIT to customize the time period allowed for a cluster event to complete before PowerHA SystemMirror issues a **config_too_long** warning for it.
reconfig_topology_start	This event marks the beginning of a dynamic reconfiguration of the cluster topology.

Table 5. Clusterwide status events (continued)

Cluster wide status event names	**Event description**
reconfig_topology_complete	This event indicates that a cluster topology dynamic reconfiguration has completed.
reconfig_resource_acquire	This event indicates that cluster resources that are affected by dynamic reconfiguration are being acquired by appropriate nodes.
reconfig_resource_release	This event indicates that cluster resources affected by dynamic reconfiguration are being released by appropriate nodes.
reconfig_resource_complete	This event indicates that a cluster resource dynamic reconfiguration has successfully completed.
cluster_notify	This event is triggered by verification when automatic cluster configuration monitoring detects errors in the cluster configuration. The output of this event is logged in the **hacmp.out** log file throughout the cluster on each node that is running cluster services.
event_error	All cluster nodes run the **event_error** event if any node has a fatal error. All nodes log the error and call out the failing node name in the **hacmp.out** log file.

Resource group event handling and recovery

The cluster manager keeps track of the resource group node priority policies, and any dependencies configured as well as the necessary topology information and resource group status so that it can take a greater variety of recovery actions, often avoiding the need for user intervention. Event logging includes a detailed summary for each high-level event to help you understand exactly what actions were taken for each resource group during the handling of failures.

For more information about how resource groups are handled in PowerHA SystemMirror, see Resource group behavior during cluster events. This topic contains information about the following PowerHA SystemMirror functions:

- Selective fallover for handling resource groups
- Handling of resource group acquisition failures
- Handling of resource groups configured with service IP resources
- Handling of PowerHA SystemMirror Enterprise Edition resource groups

Related information

Resource group behavior during cluster events

Resource group events

The cluster manager might move resource groups as a result of recovery actions taken during the processing of events such as node down.

Notes:

- If dependencies between resource groups or sites are specified, PowerHA SystemMirror processes events in a different sequence than usual.
- The lists in the following table do not include all possible resource group states. Also, the resource group instances could be in the process of acquiring or releasing. The corresponding resource group states are not listed here, but have descriptive names that explain which actions take place.

Table 6. Resource group events

Resource group event name	Event description
rg_move	This event moves a specified resource group from one node to another.
rg_move_complete	This action indicates that the **rg_move** event has successfully completed.
resource_state_change	This trigger event is used for resource group recovery if resource group dependencies are configured in the cluster. This action indicates that the cluster manager needs to change the state of one or more resource groups, or there is a change in the state of a resource managed by the cluster manager. This event runs on all nodes if one of the following situations occurs: • Application monitoring failure • Selective fallover for loss of volume group • Local network down • WAN failure • Resource group acquisition failure • Resource group recovery on IP interface availability • Expiration of settling timer for a resource group • Expiration of fallback timer for a resource group.
resource_state_ change_complete	This event runs when the **resource_state_change** event completes successfully. You can add pre-event or post-events here if necessary. You might want to be notified about resource state changes, for example.
external_resource_state_ change	This event runs when you move a resource group and PowerHA SystemMirror uses the dynamic processing path to handle the request because the resource group dependencies are configured in the cluster.
external_resource_state_ change_complete	This event runs when the **external_resource_state_change** event completes successfully.

Resource group subevents

Handling of individual resources during the processing of an event might include the following actions. For example, when a file system is in the process of being unmounted and mounted it is taken offline and then released by one node. Then it is acquired by another node and brought online.

The following table includes some but not all possible states of resource groups:

Table 7. Resource group subevents

Resource group subevents	Event description
releasing	This action indicates that a resource group is being released either to be brought offline or to be acquired on another node.
acquiring	This action is used when a resource group is being acquired on a node.
rg_up	This action indicates that the resource group is online.
rg_down	This action indicates that the resource group is offline.
rg_error	This action indicates that the resource group is in error state.

Table 7. Resource group subevents (continued)

Resource group subevents	Event description
rg_acquiring _secondary	This action indicates that the resource group is coming online at the target site (only the replicated resources are online).
rg_up_secondary	This action indicates that the resource group is online in the secondary role at the target site (only replicated resources are online).
rg_error_ secondary	This action indicates that the resource group at the site receiving the mirror data is in error state.
rg_temp_error_ state	This action indicates that the resource group is in a temporary error state. For example, it occurs due to a local network or an application failure. This state informs the cluster manager to initiate an **rg_move** event for this resource group. Resource groups should not be in this state when the cluster is stable.

After the completion of an event, the cluster manager has the state of resources and resource groups involved in the event. The cluster manager then analyzes the resource group information that it maintains internally and determines whether recovery events need to be queued for any of the resource groups. The cluster manager also uses the status of individual resources in resource groups to print out a comprehensive event summary to the **hacmp.out** log file.

For each resource group, the cluster manager keeps track of the nodes on which the resource group has tried to come online and failed. This information is updated when recovery events are processed. The cluster manager resets the node list for a resource group as soon as the resource group moves to the online or error states.

In PowerHA SystemMirror, the resource group ERROR states are displayed with detail:

Table 8. Resource groups in an ERROR state

Causes for resource group ERROR states	PowerHA SystemMirror displays this message
Parent group is NOT ONLINE; as a result, the child resource group is unavailable	`OFFLINE due to parent offline`
Higher-priority **different-node dependency** group is ONLINE	`OFFLINE due to lack of available node`
Another distributed group was acquired	`OFFLINE`
Group is falling over and in the OFFLINE state temporarily	`OFFLINE`

Manual intervention is only required when a resource group remains in ERROR state after the event processing finishes.

When a resource group is in the process of being moved, application monitoring is suspended and resumed appropriately. The application monitor recognizes that the application is in recovery state while the event is being processed.

Table 9. Application monitoring events

Application monitoring events	Event description
resume_appmon	This action is used by the application monitor to resume monitoring of an application.
suspend_appmon	This action is used by the application monitor to suspend monitoring of an application.

Related information

Resource group behavior during cluster events

Customizing cluster event processing

The cluster manager's ability to recognize a specific series of events and subevents permits a flexible customization scheme. With the PowerHA SystemMirror event customization facility, you can customize cluster event processing for your site. With customized event processing, you can provide a highly efficient path to the most critical resources in the event of a failure. However, this efficiency depends on your configuration.

As part of the planning process, you need to decide whether to customize event processing. If the actions taken by the default scripts are sufficient for your purposes, you do not need to do anything further to configure events during the configuration process.

If you do decide to customize event processing to your environment, use the PowerHA SystemMirror event customization facility described in this section. If you customize event processing, register these user-defined scripts with PowerHA SystemMirror during the configuration process.

The event customization facility includes the following functions:

- Event notification
- Pre-event and post-event processing

Complete customization of an event includes a notification to the system administrator (before and after event processing), and user-defined commands or scripts that run before and after event processing, as shown in the following example:

```
Notify sysadmin of event to be processed
Pre-event script or command
PowerHA SystemMirror event script
Post-event script or command
Notify sysadmin that event processing is complete
```

Event notification

You can specify a **notify** command that sends mail to indicate that an event is about to happen (or has just occurred), and that an event script succeeded or failed.

You configure notification methods for cluster events in SMIT under the **Custom Cluster Configuration** > **Events** > **Cluster Events** > **Change/Show Pre-Defined Events** menu. For example, a cluster might want to use a network failure notification event to inform system administrators that traffic might have to be rerouted. Afterwards, you can use a **network_up** notification event to tell system administrators that traffic can again be serviced through the restored network.

Event notification in a PowerHA SystemMirror cluster can also be done using pre-event and post-event scripts.

You can also configure a custom remote notification in response to events.

Related reference

Custom remote notification of events
You can define a notification method through the SMIT interface to issue a customized page in response to a cluster event. You can send text messaging notification to any number including a cell phone, or you can send notification to an email address.

Pre-event and post-event scripts

You can specify commands or multiple user-defined scripts that execute before and after the cluster manager calls an event script.

For example, you can specify one or more pre-event scripts that run before the **node_down** event script is processed. When the cluster manager recognizes that a remote node is down, it first processes these user-defined scripts. One such script might designate that a message be sent to all users to indicate that performance might be affected (when adapters are swapped and when application controllers are stopped and restarted). Following the **node_down** event script, a post processing event script for **network_up** notification might be included to broadcast a message to all users that a certain system is now available at another network address.

The following scenarios are other examples of where pre-event and post-event processing is useful:

- If a **node_down** event occurs, this script could notify users on the server about to takeover for the downed application controller that performance might vary, or that they should seek alternate systems for certain applications.
- Due to a network being down, a custom installation might be able to reroute traffic through other machines by creating new IP routes. The **network_up** and **network_up_complete** event scripts could reverse the procedure, ensuring that the correct routes exist after all networks are functioning.
- You can stop cluster services and move resource groups to another node as a post-event script if a network failed on the local node (but otherwise the network is functioning).

Note that when writing your PowerHA SystemMirror pre-event or post-event scripts, none of the shell environment variables defined in **/etc/environment** are available to your program. If you need to use any of these variables, explicitly source them by including this line in your script:

```
". /etc/environment"
```

If you plan to create pre-event or post-event scripts for your cluster, be aware that your scripts will be passed the same parameters used by the PowerHA SystemMirror event script you specify. For pre-event and post-event scripts, the arguments passed to the event command are the event name, event exit status, and the trailing arguments passed to the event command.

All PowerHA SystemMirror event scripts are maintained in the **/usr/es/sbin/cluster/events** directory. The parameters passed to your script are listed in the event script headers.

CAUTION: Be careful not to kill any PowerHA SystemMirror processes as part of your script. If you are using the output of the **ps** command and using a **grep** to search for a certain pattern, make sure the pattern does not match any of the PowerHA SystemMirror, Cluster Aware AIX (CAA), or Reliable Scalable Cluster Technology (RSCT) processes.

Pre-event and post-event scripts might not be needed

If you migrated from a previous version of PowerHA SystemMirror, some of your existing pre-event and post-event scripts might no longer be needed. PowerHA SystemMirror itself handles more situations.

Using the forced varyon attribute instead of pre-event or post-event scripts

If the forced varyon attribute is specified for a volume group, special scripts to force a varyon operation are no longer required.

The event_error event indicates failure on a remote node

Historically, nonrecoverable event script failures result in the **event_error** event being run on the cluster node where the failure occurred. The remaining cluster nodes did not indicate the failure. With PowerHA SystemMirror, all cluster nodes run the **event_error** event if any node has an unrecoverable error. All nodes log the error and record the failing node name in the **hacmp.out** log file.

If you have added pre-event or post-event for the **event_error** event, be aware that those event methods are called on every node, not just the failing node.

A Korn shell environment variable indicates the node where the event script failed: EVENT_FAILED_NODE is set to the name of the node where the event failed. Use this variable in your pre-event or post-event script to determine where the failure occurred.

The variable LOCALNODENAME identifies the local node. If LOCALNODENAME is not the same as EVENT_FAILED_NODE, the failure occurred on a remote node.

Resource groups processed in parallel and using pre-event and post-event scripts

Resource groups are processed in parallel by default in PowerHA SystemMirror unless you specify a customized serial processing order for all or some of the resource groups in the cluster.

When resource groups are processed in parallel, fewer cluster events occur in the cluster and appear in the event summaries.

The use of parallel processing reduces the number of particular cluster events for which you can create customized pre-event or post-event scripts. If you start using parallel processing for a list of resource groups in your configuration, be aware that some of your existing pre-event and post-event scripts might not work for these resource groups.

In particular, only the following events take place during parallel processing of resource groups:

acquire_svc_addr
acquire_takeover_addr
node_down
node_up
release_svc_addr
release_takeover_addr
start_server
stop_server

Note: In parallel processing, these events apply to an entire list of resource groups that are being processed in parallel, and not to a single resource group, as in serial processing. If you have pre-event and post-event scripts configured for these events, then after migration, these event scripts are launched not for a single resource group but for a list of resource groups, and might not work as expected.

The following events do not occur in parallel processing of resource groups:

get_disk_vg_fs
node_down_local
node_down_remote
node_down_local_complete
node_down_remote_complete
node_up_ local
node_up_remote
node_up_local_complete
node_up_remote_complete
release_vg_fs

Consider these events that do not occur in parallel processing if you have pre-event and post-event scripts and plan to upgrade to the current version.

If you want to continue using pre-event and post-event scripts, you could have one of the following cases.

Scenario	What You Should Do
You would like to use pre-event and post-event scripts for newly added resource groups.	All newly added resource groups are processed in parallel, which results in fewer cluster events. Therefore, there is a limited choice of events for which you can create pre-event and post-event scripts. In this case, if you have resources in resource groups that require handling by pre-event and post-event scripts written for specific cluster events, include these resource groups in the serial processing lists in SMIT to ensure that specific pre-event and post-event scripts can be used for these resources. For information about specifying serial or parallel processing of resource groups, see the section Configuring processing order for resource groups.
You upgrade to PowerHA SystemMirror 4.5 or later and choose parallel processing for some of the pre-existing resource groups in your configuration.	If, before migration you had configured customized pre-event or post-event scripts in your cluster, then now that these resource groups are processed in parallel after migration, the event scripts for a number of events cannot be used for these resource groups, since these events do not occur in parallel processing. If you want existing event scripts to continue working for the resource groups, include these resource groups in the serial ordering lists in SMIT, to ensure that the pre-event and post-event scripts can be used for these resources. For information about specifying serial or parallel processing of resource groups, see Configuring processing order for resource groups.

Related reference

Using forced varyon

PowerHA SystemMirror provides a forced varyon function to use in conjunction with AIX automatic error notification methods. The forced varyon function enables you to have the highest possible data availability.

Dependent resource groups and pre-event and post-event scripts

Historically, to achieve resource group and application sequencing, system administrators had to build the application recovery logic in their pre-event and post-event processing scripts. Every cluster would be configured with a pre-event script for all cluster events, and a post-event script for all cluster events. The latest releases of PowerHA SystemMirror include many more configuration options that allow customers to use built in policies for achieving the same sequencing and placement of resource groups. If you are currently using pre and post events to achieve resource group sequencing, you may want to review your implementation and transition to the built-in mechanisms.

Such scripts could become all-encompassing case statements. For instance, if you want to take an action for a specific event on a specific node, you need to edit that individual case, add the required code for pre-event and post-event scripts, and also ensure that the scripts are the same across all nodes.

To summarize, even though the logic of such scripts captures the desired behavior of the cluster, they can be difficult to customize and even more difficult to maintain later on, when the cluster configuration changes.

If you are using pre-event and post-event scripts or other methods, such as resource group processing ordering to establish dependencies between applications that are supported by your cluster, these methods might no longer be needed or can be significantly simplified. Instead, you can specify dependencies between resource groups in a cluster. For more information on planning dependent resource groups, see Resource group dependencies.

If you have applications included in dependent resource groups and still plan to use pre-event and post-event scripts in addition to the dependencies, additional customization of pre-event and post-event scripts might be needed. To minimize the chance of data loss during the application stop and restart process, customize your application controller scripts to ensure that any uncommitted data is stored to a shared disk temporarily during the application stop process and read back to the application during the application restart process. It is important to use a shared disk because the application might be restarted on a node other than the one on which it was stopped.

Related reference

Resource group dependencies
PowerHA SystemMirror offers a wide variety of configurations where you can specify the relationships between resource groups that you want to maintain at startup, fallover, and fallback.

Custom remote notification of events

You can define a notification method through the SMIT interface to issue a customized page in response to a cluster event. You can send text messaging notification to any number including a cell phone, or you can send notification to an email address.

You can use the verification automatic monitoring **cluster_notify** event to configure a PowerHA SystemMirror remote notification method to send out a message in case of detected errors in cluster configuration. The output of this event is logged in the **hacmp.out** file throughout the cluster on each node that is running cluster services.

You can configure any number of notification methods, for different events and with different text or numeric messages and telephone numbers to dial. The same notification method can be used for several different events, as long as the associated text message conveys enough information to respond to all of the possible events that trigger the notification.

After configuring the notification method, you can send a test message to make sure everything is configured correctly and that the expected message will be sent for a given event.

Planning for custom remote notification

Remote notification requires the following conditions:

- Any port specified must be defined to the AIX operating system and must be available.
- Each node that can send a page or text messages must have an appropriate modem installed and enabled.

  Note: PowerHA SystemMirror checks the availability of the port when the notification method is configured and before a page is issued. Modem status is not checked.
- Each node that can send email messages from the SMIT panel using AIX mail must have a TCP/IP connection to the Internet.
- Each node that can send text messages to a cell phone must have an appropriate Hayes-compatible dialer modem installed and enabled.

Customizing event duration time until warning

Depending on the cluster configuration, the speed of cluster nodes and the number and types of resources that need to move during cluster events, certain events might take different time intervals to complete. For such events, you might want to customize the time period PowerHA SystemMirror waits for an event to complete before issuing the **config_too_long** warning message.

Cluster events that include acquiring and releasing resource groups take a longer time to complete. The following cluster events are considered *slow* events:

- **node_up**
- **node_down**
- **reconfig_resource**

- **rg_move**

Customize event duration time for *slow* cluster events to avoid getting unnecessary system warnings during normal cluster operation.

All other cluster events are considered *fast* events. These events typically take a shorter time to complete and do not involve acquiring or releasing resources. Examples of fast events include:

- **swap_adapter**
- Events that do not handle resource groups

You can customize event duration time before receiving a warning for fast events to take corrective action faster.

Consider customizing **Event Duration Time Until Warning** if, in the case of slow cluster events, PowerHA SystemMirror issues warning messages too frequently. In the case of fast events, you want to speed up detection of a possible problem event.

Note: Dependencies between resource groups offer a predictable and reliable way of building clusters with multitier applications. However, processing of some cluster events (such as **node_up**) in clusters with dependencies could take more time than processing of those events where all resource groups are processed in parallel. Whenever resource group dependencies allow, PowerHA SystemMirror processes multiple nonconcurrent resource groups in parallel, and processes multiple concurrent resource groups on all nodes at once. However, a resource group that is dependent on other resource groups cannot be started until the others have been started first. The **config_too_long** warning timer for **node_up** events should be set large enough to allow for this.

User-defined events

You can define your own events for which PowerHA SystemMirror can run your specified recovery programs. This process adds a new dimension to the predefined PowerHA SystemMirror pre-event and post-event script customization facility.

You specify the mapping between events that you define and the recovery programs that define the event recovery actions through the SMIT interface. With this mapping, you control both the scope of each recovery action and the number of event steps synchronized across all nodes.

An RMC *resource* refers to an instance of a physical or logical entity that provides services to some other component of the system. The term resource is used very broadly to refer to software and hardware entities. For example, a resource could be a particular file system or a particular host machine. A *resource class* refers to all resources of the same type, such as processors or host machines.

A *resource manager* (daemon) maps actual entities to RMC's abstractions. Each resource manager represents a specific set of administrative tasks or system functions. The resource manager identifies the key physical or logical entity types related to that set of administrative tasks or system functions, and defines resource classes to represent those entity types.

For example, the host resource manager contains a set of resource classes for representing aspects of a individual host machine. It defines resource classes to represent:

- Individual machines (IBM.Host)
- Paging devices (IBM.PagingDevice)
- Physical volumes (IBM.PhysicalVolume)
- Processors (IBM.Processor)
- A host's identifier token (IBM.HostPublic)
- Programs running on the host (IBM.Program)
- Each type of Ethernet adapter supported by the host

The AIX *resource monitor* generates events for OS-related resource conditions such as the percentage of CPU that is idle (`IBM.Host.PctTotalTimeIdle`) or the percentage of disk space in use (`IBM.PhysicalVolume.PctBusy`). The *program resource monitor* generates events for process-related occurrences such as the unexpected end of a process. The program resource monitor uses the resource attribute `IBM.Program.ProgramName`.

Writing recovery programs

A recovery program has a sequence of recovery command specifications, possibly interspersed with **barrier** commands.

The format of these specifications follows:

```
:node_set recovery_command expected_status NULL
```

where:

- *node_set* is a set of nodes on which the recovery program is to run
- *recovery_command* is a quote-delimited string specifying a full path to the executable program. The command cannot include any arguments. Any executable program that requires arguments must be a separate script. The recovery program must be in this path on all nodes in the cluster. The program must specify an exit status.
- *expected_status* is an integer status to be returned when the recovery command completes successfully. The cluster manager compares the actual status returned to the expected status. A mismatch indicates unsuccessful recovery. If you specify the character X in the expected status field, the cluster manager omits the comparison.
- *NULL* is currently not used.

You specify node sets by dynamic relationships. PowerHA SystemMirror supports the following dynamic relationships:

All
The recovery command runs on all nodes in the current membership.

Event
The node on which the event occurred.

Other
All nodes except the one on which the event occurred.

The specified dynamic relationship generates a set of recovery commands identical to the original, except that a node ID replaces *node_set* in each set of commands.

The command string for user-defined event commands must start with a slash (/). The **`clcallev`** command runs commands that do not start with a slash.

Useful commands and reference for RMC information

To list all persistent attribute definitions for the `IBM.Host` RMC resource (*selection string* field):

```
lsrsrcdef -e -A p IBM.Host
```

To list all dynamic attribute definitions for the `IBM.Host` RMC resource (*Expression* field):

```
lsrsrcdef -e -A d IBM.Host
```

Example: Recovery program

A sample program sends a message to **/tmp/r1.out** that paging space is low on the node where the event occurred. For recovery program **r1.rp**, the SMIT fields would be filled in as follows.

Example: Recovery program fields

Field	Value
Event Name	`E_page_space(User-defined name)`
Recovery program path	`/r1.rp`
Resource name	`IBM.Host (cluster node)`
Selection string	`Name = ?" (name of node)`

Expression	`TotalPgSpFree < 256000`
	`(VMM is within 200 MB of paging space warning level).`
	The resource attribute plus the condition you want to flag.

Rearm expression	`TotalPgSpFree >256000` The resource attribute plus the adjusted condition.

Where recovery program **r1.rp** is as follows:

```
#format:
#relationship  >command to run  >expected status NULL
#
event  “/tmp/checkpagingspace”  0 NULL
```

The recovery program does not execute a command with arguments itself. Instead, it points to a shell script, **/tmp/checkpagingspace**, which contains:

```
#!/bin/ksh
/usr/bin/echo “Paging Space LOW!” > /tmp/r1.out
exit 0
```

Recovery program for node_up event example

The following example is a recovery program for the **node_up** event:

```
#format:
#relationshipcommand to run expected status NULL
#
other  “node_up”  0 NULL
#
barrier
#
event “node_up” 0 NULL
#
barrier
#
all “node_up_complete” X NULL
```

Barrier commands

You can put any number of barrier commands in the recovery program. All recovery commands before a barrier start in parallel. After a node encounters a barrier command, all nodes must reach it before the recovery program continues.

The syntax of the barrier command is **barrier**.

Event roll-up

If multiple events are outstanding simultaneously, you only see the highest priority event. Node events are higher priority than network events. But user-defined events, the lowest priority, do not roll up at all, so you see all of them.

Event summaries and preamble

When events are logged to a node's **hacmp.out** log file, the verbose output contains numerous lines of event details followed by a concise event summary. The event summaries make it easier to scan the log for important cluster events.

You can view a compilation of just the event summary portions of the past seven days of **hacmp.out** log files by using the **View Event Summaries** option in the **Problem Determination Tools** SMIT panel. The event summaries can be compiled even if you have redirected the **hacmp.out** file to a nondefault location. The **Display Event Summaries** report also includes resource group information generated by the **clRGinfo** command. You can also save the event summaries to a specified file instead of viewing them through SMIT.

When events handle resource groups with dependencies, a preamble is written to the **hacmp.out** log file listing the plan of sub events for handling the resource groups.

Planning for PowerHA SystemMirror clients

These topics discuss planning considerations for PowerHA SystemMirror clients. This is the last step before proceeding to the installation of your PowerHA SystemMirror software.

PowerHA SystemMirrorclients are end-user devices that can access the nodes in a PowerHA SystemMirror cluster. For planning purposes, it is important that you evaluate the cluster from the point of view of the clients.

Clients running Clinfo

The Clinfo program calls the **/usr/es/sbin/cluster/etc/clinfo.rc** script whenever a network or node event occurs. By default, this action updates the system's address resolution protocol (ARP) cache to reflect changes to network addresses. You can customize this script if further action is needeed.

Reconnecting to the cluster

Clients running the Clinfo daemon can reconnect to the cluster quickly after a cluster event. If you have hardware other than IBM System p between the cluster and the clients, make sure that you can update the ARP cache of those network components after a cluster event occurs.

If you configure the cluster to swap hardware addresses as well as IP addresses, you do not need to be concerned about updating the ARP cache. However, be aware that this option causes a longer delay.

If you are using IPAT via IP aliases, make sure all your clients support TCP/IP gratuitous ARP.

Customizing the clinfo.rc script

For clients running the Clinfo daemon, decide whether to customize the **/usr/es/sbin/cluster/etc/clinfo.rc** script to do more than update the ARP cache when a cluster event occurs.

Clients not running Clinfo

On clients not running the Clinfo daemon, you may have to update the local address resolution protocol (ARP) cache indirectly by pinging the client from the cluster node.

On the cluster nodes, add the name or address of a client host that you want to notify to the **PING_CLIENT_LIST** variable in the **clinfo.rc** script. When a cluster event occurs, the **clinfo.rc** script runs the following command for each host that is specified in the PING_CLIENT_LIST variable:

```
ping -c1 $host
```

This assumes the client is connected directly to one of the cluster networks.

Network components

If you configured the network so that clients attach to networks on the other side of a router, bridge, or gateway rather than to the cluster's local networks, be sure that you can update the ARP cache of those network components after a cluster event occurs.

Applications and PowerHA SystemMirror

This topic addresses some of the key issues to consider when making your applications highly available under PowerHA SystemMirror.

With PowerHA SystemMirror you can configure clusters with multitiered applications by establishing dependencies between resource groups containing different applications. These topics describe resource group dependencies and how they can help with keeping dependent applications highly available.

Related reference

Initial cluster planning
This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

Overview of applications and PowerHA SystemMirror

Besides understanding the hardware and software needed to make a cluster highly available, you must be aware of application availability considerations when planning your PowerHA SystemMirror environment. The goal of clustering is to keep your important applications available despite any single point of failure. To achieve this goal, consider the aspects of an application that make it recoverable under PowerHA SystemMirror.

There are few requirements that an application must meet to recover well under PowerHA SystemMirror. Some required characteristics, as well as a number of suggestions, are discussed here. These are grouped according to key points that applies to all PowerHA SystemMirror environments. This topic covers the following application considerations:

- *Automation*. Making sure your applications start and stop without user intervention
- *Dependencies*. Knowing what factors outside PowerHA SystemMirror affect the applications
- *Interference*. Knowing that applications themselves can hinder PowerHA SystemMirror functioning
- *Robustness*. Choosing strong, stable applications
- *Implementation*. Using appropriate scripts, file locations, and cron schedules.

You should add an application monitor to detect a problem with application startup. An application monitor in startup monitoring mode checks an application controller’s successful startup within the specified stabilization interval and exits after the stabilization period ends.

You can start the PowerHA SystemMirror cluster services on the nodes without stopping your applications, by selecting an option from a SMIT panel **PowerHA SystemMirror Services** > **Start Cluster Services**. When starting, PowerHA SystemMirror relies on the application startup scripts and configured application monitors to ensure that PowerHA SystemMirror is aware of the running application and does not start a second instance of the application.

Similarly, you can stop PowerHA SystemMirror cluster services and leave the applications running on the nodes. When the node that has been stopped and placed in an unmanaged state rejoins the cluster, the state of the resources is assumed to be the same unless a user initiates a PowerHA SystemMirror resource group command to bring the resource group into another state (for example, online on an active node).

Application automation: Minimizing manual intervention

One key requirement for an application to function successfully under PowerHA SystemMirror is that the application be able to start and stop without any manual intervention.

Application start scripts

Create a start script that starts the application. The start script should perform any clean-up or preparation necessary to ensure proper startup of the application, and also to properly manage the number of instances of the application that need to be started. When the application controller is added to a resource group. PowerHA SystemMirror calls this script to bring the application online as part of processing the resource group. Because the cluster daemons call the start script, there is no option for interaction. Additionally, upon a PowerHA SystemMirror fallover, the recovery process calls this script to bring the application online on a standby node. This allows for a fully automated recovery, and is why any necessary cleanup or preparation should be included in this script.

PowerHA SystemMirror calls the start script as the root user. It might be necessary to change to a different user in order to start the application. The **su** command can accomplish this. Also, it might be necessary to run the**nohup** command on commands that are started in the background and have the potential to be ended upon exit of the shell.

For example, a PowerHA SystemMirror cluster node might be a client in a Network Information Service (NIS) environment. If this is the case and you need to use the **su** command to change the user ID, there must be a route to the NIS server at all times. In the event that a route does not exist and the **su** command is attempted, the application script hangs. You can avoid this situation by enabling the PowerHA SystemMirror cluster node to be an NIS client. That way, a cluster node has the ability to access its own NIS map files to validate a user ID.

The start script should also check for the presence of required resources or processes. This will ensure an application can start successfully. If the necessary resources are not available, a message can be sent to the administration team to correct this and restart the application.

Start scripts should be written so that they determine whether one instance of the application is already running and not start another instance unless multiple instances are desired. Keep in mind that the start script might be run after a primary node has failed. There might be recovery actions necessary on the backup node in order to restart an application. This is common in database applications. Again, the recovery must be able to run without any interaction from administrators.

Application stop scripts

The most important aspect of an application stop script is that it completely stops an application. Failure to do so might prevent PowerHA SystemMirror from successfully completing a takeover of resources by the backup nodes. In stopping, the script might need to address some of the same concerns that the start script addresses, such as NIS and the **su** command.

The application stop script should use a phased approach. The first phase should be an attempt to stop the cluster services and bring resource groups offline. If processes refuse to end, the second phase should be used to forcefully ensure that all processing is stopped. Finally, a third phase can use a loop to repeat any steps necessary to ensure that the application has ended completely.

Be sure that your application stop script exits with the value 0 when the application has been successfully stopped. In particular, examine what happens if you run your stop script when the application is already stopped. Your script must exit with 0 in this case as well. If your stop script exits with a different value, this tells PowerHA SystemMirror that the application is still running, although possibly in a damaged state. The **event_error** event will be run and the cluster will enter an error state. This check alerts administrators that the cluster is not functioning properly.

Keep in mind that PowerHA SystemMirror allows 360 seconds by default for events to complete processing. A message indicates that the cluster has been in reconfiguration too long appears until the cluster completes its reconfiguration and returns to a stable state. This warning might be an indication that a script is hung and requires manual intervention. If this is a possibility, you might want to consider stopping an application manually before stopping PowerHA SystemMirror.

You can change the time period before the **config_too_long** event is called.

Application start and stop scripts and dependent resource groups

In PowerHA SystemMirror, support for dependent resource groups allows you to configure the following options:

- Three levels of dependencies between resource groups, for example a configuration in which node A depends on node B, and node B depends on node C. PowerHA SystemMirror prevents you from configuring circular dependencies.
- A type of dependency in which a parent resource group must be online on any node in the cluster before a child (dependent) resource group can be activated on a node.

If two applications must run on the same node, both applications must reside in the same resource group.

If a child resource group contains an application that depends on resources in the parent resource group and, then upon fallover conditions, and if the parent resource group falls over to another node, the child resource group is temporarily stopped and automatically restarted. Similarly, if the child resource group is concurrent, PowerHA SystemMirror takes it offline temporarily on all nodes, and brings it back online on all available nodes. If the fallover of the parent resource group is not successful, both the parent and the child resource groups go into an ERROR state.

Note that when the child resource group is temporarily stopped and restarted, the application that belongs to it is also stopped and restarted. Therefore, to minimize the chance of data loss during the application stop and restart process, customize your application controller scripts to ensure that any uncommitted data is stored to a shared disk temporarily during the application stop process and read back to the application during the application restart process. It is important to use a shared disk because the application might be restarted on a node other than the one on which it was stopped.

Application tier issues

Often, applications have a multitiered architecture (for example, a database tier, an application tier, and a client tier). Consider all tiers of an architecture if one or more is made highly available through the use of PowerHA SystemMirror.

For example, if the database is made highly available and a fallover occurs, consider whether actions should be taken at the higher tiers in order to automatically return the application to service. If so, it might be necessary to stop and restart application or client tiers. This can be facilitated in one of two ways. One way is to run the **`cli_on_node`** command on the tiers, and the other is to use a remote execution command such as **rsh**, **rexec**, or **ssh**.

Note: Certain methods, such as the use of ~**/.rhosts** files, pose a security risk.

Using dependent resource groups

To configure complex clusters with multitiered applications, you can use parent-child dependent resource groups. You might also want to consider using location dependencies.

Using the Clinfo API

Clinfo API is the cluster information daemon. You can write a program using the Clinfo API to run on any tiers that would stop and restart an application after a fallover has completed successfully. In this sense, the tier, or application, becomes cluster aware, responding to events that take place in the cluster.

Using pre-event and post-event scripts

Another way to address the issue of multitiered architectures is to use pre-event and post-event scripts around a cluster event. These scripts would call a remote execution command, such as **rsh**, **rexec**, or **ssh**, to stop and restart the application.

Related concepts

Applications and PowerHA SystemMirror
This topic addresses some of the key issues to consider when making your applications highly available under PowerHA SystemMirror.

Related reference

Writing effective scripts
Writing smart application start scripts can also help reduce the likelihood of problems when you bring applications online.

Planning resource groups
These topics describe how to plan resource groups within a PowerHA SystemMirror cluster.

Application dependencies
Historically, to achieve resource group and application sequencing, system administrators had to build the application recovery logic in their pre-event and post-event processing scripts. Every cluster would be configured with a pre-event script for all cluster events, and with a post-event script for all cluster events.

Application dependencies

Historically, to achieve resource group and application sequencing, system administrators had to build the application recovery logic in their pre-event and post-event processing scripts. Every cluster would be configured with a pre-event script for all cluster events, and with a post-event script for all cluster events.

Such scripts could become all-encompassing case statements. For example, if you want to take an action for a specific event on a specific node, you need to edit that individual case, add the required code for pre-event and post-event scripts, and also ensure that the scripts are the same across all nodes.

To summarize, even though the logic of such scripts captures the desired behavior of the cluster, the scripts can be difficult to customize and even more difficult to maintain later on when the cluster configuration changes.

If you are using pre-event and post-event scripts or other methods, such as resource group processing ordering to establish dependencies between applications that are supported by your cluster, then these methods might no longer be needed or can be significantly simplified. Instead, you can specify dependencies between resource groups in a cluster.

Note: In many cases, applications depend on more than data and an IP address. For the success of any application under PowerHA SystemMirror, it is important to know what the application should not depend in order to function properly. This topic outlines many of the major dependency issues. Keep in mind that these dependencies might come from outside the PowerHA SystemMirror and application environment. They might be incompatible products or external resource conflicts. Look beyond the application itself for potential problems within the enterprise.

Locally attached devices

Locally attached devices can pose a clear dependency problem. In the event of a fallover, if these devices are not attached and accessible to the standby node, an application might fail to run properly. These might include a CD-ROM device, a tape device, or an optical juke box. Consider whether your application depends on any of these and if they can be shared between cluster nodes.

Hard coding

Hard coding an application to a particular device in a particular location creates a potential dependency issue. For example, the console is typically assigned as **/dev/tty0**. Although this assigned name is common, it is by no means guaranteed. If your application assumes the name **/dev/tty0**, ensure that all possible standby nodes have the same configuration.

Host name dependencies

Some applications are written to be dependent on the AIX host name. They issue a command in order to validate licenses or name file systems. The host name is not an IP address label. The host name is specific to a node and is not failed over by PowerHA SystemMirror. The host name cannot be changed. Any application that changes the host name is not supported by PowerHA SystemMirror.

Software licensing

Another possible problem is software licensing. Software can be licensed to a particular CPU ID. If this is the case with your application, a fallover of the software will not successfully restart. You might be able to avoid this problem by having a copy of the software on all cluster nodes. Know whether your application uses software that is licensed to a particular CPU ID.

Related reference

Planning considerations for multitiered applications
Business configurations that use multitiered applications can use parent and child dependent resource groups. For example, the database must be online before the application controller. In this case, if the database goes down and is moved to a different node the resource group containing the application controller would have to be brought down and back up on any node in the cluster.

Application interference

Sometimes an application or an application environment might interfere with the proper functioning of PowerHA SystemMirror. An application might run properly on both the primary and standby nodes. However, when PowerHA SystemMirror is started, a conflict with the application or environment could arise that prevents PowerHA SystemMirror from functioning successfully.

Products manipulating network routes

Additionally, products that manipulate network routes can keep PowerHA SystemMirror from functioning as it was designed. These products can find a secondary path through a network that has had an initial failure. This routing might prevent PowerHA SystemMirror from properly diagnosing a failure and taking appropriate recovery actions.

Related reference

Initial cluster planning
This section describe the initial steps you take to plan a PowerHA SystemMirror cluster to make applications highly available.

Robustness of application

Of primary importance to the success of any application is the health, or robustness, of the application. If the application is unstable or crashing intermittently, resolve these issues before placing it in a high availability environment.

Beyond basic stability, an application under PowerHA SystemMirror should meet other robustness characteristics.

Successful start after hardware failure

A good application candidate for PowerHA SystemMirror should be able to restart successfully after a hardware failure. Run a test on an application before managing it with PowerHA SystemMirror. Run the application under a heavy load and fail the node. What does it take to recover after the node is back online? Can this recovery be completely automated? If not, the application might not be a good candidate for high availability.

Survival of real memory loss

Applications should regularly save to disk any information necessary to restart. If a failure occurs, the application can pick up from where it was before, rather than completely starting over.

Application implementation strategies

There are a number of aspects of an application to consider as you plan for implementing it under PowerHA SystemMirror.

Consider characteristics such as time to start, time to restart after failure, and time to stop. Your decisions in a number of areas, such as script writing, file storage, **/etc/inittab** file and cron schedule issues, can improve the probability of successful application implementation.

Writing effective scripts

Writing smart application start scripts can also help reduce the likelihood of problems when you bring applications online.

A good practice for start scripts is to check prerequisite conditions before you start an application. The prerequisite conditions might include access to a file system, adequate paging space, and free file system space. The start script should exit and run a command to notify system administrators if requirements are not met.

In pre-event and post-event scripts, on the first line you must specify the shell environment. For example, if you are using the Korn shell environment, the first line in the event script must be `#!/bin/ksh93`.

When you start a database, it is important to consider whether there are multiple instances within the same cluster. In this scenario, you want to start only the instances applicable for each node. Certain database startup commands read a configuration file and start all known databases at the same time. This behavior might not be an ideal configuration for all environments.

Be careful not to kill any PowerHA SystemMirror processes as part of your script. If you are using the output of the **ps** command and you are using the **grep** command to search for a certain pattern, verify that the pattern does not match any of the PowerHA SystemMirror or Reliable Scalable Cluster Technology (RSCT) processes.

Considering file storage locations

Give thought to where the configuration files reside. They could either be on ashared disk, and therefore potentially accessed by whichever node has the volume group varied on, or on each node's internal disks. This holds true for all aspects of an application. Certain files must be on shared drives. These files include data, logs, and anything that could be updated by the execution of the application. Files such as configuration files or application binaries could reside in either location.

There are advantages and disadvantages to storing optional files in either location. Having files stored on each node's internal disks implies that you have multiple copies of, and potentially multiple licenses for, the application. This could require additional cost as well as maintenance in keeping these files synchronized. However, in the event that an application needs to be upgraded, the entire cluster need not be taken out of production. One node could be upgraded while the other remains in production. The best solution is the one that works best for a particular environment.

Considering /etc/inittab and cron table issues

Also give thought to applications, or resources needed by an application that either start out of the /etc/inittab file or out of the cron table.

The inittab file starts applications when you start the system. If cluster resources are needed for an application to function, they will not become available until after PowerHA SystemMirror is started. It is better to use the PowerHA SystemMirror application controller facility that allows the application to be a resource that is started only after all dependent resources are online.

Note: It is important that the following settings are correct in /etc/inittab file:

```
hacmp:2:once:/usr/es/sbin/cluster/etc/rc.init
```

- The clinit and pst_clinit entries must be the last entries of run level "2".
- The clinit entry must proceed the pst_clinit entry.

An incorrect entry for these prevents PowerHA SystemMirror from starting.

In the cron table, jobs are started according to a schedule set in the table and the date setting on a node. This information is maintained on internal disks and thus cannot be shared by a standby node. Synchronize these cron tables so that a standby node can perform the necessary action at the appropriate time. Also, ensure that the date is set the same on the primary node and any of its standby nodes.

Examples: Oracle database and SAP R/3

Here are two examples that illustrates issues to consider in order to make the applications Oracle Database and SAP R/3 function well under PowerHA SystemMirror.

Example 1: Oracle Database

The Oracle Database, like many databases, functions well under PowerHA SystemMirror. It is a robust application that handles failures well. It can roll back uncommitted transactions after a fallover and return to service in a timely manner. However, there are a few things to keep in mind when using Oracle Database under PowerHA SystemMirror.

Starting Oracle

Oracle must be started by the Oracle user ID. Thus, the start script should contain the following code `su - oracleuser`. The dash (-) is important since the **su** command needs to take on all characteristics of the Oracle user and reside in the Oracle users home directory. The command would look something like this:

```
su - oracleuser -c /apps/oracle/startup/dbstart
```

The **`dbstart`** command and the **`dbshut`** command read the **/etc/oratabs** file for instructions on which database instances are known and should be started. In certain cases it is inappropriate to start all of the instances, because they might be owned by another node. This would be the case in the mutual takeover of two Oracle instances. The **oratabs** file typically resides on the internal disk and thus cannot be shared. If appropriate, consider other ways of starting different Oracle instances.

Stopping Oracle

The stopping of Oracle is a process of special interest. There are several different ways to ensure Oracle has completely stopped. The suggested sequence is this: first, implement a graceful shutdown; second, call a shutdown immediate, which is a somewhat more forceful method; finally, create a loop to check the process table to ensure all Oracle processes have exited.

Oracle file storage

The Oracle product database contains several files as well as data. It is necessary that the data and redo logs be stored on shared disk so that both nodes might have access to the information. However, the Oracle binaries and configuration files could reside on either internal or shared disks. Consider what solution is best for your environment.

Example 2: SAP R/3, a multitiered application

SAP R/3 is an example of a three-tiered application. It has a database tier, an application tier, and a client tier. Most frequently, it is the database tier that is made highly available. In such a case, when a fallover occurs and the database is restarted, it is necessary to stop and restart the SAP application tier. You can do this in one of two ways:

- Using a remote execution command, such as **rsh**, **rexec**, or **ssh**

  Note: Certain methods, such as the use of ~/.rhosts files, pose a security risk.

- Making the application tier nodes cluster aware.

Using a remote execution command

The first way to stop and start the SAP application tier is to create a script that performs remote command execution on the application nodes. The application tier of SAP is stopped and then restarted. This is done for every node in the application tier. Using a remote execution command requires a method of allowing the database node access to the application node.

Note: Certain methods, such as the use of ~/.rhosts files, pose a security risk.

Making application tier nodes cluster aware

A second method for stopping and starting the application tier is to make the application tier nodes cluster aware. This means that the application tier nodes are aware of the clustered database and know when a fallover occurs. You can implement this by making the application tier nodes either PowerHA SystemMirror servers or clients. If the application node is a server, it runs the same cluster events as the database nodes to indicate a failure. Pre-event and post-event scripts could then be written to stop and restart the SAP application tier. If the application node is a PowerHA SystemMirror client, it is notified of the database fallover using SNMP through the cluster information daemon (Clinfo). A program could be written using the Clinfo API to stop and restart the SAP application tier.

Related information

Programming client applications for the Clinfo API

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Index

IBM®