IBM PowerHA SystemMirror for AIX

Standard Edition

Version 7.2

PowerHA SystemMirror concepts

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 107.

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.

© Copyright International Business Machines Corporation 2017, 2023.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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.

© Copyright IBM Corp. 2017, 2023

PowerHA SystemMirror concepts

The following information introduces important concepts you must understand before you can use the PowerHA SystemMirror software for the AIX operating system.

What's new in PowerHA SystemMirror concepts

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

How to see what's new or changed

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

December 2023

1. Updated the Tie breaker option for split and merge policies topic with Site Priority configuration for split and merge policies.
2. Added new feature “Ansible integration with PowerHA SystemMirror” on page 101 topic.
3. Updated a note for cluster tool support in following topics.
   - Steps for configuring a cluster
   - “Cluster test tool” on page 100

December 2020

Updated the following topics with information about the Split and Merge Management Policy field:

- Merge policy
- Split policy

December 2019

Added the Cross-cluster verification utility topic that provides an overview of the cross-cluster verification utility.

December 2018

The following information is a summary of the updates that were made to this topic collection:

- Added information about pre-events and post-events in the “Customizing event processing” on page 73 topic.
- Updated the examples of events that the cluster manager recognizes in the “Cluster events” on page 71 topic.

© Copyright IBM Corp. 2017, 2023

PowerHA SystemMirror for AIX

The following information discusses the concepts of high availability and clustering, presents the PowerHA SystemMirror cluster diagram, and describes a PowerHA SystemMirror cluster from a functional perspective.

High availability clustering for AIX

The IBM PowerHA SystemMirror software provides a low-cost commercial computing environment that ensures quick recovery of mission-critical applications from hardware and software failures.

With PowerHA SystemMirror software, critical resources remain available. For example, a PowerHA SystemMirror cluster could run a database server program that services client applications. The clients send queries to the server program that responds to their requests by accessing a database, stored on a shared external disk.

This high availability system combines custom software with industry-standard hardware to minimize downtime by quickly restoring services when a system, component, or application fails. Although not instantaneous, the restoration of service is rapid, usually within 30 to 300 seconds.

In a PowerHA SystemMirror cluster, to ensure the availability of these applications, the applications are put under PowerHA SystemMirror control. PowerHA SystemMirror takes measures to ensure that the applications remain available to client processes even if a component in a cluster fails. To ensure availability, in case of a component failure, PowerHA SystemMirror moves the application (along with resources that ensure access to the application) to another node in the cluster.

High availability and hardware availability

High availability is sometimes confused with simple hardware availability. Fault tolerant, redundant systems (such as RAID) and dynamic switching technologies (such as DLPAR) provide recovery of certain hardware failures, but do not provide the full scope of error detection and recovery required to keep a complex application highly available.

A modern, complex application requires access to all of these components:

- Nodes (CPU, memory)
- Network interfaces (including external devices in the network topology)
- Disk or storage devices.

Recent surveys of the causes of downtime show that actual hardware failures account for only a small percentage of unplanned outages. Other contributing factors include:

- Operator errors
- Environmental problems
- Application and operating system errors.

Reliable and recoverable hardware simply cannot protect against failures of all these different aspects of the configuration. Keeping these varied elements, and therefore the application, highly available requires:

- Thorough and complete planning of the physical and logical procedures for access and operation of the resources on which the application depends. These procedures help to avoid failures in the first place.
- A monitoring and recovery package that automates the detection and recovery from errors.
- A well-controlled process for maintaining the hardware and software aspects of the cluster configuration while keeping the application available.

High availability versus fault tolerance

The difference between fault tolerance and high availability, is this: A fault tolerant environment has no service interruption but a significantly higher cost, while a highly available environment has a minimal service interruption.

Fault tolerance relies on specialized hardware to detect a hardware fault and instantaneously switch to a redundant hardware component—whether the failed component is a processor, memory board, power supply, I/O subsystem, or storage subsystem. Although this cutover is apparently seamless and offers non-stop service, a high premium is paid in both hardware cost and performance because the redundant components do no processing. More importantly, the fault tolerant model does not address software failures, by far the most common reason for downtime.

High availability views availability not as a series of replicated physical components, but rather as a set of system-wide, shared resources that cooperate to guarantee essential services. High availability combines software with industry-standard hardware to minimize downtime by quickly restoring essential services when a system, component, or application fails. While not instantaneous, services are restored rapidly, often in less than a minute.

Many sites are willing to absorb a small amount of downtime with high availability rather than pay the much higher cost of providing fault tolerance. Additionally, in most highly available configurations, the backup processors are available for use during normal operation.

High availability systems are an excellent solution for applications that must be restored quickly and can withstand a short interruption should a failure occur. Some industries have applications so time-critical that they cannot withstand even a few seconds of downtime. Many other industries, however, can withstand small periods of time when their database is unavailable. For those industries, PowerHA SystemMirror can provide the necessary continuity of service without total redundancy.

Role of PowerHA SystemMirror

PowerHA SystemMirror has many benefits.

PowerHA SystemMirror helps you with the following:

- The PowerHA SystemMirror planning process and documentation include tips and advice on the best practices for installing and maintaining a highly available PowerHA SystemMirror cluster.
- Once the cluster is operational, PowerHA SystemMirror provides the automated monitoring and recovery for all the resources on which the application depends.
- PowerHA SystemMirror provides a full set of tools for maintaining the cluster while keeping the application available to clients.

PowerHA SystemMirror allows you to:

- Quickly and easily setup a basic two-node cluster by using the typical initial cluster configuration SMIT path or the application configuration assistants (Smart Assists).
- Test your PowerHA SystemMirror configuration by using the Cluster Test Tool. You can evaluate how a cluster behaves under a set of specified circumstances, such as when a node becomes inaccessible, a network becomes inaccessible, and so forth.
- Ensure high availability of applications by eliminating single points of failure in a PowerHA SystemMirror environment.
- Leverage high availability features available in AIX.
- Manage how a cluster handles component failures.
- Secure cluster communications.
- Monitor PowerHA SystemMirror components and diagnose problems that might occur.

Related information

PowerHA SystemMirror for AIX

Application clusters

An *application cluster* is a group of loosely coupled machines networked together, sharing disk resources.

In a cluster, multiple server machines cooperate to provide a set of services or resources to clients.

Clustering two or more servers to back up critical applications is a cost-effective high availability option. You can use more of your site's computing power while ensuring that critical applications resume operations after a minimal interruption caused by a hardware or software failure.

Application clusters also provides a gradual, scalable growth path. It is easy to add a processor to the cluster to share the growing workload. You can also upgrade one or more of the processors in the cluster to a more powerful model. If you were using a fault tolerant strategy, you must add *two* processors, one as a redundant backup that does no processing during normal operations.

PowerHA SystemMirror use of Cluster Aware AIX

PowerHA SystemMirror is built in addition to the core clustering capabilities that are supported in the AIX operating system. PowerHA SystemMirror is supported for all editions of AIX that support Cluster Aware AIX (CAA) capabilities.

CAA and PowerHA SystemMirror use Universal IDs (UID and UUID) to track disks and nodes. Dynamically changing UID and UUID is not supported. The UID and UUID are normally invariant under most circumstances. However, there are known scenarios such as reinstalling the operating system where the UID and UUID can change. If you make changes to the UID and UUID, you must remove and recreate the CAA cluster to ensure all UID and UUIDs are updated.

In AIX Version 7.2, or later, or in IBM AIX 7.1 with Technology Level 4, or later, CAA detects and handles network failures after 20 seconds (default value). To change the default value from 20 seconds, run the **`clmgr modify cluster NETWORK_FAILURE_DETECTION_TIME=<xxx>`** command, where xxx is the number of seconds, in the range 5 - 590.

The following information is about the key components of Cluster Aware AIX that are used as the foundation to build a PowerHA SystemMirror solution stack:

Heartbeat management

By default, PowerHA SystemMirror uses unicast communications for heartbeat. As an alternative, multicast communications may be configured instead of unicast. For multicast, you can optionally select a multicast address, or let Cluster Aware AIX (CAA) automatically assign one. You can specify a multicast address while configuring the cluster, or have a multicast setup by Cluster Aware AIX (CAA) during the configuration based on the network environment. Cluster communication is achieved by communicating over multiple redundant paths of communication. The following redundant paths of communication provide a robust clustering foundation that might not be prone to cluster partitioning:

TCP/IP Networks

PowerHA SystemMirror and Cluster Aware AIX use all network interfaces that are available for cluster communication. All of these interfaces are discovered by default and used for health management and other cluster communication. You can use the PowerHA SystemMirror management interfaces to remove any interface that you do not want to be used for application availability. You can also define the interfaces that you do not want to be used as private interfaces with PowerHA SystemMirror.

SAN based communication

CAA supports storage area network (SAN) fabric-based cluster communication, including heartbeating, for a limited number of adapters. This type of heartbeating is optional and might not work with most environments because of network zoning requirements that allow packets to move from one client to another client by using Small Computer System Interface (SCSI) protocol.

Central cluster-repository based communication

Cluster health and other cluster communication is achieved through the central repository disk. PowerHA SystemMirror 7.2, or later, provides an Automatic Repository Disk Replacement (ARR) function that automatically replaces a failed repository disk with a backup repository disk. The

ARR function is available only if you configure and identify a backup repository disk by using PowerHA SystemMirror.

Network interface failure detection time

PowerHA SystemMirror relies on CAA to monitor and detect network interface failures and node failures. In IBM AIX 7.1 with Technology Level 4, or earlier, CAA detected network failures within a fixed amount of time (5 seconds). If a hardware failure occurred in these versions of the AIX operating system, the failures were reported immediately. This type of reporting is called quick failure process. This detection and reporting process in the AIX operating system is different than how PowerHA SystemMirror Version 6.1 reports and detects failures. In PowerHA SystemMirror 6.1, failures are not declared until the full network failure detection time occurs. This process is called full wait time based on relaxed failure detection.

In AIX Version 7.2, or later, or in IBM AIX 7.1 with Technology Level 4, or later, you can use the `NETWORK_FAILURE_DETECTION_TIME` option with the **`clmgr`** command to set the failure detection time for the network interface. The default value for the `NETWORK_FAILURE_DETECTION_TIME` option is 20 seconds. In AIX Version 7.2, or later, or in IBM AIX 7.1 with Technology Level 4, or later, the failure detection process occurs after the full wait period of the failure detection time. These version of the AIX operating system do not use the quick failure detection process.

To change the default value from 20 seconds for the `NETWORK_FAILURE_DETECTION_TIME` option, run the **`clmgr modify cluster NETWORK_FAILURE_DETECTION_TIME=<xxx>`** command, where `xxx` is one of the following values:

0

If you specify this value and the cluster is synchronized, then the network failure detection occurs after 5 seconds and uses the quick failure detection process. This option was used in IBM AIX 7.1 with Technology Level 4, or earlier.

5 - 590 seconds

If you specify a value in this range and if the cluster is synchronized, the network failure detection occurs after the specified value and uses the full wait time process.

Node failure detection time

PowerHA SystemMirror and CAA can detect failure of a partner node in a cluster when heartbeats are missing from network communication and disk communication. When these communication channels are lost, monitoring is enabled for a set period of time. This monitoring is known as node failure detection time.

To configure node failure detection time, you can use one of the following options:

SMIT

To configure node failure detection time, complete the following steps:

1. From the command line, enter **`smit sysmirror`**.
2. In the SMIT interface, select **Custom Cluster Configurations** > **CLuster Nodes and Networks** > **Manage the Cluster** > **Cluster heartbeat settings**, and press Enter.
3. Complete all required field, and press Enter.

Command line

From the command line, run the **`clmgr modify cluster HEARTBEAT_FREQUENCY=<v1> GRACE_PERIOD=<v2>`** command, where v1 and v2 are values in seconds.

The `HEARTBEAT_FREQUENCY` option is the node communication time-out value. This value is the number of seconds that CAA waits to receive packets from the partner node before completing the next step in the process to determine whether the partner node has failed. Valid values for the The `HEARTBEAT_FREQUENCY` option are 20 - 600 seconds. The default value is 30 seconds. The value for the `HEARTBEAT_FREQUENCY` options must be 10 seconds more than the value used for the `NETWORK_FAILURE_DETECTION_TIME` option.

The `GRACE_PERIOD` option is the additional time for which CAA waits after the value specified for the `HEARTBEAT_FREQUENCY` option. The default value of the `GRACE_PERIOD` option is 10 seconds.

Enhanced event management

CAA generates fine granular storage and network events that are used by PowerHA SystemMirror to provide a better decision-making capability for high availability management.

Manage storage across the nodes

PowerHA SystemMirror uses the storage fencing capabilities of AIX for better storage management across the nodes in the cluster. The fencing capabilities are supported for only disks that are configured with native AIX Multipath I/O (MPIO). PowerHA SystemMirror manages shared disks through the enhanced concurrent volume management method.

Note: PowerHA SystemMirror attempts to use the CAA storage framework fencing capability to prevent access of shared disks by nodes that do not have access to the owning shared volume group. This fencing capability prevents data corruption because of inadvertent access to shared disks from multiple nodes. However, the CAA storage framework fencing capability is supported only for native AIX MPIO.

Related concepts

PowerHA SystemMirror multiple-site solutions
PowerHA SystemMirror supports different types of definitions for sites and site-specific policies for high availability 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 linked cluster
A linked cluster is ideal for situations where each site is at a different geographical location. Typically, the sites are far enough apart so that they cannot conveniently share a common storage area network (SAN). Each site must have its own active repository disk, and any backup repositories. Linked clusters always use unicast to communicate between sites. Linked clusters are a useful part of high availability disaster recovery (HADR).

PowerHA SystemMirror stretched cluster
A stretched cluster is ideal for situations where each site is located in the same geographical location. Typically, all nodes are on a common storage area network (SAN). At a minimum, the active repository and any backup repositories are shared between sites, as are any disks in resource groups that can run at either site. Stretched clusters can support LVM cross-site mirroring, and HyperSwap.

Cluster repository disks
PowerHA SystemMirror uses a shared disk as a central repository for managing the configuration of the cluster. This disk must be accessible by all of the nodes in the cluster by using the standard or stretched cluster deployment method, or must be accessible by all nodes in a site by using the linked cluster deployment method.

Central configuration management
A key element of the cluster configuration is the cluster repository disk. The cluster repository disk is used as the central repository for the cluster configuration data.

Related information

Setting up cluster storage communication

Central configuration management

A key element of the cluster configuration is the cluster repository disk. The cluster repository disk is used as the central repository for the cluster configuration data.

The cluster repository disk must be accessible from all nodes in the cluster and it must be at least 512 MB and no more than 460 GB for a PowerHA SystemMirror cluster configuration. It must be backed up by a redundant and highly available SAN configuration. The central repository disk must be appropriately configured for Redundant Array of Independent Disks (RAID) to avoid having a single point of failure.

The cluster repository disk is a special device for the cluster. It has special characteristics and is uniquely managed by Cluster Aware AIX (CAA) and PowerHA SystemMirror. You must not directly use LVM or file system commands and interfaces because they can cause corruption of the cluster repository disk. The cluster repository disk is privately managed by CAA. The cluster repository disk must not be administered outside the CAA environment or PowerHA SystemMirror environment.

The cluster repository serves the following purposes for cluster management:

- The configuration of the CAA cluster is centrally maintained and managed on the repository disk.
- For cluster-based communication, a portion of the repository disk is reserved for node-to-node heartbeat and message communication.

  Note: This form of communication is used when all other forms of communications fail.

Related concepts

PowerHA SystemMirror use of Cluster Aware AIX
PowerHA SystemMirror is built in addition to the core clustering capabilities that are supported in the AIX operating system. PowerHA SystemMirror is supported for all editions of AIX that support Cluster Aware AIX (CAA) capabilities.

Setting up cluster SAN communication

PowerHA SystemMirror and Cluster Aware AIX support an additional method of communication between the hosts called storage area network (SAN) communication.

SAN communications exploits the SAN physical links that exist in a typical data center to provide high speed communications between cluster nodes. This communications mechanism provides a fall back function if a failure occurs in the network communications paths between the cluster nodes.

SAN communication works for a set of disk adapters. You must have SAN zoning setup so that the Small Computer System Interface (SCSI) packets flow from host to host through the various switches in between each host.

Setting up SAN communication involves invoking a set of commands on the supported disk adapter to make it SAN communication capable.

In VIOS 2.2.0.11, or later, you can use SAN communication between logical partitions by establishing a virtual local area network though a virtual Ethernet adapter on each VIOS client. You can set up SAN communication through VIOS for both NPIV and vSCSI environments.

Related information

Setting up cluster storage communication

Defining a virtual Ethernet adapter

Availability costs and benefits continuum

PowerHA SystemMirror availability has many costs and benefits.

The following figure shows the costs and benefits of availability technologies.

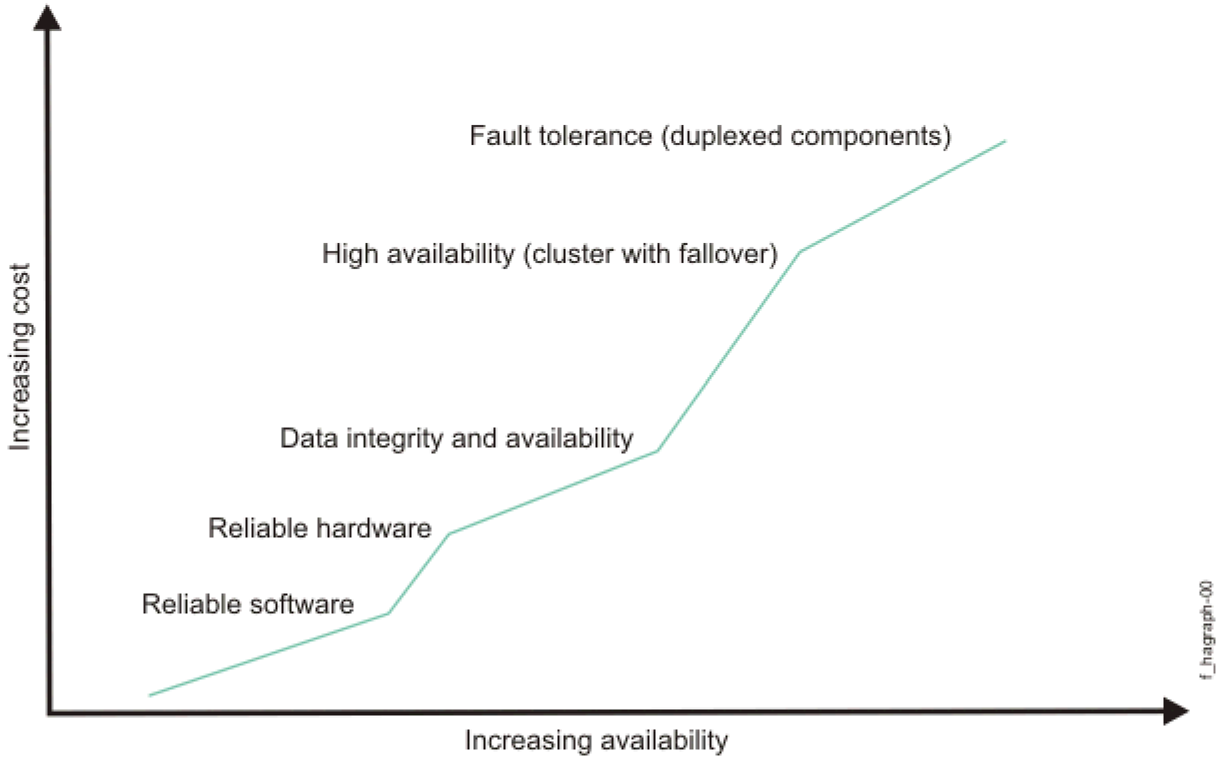


Figure 1. Costs and benefits of available technologies

As you can see, availability is not an all-or-nothing proposition. Think of availability as a continuum. Reliable hardware and software provide the base level of availability. Advanced features such as RAID devices provide an enhanced level of availability. High availability software provides near continuous access to data and applications. Fault tolerant systems ensure the constant availability of the entire system, but at a higher cost.

Enhancing availability with the AIX software

PowerHA SystemMirror takes advantage of the features in AIX - the high-performance UNIX operating system. AIX adds new functionality to further improve security and system availability.

This includes improved availability of mirrored data and enhancements to Workload Manager that help solve problems of mixed workloads by dynamically providing resource availability to critical applications. PowerHA SystemMirror provides both horizontal and vertical scalability without downtime for your system. The AIX operating system provides numerous features designed to increase system availability by lessening the impact of both planned (data backup, system administration), unplanned (hardware or software failure) downtime, and flexibility in hardware resource management by using Capacity on Demand (CoD) functions.

The AIX operating system provides the following features:

- Journaled File System and Enhanced Journaled File System
- Disk mirroring
- Process control
- DLPAR and CoD
- Workload Partitions

Journaled file system and enhanced journaled file system

The AIX native file system, the Journaled File System (JFS), uses database journaling techniques to maintain its structural integrity. System or software failures do not leave the file system in an unmanageable condition. When rebuilding the file system after a major failure, AIX uses the JFS log to restore the file system to its last consistent state. Journaling thus provides faster recovery than the standard UNIX file system consistency check (fsck) utility. In addition, the Enhanced Journaled File System (JFS2) is available in AIX.

Disk mirroring

Disk mirroring software provides data integrity and online backup capability. It prevents data loss due to disk failure by maintaining up to three copies of data on separate disks so that data is still accessible after any single disk fails. Disk mirroring is transparent to the application. No application modification is necessary because mirrored and conventional disks appear the same to the application.

Process control

The AIX System Resource Controller (SRC) monitors and controls key processes. The SRC can detect when a process terminates abnormally, log the termination, pass messages to a notification program, and restart the process on a backup processor.

Dynamic LPAR management

PowerHA SystemMirror can move application resources between LPARs and can perform the necessary dynamic resource adjustments through the Resource Optimized High Availability (ROHA) function. The ROHA uses the features that are available with IBM Power Systems servers to dynamically manage the following types of hardware resources:

- Capacity on Demand (CoD) functions (including On/Off CoD and Enterprise Pool CoD) manage memory and CPU resources at the frame (CEC) level.
- DLPAR functions manage memory and CPU resources at the logical partition level.

With the ROHA function, you can use PowerHA SystemMirror to optimize the amount of resources for each application. For example, during a takeover the hardware resources (CPU and memory) are dynamically released from the active node, and dynamically acquired and allocated to the standby node.

Workload Partitions

Workload Partitions allow multiple instances of an application to run within the same operating system environment while providing isolation between those environments, thus providing protection and isolation between instances.

Cluster Aware AIX (CAA)

PowerHA SystemMirror uses the AIX clustering capabilities to provide for an advanced high availability solution. CAA provides the AIX kernel with heartbeating and health management. PowerHA SystemMirror monitors for fine granular storage and network events and handles the critical situations in the AIX operating system. PowerHA SystemMirror can discover hardware components, thus making it easier to manage and deploy clusters.

Physical components of a PowerHA SystemMirror cluster

PowerHA SystemMirror provides a highly available environment by identifying a set of resources essential to uninterrupted processing. It also defines a protocol that nodes use to collaborate to ensure that these resources are available.

PowerHA SystemMirror extends the clustering model by defining relationships among cooperating processors where one processor provides the service offered by a peer should the peer be unable to do so. As shown in the following figure, a PowerHA SystemMirror cluster is made up of the following physical components:

- Nodes
- Shared external disk devices

- Networks
- Network interfaces
- Cluster repository disk
- Clients.

The PowerHA SystemMirror software allows you to combine physical components into a wide range of cluster configurations, providing you with flexibility in building a cluster that meets your processing requirements. This figure shows one example of a PowerHA SystemMirror cluster. Other PowerHA SystemMirror clusters could look very different - depending on the number of processors, the choice of networking and disk technologies, and so on.

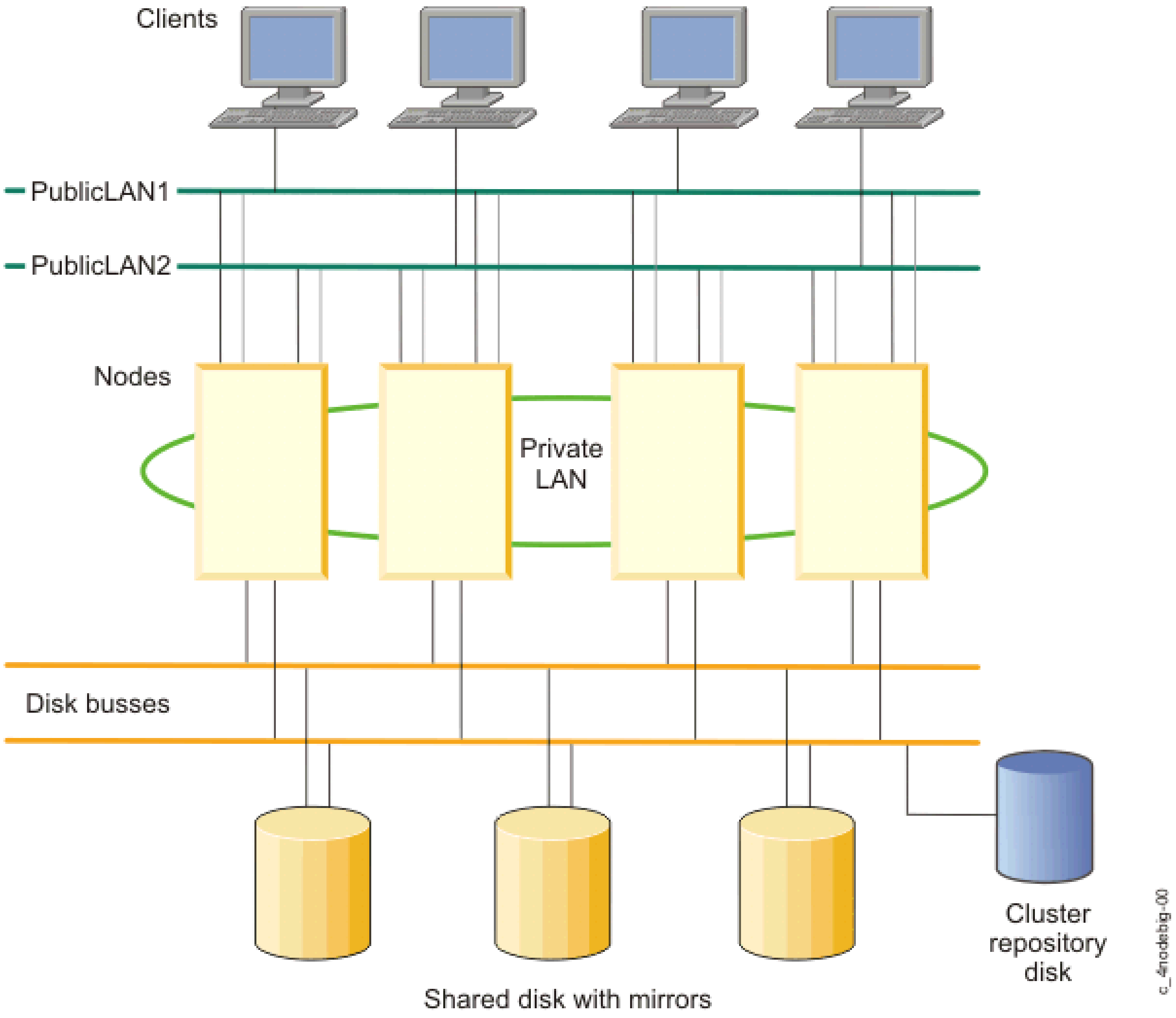


Figure 2. Example of a PowerHA SystemMirror cluster

PowerHA SystemMirror cluster configurations provides examples of cluster configurations supported by the PowerHA SystemMirror software.

Related concepts

PowerHA SystemMirror cluster configurations

This chapter provides examples of the types of cluster configurations supported by the PowerHA SystemMirror software.

PowerHA SystemMirror nodes

Nodes form the core of a PowerHA SystemMirror cluster. A node is a processor that runs AIX, the PowerHA SystemMirror software, and the application software.

Clustering these servers to back up critical applications is a cost-effective high availability option. A business can use more of its computing power while ensuring that its critical applications resume running after a short interruption caused by a hardware or software failure.

In a PowerHA SystemMirror cluster, each node is identified by a unique name. A node can own a set of resources: disks, volume groups, file systems, networks, network addresses, and applications. Typically, a node runs a server or a back end application that accesses data on the shared external disks.

Related concepts

Ensuring application availability
This section describes how the PowerHA SystemMirror software ensures application availability by ensuring the availability of cluster components. PowerHA SystemMirror eliminates single points of failure for all key system components, and eliminates the need for scheduled downtime for most routine cluster maintenance tasks.

PowerHA SystemMirror resources and resource groups
Look here for resource-related concepts and definitions that are used throughout the documentation and also in the PowerHA SystemMirror user interface.

Shared external disk devices

Each node has access to one or more shared external disk devices. A *shared external disk device* is a disk physically connected to multiple nodes.

The shared disk stores mission-critical data, typically mirrored or RAID-configured for data redundancy. A node in a PowerHA SystemMirror cluster must also have internal disks that store the operating system and application binaries, but these disks are not shared. Depending on the type of disk used, the PowerHA SystemMirror software supports the following types of access to shared external disk devices - nonconcurrent access and concurrent access.

- In *nonconcurrent access environments*, only one connection is active at any given time, and the node with the active connection owns the disk. When a node fails, the node that currently owns the disk leaves the cluster, disk takeover occurs and a surviving node assumes ownership of the shared disk. This typical cluster configuration is used by most applications.
- In *concurrent access environments*, the shared disks are actively connected to more than one node simultaneously. Therefore, when a node fails, disk takeover is not required. This type of access is used only by applications that can manage and coordinate simultaneous access to shared data from multiple nodes.

Note that in such environments, either all nodes defined in the cluster can be part of the concurrent access, or only a subset of cluster nodes can have access to shared disks. In the second case, you configure resource groups only on those nodes that have shared disk access. The differences between these methods are explained more fully in the section Shared external disk access.

Related concepts

Shared external disk access

The PowerHA SystemMirror software supports two methods of shared external disk access: nonconcurrent and concurrent.

Networks

As an independent, layered component of the AIX operating system, the PowerHA SystemMirror software is designed to work with any TCP/IP-based network.

Nodes in a PowerHA SystemMirror cluster use the network to:

- Allow clients to access the cluster nodes
- Enable cluster nodes to exchange heartbeat messages
- Serialize access to data (in concurrent access environments)

The PowerHA SystemMirror software has been tested with Ethernet and storage area networks.

Types of networks

The PowerHA SystemMirror software defines two types of communication networks, characterized by whether these networks use communication interfaces based on the TCP/IP subsystem (TCP/IP-based) or storage area networks.

TCP/IP-based network

Connects two or more server nodes, and optionally allows client access to these cluster nodes, using the TCP/IP protocol. By default, PowerHA SystemMirror uses unicast communications for heartbeat. You can optionally select to use multicast communications if your network is configured to support multicast.

Storage Area Network (SAN)

Provides communication between PowerHA SystemMirror cluster nodes for control message and heartbeat traffic. This is an alternate communications path to the TCP/IP network.

Cluster repository disks

PowerHA SystemMirror uses a shared disk as a central repository for managing the configuration of the cluster. This disk must be accessible by all of the nodes in the cluster by using the standard or stretched cluster deployment method, or must be accessible by all nodes in a site by using the linked cluster deployment method.

Use the System Management Interface Tool (SMIT) interface to manage and configure the cluster repository disk and the backup repository disks.

You must have at least 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.

With Cluster Aware AIX the cluster repository disk is used for the following purposes:

- Cluster-wide configuration management
- Cluster messaging and heartbeating. The repository disk is used as another redundant path of communication between the nodes.

You must have at least 512 MB and no more than 460 GB of disk space that is allocated for the cluster repository disk.

Verify that the disk you select as the repository disk does not have a reservation policy. To check the disks reservation policy, run the **`lsattr -El hdisk - a reserve_policy`** command. To change the disk reservation policy to `no_reserve`, run the **`chdev –a reserve_policy=no_reserve –l hdisk`** command.

After you assign a disk as the repository disk, the disk cannot be used for any other purposes. Verify that the disk you use as the repository disk does not contain any user data. When the disk is identified as a repository disk by PowerHA SystemMirror, all information on the disk is erased.

When you plan to use the disks as repository disks, you must plan for 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 might be in a different physical storage disk. Update your administrative procedures and documentation with the backup disk information. The cluster tolerates the loss or failure of the repository disk. Critical operations continue without a repository disk. However, you must quickly correct the problem with the repository disk. If you are using PowerHA SystemMirror 7.2, or earlier, you must manually replace the failed repository disk with a repository disk from the backup list. You can also replace a working repository disk with a new disk to increase the size or to change to a different storage subsystem.

You can configure PowerHA SystemMirror 7.2, or later, to use the Automatic Repository Disk Replacement (ARR) function in Cluster Aware AIX (CAA) during the failure of an active repository disk. The ARR function automatically replaces the failed repository disk with a disk from the backup repository disks. The first backup repository disk in the list replaces the failed repository disk.

You should configure all repository disks with AIX native Mulitple Path I/O (MPIO). The AIX MPIO enables PowerHA SystemMirror to monitor and respond better to disk failures by using repository disks.

Mirror pools and repository disks

If you use PowerHA SystemMirror and Logical Volume Manager (LVM) for managing mirror pools across different locations and the repository disk fails, the new repository disk must not be from the same location where the failure occurred.

Related concepts

PowerHA SystemMirror linked cluster
A linked cluster is ideal for situations where each site is at a different geographical location. Typically, the sites are far enough apart so that they cannot conveniently share a common storage area network (SAN). Each site must have its own active repository disk, and any backup repositories. Linked clusters always use unicast to communicate between sites. Linked clusters are a useful part of high availability disaster recovery (HADR).

PowerHA SystemMirror stretched cluster
A stretched cluster is ideal for situations where each site is located in the same geographical location. Typically, all nodes are on a common storage area network (SAN). At a minimum, the active repository and any backup repositories are shared between sites, as are any disks in resource groups that can run at either site. Stretched clusters can support LVM cross-site mirroring, and HyperSwap.

PowerHA SystemMirror use of Cluster Aware AIX
PowerHA SystemMirror is built in addition to the core clustering capabilities that are supported in the AIX operating system. PowerHA SystemMirror is supported for all editions of AIX that support Cluster Aware AIX (CAA) capabilities.

Related information

Planning for repository disk

Replacing a repository disk with SMIT

Repository disks and multipathing

The AIX operating system manages repository disks in a special way during configuration of a cluster.

AIX cluster services recognize the repository disk during the AIX device configuration process and can perform special actions on the disk. For example, the repository disk is used by the cluster services to enable disk-based communication across hosts. Disk communication involves one host writing to the disk and the other host or hosts retrieving the message through a polling mechanism.

Cluster services also generate detailed events in relation to the health of the repository disk. If a repository disk fails, CAA uses an Automatic Repository Replacement (ARR) function that automatically replaces a failed repository disk with a backup repository disk. The ARR function is available only if you identify a backup repository disk with PowerHA SystemMirror.

You can configure the following types of disks as repository disks:

AIX multipath disks (AIX local MPIO)
These disks can be automatically created and used as repository disks. This is the recommended type of disk to use as a repository disk.

virtual SCSI (vSCSI) disk
These disks can be mapped through Virtual I/O Server (VIOS) as vSCSI disk to the client logical partition (LPAR). When the vSCSI disk is used as a repository disk, it must be mapped to a different PowerHA SystemMirror node or a different Cluster Aware AIX (CAA) node using the same vSCSI methods.

Third-party multipath disks
These disks follow the guidelines of AIX multipathing concepts, but provide their own multipathing device drivers and software. These disks can be configured as repository disks when the relevant Object Data Manager (ODM) information is available. Disks managed by EMC PowerPath and Hitachi Dynamic Link Manager (HDLM) can be configured as repository disks using this method.

To upgrade the MPIO software or third-party multipath software, you must stop CAA cluster services by entering the **`clmgr offline cluster STOP_CAA=yes`** command.

Repository disk failure

You must plan correctly for repository disk failure and what is needed to correct issues related to repository disk failure.

Repository disk failure is tolerated by a PowerHA SystemMirror cluster. If any node in the cluster encounters errors with the repository disk or with accessing the repository disk, the cluster enters a limited or restricted mode. In this mode of operation, you cannot use most topology-related operations. For example, a node cannot be added or a node cannot join the cluster. However, critical cluster functions can be performed. For example, you can move a resource group from an active node to a standby node.

When the repository disk fails, the administrator is notified about the disk failure. PowerHA SystemMirror continues to notify the administrator about the repository disk failure until it is resolved. To get notifications or customize event processing when a repository disk fails, see the Configuring pre-event and post-event processing topic.

PowerHA SystemMirror and Cluster Aware AIX (CAA) support live repository disk replacement, which you can use to replace a failed or working repository disk. CAA repopulates the new disk with cluster information and starts to use the disk as the repository disk.

PowerHA SystemMirror 7.2.0, or later, supports Automatic Repository Disk Replacement (ARR) function. ARR uses CAA and automatically replaces a failed repository disk with a backup repository disk. The ARR function is available only if you configure a backup repository disk with PowerHA SystemMirror.

To use the ARR function, your environment must meet the following requirements:

- A cluster or site has a backup repository disk identified.
- PowerHA SystemMirror Version 7.2.0, or later, is installed.
- One of the following versions of the AIX operating system is installed:
  - AIX Version 7.1.4, or later
  - AIX Version 7.2.0, or later

CAA monitors repository disk failure by checking I/O errors and by verifying that the disk is in an active state. These verification checks occur periodically and are not performed every time the repository disk is read from or written to. Do not write directly to the repository disk, even for testing purposes. Writing directly to the repository disk asynchronously might cause the operating system and CAA operations to be disrupted abruptly, resulting in unpredictable results.

Related information

dd command

Planning for repository disk

Replacing a repository disk with SMIT

Clients

A client is a processor that can access the nodes in a cluster over a local area network.

Clients each run a "front end" or client application that queries the server application running on the cluster node. The PowerHA SystemMirror software provides a highly available environment for critical data and applications on cluster nodes. The PowerHA SystemMirror software does not make the clients themselves highly available. AIX clients can use the Cluster Information (Clinfo) services to receive notice of cluster events. Clinfo provides an API that displays cluster status information. PowerHA SystemMirror provides a cluster status utility, the **/usr/es/sbin/cluster/clstat**. It is based on Clinfo and reports the status of key cluster components: the cluster itself, the nodes in the cluster, the network interfaces connected to the nodes, and the resource groups on each node. The cluster status interface of the **clstat** utility includes web-based, Motif-based and ASCII-based versions.

Related concepts

Cluster information program
The Cluster Information Program (Clinfo), the **clinfo** daemon, is an SNMP-based monitor.

Goal of PowerHA SystemMirror: Eliminating scheduled downtime

The primary goal of high availability clustering software is to minimize, or ideally, eliminate, the need to take your resources out of service during maintenance and reconfiguration activities.

PowerHA SystemMirror software optimizes availability by allowing for the dynamic reconfiguration of running clusters. Most routine cluster maintenance tasks, such as adding or removing a node or changing the priority of nodes participating in a resource group, can be applied to an active cluster without stopping and restarting cluster services. In addition, you can keep a PowerHA SystemMirror cluster online while making configuration changes by using the Cluster Single Point of Control (C-SPOC) facility. *C-SPOC* makes cluster management easier because you can change to shared volume groups, users, and groups across the cluster from a single node. The changes are propagated transparently to other cluster nodes.

PowerHA SystemMirror cluster nodes, networks, and heartbeating concepts

This section introduces major cluster topology-related concepts and definitions that are used throughout the documentation and in the PowerHA SystemMirror user interface.

Nodes

A node is a processor that runs both AIX and the PowerHA SystemMirror software.

Nodes might share a set of resources such as, disks, volume groups, file systems, networks, network IP addresses, and applications. The PowerHA SystemMirror software supports up to 16 nodes in a cluster. In a PowerHA SystemMirror cluster, each node is identified by a unique name. In PowerHA SystemMirror, a node name and a hostname can usually be the same. Nodes serve as core physical components of a PowerHA SystemMirror cluster. For more information on nodes and hardware, see the section Nodes. Two types of nodes are defined:

- Server nodes form the core of a PowerHA SystemMirror cluster. Server nodes run services or back end applications that access data on the shared external disks.
- Client nodes run front end applications that retrieve data from the services provided by the server nodes. Client nodes can run PowerHA SystemMirror software to monitor the health of the nodes, and to react to failures.

Server nodes

A cluster server node usually runs an application that accesses data on the shared external disks. Server nodes run PowerHA SystemMirror daemons and keep resources highly available. Typically, applications are run, storage is shared between these nodes, and clients connect to the server nodes through a service IP address.

Client nodes

A full high availability solution typically includes the client machine that uses services provided by the servers. Client nodes can be divided into two categories: naive and intelligent.

- A naive client views the cluster as a single entity. If a server fails, the client must be restarted, or at least must reconnect to the server.
- An intelligent client is cluster-aware. A cluster-aware client reacts appropriately in the face of a server failure, connecting to an alternate server, perhaps masking the failure from the user. Such an intelligent client must have knowledge of the cluster state.

PowerHA SystemMirror extends the cluster paradigm to clients by providing both dynamic cluster configuration reporting and notification of cluster state changes, such as changes in subsystems or node failure.

Sites

You can define a group of one or more server nodes as belonging to a site.

The site becomes a component, such as a node or a network, that is known to the PowerHA SystemMirror software. PowerHA SystemMirror supports clusters that are divided into two sites.

You can configure split-site Logical Volume Manager (LVM) mirroring by associating specific LVM mirror pools with a physical site. If you then specify to PowerHA SystemMirror, which physical volumes are located at each site, C-SPOC displays the site information when selecting volumes from sites for LVM mirrors pools. During cluster verification, PowerHA SystemMirror performs additional checks to make sure that the mirror definitions are consistent with the site definitions.

In addition, the Extended Distance function of PowerHA SystemMirror Enterprise Edition provides two distinct software solutions for disaster recovery. These solutions enable a PowerHA SystemMirror cluster to operate over extended distances at two sites.

PowerHA SystemMirror Enterprise Edition for Metro Mirror increases data availability for DS8000® volumes, DS6000™ volumes, and IBM TotalStorage™ Enterprise Storage Server® (ESS) volumes that use Peer-to-Peer Remote Copy (PPRC) to copy data to a remote site for disaster recovery purposes. PowerHA SystemMirror Enterprise Edition for Metro Mirror takes advantage of the PPRC fallover and fallback functions and of PowerHA SystemMirror cluster management to reduce downtime and recovery time during disaster recovery.

When PPRC is used for data mirroring between sites, the physical distance between sites is limited to the capabilities of the ESS hardware.

PowerHA SystemMirror Enterprise Edition for Geographic Logical Volume Manager (GLVM) increases data availability for IBM volumes that use GLVM to copy data to a remote site for disaster recovery purposes. PowerHA SystemMirror Enterprise Edition for GLVM takes advantage of the following components to reduce downtime and recovery time during disaster recovery:

AIX and PowerHA SystemMirror Enterprise Edition for GLVM data mirroring and synchronization. Both standard and enhanced concurrent volume groups can be geographically mirrored with the GLVM utilities.

TCP/IP-based unlimited distance network support up to four XD_data data mirroring networks can be configured.

If a component fails, PowerHA SystemMirror ensures that a mirrored copy of the data is accessible at either a local or remote site. Both concurrent and nonconcurrent resource groups can be configured in a PowerHA SystemMirror cluster with GLVM. However, intersite policy cannot be concurrent.

Related information

Planning guide

Metro Mirror: Planning and administration

Geographic LVM Planning and administration

Split policy

A cluster split event can occur between sites when a group of nodes cannot communicate with the remaining nodes in a cluster. For example, in a linked cluster, a split occurs if all communication links between the two sites fail. A cluster split event splits the cluster into two or more partitions.

You can use PowerHA SystemMirror to configure a split policy that specifies the response to a cluster split event.

The following options are available for configuring a split policy:

None

This option indicates that no action occurs when a cluster split event is detected. Each partition that is created by the cluster split event becomes an independent cluster. Each partition can start a workload independent of the other partition. If shared volume groups are in use, it can potentially lead to data corruption. This option is the default setting, since manual configuration is required to establish an alternative policy. Do not use this option if your environment is configured to use HyperSwap for PowerHA SystemMirror.

Tie breaker

You can use this option to specify a disk or an NFS file.

If you specify a disk for the tie breaker, each partition attempts to acquire the tie breaker disk by placing a lock on the tie breaker disk. If you specify a disk for the tie breaker, a SCSI disk that is assessable to all nodes in the cluster is used. The partition that cannot lock the disk is rebooted, as specified in the action plan.

If you specified an NFS file for the tie-breaker, the NFS mount must exist on each of the nodes in the cluster from the selected NFS server. The partition that first reserves the NFS file continues to function. The partition that cannot lock the NFS file is rebooted, as specified in the action plan.

Note: The default NFS mount options are `vers=4,fg,soft,retry=1,timeo=10`. Modifying the default values might lead to failure in acquiring the NFS lock.

Cloud is another tiebreaker option and you must have cloud communication on all the nodes of the cluster for this option. During cluster split event, each partition attempts to acquire a lock by uploading a file to the configured Cloud service. The partition that successfully uploads the file to the configured Cloud service continues to function. The partition that cannot upload the file to the configured Cloud service is rebooted or the cluster services are restarted as specified by the chosen action plan in the policy setting.

If you use the **Cloud** option for the split policy, the merge policy must also be configured to use the **Cloud** option.

Manual

This option indicates that you want to manually fix the problem when a cluster split occurs.

Each node in the partition presents a message to choose to continue running cluster services or recover cluster services (which restarts the node). With this option, you can specify the number of attempts and the frequency of attempts that require your input. You can also specify a default action to occur after the number of attempts that require your input is reached and you have not provided any input.

The following message is displayed for a linked cluster that specifies the manual option when a cluster split event occurs:

```
Broadcast message from root@e08m138.ausprv.stglabs.ibm.com (tty) at 04:09:48 ...
A cluster split has been detected.
You must decide if this side of the partitioned cluster is to continue.
To have it continue, enter
        /usr/es/sbin/cluster/utilities/cl_sm_continue
To have the recovery action - Reboot - taken on all nodes on this partition, enter
        /usr/es/sbin/cluster/utilities/cl_sm_recover
LOCAL_PARTITION 1 e08m138 OTHER_PARTITION 2 e08m140
```

In this example, you can use the manual option to check whether a split event or a merger event is waiting for a manual response from the **Problem Determination Tools** > **Manual Response to Split or Merge** > **Display any needed Manual Response** SMIT menu.

If you want to use the manual option for stretched clusters and standard clusters, your environment must be running the following versions of software:

- IBM AIX 7.2 with Technology Level 1, or later
- PowerHA SystemMirror Version 7.2.1, or later

Note: For any type of cluster that uses the manual option after the number of attempts specified is reached and you have not provided any input, the partition that has the lowest node ID is chosen as the winning partition.

Related concepts

Merge policy
Depending on the cluster split policy, the cluster might have two partitions that run independently of each other. You can use PowerHA SystemMirror to configure a merge policy that allows the partitions to operate together again after communications are restored between the partitions.

Tie breaker option for split and merge policies
You can use the tie breaker option to specify a SCSI disk or a Network File System (NFS) file that is used by the split and merge policies.

PowerHA SystemMirror linked cluster
A linked cluster is ideal for situations where each site is at a different geographical location. Typically, the sites are far enough apart so that they cannot conveniently share a common storage area network (SAN). Each site must have its own active repository disk, and any backup repositories. Linked clusters always use unicast to communicate between sites. Linked clusters are a useful part of high availability disaster recovery (HADR).

Related information

Configuring a split and merge policy

Merge policy

Depending on the cluster split policy, the cluster might have two partitions that run independently of each other. You can use PowerHA SystemMirror to configure a merge policy that allows the partitions to operate together again after communications are restored between the partitions.

The following options are available for configuring a merge policy:

Majority

The partition with the highest number of nodes remains online. If each partition has the same number of nodes, then the partition that has the lowest node ID is selected for standard and stretched clusters. The lowest site ID is selected for linked cluster. The partition that does not remain online is rebooted, as specified by the selected action plan. For stretched clusters to use the majority option, your environment must be running one of the following version of the AIX operating system:

- IBM AIX 7.1 with Technology Level 4, or later
- AIX Version 7.2, or later

Tie breaker

You can use a disk or an NFS file for a tie-breaker. If you use a disk for the tie-breaker, each partition attempts to acquire the tie breaker by placing a lock on the tie breaker disk. The tie breaker is a SCSI disk that is accessible to all nodes in the cluster. The partition that cannot lock the disk is rebooted, or cluster services are restarted, as specified by the chosen action plan.

If you use an NFS file for the tie-breaker, the NFS mount must exist on each of the nodes in the cluster from the selected NFS server. The partition that first reserves the NFS file continues to function. The partition that cannot lock the NFS file is rebooted, or cluster services are restarted, as specified by the chosen action plan.

If you use this option, your split policy configuration must also use the tie breaker option.

Note: The default NFS mount options are `vers=4,fg,soft,retry=1,timeo=10`. Modifying the default values might lead to failure in acquiring the NFS lock.

Cloud is another tie breaker option and you must have cloud communication on all the nodes of the cluster for this option. During cluster merge event, each partition attempts to acquire a lock by uploading a file to the configured Cloud service. The partition that successfully uploads the file to the configured Cloud service continues to function. The partition that cannot upload the file to the configured Cloud service is rebooted or the cluster services are restarted as specified by the chosen action plan in the policy setting.

If you use the **Cloud** option for the merge policy, the split policy must also be configured to use the **Cloud** option.

Manual

This policy option requires that you select the winning site during a merge event.

Each node in the partition presents a message to choose to continue running cluster services or recover cluster services (which restarts the node). With this option, you can specify the number of attempts and the frequency of attempts that require your input. You can also specify a default action to occur after the number of attempts that require your input is reached and you have not provided any input.

The following message is displayed for a linked cluster that specifies the manual option when a cluster split event occurs:

```
Broadcast message from root@e08m138.ausprv.stglabs.ibm.com (tty) at 04:09:48 ...
A cluster split has been detected.
You must decide if this side of the partitioned cluster is to continue.
To have it continue, enter
        /usr/es/sbin/cluster/utilities/cl_sm_continue
To have the recovery action - Reboot - taken on all nodes on this partition, enter
        /usr/es/sbin/cluster/utilities/cl_sm_recover
LOCAL_PARTITION 1 e08m138 OTHER_PARTITION 2 e08m140
```

In this example, you can use the manual option to check whether a split event or a merger event is waiting for a manual response from the **Problem Determination Tools** > **Manual Response to Split or Merge** > **Display any needed Manual Response** SMIT menu.

If you want to use the manual option for stretched clusters and standard clusters, your environment must be running the following versions of software:

- IBM AIX 7.2 with Technology Level 1, or later
- PowerHA SystemMirror Version 7.2.1, or later

Note: For any type of cluster that uses the manual option after the number of attempts specified is reached and you have not provided any input, the partition that has the lowest node ID is chosen as the winning partition.

None

This option indicates that no action occurs when a cluster merge event occurs. To avoid any data corruption after a merge occurs, you must reboot the losing partition node manually. This option is only available from the **`clmgr`** command. If you specify none for the merge policy, you must select none for the split policy.

If you want to use the none option for stretched clusters and standard clusters, your environment must be running the following versions of software:

- IBM AIX 7.2 with Technology Level 1, or later
- PowerHA SystemMirror Version 7.2.1, or later

Related concepts

Split policy

A cluster split event can occur between sites when a group of nodes cannot communicate with the remaining nodes in a cluster. For example, in a linked cluster, a split occurs if all communication links between the two sites fail. A cluster split event splits the cluster into two or more partitions.

Tie breaker option for split and merge policies
You can use the tie breaker option to specify a SCSI disk or a Network File System (NFS) file that is used by the split and merge policies.

PowerHA SystemMirror linked cluster
A linked cluster is ideal for situations where each site is at a different geographical location. Typically, the sites are far enough apart so that they cannot conveniently share a common storage area network (SAN). Each site must have its own active repository disk, and any backup repositories. Linked clusters always use unicast to communicate between sites. Linked clusters are a useful part of high availability disaster recovery (HADR).

Related information

Configuring a split and merge policy

Tie breaker option for split and merge policies

You can use the tie breaker option to specify a SCSI disk or a Network File System (NFS) file that is used by the split and merge policies.

A tie breaker disk or an NFS file is used when the sites in the cluster can no longer communicate with each other. This communication failure results in the cluster splitting the sites into two, independent partitions. If failure occurs because the cluster communication links are not responding, both partitions attempt to lock the tie breaker disk or the NFS file. The partition that acquires the tie breaker disk continues to function, while the other partition reboots, or has cluster services restarted, depending on the selected action plan.

The disk or NFS-mounted file that is identified as the tie breaker must be accessible to all nodes in the cluster.

When partitions that were part of the cluster are brought back online after the communication failure, they must be able to communicate with the partition that owns the tie breaker disk or NFS file. If a partition that is brought back online cannot communicate with the tie breaker disk or the NFS file, it does not join the cluster. The tie breaker disk or NFS file is released when all nodes in the configuration rejoin the cluster.

When you configure a tie breaker disk for split and merge recovery handling, the disk must also be supported by the **devrsrv** command, which is part of the AIX operating system. The SMIT interface that you use for selecting the tie breaker disk filters out any disks that do not meet this requirement. To use EMC disk as a tiebreaker disk, configure the EMC disk by using the IBM AIX Multi Path IO (MPIO).

EMC PowerPath disks are not supported for use as a tiebreaker disk.

Site Priority configuration for split and merge policies

Starting with PowerHA SystemMirror Version 7.2.8, or later, and Reliable Scalable Cluster Technology (RSCT) Version 3.3.2.0, or later, configuration of priority is enabled for any PowerHA cluster site. Site Priority configuration enables RSCT to make quorum decisions. The Site Priority configuration also provides an option to add time delay for the low-priority sites so that the higher priority sites have more chance to win the tie breaker.

Note: The Site Priority feature is functional only for tie breaker configuration of SCSI PR disks.

Limitation on the tie breaker option with more than two nodes in a cluster

In a cluster with more than two nodes, after a cluster split occurs, the tie breaker disk or NFS file is reserved by the winning partition. From the winning partition, a single node makes a reservation on the tie breaker disk or NFS file. If this node fails without releasing the reservation on the tie breaker disk or

NFS file, the remaining nodes cannot obtain the reservation on the tie breaker disk or NFS file and lose. Therefore, all nodes are rebooted.

After the cluster split occurs, you must quickly resolve the problem with the failed node so that the tie breaker disk reservation is released.

Note: The default NFS mount options are `vers=4,fg,soft,retry=1,timeo=10`. Modifying the default values might lead to failure in acquiring the NFS lock.

Related concepts

Split policy
A cluster split event can occur between sites when a group of nodes cannot communicate with the remaining nodes in a cluster. For example, in a linked cluster, a split occurs if all communication links between the two sites fail. A cluster split event splits the cluster into two or more partitions.

Merge policy
Depending on the cluster split policy, the cluster might have two partitions that run independently of each other. You can use PowerHA SystemMirror to configure a merge policy that allows the partitions to operate together again after communications are restored between the partitions.

Quarantine policy

A quarantine policy isolates the previously active node that was hosting a critical resource group after a cluster split event or node failure occurs. The quarantine policy ensures that your application data is not corrupted or lost.

Quarantine policy includes two options, active node halt policy and disk fencing.

Active node halt policy

If a cluster split occurs, the active node halt policy stops the previous active LPAR so that the LPAR hosting the application is completely quieced before the application is brought back online by the standby LPAR.

This process ensures that the application is operating on only one node at a time in the cluster. PowerHA SystemMirror brings the active LPAR offline by running commands from the Hardware Management Console (HMC). If you configured the LPAR policy to reboot automatically in the AIX operating system, you must configure the PowerHA SystemMirror resource group boot time policy to be set up for manual restart.

The following figure displays the HMC-based active node halt policy implementation. In the following figure, the standby LPAR runs a command on the HMC to stop the active LPAR. When the command successfully takes the active LPAR offline, the standby LPAR starts the resource group on the standby node.

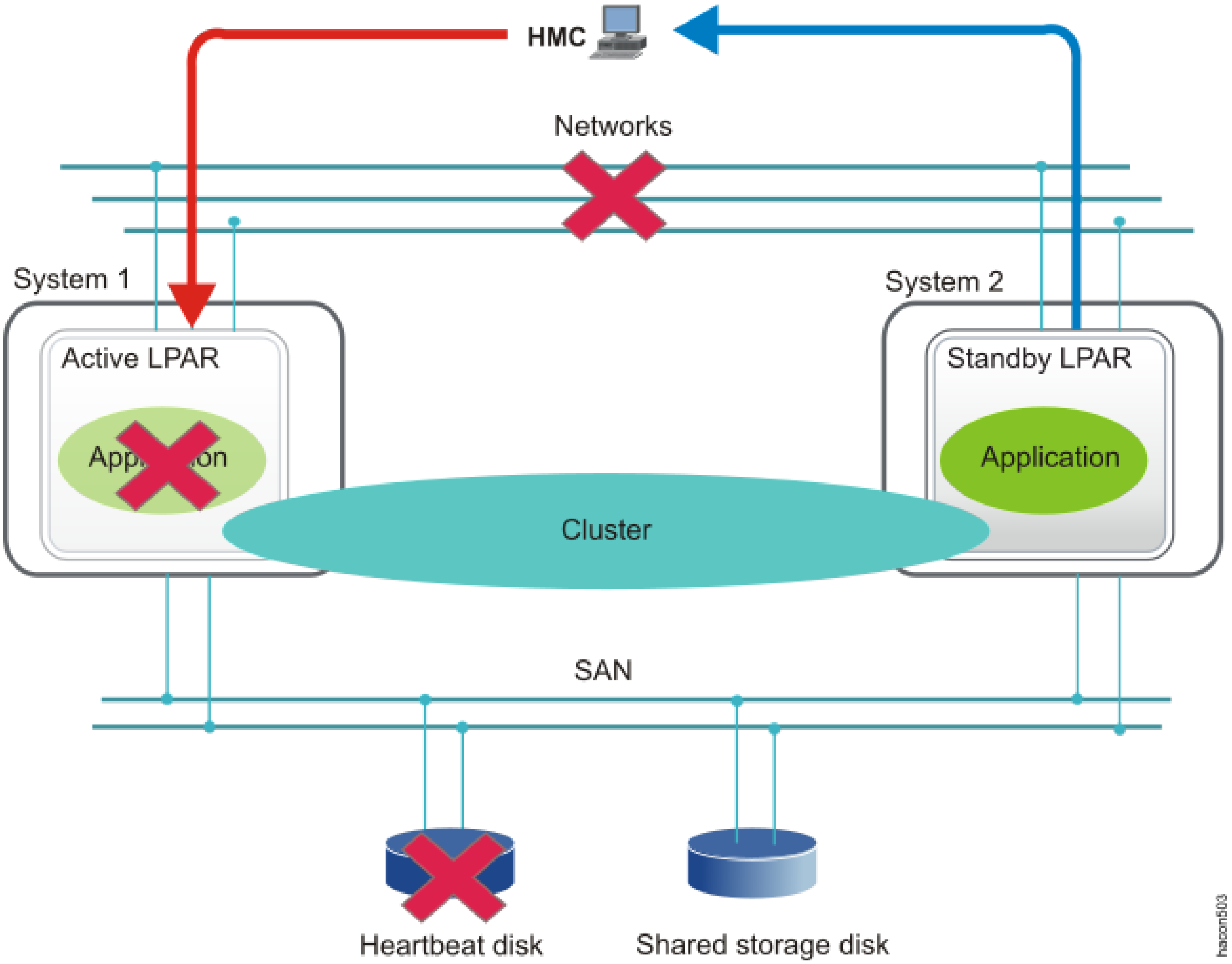


Figure 3. Active node halt policy

Note: If an error occurs that results in the active LPAR not being brought offline, the standby LPAR enters an error state and requires manual intervention to fix the problem.

Disk fencing policy

PowerHA SystemMirror provides protection against a split cluster by using a disk fencing policy. The disk fencing policy uses an SCSI-3 reservation function to separate (fence out) the node with problems that is hosting the workload from the cluster. In this scenario, the workload that was running on the problematic node is started on the standby LPAR.

The fencing process ensures that standalone nodes have access to the disks and that the data remains protected. The disk fencing policy is supported for an Active-Passive deployment model. Disk fencing ensures that a workload can run with write access on only one node in the cluster. PowerHA SystemMirror registers the disks of all the volume groups that are part of any resource group.

The following figure displays what occurs when a cluster split occurs when using a disk fencing policy. In the following figure, the standby LPAR communicates with the shared storage disk and requests that access to the active LPAR disk is revoked. The shared storage disk blocks any write access from the previously active LPAR, even if the active LPAR is restarted. The standby LPAR brings the application or resource group online if PowerHA SystemMirror can fence out the disks for the resource groups in the active LPAR. If errors occur in the standby LPAR while fencing out the disks in the active LPAR, the applications are not brought online and you must correct the problems and manually bring the resource groups back online.

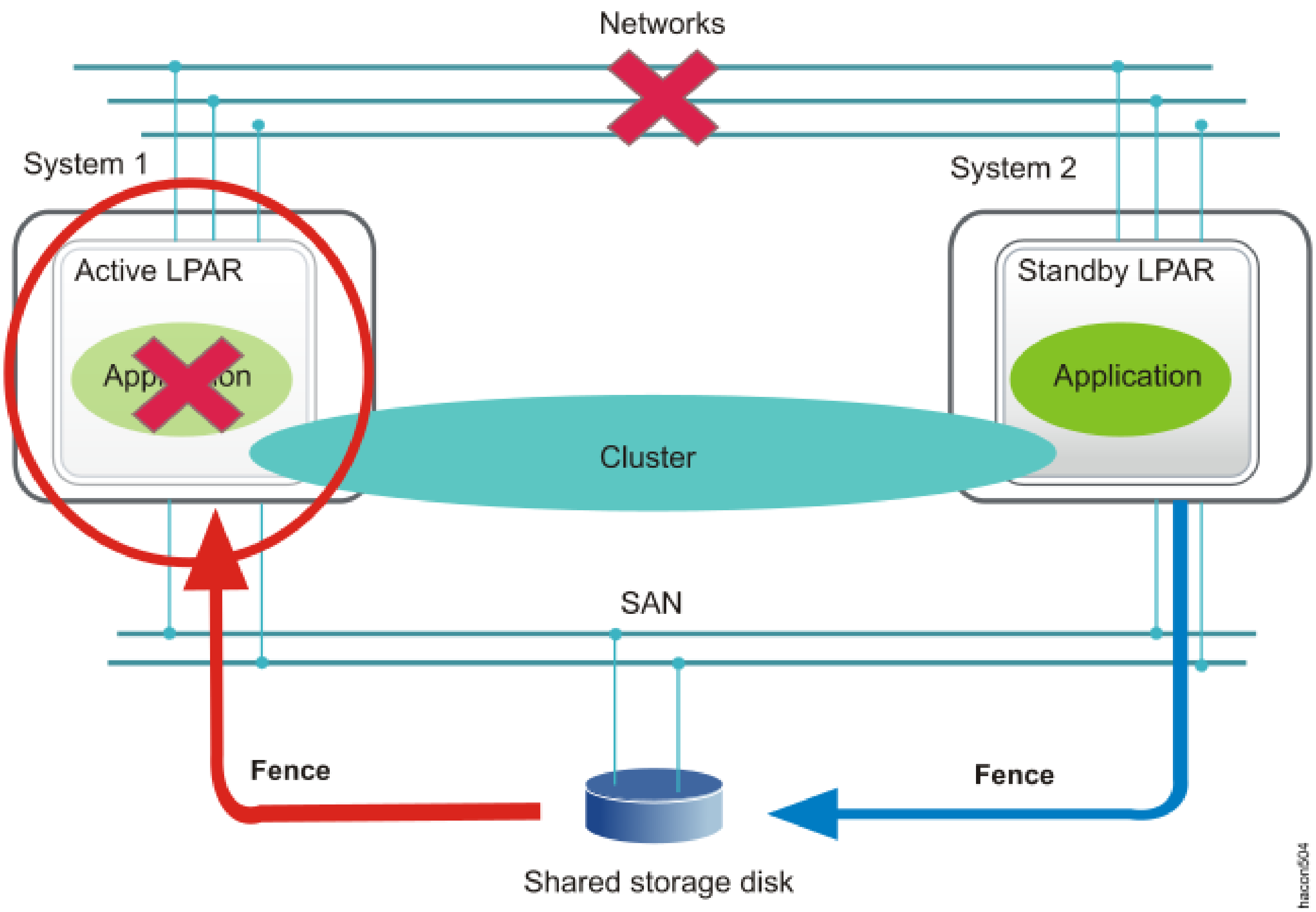


Figure 4. Disk fencing policy

The following key attributes apply to the disk fencing policy in PowerHA SystemMirror:

- Disk fencing applies to only active-passive cluster environments.
- Disk fencing is not supported for resource group that use a startup policy of **Online on All Available Nodes**.
- Disk fencing is supported at the cluster level. Therefore, you can enable or disable disk fencing policy at the cluster level.
- Disk fencing manages all disks that are part of volume groups included in resource groups.
- Disk fencing is supported in mutual takeover configurations. If multiple resource groups exist in the cluster, you must choose one resource group to be the most critical resource group. When the cluster split event occurs, the relative location of the critical resource group determines which site wins. The site that wins is the site that was not running the critical resource group before the cluster split event occurred.

PowerHA SystemMirror uses as much information as possible from the cluster to determine the heath of the LPARs. For example, if the active LPAR was going to crash, PowerHA SystemMirror sends a message to the standby LPAR before the active LPAR goes offline. These notifications ensue that the standby LPAR is certain that the active LPAR has gone offline, and that the standby LPAR can bring the application online.

Note: In certain cases, the standby LPAR is aware that the active LPAR is not sending heartbeats but cannot determine the actual status of the active LPAR. In this case, the standby LPAR declares that the active LPAR has failed after waiting for the time you specified in the **Node Failure Detection Timeout** field. At this time, the standby LPAR fences out all the disks before bringing the resource groups online. If a single disk is not correctly fenced out, the resource group is not brought online.

PowerHA SystemMirror multiple-site solutions

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

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

Stretched cluster
Contains nodes from sites that are located at the same geographical locations. Stretched clusters do not support HADR with storage replication management.

Linked cluster
Contains nodes from sites that are located at different geographical locations. Linked clusters support cross-site LVM mirroring and HyperSwap.

The following table displays the difference between stretched clusters and linked clusters.

Table 1. Stretched clusters and linked cluster differences

Function	Stretched clusters	Linked clusters
Site communication	Multicast	Unicast
Repository disk	Shared	Separate
Cluster communication	• Network • Storage area network (SAN) • Disk	Network
Cross-site Logical Volume Manager mirroring	Available	Available with SAN
HyperSwap	Available	Available with SAN
Concurrent resource group with HyperSwap	Available	Available with SAN

Related concepts

PowerHA SystemMirror use of Cluster Aware AIX
PowerHA SystemMirror is built in addition to the core clustering capabilities that are supported in the AIX operating system. PowerHA SystemMirror is supported for all editions of AIX that support Cluster Aware AIX (CAA) capabilities.

PowerHA SystemMirror linked cluster

A linked cluster is ideal for situations where each site is at a different geographical location. Typically, the sites are far enough apart so that they cannot conveniently share a common storage area network (SAN). Each site must have its own active repository disk, and any backup repositories. Linked clusters always use unicast to communicate between sites. Linked clusters are a useful part of high availability disaster recovery (HADR).

Cluster Aware AIX (CAA) can use linked clusters to support multiple sites that are geographically far apart (in different cities). Linked clusters link the individual CAA clusters to the sites that are at different locations. The links between the sites are used for heartbeat and cluster communication.

You can use linked clusters for communication across short geographical distances, such as between different buildings within the same city.

The following table displays the supported linked cluster configurations for split policy options and merge policy options. For example, in a linked cluster you can have a configuration with a split policy value of

None and a merge policy value of either **Majority** or **Priority**. However, you cannot have a configuration with a split policy value of **Tie-breaker** and a merge policy value of **Manual**.

Table 2. Linked cluster options for split and merge policies

Split policy options	**Merge policy options**			
	Majority	**Priority**	**Tie-breaker**	**Manual**
None	X	X		
Tie-breaker			X	
Manual				X

The following figure displays a linked cluster where the sites are at different cities and each site has its own repository disk.

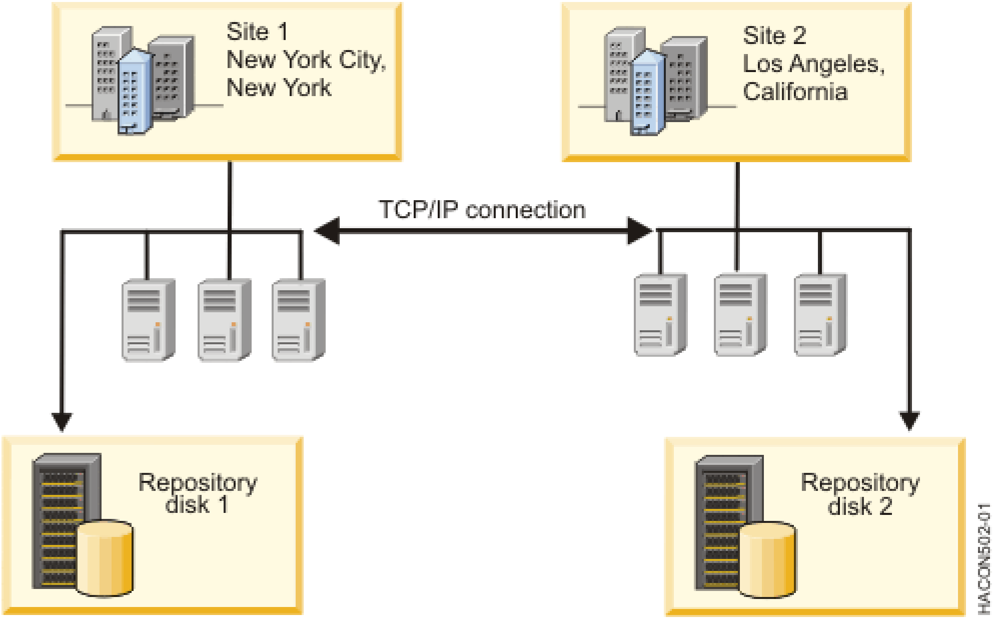


Figure 5. Typical linked cluster configuration

Related concepts

PowerHA SystemMirror use of Cluster Aware AIX
PowerHA SystemMirror is built in addition to the core clustering capabilities that are supported in the AIX operating system. PowerHA SystemMirror is supported for all editions of AIX that support Cluster Aware AIX (CAA) capabilities.

PowerHA SystemMirror stretched cluster
A stretched cluster is ideal for situations where each site is located in the same geographical location. Typically, all nodes are on a common storage area network (SAN). At a minimum, the active repository and any backup repositories are shared between sites, as are any disks in resource groups that can run at either site. Stretched clusters can support LVM cross-site mirroring, and HyperSwap.

Cluster repository disks
PowerHA SystemMirror uses a shared disk as a central repository for managing the configuration of the cluster. This disk must be accessible by all of the nodes in the cluster by using the standard or stretched

cluster deployment method, or must be accessible by all nodes in a site by using the linked cluster deployment method.

PowerHA SystemMirror stretched cluster

A stretched cluster is ideal for situations where each site is located in the same geographical location. Typically, all nodes are on a common storage area network (SAN). At a minimum, the active repository and any backup repositories are shared between sites, as are any disks in resource groups that can run at either site. Stretched clusters can support LVM cross-site mirroring, and HyperSwap.

When you use a stretched cluster in your environment, a single Cluster Aware AIX (CAA) cluster is deployed across sites in the cluster.

If IP communication fails and your environment is using stretched clusters, all nodes in the cluster can use the disk heartbeat function to communicate and to keep your environment operational.

Stretched clusters must meet the following requirements:

- Site communication can use unicast or multicast IP addresses.
- Share at least one repository disk between the sites.

The following table displays the supported stretched cluster configurations for split policy options and merge policy options. For example, in a stretched cluster you can have a configuration with a split policy value of **None** and a merge policy value of **Majority**. However, you cannot have a configuration with a split policy value of **Tie-breaker** and a merge policy value of **Majority**.

Table 3. Stretched cluster options for split and merge policies

Split policy options	**Merge policy options**	
	Majority	**Tie-breaker**
None	X	
Tie-breaker		X

The following figure is an example of a stretched cluster that has two sites that are in different buildings, which are geographically close to each other in the same city and that share a repository disk.

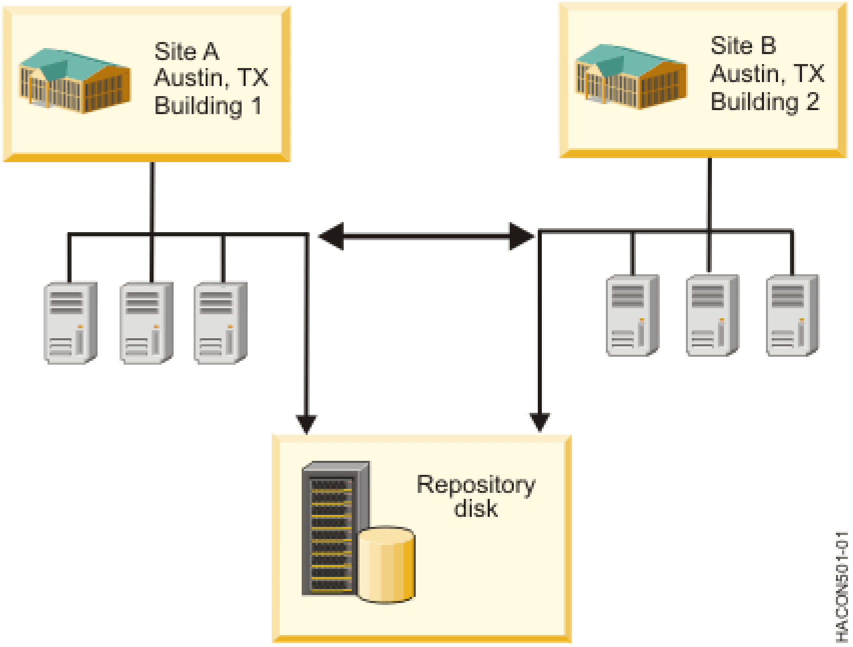


Figure 6. Typical stretched cluster configuration

Related concepts

PowerHA SystemMirror linked cluster
A linked cluster is ideal for situations where each site is at a different geographical location. Typically, the sites are far enough apart so that they cannot conveniently share a common storage area network (SAN). Each site must have its own active repository disk, and any backup repositories. Linked clusters always use unicast to communicate between sites. Linked clusters are a useful part of high availability disaster recovery (HADR).

PowerHA SystemMirror use of Cluster Aware AIX
PowerHA SystemMirror is built in addition to the core clustering capabilities that are supported in the AIX operating system. PowerHA SystemMirror is supported for all editions of AIX that support Cluster Aware AIX (CAA) capabilities.

Cluster repository disks
PowerHA SystemMirror uses a shared disk as a central repository for managing the configuration of the cluster. This disk must be accessible by all of the nodes in the cluster by using the standard or stretched cluster deployment method, or must be accessible by all nodes in a site by using the linked cluster deployment method.

Cluster networks

Cluster nodes communicate with each other over communication networks.

If one of the physical network interface cards on a node on a network fails, PowerHA SystemMirror preserves the communication to the node by transferring the traffic to another physical network interface card on the same node. If a "connection" to the node fails, PowerHA SystemMirror transfers resources to another node to which it has access.

In addition, RSCT sends heartbeats between the nodes over the cluster networks to periodically check on the health of the cluster nodes themselves. If PowerHA SystemMirror detects no heartbeats from a node, a node is considered as failed and resources are automatically transferred to another node.

We highly recommend configuring multiple communication paths between the nodes in the cluster. Having multiple communication networks prevents cluster partitioning, in which the nodes within each partition form their own entity. In a partitioned cluster, it is possible that nodes in each partition could allow simultaneous non-synchronized access to the same data. This can potentially lead to different views of data from different nodes.

Related concepts

Heartbeating over TCP/IP and storage area networks
A *heartbeat* is a type of a communication packet that is sent between nodes. Heartbeats are used to monitor the health of the nodes, networks and network interfaces, and to prevent cluster partitioning.

Physical and logical networks

A physical network connects two or more physical network interfaces.

As stated in the previous section, configuring multiple TCP/IP-based networks helps to prevent cluster partitioning. PowerHA SystemMirror communicates across the storage network when necessary. This additional communication path helps prevent partitioned clusters by providing additional communications paths in cases when the TCP/IP-based network connections become congested or severed between cluster nodes.

Note: If you are considering a cluster where the physical networks use external networking devices to route packets from one network to another, consider the following: When you configure a PowerHA SystemMirror cluster, PowerHA SystemMirror verifies the connectivity and access to all interfaces defined on a particular physical network. However, PowerHA SystemMirror cannot determine the presence of external network devices such as bridges and routers in the network path between cluster nodes. If the networks have external networking devices, ensure that you are using devices that are highly available and redundant so that they do not create a single point of failure in the PowerHA SystemMirror cluster.

A *logical network* is a portion of a physical network that connects two or more logical network interfaces or devices. A logical network interface or device is the software entity that is known by an operating system. There is a one-to-one mapping between a physical network interface/device and a logical network interface/device. Each logical network interface can exchange packets with each logical network interface on the same logical network.

If a subset of logical network interfaces on the logical network needs to communicate with each other (but with no one else) while sharing the same physical network, subnets are used. A subnet mask defines the part of the IP address that determines whether one logical network interface can send packets to another logical network interface on the same logical network.

Logical networks in PowerHA SystemMirror

PowerHA SystemMirror has its own, similar concept of a logical network.

All logical network interfaces in a PowerHA SystemMirror network can communicate PowerHA SystemMirror packets with each other directly. Each logical network is identified by a unique name. If you use an automatic discovery function for PowerHA SystemMirror cluster configuration, PowerHA SystemMirror assigns a name to each PowerHA SystemMirror logical network it discovers, such as *net_ether_01*. A PowerHA SystemMirror logical network might contain one or more subnets. RSCT takes care of routing packets between logical subnets.

Related concepts

PowerHA SystemMirror supported hardware
You can use different types of IBM hardware with PowerHA SystemMirror to implement the base level of a highly available environment.

Local and global network failures

When a failure occurs on a cluster network, PowerHA SystemMirror uses *network failure events* to manage such cases. PowerHA SystemMirror watches for and distinguishes between two types of network failure events: local network failure and global network failure events.

Local network failure

A *local network failure* is a PowerHA SystemMirror event in which packets cannot be sent or received by one node over a PowerHA SystemMirror logical network. This might occur, for instance, if all of the node's network interface cards participating in the particular PowerHA SystemMirror logical network fail. In the case of a local network failure, the network is still in use by other nodes. To handle local network failures, PowerHA SystemMirror selectively moves the resources (on that network) from one node to another. This operation is referred to as selective fallover.

Global network failure

A *global network failure* is a PowerHA SystemMirror event in which packets cannot be sent or received by any node over a PowerHA SystemMirror logical network. This can occur, for instance, if the physical network is damaged. It is important to distinguish between a global network and a global network failure event. A *global network* is a combination of PowerHA SystemMirror networks. A *global network failure event* refers to a failure that affects all nodes connected to any logical PowerHA SystemMirror network, not necessarily a global network.

PowerHA SystemMirror communication interfaces

A PowerHA SystemMirror communication interface is a grouping of a logical network interface, a service IP address and a service IP label that you defined in PowerHA SystemMirror.

PowerHA SystemMirror communication interfaces combine to create IP-based networks. A PowerHA SystemMirror communication interface is a combination of:

- A *logical network interface* is the name to which AIX resolves a port (for example, `en0`) of a physical network interface card.

- A *service IP address* is an IP address (for example, `129.9.201.1`) over which services, such as an application, are provided, and over which client nodes communicate.
- A *service IP label* is a label (for example, a hostname in the **/etc/hosts** file, or a logical equivalent of a service IP address, such as `node_A_en_service`) that maps to the service IP address.

Communication interfaces in PowerHA SystemMirror are used in the following ways:

- A communication interface refers to IP-based networks and network interface cards (NIC). The NICs that are connected to a common physical network are combined into logical networks that are used by PowerHA SystemMirror.
- Each NIC is capable of hosting several TCP/IP addresses. When configuring a cluster, you must define the IP addresses that PowerHA SystemMirror monitors (base or boot IP addresses), and the IP addresses that PowerHA SystemMirror keeps highly available (the service IP addresses).
- Heartbeating in PowerHA SystemMirror occurs over communication interfaces. PowerHA SystemMirror uses the heartbeating facility of Cluster Aware AIX (CAA) to monitor its network interfaces and IP addresses. CAA provides the network topology you create to RSCT, while RSCT provides failure notifications to PowerHA SystemMirror.

PowerHA SystemMirror non-IP communication devices

PowerHA SystemMirror also monitors network devices that are not capable of IP communications. These devices include the storage area network (SAN) and shared disks.

Subnet routing requirements in PowerHA SystemMirror

A subnet route defines a path, defined by a subnet, for sending packets through the logical network to an address on another logical network.

AIX allows you to add multiple routes for the same destination in the kernel routing table. If multiple matching routes have equal criteria, routing can be performed alternatively using one of the several subnet routes. It is important to consider subnet routing in PowerHA SystemMirror because of the following considerations:

- PowerHA SystemMirror does not distinguish between logical network interfaces that share the same subnet route. If a logical network interface shares a route with another interface, PowerHA SystemMirror has no means to determine its health. For more information on network routes, see the AIX man page for the **`route`** command.
- Various constraints are often imposed on the IP-based networks by a network administrator or by TCP/IP requirements. The subnets and routes are also constraints within which PowerHA SystemMirror must be configured for operation.

Note: You should have each communication interface on a node that belongs to a unique subnet, so that PowerHA SystemMirror can monitor each interface. This is not a strict requirement in all cases and depends on several factors. For example, when there is only one network interface per node, such as in a virtual network environment, the boot and service IP addresses can be on the same subnet because there is a single subnet route. If unique subnets are required, PowerHA SystemMirror cluster verification will flag this as an error.

Service IP label and address

A *service IP label* is a label that maps to the service IP address and is used to establish communication between client nodes and the server node.

Services, such as a database application, are provided using the connection made over the service IP label. A service IP label can be placed in a resource group as a resource, which allows PowerHA SystemMirror to monitor its health and keep it highly available, either within a node or, if IP address takeover is configured, between the cluster nodes by transferring it to another node in the event of a failure.

IP alias

An *IP alias* is an IP label or IP address that is configured onto a network interface card in addition to the originally configured IP label or IP address on the network interface card (NIC).

IP aliases are an AIX function that is supported by PowerHA SystemMirror. The AIX operating system supports multiple IP aliases on a NIC. Each IP alias on a NIC can be configured on a separate subnet.

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

The following topics contain information about how PowerHA SystemMirror binds a service IP label with a communication interface depending on which mechanism is used to recover a service IP label.

IP address takeover

IP address takeover is a mechanism for recovering a service IP label by moving it to another network interface card (NIC) on another node, when the initial NIC fails.

IPAT occurs if the physical network interface card on one node fails and if there are no other accessible physical network interface cards on the same network on the same node. Therefore, swapping IP labels of these NICs within the same node cannot be performed and PowerHA SystemMirror will use IPAT to recover the service IP address by using a NIC on a backup node. IP address takeover keeps the IP address highly available by recovering the IP address after failures. PowerHA SystemMirror uses a method called IPAT via IP aliases.

IPAT and service IP labels

IPAT manipulates the service IP label.

When IPAT via IP aliases is used, the service IP label or IP address is aliased or added as an additional address to the same network interface. That is, multiple IP addresses or IP labels are configured on the same network interface at the same time. In this configuration, all IP addresses or labels that you define must be configured on different subnets unless there is a single network interface per node. This method can save hardware, but requires additional subnets.

IP address takeover via IP aliases

You can configure IP address takeover on certain types of networks using the IP aliasing network capabilities of the AIX operating system.

Defining IP aliases to network interfaces allows creation of more than one IP label and address on the same network interface. IPAT via IP aliases uses the gratuitous Address Resolution Protocol (ARP) capabilities available on many types of networks.

When a resource group containing the service IP label falls over from the primary node to the target node, the service IP labels are added (and removed) as alias addresses to the base IP addresses on an available NIC. This allows a single NIC to support more than one service IP label placed on it as an alias. Therefore, the same node can host more than one resource group at the same time.

When there a multiple interfaces on the same node connected to the same network, and those interfaces are not combined into a Ethernet Aggregation, all boot addresses must all be on different subnets. Also, any persistent addresses or service addresses must be on different subnets than the boot addresses.

Because IP aliasing allows coexistence of multiple service labels on the same network interface, you can use fewer physical network interface cards in your cluster. Upon fallover, PowerHA SystemMirror equally distributes aliases between available network interface cards.

Distribution preference for service IP label aliases

PowerHA SystemMirror uses the IP address takeover (IPAT) via IP aliases method for keeping the service IP labels in resource groups highly available.

At cluster startup, PowerHA SystemMirror (by default) distributes all service IP label aliases across all available boot interfaces on a network by using the principle of the least load. PowerHA SystemMirror assigns any new service address to the interface that has the least number of aliases or persistent IP labels already assigned to it.

However, in some cases, it might be desirable to specify other types of allocation, or to ensure that the labels continue to be allocated in a particular manner, not only during startup but also during the subsequent cluster events.

For instance, you might want to allocate all service IP label aliases to the same boot interface as the one currently hosting the persistent IP label for that node. This option might be useful in VPN firewall configurations where only one interface is granted external connectivity and all IP labels (persistent and service IP label aliases) must be placed on the same interface to enable the connectivity.

You can configure a distribution preference for the aliases of the service IP labels that are placed under PowerHA SystemMirror control.

A distribution preference for service IP label aliases is a network-wide attribute used to control the placement of the service IP label aliases on the physical network interface cards on the nodes in the cluster. Configuring a distribution preference for service IP label aliases does the following:

- Allows you to 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 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 as long as acceptable network interfaces are available in the cluster. PowerHA SystemMirror always keeps service IP labels active, even if the preference cannot be satisfied.

Related concepts

Persistent node IP labels

A *persistent node IP label* is a useful administrative tool that lets you contact a node even if the PowerHA SystemMirror cluster services are down on that node.

Related information

Administering PowerHA SystemMirror

Planning PowerHA SystemMirror

Heartbeating over TCP/IP and storage area networks

A *heartbeat* is a type of a communication packet that is sent between nodes. Heartbeats are used to monitor the health of the nodes, networks and network interfaces, and to prevent cluster partitioning.

In order for a PowerHA SystemMirror cluster to recognize and respond to failures, it must continually check the health of the cluster. Some of these checks are provided by the heartbeat function.

Each cluster node sends heartbeat messages at specific intervals to other cluster nodes, and expects to receive heartbeat messages from the nodes at specific intervals. If messages stop being received, PowerHA SystemMirror recognizes that a failure has occurred. Heartbeats can be sent over:

- TCP/IP networks
- Storage Area Networks
- Cluster repository disk

Cluster Aware AIX (CAA) uses heartbeat communication on all available TCP/IP networks and storage area networks (SAN). If TCP/IP networks and SAN networks fail, CAA attempts to use the repository disk as an alternative heartbeat mechanism. The heartbeat path for the backup repository disk is displayed as a dpcom interface in the output of the **lscluster** command. If TCP/IP networks and SAN networks and working, the **lscluster -i** command displays the dpcom interface as `restricted`.

The heartbeat function is configured to use specific paths between nodes. This allows heartbeats to monitor the health of all PowerHA SystemMirror networks and network interfaces, as well as the cluster nodes themselves.

The heartbeat paths are set up automatically by CAA; you have the option to configure point-to-point and disk paths as part of PowerHA SystemMirror configuration.

Related concepts

Cluster networks
Cluster nodes communicate with each other over communication networks.

Heartbeating over Internet Protocol networks

PowerHA SystemMirror relies on Cluster Aware AIX (CAA) to provide heartbeating between cluster nodes over Internet Protocol networks. By default, CAA uses unicast communications for heartbeat. You can optionally select to use multicast communications if your network is configured to support multicast.

PowerHA SystemMirror 7.1.2, or later, supports Internet Protocol version 6 (IPv6). However, you cannot explicitly specify the IPv6 multicast address. CAA uses an IPv6 multicast address which is derived from the Internet Protocol 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**.

By default CAA provides heartbeating over all configured AIX interfaces (and not just the interfaces that are configured in the PowerHA SystemMirror topology).

Note: To stop CAA heartbeating over a specific network, you can define it as **private**, instead of **public**. This note applies to both IPv4 and IPv6 interfaces.

Related concepts

Subnet routing requirements in PowerHA SystemMirror
A subnet route defines a path, defined by a subnet, for sending packets through the logical network to an address on another logical network.

Related information

Steps for changing an IP-based network

Heartbeating over storage area network

Cluster Aware AIX (CAA) automatically exchanges heart beats over the common storage area network (SAN), spanning the cluster nodes. These connections can provide an alternate heartbeat path for a cluster that uses a single TCP/IP-based network.

PowerHA SystemMirror multicasting

PowerHA SystemMirror uses a cluster health management layer that is embedded as part of the AIX operating system called Cluster Aware AIX (CAA). CAA uses AIX kernel-level code to exchange heartbeats over a network by using Fibre Channel adapters and by using disk-based messaging through the central repository.

By default, CAA uses unicast communications for heartbeat. You can optionally select to use multicast communications if your network is configured to support multicast. For cluster communication, you can manually configure a multicast address or have CAA automatically select the multicast address.

Multicast communication is not used when you initially create a cluster, but it is required for normal operations of the cluster. You must test and verify that multicast communication can travel across your network between all nodes in a cluster.

Related information

Administratively Scoped IP Multicast

Multicast address range and assignments

Multicast router information

Multicast packet communication

Multicasting is a form of addressing, where nodes form a group and exchange messages.

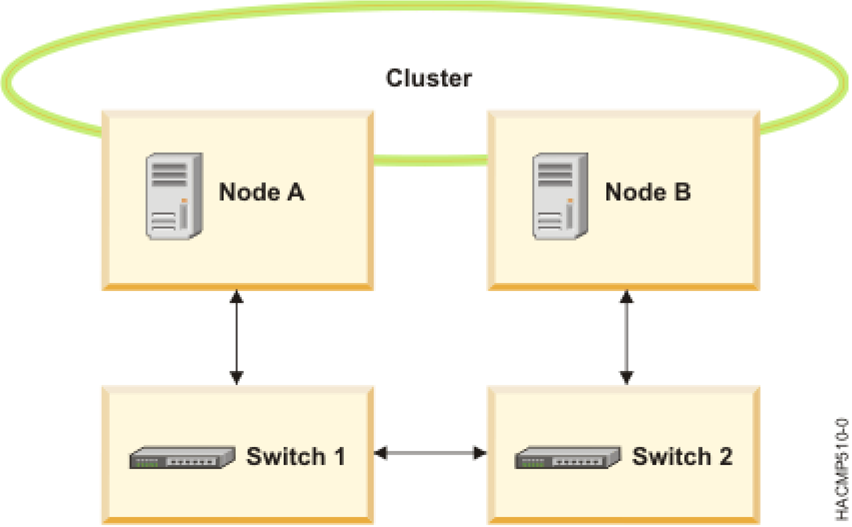
A multicast message sent by one node in the group is received by all other nodes in the group. This function enables efficient cluster communication. For example, a node in a cluster can send a single multicast packet that can notify the other nodes about a critical event.

Multicast network switches

A switch connects different nodes and network segments, and sends network data to the correct node. A *switch* is a multiport network bridge that processes and routes data at the data link layer (layer 2) of the Open Systems Interconnection (OSI) model. Some switches can also process data at the network layer (layer 3).

Typically in a data center environment, multiple nodes are interconnected by switches. When switches cascade, a multicast packet is sent from one node through a switch and then to another switch until it reaches the final destination node in the cluster. Switches handle multicast packets differently than regular network communication. Therefore, switch-to-switch communication might not occur for multicast packets if the network is not set up correctly.

The following figure displays a configuration where Node A is part of a cluster that connects to Node B through Switch 1 and Switch 2. For multicast packets to flow between Node A and Node B, the network connection between Switch 1 and Switch 2 must be enabled for multicast communication.



Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is a communications protocol that enables a node (receiver) to inform a multicast router (IGMP querier) of the node's intention to receive particular multicast traffic.

IGMP runs between a router and a node that enables the following actions:

- Routers ask nodes if they need a particular multicast stream (IGMP query).
- Nodes respond to the router if they are seeking a particular multicast stream (IGMP reports).

The IGMP communication protocol is used by the nodes and the adjacent routers on IP networks to interact and to establish ground rules for multicast communication and establish multicast group membership.

IGMP snooping

IGMP snooping is an activity performed by switches to track the IGMP communications related packet exchanges and adapt to filtering the multicast packets. Switches featuring IGMP snooping derive useful information by observing these IGMP transactions between the nodes and routers. This function enables the switches to correctly forward the multicast packets, when needed, to the next switch in the network path.

Switches monitor the IGMP traffic and only send out multicast packets when necessary. A switch typically builds an IGMP snooping table that has a list of all the ports that have requested a particular multicast group. The IGMP snooping table is used to allow multicast packets to travel across the network or to disallow them from traveling across the network. You can configure your switch to avoid IGMP snooping.

Multicast routing

Multicast routing is the process by which network devices decide how multicast packets are delivered to all intended recipients of the multicast traffic.

Read your router documentation to determine whether your router implements multicast routing.

Network entities that forward multicast packets using special routing algorithms are called *mrouters*. Nodes and other network elements implement mrouters to enable multicast traffic to travel across the network.

Note: Some routers also support multicasting package routing.

When switches cascade, you might need to set up your network switches to forward the packets, as necessary, to implement the mrouters. If you are experiencing troubles with multicast communications in your cluster, check the network devices for correct settings for IGMP and snooping. For more information about setting up your network for multicast traffic, see the documentation for your network devices.

PowerHA SystemMirror resources and resource groups

Look here for resource-related concepts and definitions that are used throughout the documentation and also in the PowerHA SystemMirror user interface.

Identifying and keeping available your cluster resources

The PowerHA SystemMirror software provides a highly available environment.

The PowerHA SystemMirror software does this by:

- Identifying the set of cluster resources that are essential to the operation of an application, and combining those resources into a resource group.
- Defining the resource group policies and attributes that dictate how PowerHA SystemMirror manages resources to keep them highly available at different stages of cluster operation (startup, fallover and fallback).

By identifying resources and defining resource group policies, the PowerHA SystemMirror software makes numerous cluster configurations possible, providing tremendous flexibility in defining a cluster environment tailored to individual requirements.

Identifying cluster resources

The following cluster resources can include both hardware resources and software resources:

- Disks
- Volume groups

- Logical volumes
- File systems
- Service IP labels or addresses
- Applications
- Tape resources

A processor running PowerHA SystemMirror owns a user-defined set of resources: disks, volume groups, file systems, IP addresses, and applications. For the purpose of keeping resources highly available, sets of interdependent resources might be configured into resource groups. *Resource groups* allow you to combine related resources into a single logical entity for easier configuration and management. The PowerHA SystemMirror software handles the resource group as a unit, thus keeping the interdependent resources together on one node and keeping them highly available.

Related concepts

Volume groups
A volume group is a set of physical volumes that AIX treats as a contiguous, addressable disk region.

Logical volumes
A *logical volume* is a set of logical partitions that AIX makes available as a single storage unit - that is, the logical volume is the logical view of a physical disk. Logical partitions might be mapped to one, two, or three physical partitions to implement mirroring.

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.

Applications
The purpose of a highly available system is to ensure that critical services are accessible to users. Applications usually need no modification to run in the PowerHA SystemMirror environment. Any application that can be successfully restarted after an unexpected shutdown is a candidate for PowerHA SystemMirror.

Service IP labels and IP addresses
A service IP label is used to establish communication between client nodes and the server node. Services, such as a database application, are provided using the connection made over the service IP label.

Tape resources
You can configure a SCSI or a Fibre Channel tape drive as a cluster resource in a nonconcurrent resource group, making it highly available to two nodes in a cluster.

Types of cluster resources

This section provides a brief overview of the resources that you can configure in PowerHA SystemMirror and include into resource groups to let PowerHA SystemMirror keep them highly available.

Volume groups

A volume group is a set of physical volumes that AIX treats as a contiguous, addressable disk region.

Volume groups are configured to the AIX operating system, and can be included in resource groups in PowerHA SystemMirror. 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. Shared disks are those that are physically attached to the cluster nodes and logically configured on all cluster nodes.

Logical volumes

A *logical volume* is a set of logical partitions that AIX makes available as a single storage unit - that is, the logical volume is the logical view of a physical disk. 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 (nonconcurrent access), or a raw device (concurrent access). Concurrent access does not support file systems. Databases and applications in concurrent access environments must access raw logical volumes (for example, `/dev/rsharedlv`).

A shared logical volume must have a unique name within a PowerHA SystemMirror cluster.

Note: A shared volume group cannot contain an active paging space.

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.

Shared file systems

In the PowerHA SystemMirror system, a shared file system is a journaled file system that resides entirely in a shared logical volume. For nonconcurrent access, you want to plan shared file systems so that they will be placed on external disks shared by cluster nodes. Data resides in file systems on these external shared disks in order to be made highly available. For concurrent access, you cannot use journaled file systems. Instead, use raw logical volumes.

Journaled file system and enhanced journaled file system

An Enhanced Journaled File System (JFS2) provides the capability to store much larger files than the Journaled File System (JFS). JFS2 is the default file system for the 64-bit kernel. You can choose to implement either JFS, which is the recommended file system for 32-bit environments, or JFS2, which offers 64-bit functionality. JFS2 is more flexible than JFS because it allows you to dynamically increase and decrease the number of files you can have in a file system. JFS2 also lets you include the file system log in the same logical volume as the data, instead of allocating a separate logical volume for logs for all file systems in the volume group.

Applications

The purpose of a highly available system is to ensure that critical services are accessible to users. Applications usually need no modification to run in the PowerHA SystemMirror environment. Any application that can be successfully restarted after an unexpected shutdown is a candidate for PowerHA SystemMirror.

For example, all commercial DBMS products provide a checkpoint on the state of the disk in some sort of transaction journal. In the event of a server failure, the fallover server restarts the DBMS, which reestablishes database consistency and then resumes processing.

You can use AIX Fast Connect to share resources with systems that are not running AIX as the operating system. If you configure Fast Connect as a SystemMirror resource, PowerHA SystemMirror keeps it highly available and recover from node and network interface failures. PowerHA SystemMirror also verifies the configuration of Fast Connect during cluster verification. Applications are managed by defining the application to PowerHA SystemMirror as an application controller resource. The application controller includes application start and stop scripts. PowerHA SystemMirror uses these scripts when the application needs to be brought online or offline on a particular node, to keep the application highly available.

Note: The start and stop scripts are the main points of control for PowerHA SystemMirror over an application. It is very important that the scripts you specify operate correctly to start and stop all aspects of the application. If the scripts fail to properly control the application, other parts of the application recovery might be affected. For example, if the stop script you use fails to completely stop the application and a process continues to access a disk, PowerHA SystemMirror will not be able to bring the volume group offline on the node that failed or recover it on the backup node.

Add your application controller to a PowerHA SystemMirror resource group only after you have thoroughly tested your application start and stop scripts.

The resource group that contains the application controller should also contain all the resources that the application depends on, including service IP addresses, volume groups, and file systems. Once such a resource group is created, PowerHA SystemMirror manages the entire resource group and, therefore, all the interdependent resources in it as a single entity. PowerHA SystemMirror coordinates the application recovery and manages the resources in the order that ensures activating all interdependent resources before other resources.

In addition, PowerHA SystemMirror includes application monitoring capability, whereby you can define a monitor to detect the unexpected termination of a process or to periodically poll the termination of an application and take automatic action upon detection of a problem.

You can configure multiple application monitors and associate them with one or more application controllers. 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. Or, you can configure a custom monitor to check the health of the database, and a process termination monitor to instantly detect termination of the database process.

You can also specify a mode for an application monitor. It can either track how the application is being run (running mode), or whether the application has started successfully (application startup mode). Using a monitor to watch the application startup is especially useful for complex cluster configurations.

Service IP labels and IP addresses

A service IP label is used to establish communication between client nodes and the server node. Services, such as a database application, are provided using the connection made over the service IP label.

A service IP label can be placed in a resource group as a resource that allows PowerHA SystemMirror to monitor its health and keep it highly available, either within a node or, if IP address takeover is configured, between the cluster nodes by transferring it to another node in the event of a failure.

Note: Certain subnet requirements apply for configuring service IP labels as resources in different types of resource groups.

Related concepts

Service IP label and address
A *service IP label* is a label that maps to the service IP address and is used to establish communication between client nodes and the server node.

Related information

Planning PowerHA SystemMirror

Tape resources

You can configure a SCSI or a Fibre Channel tape drive as a cluster resource in a nonconcurrent resource group, making it highly available to two nodes in a cluster.

Management of shared tape drives is simplified by the following PowerHA SystemMirror functionality:

- Configuration of tape drives using the SMIT configuration tool
- 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 a dynamic reconfiguration of cluster resources.

Cluster resource groups

To be made highly available by the PowerHA SystemMirror software, each resource must be included in a resource group. Resource groups allow you to combine related resources into a single logical entity for easier management.

The maximum number of resource groups that are supported is 64.

Participating node list

A participating node list defines a list of nodes that can host a particular resource group.

You define a node list when you configure a resource group. The participating node list can contain some or all nodes in the cluster.

Typically, this list contains all nodes sharing the same data and disks.

Default node priority

Default node priority is identified by the position of a node in the node list for a particular resource group.

The first node in the node list has the highest node priority. This node is also called the home node for a resource group. The node that is listed before another node has a higher node priority than the current node.

Depending on the fallback policy for a resource group, when a node with a higher priority for a resource group (that is currently being controlled by a less priority node) joins or reintegrates into the cluster, it takes control of the resource group. That is, the resource group moves from nodes with less priorities to the higher priority node.

At any given time, the resource group can have a default node priority specified by the participating node list. However, various resource group policies you select can override the default node priority and "create" the actual node priority according to which a particular resource group would move in the cluster.

Dynamic node priority

By setting a dynamic node priority policy you to use an RSCT resource variable such as lowest CPU load to select the takeover node for a nonconcurrent resource group. 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 RSCT resource variable. You can set different policies for different groups or the same policy for several groups.

Home node

The home node (the highest priority node for this resource group) is the first node that is listed in the participating node list for a nonconcurrent resource group.

The home node is a node that normally owns the resource group. A nonconcurrent resource group might or might not have a home node. Whether a nonconcurrent resource group has a home node depends on the startup, fallover, and fallback behaviors of a resource group.

Due to different changes in the cluster, the group might not always start on the home node. It is important to differentiate between the home node for a resource group and the node that currently owns it.

The term home node is not used for concurrent resource groups because they are owned by multiple nodes.

Startup, fallover, and fallback

PowerHA SystemMirror ensures the availability of cluster resources by moving resource groups from one node to another when the conditions in the cluster change.

PowerHA SystemMirror manages resource groups by activating them on a particular node or multiple nodes at cluster startup, or by moving them to another node if the conditions in the cluster change. These are the stages in a cluster lifecycle that affect how PowerHA SystemMirror manages a particular resource group:

Cluster startup
Nodes are up and resource groups are distributed between them according to the resource group startup policy you selected.

Node failure

Resource groups that are active on this node fall over to another node.

Node recovery

A node reintegrates into the cluster and resource groups could be reacquired, depending on the resource group policies you select.

Resource failure and recovery

A resource group might fall over to another node, and be reacquired, when the resource becomes available.

Cluster shutdown

There are different ways of shutting down a cluster, one of which ensures that resource groups move to another node.

During each of these cluster stages, the behavior of resource groups in PowerHA SystemMirror is defined by the following:

- Which node, or nodes, activate the resource group at cluster startup
- How many resource groups are allowed to be acquired on a node during cluster startup
- Which node takes over the resource group when the node that owned the resource group fails and PowerHA SystemMirror needs to move a resource group to another node
- Whether a resource group falls back to a node that has just joined the cluster or stays on the node that currently owns it.

The resource group policies that you select determine which cluster node originally controls a resource group and which cluster nodes take over control of the resource group when the original node relinquishes control.

Each combination of these policies allows you to specify varying degrees of control over which node, or nodes, control a resource group.

To summarize, the focus of PowerHA SystemMirror on resource group ownership makes numerous cluster configurations possible and provides tremendous flexibility in defining the cluster environment to fit the particular needs of the application. The combination of startup, fallover and fallback policies summarizes all the management policies available in previous releases without the requirement to specify the set of options that modified the behavior of "predefined" group types.

When defining resource group behaviors, keep in mind that a resource group can be taken over by one or more nodes in the cluster.

Startup, fallover, and fallback are specific behaviors that describe how resource groups behave at different cluster stages. It is important to keep in mind the difference between fallover and fallback. These terms appear frequently in discussion of the various resource group policies.

Startup

Startup is the activation of a resource group on a node (or multiple nodes) on which it currently resides, or on the home node for this resource group. Resource group startup occurs during cluster startup, or initial acquisition of the group on a node.

Fallover

Fallover is the movement of a resource group from the node that currently owns the resource group to another active node after the current node experiences a failure. The new owner is not a reintegrating or joining node.

Fallover is valid only for nonconcurrent resource groups.

Fallback

Fallback is the movement of a resource group from the node on which it currently resides (which is not a home node for this resource group) to a node that is joining or reintegrating into the cluster.

For example, when a node with a higher priority for that resource group joins or reintegrates into the cluster, it takes control of the resource group. That is, the resource group falls back from nodes with lesser priorities to the higher priority node.

Defining a fallback behavior is valid only for nonconcurrent resource groups.

Resource group policies and attributes

You can configure resource groups to use specific startup, fallover and fallback policies. This section describes resource group attributes and scenarios, and helps you to decide which resource groups suit your cluster requirements.

You can use resource group policies in the following ways:

- Configure resource groups to ensure that they are brought back online on reintegrating nodes during off-peak hours.
- Specify that a resource group that contains a certain application is the only one that will be given preference and be acquired during startup on a particular node. You do so by specifying the node distribution policy. This is relevant if multiple nonconcurrent resource groups can potentially be acquired on a node, but a specific resource group owns an application that is more important to keep available.
- Specify that specific resource groups be kept together online on the same node, or kept apart online on different nodes, at startup, fallover, and fallback.

Resource group startup, fallover, and fallback

Several policies exist for individual resource groups.

These policies are:

Table 4. Resource group policies

Resource group	**Policy**
Startup	• **Online on Home Node Only**. The resource group is brought online only on its home (highest priority) node during the resource group startup. This requires the highest priority node to be available (first node in the resource group's node list). • **Online on First Available Node**. The resource group comes online on the first participating node that becomes available. • **Online on All Available Nodes**. The resource group is brought online on all nodes. • **Online Using Distribution Policy**. Only one resource group is brought online on each node.
Fallover	• **Fallover to Next Priority Node in the List**. The resource group follows the default node priority order specified in the resource group's node list. • **Fallover Using Dynamic Node Priority**. Before selecting this option, select one of the three predefined dynamic node priority policies. These are based on RSCT variables, such as the node with the most memory available. Fallover for this option is not supported when sites are configured. • **Bring Offline (on Error Node Only)**. Select this option to bring a resource group offline on a node during an error condition.

Table 4. Resource group policies (continued)

Resource group	Policy
Fallback	• **Fallback to Higher Priority Node in the List**. A resource group falls back when a higher priority node joins the cluster. If you select this option, you can use the delayed fallback timer. If you do not configure a delayed fallback policy, the resource group falls back immediately when a higher priority node joins the cluster. • **Never Fallback**. A resource group does not fall back to a higher priority node when it joins the cluster.

Related information

Administering PowerHA SystemMirror

Planning PowerHA SystemMirror

Settling time, dynamic node priority, and fallback timer

You can configure some additional parameters for resource groups that dictate how the resource group behaves at startup, fallover, and fallback.

The following are additional parameters for resource groups that you can configure:

Settling Time

You can configure a startup behavior of a resource group by specifying the settling time for a resource group that is currently offline. When the settling time is not configured, the resource group starts on the first available higher priority node that joins the cluster. If the settling time is configured, PowerHA SystemMirror waits for the duration of the settling time period for a higher priority node to join the cluster before it activates a resource group. Specifying the settling time enables a resource group to be acquired on a node that has a higher priority, when multiple nodes are joining simultaneously. The settling time is a cluster-wide attribute that, if configured, affects the startup behavior of all resource groups in the cluster for which you selected Online on First Available Node startup behavior.

Distribution Policy

You can configure the startup behavior of a resource group to use the node-based distribution policy. This policy ensures that during startup, a node acquires only one resource group. See the following section for more information.

Dynamic Node Priority

You can configure a fallover behavior of a resource group to use one of three dynamic node priority policies. These policies are based on RSCT variables such as the most free memory or lowest use of CPU. To recover the resource group PowerHA SystemMirror selects the node that best fits the policy at the time of fallover.

Note: Fallover that uses the Dynamic Node Priority option is not supported when sites are configured.

Delayed Fallback Timer

You can configure a fallback behavior of a resource group 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. This is useful, for instance, for scheduling the resource group fallbacks to occur during off-peak business hours.

Distribution policy

On cluster startup, you can use a node-based distribution policy for resource groups.

If you select this policy for several resource groups, PowerHA SystemMirror tries to have each node acquire only one of those resource groups during startup. This allows you to distribute your CPU-intensive applications on different nodes.

Related information

Administering PowerHA SystemMirror

Planning PowerHA SystemMirror

Networks and resource groups

All resource groups support service IP labels configured on aliased networks.

A service IP label can be included in any nonconcurrent resource group - that resource group could have any of the allowed startup policies except Online on All Available Nodes.

Resource group dependencies

PowerHA SystemMirror supports resource group ordering and customized serial processing of resources to accommodate cluster configurations where a dependency exists between applications residing in different resource groups.

With customized serial processing, you can specify that a given resource group be processed before another resource group. PowerHA SystemMirror offers an easy way to configure parent and child dependencies between resource groups (and applications that belong to them) to ensure proper processing during cluster events.

Location dependency policies are available for you to configure resource groups so they are distributed the way you expect not only when you start the cluster, but also during fallover and fallback. You can configure dependencies so that specified groups come online on different nodes or on the same nodes. PowerHA SystemMirror processes the dependent resource groups in the proper order using parallel processing where possible and serial as necessary. You do not have to customize the processing. You can configure different types of dependencies among resource groups:

- Parent and child dependencies
- Location dependencies.

The dependencies between resource groups that you configure are:

- Explicitly specified using the SMIT interface
- Established cluster-wide, not just on the local node
- Guaranteed to be honored in the cluster.

Child and parent resource groups dependencies

Configuring a resource group parent and child dependency allows for easier cluster configuration and 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 and child dependency behavior:

- If resource group A depends on resource group B, upon node startup, resource group B must be brought online before resource group A is acquired on any node in the cluster. Upon fallover, the order is reversed: Resource group A must be taken offline before resource group B is taken offline.
- In addition, if resource group A depends on resource group B, during a node startup or node reintegration, resource group A cannot be taken online before resource group B is brought online. If resource group B is taken offline, resource group A will be taken offline too, since it depends on resource group B.

Dependencies between resource groups offer a predictable and reliable way of building clusters with multi-tier applications. For more information on typical cluster environments that can use dependent resource groups, see Cluster configurations with multitiered applications. These terms describe parent and child dependencies between resource groups:

- A *parent resource group* has to be in an online state before the resource group that depends on it (child) can be started.

- A *child resource group* depends on a parent resource group. It will get activated on any node in the cluster only after the parent resource group has been activated. Typically, the child resource group depends on some application services that the parent resource group provides.

Upon resource group release (during fallover or stopping cluster services, for instance) PowerHA SystemMirror brings offline a child resource group before a parent resource group is taken offline. The following graphic illustrates the parent and child dependency relationship between resource groups.

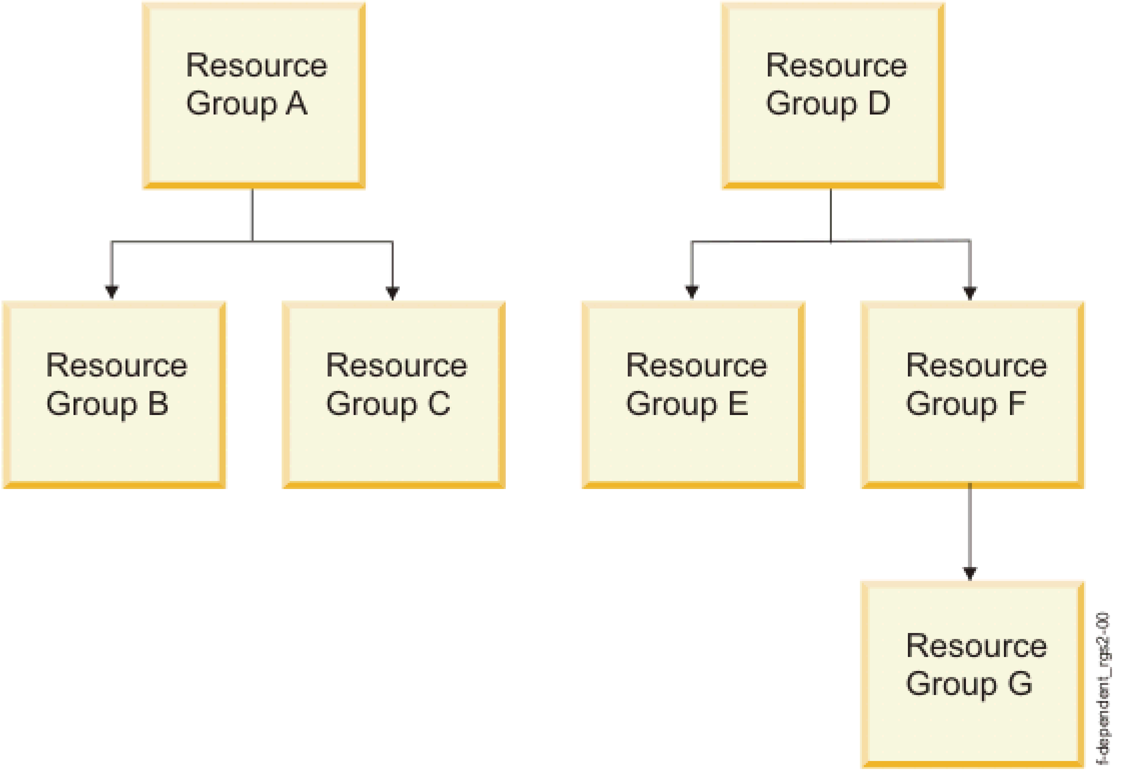


Figure 7. Example of two and three levels of dependencies between resource groups

The example shows relationships that were structured under these guidelines and limitations:

- You can configure a type of dependency where 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.
- A resource group can serve as both a parent and a child 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.

These guidelines and limitations also apply to parent and child dependencies between resource groups:

- You can add, change or delete a dependency between resource groups, while the cluster services are running.
- When you delete a dependency between two resource groups, only the link between these resource groups is removed from the PowerHA SystemMirror Configuration Database. The resource groups are not deleted.
- During fallover of a parent resource group, a child resource group containing the application temporarily goes offline and then online on any available node. The application that belongs to the child resource group is also stopped and restarted.

Related concepts

Cluster configurations with multitiered applications

A typical cluster configuration that could utilize parent and child dependent resource groups is the environment in which an application such as WebSphere depends on another application such as DB2.

Resource group location dependencies

In addition to various policies for individual resource groups and parent and child dependencies, PowerHA SystemMirror offers policies to handle overall resource group interdependencies. PowerHA SystemMirror recognizes these relationships and processes the resource groups in the proper order.

You can configure resource groups so that:

- Two or more specified resource groups are always be online on the same node. They start up, fall over, and fall back to the same node.
- Two or more specified resource groups are always be online on different nodes. They start up, fall over, and fall back to different nodes. You assign priorities to the resource groups so that the most critical ones are handled first in case of fallover and fallback.
- Two or more specified resource groups (with replicated resources) are always be online on the same site.

After you configure individual resource groups with a given location dependency, they form a set that is handled as a unit by the Cluster Manager. The following rules apply when you move a resource group explicitly with the **`clRGmove`** command:

- If a resource group participates in an *Online On Same Node Dependency* set, it can be brought online only on the node where all other resource groups from the same node set are currently online. This is the same rule for the Cluster Manager.
- If a resource group participates in an *Online On Same Site Dependency* set, you can bring it online only on the site where the other resource groups from the same site set are currently online. This is the same rule for the Cluster Manager.
- If a resource group participates in an *Online On Different Nodes Dependency* set, you can bring it online only on a node that does not host any other resource group in the different node dependency set. (This is the same rule for the Cluster Manager.) However, when you move a resource group that belongs to this set, priorities are treated as of equal value, whereas when PowerHA SystemMirror brings these groups online it takes priorities into account.

Sample location dependency model

The fictional company, XYZ Publishing, company follows a business continuity model that involves prioritizing the different platforms used to develop the web content. Location policies are used to keep some resource groups strictly on separate nodes and others together on the same node.

The following figure displays an example of a location dependency model that has three nodes that are used for three different applications and their associated databases. The database for each application must always run on the same node as the application. For example, the production database must contain the production application.

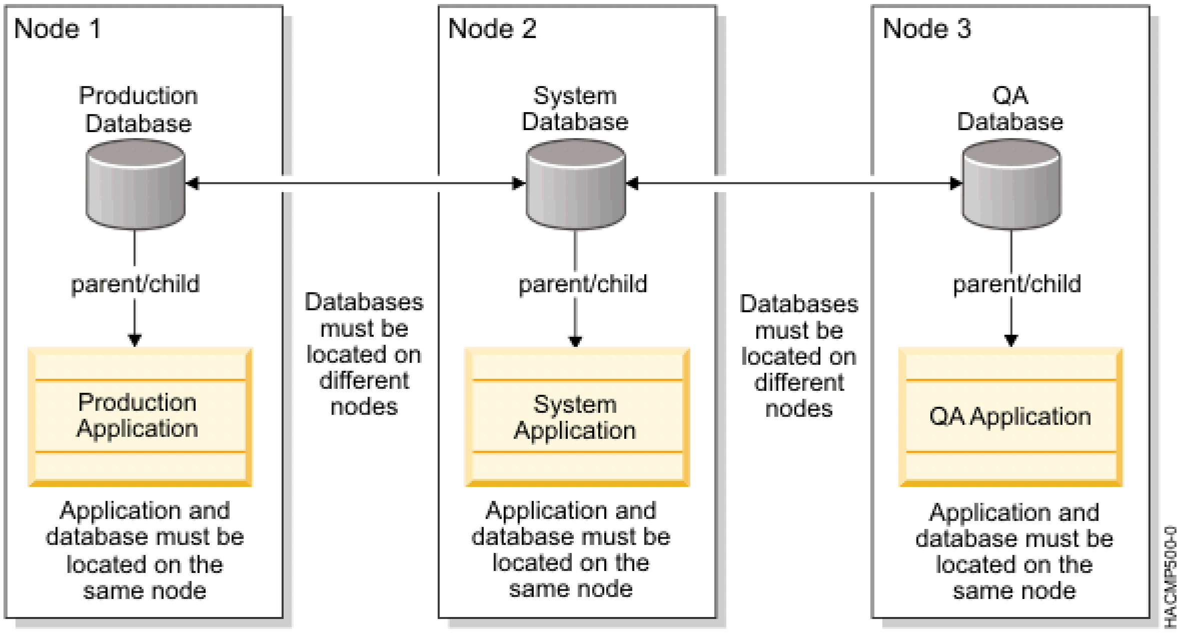

Node 1
Production Database
parent/child
Production Application
Application and database must be located on the same node
Databases must be located on different nodes
Node 2
System Database
parent/child
System Application
Application and database must be located on the same node
Databases must be located on different nodes
Node 3
QA Database
parent/child
QA Application
Application and database must be located on the same node
HACMP500-0

Figure 8. Example of nodes 1, 2, and 3 being used for separate databases

Related information

Administering PowerHA SystemMirror

Planning PowerHA SystemMirror

Sites and resource groups

Most PowerHA SystemMirror configurations do not include sites and use the IGNORE management policy, which is the default. If you have installed a PowerHA SystemMirror Enterprise Edition component for disaster recovery, you distribute the cluster nodes between geographically separated sites and select one of the inter-site management policies.

You include the resources that you want to replicate in resource groups. You define the startup, fallover, and fallback policies for the primary instance of a replicated resource group. The primary instance is where the resource group is online. The node with the primary instance of the resource group activates all the group's resources. The secondary instance (the replication) is activated on a node that is located on the other site as a backup. The inter-site management policy in combination with the resource group startup, fallover, fallback policies determines the site where the primary instance is first located, and how fallover and fallback between sites are handled.

The following options exist for configuring resource group inter-site management policies:

- Prefer primary site
- Online on either site
- Online on both sites

If you define sites for the cluster, then when you define the startup, fallover, and fallback policies for each resource group you want to replicate, you can assign the resource group to a node on the primary site and to a node at the other (secondary) site. The primary instance of the resource group runs on the primary site and the secondary instance runs on the secondary site.

If you have a concurrent resource group, you define it to run on all nodes. In this case, you can select the Online on both sites option for the inter-site management policy. Then, the instances on both sites are active (there are no secondary instances). You can also select the other inter-site management policies so that a concurrent resource group is online on all nodes at one site and has backup instances on the other site.

Note: When you have sites that are defined and a non-concurrent resource group, stopping cluster services by using the unmanage option on one site node brings the resource group to the UNMANAGED state on all nodes for that site.

You can also move the primary instance of a resource group across site boundaries with the **clRGmove** utility. PowerHA SystemMirror then redistributes the peer secondary instances as necessary (or gives you a warning if the move operation is disallowed due to a configuration requirement).

When you move the primary instance of a resource group while its secondary instance is offline, the secondary instance remains offline. PowerHA SystemMirror does not automatically bring the secondary instance of the resource group online.

Related information

Administration guide

Planning guide

Log Analyzer

The Log Analyzer function is available in PowerHA SystemMirror 7.2.2, or later. The Log Analyzer function provides capabilities for scanning and extracting detailed information about different types of errors such as disk failures or interface failures from log files of PowerHA SystemMirror, AIX, and other system components. The

The Log Analyzer function performs the following tasks:

- Analyzes the log files and provides an error report based on error strings or time stamps.
- Analyzes the core dump file from the AIX error log.
- Analyzes the log files that are collected through the **snap** and **clsnap** utility.
- Analyzes user-specified **snap** file based on error strings that are provided and generates a report.

The Log Analyzer function helps during the problem determination process to proceed quickly and effectively without the need for manually locating and assembling the error information.

PowerHA SystemMirror supported hardware

You can use different types of IBM hardware with PowerHA SystemMirror to implement the base level of a highly available environment.

For a summary of hardware that is supported for PowerHA SystemMirror, see IBM Techdocs: PowerHA hardware support matrix (http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101347).

Related information

Installing PowerHA SystemMirror

OEM disk, volume group, and file systems accommodation

PowerHA SystemMirror cluster software

This section describe the PowerHA SystemMirror software that implements a highly available environment.

Software components of a PowerHA SystemMirror node

The software components of a PowerHA SystemMirror node has many layers.

The following figure shows the layers of software on a node in a PowerHA SystemMirror cluster:

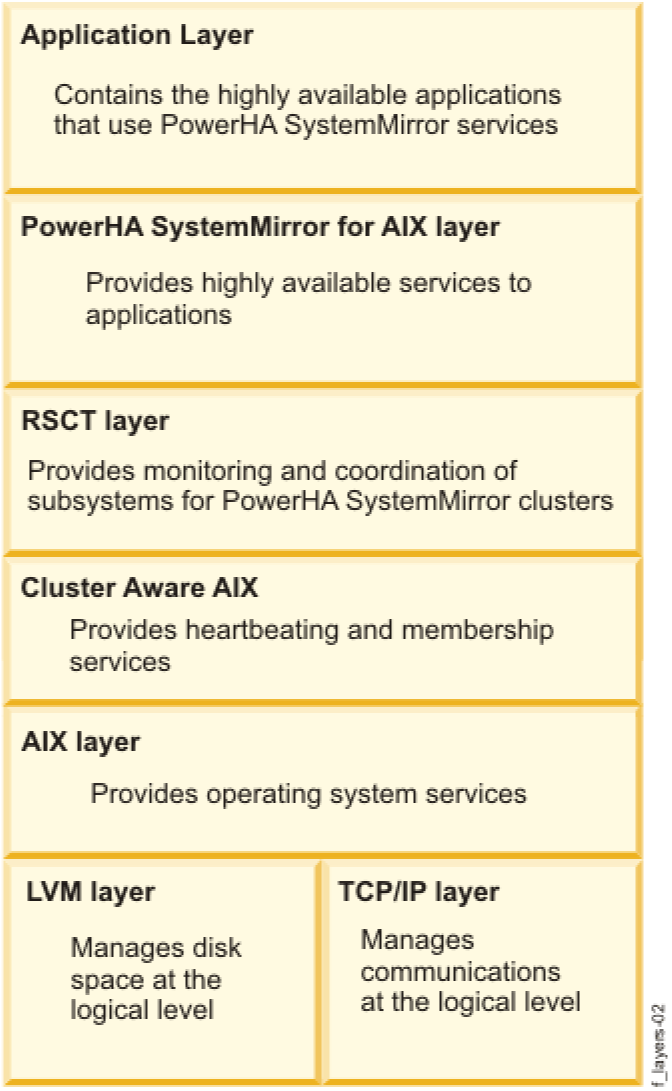

Application Layer
Contains the highly available applications that use PowerHA SystemMirror services
PowerHA SystemMirror for AIX layer
Provides highly available services to applications
RSCT layer
Provides monitoring and coordination of subsystems for PowerHA SystemMirror clusters
Cluster Aware AIX
Provides heartbeating and membership services
AIX layer
Provides operating system services
LVM layer
Manages disk space at the logical level
TCP/IP layer
Manages communications at the logical level
f_layers-02

Figure 9. Model of a PowerHA SystemMirror cluster node

The following list describes each layer:

Application layer
Any applications made highly available through services that are provided by PowerHA SystemMirror for AIX.

PowerHA SystemMirror for AIX layer
Software that recognizes changes within a cluster and coordinates the use of AIX features to create a highly available environment for critical data and applications. PowerHA SystemMirror complements other layers by providing more services to enable applications with their associated communications, and data storage services to be highly available.

RSCT layer
The IBM Reliable Scalable Cluster Technology (RSCT) services are packaged with AIX. The RSCT layer provides facilities for synchronization and coordination of clustering activity by using reliable messaging, in distributed or cluster environments. RSCT includes the Resource Monitoring and Control (RMC), Configuration Resource Manager, and Group Services components.

Cluster Aware AIX layer
This extension of the AIX operating system provides facilities for monitoring node membership and network interface health, and event notification.

AIX layer
Software that provides the underlying support for a PowerHA SystemMirror cluster.

Logical Volume Manager (LVM) subsystem layer
Manages data storage at the logical level.

TCP/IP subsystem layer
Provides communications support for a PowerHA SystemMirror cluster.

Related concepts

IBM reliable scalable cluster technology availability services
The IBM Reliable Scalable Cluster Technology (RSCT) high availability services provide greater scalability, notify distributed subsystems of software failure, and coordinate recovery and synchronization among all subsystems in the software stack. RSCT reports node and network failures to the PowerHA cluster manager.

Cluster manager

Changes in the state of the cluster are referred to as cluster events. On each node, the Cluster Manager monitors local hardware and software subsystems for these events, such as an application failure event.

In response to such events, the Cluster Manager runs one or more event scripts, such as a restart application script. Cluster Managers on all nodes exchange messages to coordinate any actions required in response to an event.

The Cluster Manager is a daemon that runs on each cluster node. The main task of the Cluster Manager is to respond to unplanned events, such as recovering from software and hardware failures, or user-initiated events, such as a joining node event. The RSCT subsystem informs the Cluster Manager about node and network-related events.

Cluster manager connection to other PowerHA SystemMirror daemons

The Cluster Manager gathers information relative to cluster state changes of nodes and interfaces. The Cluster Information Program (Clinfo) gets this information from the Cluster Manager and allows clients communicating with Clinfo to be aware of a cluster's state changes. This cluster state information is stored in the PowerHA SystemMirror Management Information Base (MIB).

Related concepts

PowerHA SystemMirror configuration process and facilities
These topics provide an overview of the PowerHA SystemMirror cluster configuration process and the administrative tools supplied with the PowerHA SystemMirror software.

IBM reliable scalable cluster technology availability services
The IBM Reliable Scalable Cluster Technology (RSCT) high availability services provide greater scalability, notify distributed subsystems of software failure, and coordinate recovery and synchronization among all subsystems in the software stack. RSCT reports node and network failures to the PowerHA cluster manager.

PowerHA SystemMirror software components

PowerHA SystemMirror software has many components.

Cluster secure communication subsystem

PowerHA SystemMirror has a common communication infrastructure that increases the security of intersystem communications. Cluster utilities use the Cluster Communications daemon that runs on each node for communication between the nodes.

Because there is only one common communications path, all communications are reliably secured. Although most components communicate through the Cluster Communications daemon, the following components use another mechanism for inter-node communications:

Component	Communication Method
Cluster Manager	RSCT

Component	Communication Method
Heartbeat messaging	Cluster Aware AIX
Cluster Information Program (Clinfo)	SNMP

For users who require additional security, PowerHA SystemMirror provides message authentication and encryption for messages sent between cluster nodes.

Connection authentication

Standard security mode checks the source IP address against an access list, checks that the value of the source port is between 571 and 1023, and uses the principle of least-privilege for remote command execution. Standard security is the default security mode. For added security, you can set up a VPN for connections between nodes for PowerHA SystemMirror inter-node communications.

Message authentication and encryption

Message authentication and message encryption provide additional security for PowerHA SystemMirror messages sent between cluster nodes. Message authentication ensures the origination and integrity of a message. Message encryption changes the appearance of the data as it is transmitted and translates it to its original form when received by a node that authenticates the message. You can configure the security options and options for distributing encryption keys using the SMIT interface.

IBM reliable scalable cluster technology availability services

The IBM Reliable Scalable Cluster Technology (RSCT) high availability services provide greater scalability, notify distributed subsystems of software failure, and coordinate recovery and synchronization among all subsystems in the software stack. RSCT reports node and network failures to the PowerHA cluster manager.

The PowerHA SystemMirror and RSCT software stack runs on each cluster node. The PowerHA SystemMirror Cluster Manager obtains indications of possible failures from several sources:

- RSCT monitors the state of the network devices
- AIX LVM monitors the state of the volume groups and disks
- Application monitors monitor the state of the applications.

The PowerHA SystemMirror Cluster Manager drives the cluster recovery actions in the event of a component failure. RSCT running on each node exchanges a heartbeat with its peers so that it can monitor the availability of the other nodes in the cluster. If the heartbeat stops, the peer systems drive the recovery process. The peers take the necessary actions to get the critical applications running and to ensure that data has not been corrupted or lost. RSCT services include the following components:

- Resource Monitoring and Control (previous versions of PowerHA SystemMirror use the Event Management subsystem). A distributed subsystem providing a set of high availability services. It creates events by matching information about the state of system resources with information about resource conditions of interest to client programs. Client programs in turn can use event notifications to trigger recovery from system failures.
- Group Services. A system-wide, highly available facility for coordinating and monitoring changes to the state of an application running on a set of nodes. Group Services helps in both the design and implementation of highly available applications and in the consistent recovery of multiple applications. It accomplishes these two distinct tasks in an integrated framework.

The following figure shows the main components that make up the PowerHA SystemMirror architecture:

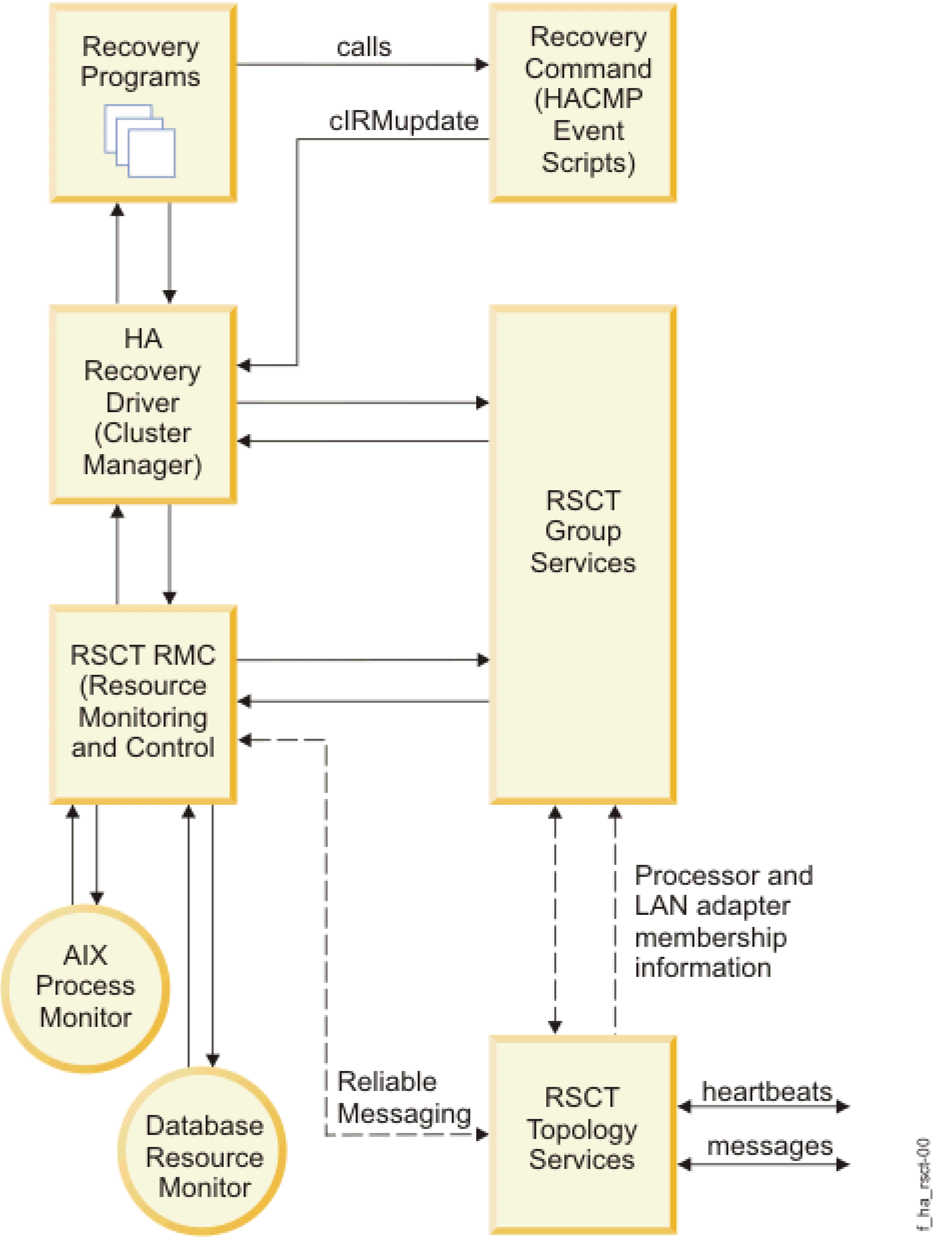

Recovery Programs
calls
Recovery Command (HACMP Event Scripts)
clRMupdate
HA Recovery Driver (Cluster Manager)
RSCT Group Services
RSCT RMC (Resource Monitoring and Control
AIX Process Monitor
Database Resource Monitor
Processor and LAN adapter membership information
Reliable Messaging
RSCT Topology Services
heartbeats
messages
f_ha_rsct-00

Figure 10. Components of PowerHA SystemMirror architecture

Cluster manager and SNMP monitoring programs

A PowerHA SystemMirror cluster is dynamic and can undergo various transitions in its state over time.

For example, a node can join or leave the cluster, or another IP label can replace a service IP label on a physical network interface card. Each of these changes affects the composition of the cluster, especially when highly available clients and applications must use services provided by cluster nodes.

SNMP support

The Cluster Manager provides Simple Network Management Protocol (SNMP) support to client applications. SNMP is an industry-standard specification for monitoring and managing TCP/IP-based networks. SNMP includes a protocol, a database specification, and a set of data objects. This set of data objects forms a Management Information Base (MIB). SNMP provides a standard MIB that includes information such as IP addresses and the number of active TCP connections. The standard SNMP agent is the **snmpd** daemon.

You can extend SNMP to include enterprise-specific MIBs that contain information relating to a discrete environment or application. In PowerHA SystemMirror, the Cluster Manager maintains information about the objects defined in its MIB and passes this information on to a specialized network monitoring or network management station.

PowerHA SystemMirror MIB

The Cluster Manager maintains cluster status information in a special PowerHA SystemMirror MIB. When the Cluster Manager starts on a cluster node, it registers with the SNMP daemon **snmpd**, and then continually gathers cluster information. The Cluster Manager maintains an updated topology of the cluster in the PowerHA SystemMirror MIB as it tracks events and the resulting states of the cluster.

Related information

Programming client applications

Cluster information program

The Cluster Information Program (Clinfo), the **clinfo** daemon, is an SNMP-based monitor.

Clinfo, running on a client machine or on a cluster node, queries the MIB for updated cluster information. Through Clinfo, information about the state of a PowerHA SystemMirror cluster, nodes, and networks can be made available to clients and applications. Clients can be divided into two categories: naive and intelligent.

- A *naive* client views the cluster complex as a single entity. If a cluster node fails, the client must be restarted (or at least must reconnect to the node), if IP address takeover (IPAT) is not enabled.
- An *intelligent* client, on the other hand, is cluster-aware - it reacts appropriately to node failure, connecting to an alternate node and perhaps masking the failure from the user. Such an intelligent client must have knowledge of the cluster state.

The PowerHA SystemMirror software extends the benefits of highly available servers, data, and applications to clients by providing notification of cluster state changes to clients through the Cluster Manager and Clinfo API functions.

Responding to cluster changes

Clinfo calls the **/usr/es/sbin/cluster/etc/clinfo.rc** script whenever a cluster, network, or node event occurs. By default, the **clinfo.rc** script flushes the system's ARP cache to reflect changes to network IP addresses, and it does not update the cache until another address responds to a *ping* request. Flushing the ARP cache typically is not necessary if the PowerHA SystemMirror **hardware address swapping** facility is enabled because hardware address swapping maintains the relationship between an IP address and a hardware address.

In a switched Ethernet network, you might need to flush the ARP cache to ensure that the new MAC address is communicated to the switch.

You can add logic to the **clinfo.rc** script if further action is desired.

Clinfo APIs

The Clinfo APIs provide application developers with both a C and a C++ language interface for accessing cluster status information. The PowerHA SystemMirror software includes two versions of the Clinfo APIs: one for single-threaded applications and one for multithreaded applications. Clinfo and its associated APIs enable developers to write applications that recognize and respond to changes in a cluster.

Related concepts

PowerHA SystemMirror cluster nodes, networks, and heartbeating concepts
This section introduces major cluster topology-related concepts and definitions that are used throughout the documentation and in the PowerHA SystemMirror user interface.

Ensuring application availability

This section describes how the PowerHA SystemMirror software ensures application availability by ensuring the availability of cluster components. PowerHA SystemMirror eliminates single points of failure for all key system components, and eliminates the need for scheduled downtime for most routine cluster maintenance tasks.

Related information

Programming client applications

Highly available NFS server

The highly available NFS server functionality is included in the PowerHA SystemMirror product subsystem.

A highly available NFS server allows a backup processor to recover current NFS activity should the primary NFS server fail. The NFS server special functionality includes highly available modifications and locks on network file systems (NFS). You can do the following:

- Use the reliable NFS server capability that preserves locks and dupcache (2-node clusters only if using NFS version 2 and version 3)
- Specify a network for NFS cross-mounting
- Define NFS exports and cross-mounts at the directory level
- Specify export options for NFS-exported directories and file systems
- Configure two nodes to use NFS.

PowerHA SystemMirror clusters can contain up to 16 nodes. Clusters that use NFS version 2 and version 3 can have a maximum of two nodes, and clusters that use NFS version 4 can have a maximum of 16 nodes.

Shared external disk access

The PowerHA SystemMirror software supports two methods of shared external disk access: nonconcurrent and concurrent.

Nonconcurrent shared external disk access

In a nonconcurrent environment, only one node has access to a shared external disk at a given time.

If this node fails, one of the peer nodes acquires the disk, mounts file systems defined as resources, and restarts applications to restore critical services. Typically, this takes from 30 - 300 seconds, depending on the number and size of the file systems.

A nonconcurrent configuration can use:

- SCSI disks
- SCSI disk arrays
- Serial disks
- Fibre Channel directly attached disks
- Fibre Channel SAN-attached disks

To prevent a failed disk from becoming a single point of failure, each logical volume in a shared volume group should be mirrored using the AIX LVM facility. If you are using an IBM Enterprise Storage System or other supported RAID array, do not use LVM mirroring. RAID devices provide their own data redundancy.

Most software that can run in single-machine mode can be managed by the PowerHA SystemMirror software without modification.

nonconcurrent access typically does not require any code changes to server programs (a database management system, for example), or to applications to provide a highly available solution. To end users, node failure looks like a very fast machine reboot. One of the surviving nodes takes ownership of the failed node's resource groups and restarts the highly available applications. The Journaled File System, the native AIX file system, guarantees file system integrity. The server program guarantees transaction data integrity.

Users simply log onto one of the surviving nodes and restart the application. The logon and application restart procedures can be driven by the PowerHA SystemMirror software. In some PowerHA SystemMirror configurations, users can continue without having to take any action - they simply experience a delay during fallover.

Concurrent shared external disk access

The concurrent access feature enhances the benefits provided by a PowerHA SystemMirror cluster. Concurrent access allows simultaneous access to a volume group on a disk subsystem attached to multiple (up to 16) nodes.

Using concurrent access, a cluster can offer nearly continuous availability of data that rivals fault tolerance, but at a much low cost. Additionally, concurrent access provides higher performance, eases application development, and allows horizontal growth. Since concurrent access provides simultaneous access to data from multiple nodes, additional tools might be required to prevent multiple nodes from modifying the same block of data in a conflicting way. The PowerHA SystemMirror software provides the Clinfo program that prepares an application to run in a concurrent access environment. The Clinfo API provides an API through which applications might become “cluster-aware”. The Clinfo tool is described earlier in this chapter.

The following list includes the benefits of concurrent shared external disk access:

- *Transparent Recovery Increases Availability*. Concurrent access significantly reduces the time for a fallover - sometimes to just a few seconds - because the peer systems already have physical access to the shared disk and are running their own instances of the application.

  In a concurrent access environment, fallover basically involves backing out in-flight transactions from the failed processor. The server software running on the surviving nodes is responsible for recovering any partial transactions caused by the crash. Since all nodes have concurrent access to the data, a client/server application can immediately retry a failed request on the surviving nodes, which continue to process incoming transactions.
- *Harnessing Multiple Processors Increases Throughput*. Applications are no longer limited to the throughput of a single processor. Instead, multiple instances of an application can run simultaneously on multiple processors. As more processing power is required, more systems can be added to the cluster to increase throughput.
- *Single Database Image Eases Application Development and Maintenance*. In a nonconcurrent environment, the only route to improving performance is to partition an application and its data. Breaking code and data into pieces makes both application development and maintenance more complex.

Splitting a database requires a high degree of expertise to make sure that the data and workload are evenly distributed among the processors. Partitioning code and data is not necessary in a concurrent access environment. To increase throughput, multiple instances of the same application running on different processors can simultaneously access a database on a shared external disk. A concurrent configuration can use:

- SCSI disks
- SCSI disk arrays
- Serial disks
- Fibre Channel direct-attached disks
- Fibre Channel SAN-attached disks

When creating concurrent access logical volumes, use LVM mirroring to avoid having the disks be a single point of failure, except for RAID disk subsystems that supply their own mirroring. Concurrent access does not support the use of the Journaled File System. Therefore, the database manager must write directly to the raw logical volumes or hdisks in the shared volume group. An application must use some method to arbitrate all requests for shared data. Most commercial UNIX databases provide a locking model that makes them compatible with the PowerHA SystemMirror software. Check with your database vendor to determine whether a specific application supports concurrent access processing.

Related concepts

Enhanced concurrent mode

AIX provides a new form of concurrent mode: *enhanced concurrent mode*. In enhanced concurrent mode, the instances of the Concurrent Logical Volume Manager (CLVM) coordinate changes between nodes through the Group Services component of the Reliable Scalable Cluster Technology (RSCT) facility in AIX. PowerHA SystemMirror automatically creates shared volume groups as enhanced concurrent mode, and converts existing shared volume groups to enhanced concurrent mode

Concurrent resource manager

The Concurrent resource manager of PowerHA SystemMirror provides concurrent access to shared disks in a highly available cluster, allowing tailored actions to be taken during takeover to suit business needs.

Concurrent Resource Manager adds enhanced-concurrent support for shared volume groups on all types of disks, and concurrent shared-access management for supported RAID disk subsystems.

Enhanced concurrent mode

AIX provides a new form of concurrent mode: *enhanced concurrent mode*. In enhanced concurrent mode, the instances of the Concurrent Logical Volume Manager (CLVM) coordinate changes between nodes through the Group Services component of the Reliable Scalable Cluster Technology (RSCT) facility in AIX. PowerHA SystemMirror automatically creates shared volume groups as enhanced concurrent mode, and converts existing shared volume groups to enhanced concurrent mode

Group Services protocols flow over the communications links between the cluster nodes. Any disk supported by PowerHA SystemMirror for attachment to multiple nodes can be included in an enhanced concurrent mode volume group.

Complementary cluster software

A broad range of additional tools aids you in efficiently building, managing and expanding high availability clusters in AIX environments.

These include:

- General Parallel File System (GPFS) for AIX, a cluster-wide file system that allows users shared access to files that span multiple disk drives and multiple nodes.
- Workload Manager for AIX provides resource balancing between applications.
- Smart Assist software for configuring Oracle, DB2, WebSphere, and other applications in PowerHA SystemMirror clusters.

Related information

Developing Smart Assist applications for PowerHA SystemMirror

Smart Assist for PowerHA SystemMirror

Ensuring application availability

This section describes how the PowerHA SystemMirror software ensures application availability by ensuring the availability of cluster components. PowerHA SystemMirror eliminates single points of failure for all key system components, and eliminates the need for scheduled downtime for most routine cluster maintenance tasks.

Overview: Application availability

The key facet of a highly available cluster is its ability to detect and respond to changes that could interrupt the essential services it provides. The PowerHA SystemMirror software allows a cluster to continue to provide application services critical to an installation even though a key system component, for example, network interface card is no longer available.

When a component becomes unavailable, the PowerHA SystemMirror software is able to detect the loss and shift the workload from that component to another component in the cluster. In planning a highly available cluster, you attempt to ensure that key components do not become single points of failure.

You can use Smart Assists for PowerHA SystemMirror to manage highly available applications such as, DB2, SAP, Oracle, and WebSphere. You can view these applications as a single PowerHA SystemMirror entity that is represented as an application name in the PowerHA SystemMirror software.

PowerHA SystemMirror software allows a cluster to continue providing application services while routine maintenance tasks are performed using a process called dynamic reconfiguration. In dynamic reconfiguration, you can change components in a running cluster, such as adding or removing a node or network interface, without having to stop and restart cluster services. The changed configuration becomes the active configuration dynamically. You can also dynamically replace a failed disk. The following sections describe conceptually how to use the PowerHA SystemMirror software to:

- Eliminate single points of failure in a cluster.
- Minimize scheduled downtime in a PowerHA SystemMirror cluster with the dynamic reconfiguration, resource group management, and cluster management (C-SPOC) utilities.
- Minimize unscheduled downtime with the fast recovery feature, and by specifying a delayed fallback timer policy for resource groups.
- Minimize the time it takes to perform disk takeover.
- Interpret and emulate cluster events.

Note: You might need to monitor the cluster activity while a key component fails and the cluster continues providing availability of an application.

Related concepts

PowerHA SystemMirror configuration process and facilities
These topics provide an overview of the PowerHA SystemMirror cluster configuration process and the administrative tools supplied with the PowerHA SystemMirror software.

Related information

Smart Assits for PowerHA SystemMirror

Eliminating single points of failure in a PowerHA SystemMirror cluster

The PowerHA SystemMirror software enables you to build clusters that are both highly available and scalable by eliminating single points of failure (SPOF). 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 to provide that function and essential services become unavailable. For example, if all the data for a critical application resides on a single disk that is not mirrored, and that disk fails, the disk is a single point of failure for the entire system. Client nodes cannot access that application until the data on the disk is restored.

Potential single points of failure in a PowerHA SystemMirror cluster

To be highly available, a cluster must have no single point of failure.

PowerHA SystemMirror provides recovery options for the following cluster components:

- Nodes
- Applications
- Networks and network interfaces
- Disks and disk adapters.

While the goal is to eliminate all single points of failure, compromises might have to be made. There is usually a cost associated with eliminating a single point of failure. For example, redundant hardware increases cost. The cost of eliminating a single point of failure should be compared to the cost of losing services should that component fail. The purpose of the PowerHA SystemMirror software is to provide a cost-effective, highly available computing environment that can grow to meet future processing demands.

Eliminating nodes as a single point of failure

Nodes leave the cluster either through a planned transition (a node shutdown or stopping cluster services on a node), or because of a failure.

Node failure begins when a node monitoring a neighbor node ceases to receive heartbeat traffic for a defined period of time. If the other cluster nodes agree that the failure is a node failure, the failing node is removed from the cluster and its resources are taken over by the nodes configured to do so. An active node might, for example, take control of the shared disks configured on the failed node. Or, an active node might masquerade as the failed node (by acquiring its service IP address) and run the processes of the failed node while still maintaining its own processes. Thus, client applications can switch over to a surviving node for shared-disk and processor services. The PowerHA SystemMirror software provides the following facilities for processing node failure:

- Disk takeover
- IP address takeover via IP aliases
- IP address takeover via IP replacement (with or without hardware address takeover)

Disk takeover

In a PowerHA SystemMirror environment, shared disks are physically connected to multiple nodes.

Disk takeover in concurrent environments

In concurrent access configurations, the shared disks are actively connected to multiple nodes at the same time. Therefore, disk takeover is not required when a node leaves the cluster.

The following figures illustrate disk takeover in concurrent environments.

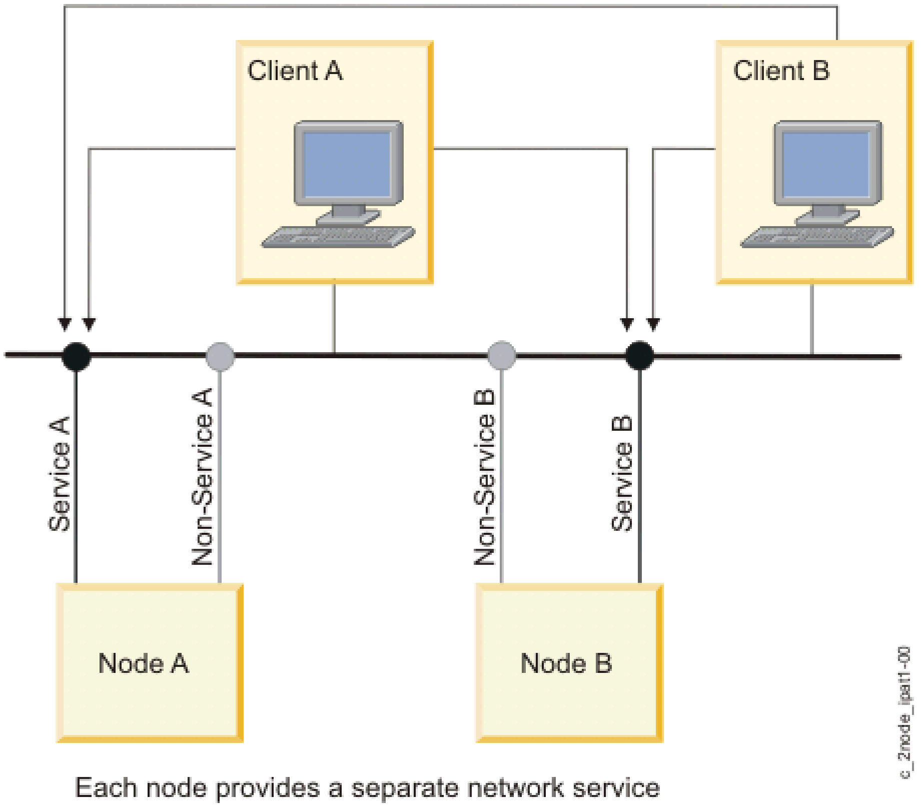


Figure 11. Concurrent access configuration before disk takeover

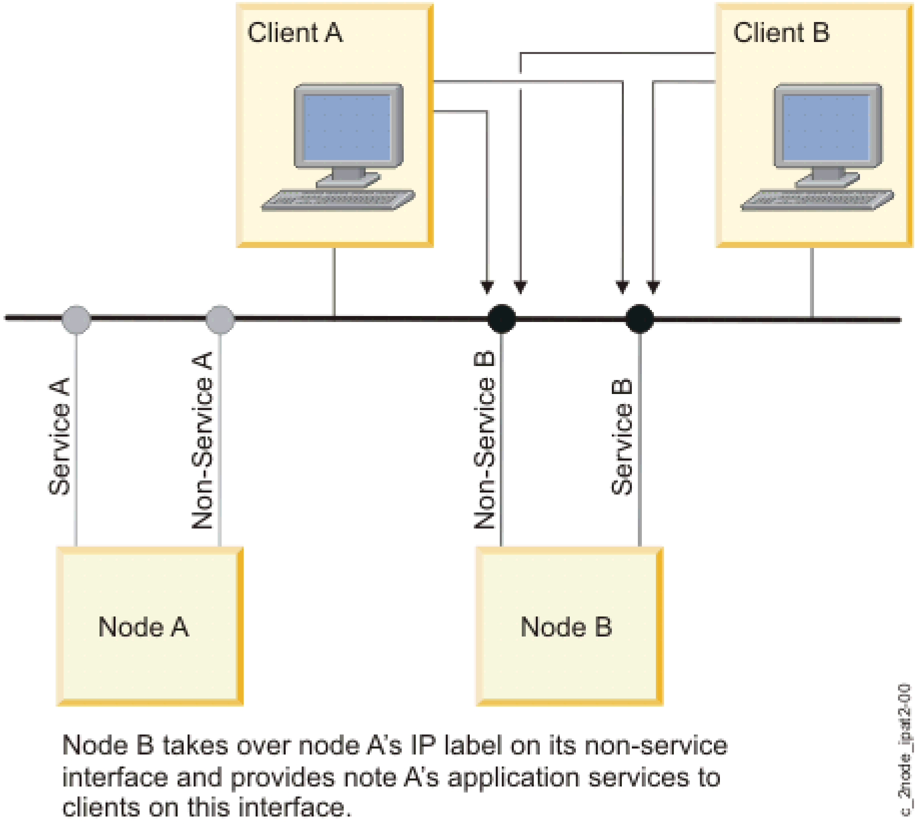

Client A
Client B
Service A
Non-Service A
Non-Service B
Service B
Node A
Node B
Node B takes over node A's IP label on its non-service interface and provides note A's application services to clients on this interface.
c_2node_ipat2-00

Figure 12. Concurrent access configuration after disk takeover

IP address takeover

IP address takeover (IPAT) is a networking capability that allows a node to acquire the network address of a node that has abandoned the cluster.

IP address takeover is necessary in a PowerHA SystemMirror cluster when a service being provided to clients is bound to a specific IP address, that is, when a service IP label through which services are provided to the clients is included as a resource in a cluster resource group. If, instead of performing an IPAT, a surviving node simply did a disk and application takeover, clients would not be able to continue using the application at the specified server IP address. PowerHA SystemMirror uses IPAT via Aliases.

The following figures illustrate IP address takeover.

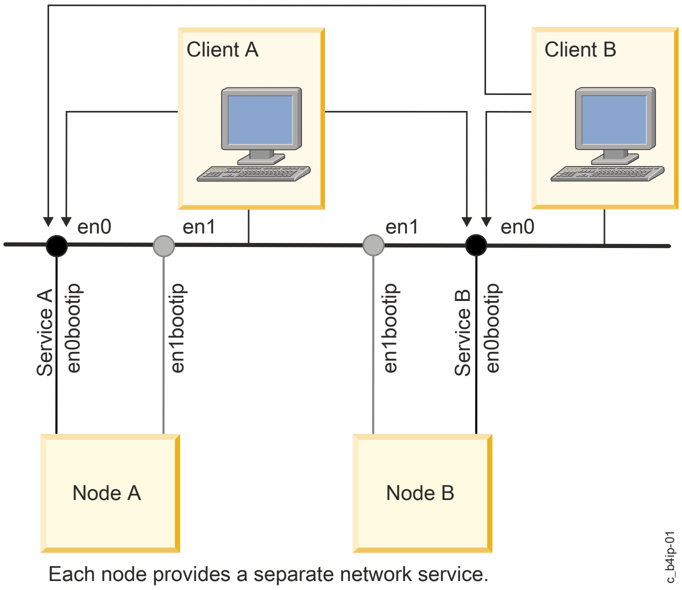


Figure 13. Configuration before IP address takeover via IP aliases

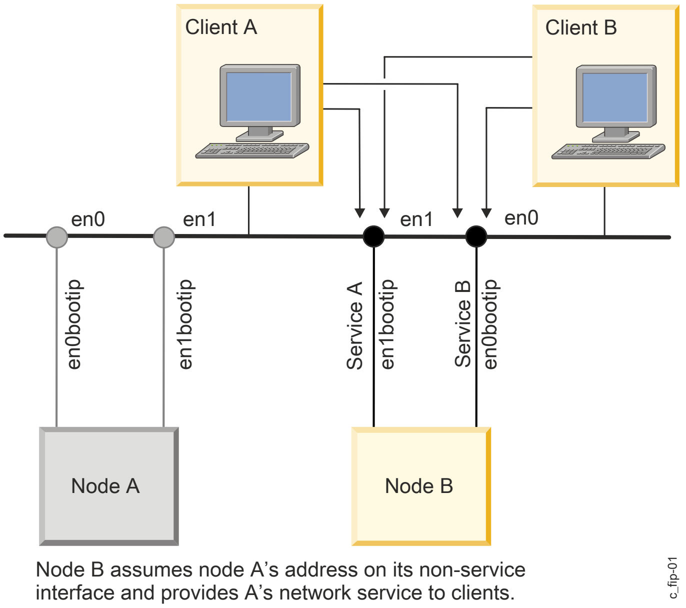

Client A
Client B
en0
en1
en1
en0
en0bootip
en1bootip
Service A
en1bootip
Service B
en0bootip
Node A
Node B
Node B assumes node A's address on its non-service interface and provides A's network service to clients.
c_fip-01

Figure 14. Configuration after IP address takeover via IP aliases

Related concepts

IP address takeover via IP aliases

You can configure IP address takeover on certain types of networks using the IP aliasing network capabilities of the AIX operating system.

Eliminating applications as a single point of failure

The primary reason to create PowerHA SystemMirror clusters is to provide a highly available environment for mission-critical applications.

For example, a PowerHA SystemMirror cluster could run a database server program that services client applications. The clients send queries to the server program that responds to their requests by accessing a database, stored on a shared external disk.

In a PowerHA SystemMirror cluster, these critical applications can be a single point of failure. To ensure the availability of these applications, the node configured to take over the resources of the node leaving the cluster should also restart these applications so that they remain available to client processes. You can make an application highly available by using:

- Application controller
- Cluster control
- Application monitors
- Application Availability Analysis Tool

To put the application under PowerHA SystemMirror control, you create an application controller cluster resource that associates a user-defined name of the server with the names of user-provided 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.

Certain applications can be made highly available without application controllers. You can place such applications under cluster control by configuring an aspect of the application as part of a resource group. For example, Fast Connect services can all be added as resources to a cluster resource group, making them highly available in the event of node or network interface failure.

Note: Application takeover is usually associated with IP address takeover. If the node restarting the application also acquires the IP service address on the failed node, the clients only need to reconnect to the same server IP address. If the IP address was not taken over, the client needs to connect to the new server to continue accessing the application.

Additionally, you can use the AIX System Resource Controller (SRC) to monitor for the presence or absence of an application daemon and to respond accordingly.

Application monitors

You can also configure an application monitor to check for process failure or other application failures and automatically take action to restart the application.

You can configure multiple application monitors and associate them with one or more application controllers. 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. Or, you can configure a custom monitor to check the health of the database, and a process termination monitor to instantly detect termination of the database process.

Application availability analysis tool

The Application Availability Analysis tool measures the exact amount of time that any of your applications have been available. The PowerHA SystemMirror software collects, timestamps, and logs extensive information about the applications you choose to monitor with this tool. Using SMIT, you can select a time period and the tool displays uptime and downtime statistics for a specific application during that period.

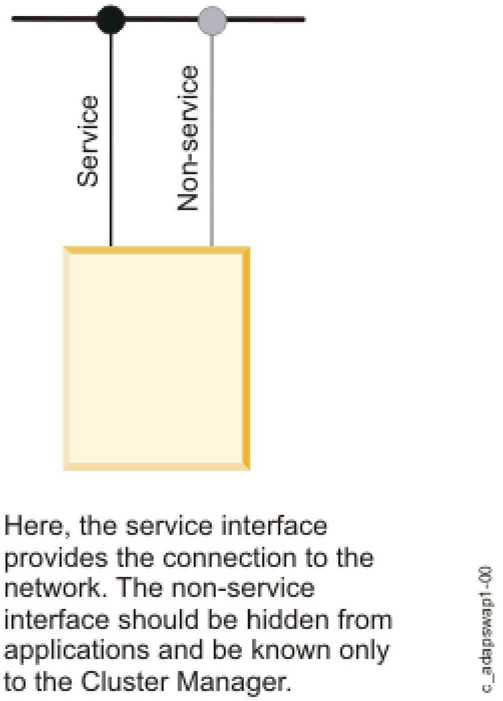
Eliminating communication interfaces as a single point of failure

The PowerHA SystemMirror software handles failures of network interfaces on which a service IP label is configured.

Two types of such failures are:

- Out of two network interfaces configured on a node, the network interface with a service IP label fails, but an additional backup network interface card remains available on the same node. In this case, the Cluster Manager swaps the roles of these two interface cards on that node. Such a network interface failure is transparent to you except for a small delay while the system reconfigures the network interface on a node.
- Out of two network interfaces configured on a node, an additional, or a backup network interface fails, but the network interface with a service IP label configured on it remains available. In this case, the Cluster Manager detects a backup network interface failure, logs the event, and sends a message to the system console. If you want additional processing, you can customize the processing for this event.

The following figures illustrate network interface swapping that occurs on the same node:



Here, the service interface provides the connection to the network. The non-service interface should be hidden from applications and be known only to the Cluster Manager.

Figure 15. Configuration before network adapter swap

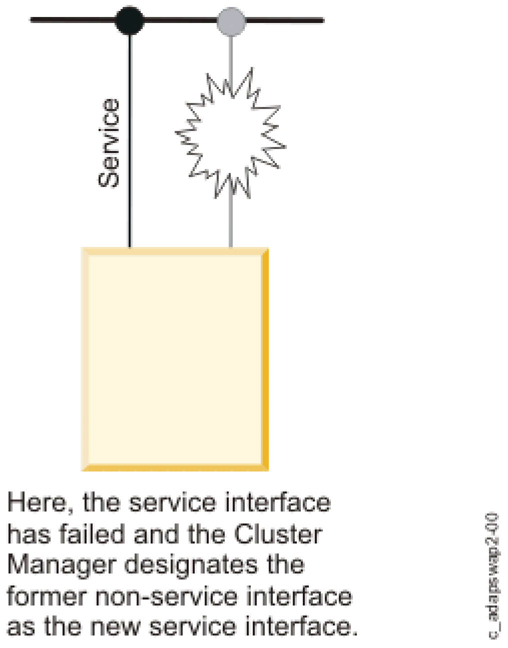

Service
Here, the service interface has failed and the Cluster Manager designates the former non-service interface as the new service interface.
c_adapswap2-00

Figure 16. Configuration after network adapter swap

Eliminating networks as a single point of failure

Network failure occurs when a PowerHA SystemMirror network fails for all the nodes in a cluster. This type of failure occurs when none of the cluster nodes can access each other using any of the network interface cards configured for a given PowerHA SystemMirror network.

The following figure illustrates a network failure:

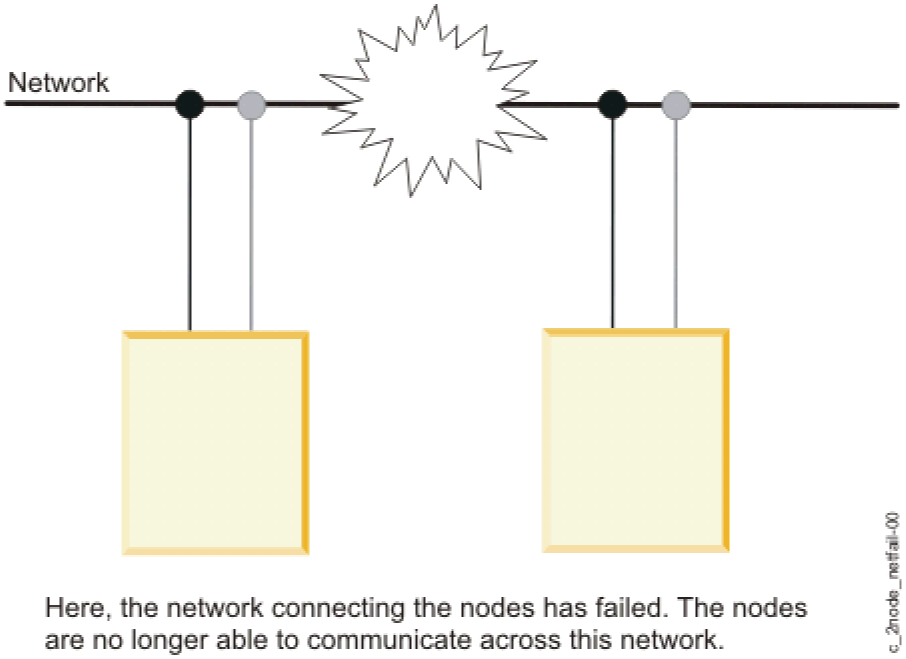

Network
Here, the network connecting the nodes has failed. The nodes are no longer able to communicate across this network.
c_2node_netfail-00

Figure 17. Network failure

The PowerHA SystemMirror software's first line of defense against a network failure is to have the nodes in the cluster connected by multiple networks. If one network fails, the PowerHA SystemMirror software uses a network that is still available for cluster traffic and for monitoring the status of the nodes.

You can specify additional actions to process a network failure - for example, rerouting through an alternate network. Having at least two networks to guard against network failure is highly recommended.

When a local network failure event occurs, the Cluster Manager takes selective recovery actions for resource groups containing a service IP label connected to that network. The Cluster Manager attempts to move only the resource groups affected by the local network failure event, rather than all resource groups on a particular node.

Node isolation and partitioned clusters

Node isolation is when all networks connecting two or more parts of the cluster fail. Each group (one or more) of nodes is completely isolated from the other groups. A partitioned cluster is a cluster in which certain groups of nodes are unable to communicate with other groups of nodes.

In the following illustration of a partitioned cluster, Node A and Node C are on one side of the partition and Node B and Node D are on the other side of the partition.

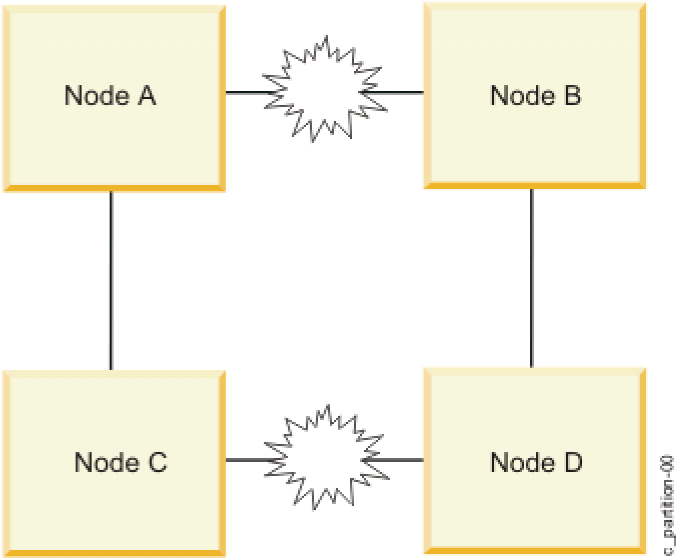


Figure 18. Partitioned cluster

The problem with a partitioned cluster is that the nodes on one side of the partition interpret the absence of heartbeats from the nodes on the other side of the partition to mean that those nodes have failed and then generate node failure events for those nodes. Once this occurs, nodes on each side of the cluster (if so configured) attempt to take over resources from a node that is still active and, therefore, still legitimately owns those resources. These attempted takeovers can cause unpredictable results in the cluster, for example, data corruption due to a disk being reset.

Using Storage Area Networks to prevent partitioning

To prevent the loss of a TCP/IP network from causing a partitioned cluster, the Cluster Aware AIX software will heartbeat through the storage area network that spans the cluster nodes.

Related information

Planning PowerHA SystemMirror

Eliminating disks and disk adapters as a single point of failure

The PowerHA SystemMirror software does not itself directly handle disk and disk adapter failures. Rather, AIX handles these failures through LVM mirroring on disks and by internal data redundancy on the disks that have that capability.

For example, by configuring the system with multiple storage adapters, heart beating through the storage area network, and LVM mirroring, any single component in the disk subsystem (adapter, cabling, disks) can fail without causing unavailability of data on the disk.

If you are using the IBM 2105 ESS or other RAID storage subsystems, the disk array itself is responsible for providing data redundancy.

AIX error notification facility

With the AIX Error Notification facility you can detect an event not specifically monitored by the PowerHA SystemMirror software. For example, a disk adapter fails to program a response to the event.

Permanent hardware errors on disk drives, controllers, or adapters can affect the fault resiliency of data. By monitoring these errors through error notification methods, you can assess the impact of a failure on the cluster's ability to provide high availability. A simple implementation of error notification would be to send a mail message to the system administrator to investigate the problem further. A more complex implementation could include logic to analyze the failure and decide whether to continue processing, stop processing, or escalate the failure to a node failure and have the takeover node make the volume group resources available to clients.

It is strongly recommended that you implement an error notification method for all errors that affect the disk subsystem. Doing so ensures that degraded fault resiliency does not remain undetected.

AIX error notification methods are automatically used in PowerHA SystemMirror to monitor certain recoverable LVM errors, such as volume group loss errors.

Automatic error notification

You can automatically configure error notification for certain cluster resources using a specific option in SMIT. If you select this option, error notification is turned on automatically on all nodes in the cluster for particular devices.

Certain nonrecoverable error types, such as disk adapter errors, are supported by automatic error notification. This feature does not support media errors, recovered errors, or temporary errors. One of two error notification methods is assigned for all error types supported by automatic error notification.

In addition, if you add a volume group to a resource group, PowerHA SystemMirror creates an AIX Error Notification method for it. In the case where a volume group loses quorum, PowerHA SystemMirror uses this method to selectively move the affected resource group to another node. Do not edit or alter the error notification methods that are generated by PowerHA SystemMirror.

The PowerHA SystemMirror error notification facility does not support MPIO disks.

Minimizing scheduled downtime with PowerHA SystemMirror

The PowerHA SystemMirror software enables you to perform most routine maintenance tasks on an active cluster dynamically, without having to stop and then restart cluster services to make the changed configuration the active configuration.

Starting cluster services without stopping applications

You can start the PowerHA SystemMirror cluster services on the node(s) without stopping your applications.

Related information

Starting and stopping cluster services

Dynamic automatic reconfiguration

Dynamic automatic reconfiguration (DARE) is triggered when you synchronize the cluster configuration after making changes on an active cluster. Applying a cluster snapshot using SMIT also triggers a dynamic reconfiguration event.

For example, to add a node to a running cluster, you simply connect the node to the cluster, add the node to the cluster topology on any of the existing cluster nodes, and synchronize the cluster. The new node is added to the cluster topology definition on all cluster nodes and the changed configuration becomes the currently active configuration. After the dynamic reconfiguration event completes, you can start cluster services on the new node.

PowerHA SystemMirror verifies the modified configuration before making it the currently active configuration to ensure that the changes you make result in a valid configuration.

How dynamic reconfiguration works

When dynamic automatic reconfiguration (DARE) is used on a running cluster, and PowerHA SystemMirror starts, DARE creates a private copy of the PowerHA SystemMirror-specific object classes stored in the system default Object Data Model (ODM).

From now on, the ODM is referred to as the PowerHA SystemMirror Configuration Database. Two directories store configuration database data:

- The Active Configuration Directory (ACD), a private directory, stores the PowerHA SystemMirror Configuration Database data for reference by all the PowerHA SystemMirror daemons, scripts, and utilities on a running node.
- The Default Configuration Directory (DCD), the system default directory, stores PowerHA SystemMirror configuration database and data.

Note: The operation of DARE is described here for completeness. No manual intervention is required to ensure that PowerHA SystemMirror carries out these operations. PowerHA SystemMirror correctly manages all dynamic reconfiguration operations in the cluster.

The DCD is the directory named **/etc/objrepos**. This directory contains the default system object classes, such as the customized device database (CuDv) and the predefined device database (PdDv), as well as the specific PowerHA SystemMirror object classes. The ACD is **/usr/es/sbin/cluster/etc/objrepos/active**.

Note: When you configure a cluster, you modify the PowerHA SystemMirror configuration database data stored in the DCD - not data in the ACD. SMIT and other PowerHA SystemMirror configuration utilities all modify the PowerHA SystemMirror configuration database data in the DCD. In addition, all user commands that display PowerHA SystemMirror configuration database data, such as the **cllsif** command, read data from the DCD.

The following figure illustrates how the PowerHA SystemMirror daemons, scripts, and utilities all reference the ACD when accessing configuration information.

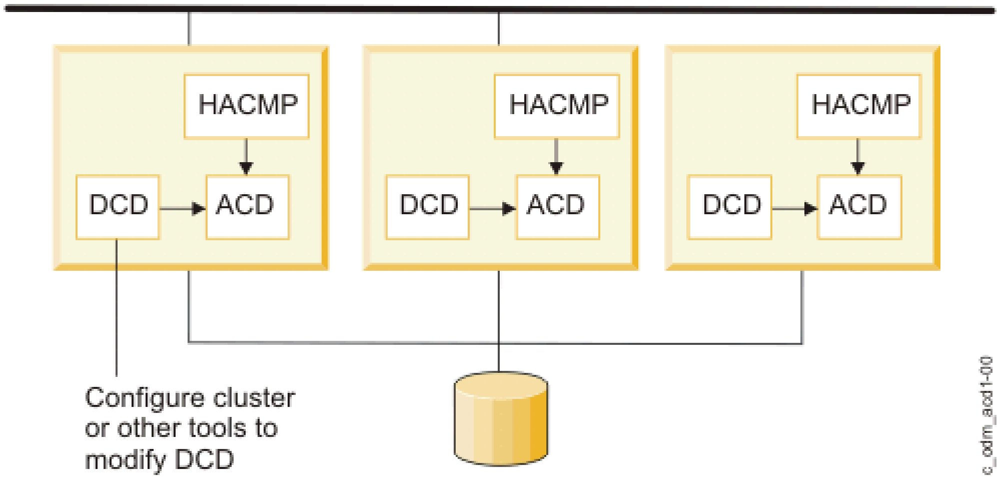


Figure 19. Relationship of PowerHA SystemMirror to ACD when a cluster starts.

Reconfiguring a cluster dynamically

The PowerHA SystemMirror software depends on the location of certain PowerHA SystemMirror configuration database repositories to store configuration data.

The presence or absence of these repositories is sometimes used to determine steps taken during cluster configuration and operation. The ODMPATH environment variable allows PowerHA SystemMirror configuration database commands and subroutines to query locations other than the default location (held in the ODMDIR environment variable) if the queried object does not exist in the default location. You can set this variable, but it must not be set to include the **/etc/objrepos** directory or you will lose the integrity of the PowerHA SystemMirror configuration information.

To change the configuration of an active cluster, you modify the cluster definition stored in the specific PowerHA SystemMirror configuration database classes stored in the default configuration directory (DCD) using SMIT. When you change the cluster configuration in an active cluster, you use the same SMIT paths to make the changes, but the changes do not take effect immediately. Therefore, you can make several changes in one operation. When you synchronize your configuration across all cluster nodes, a cluster-wide dynamic reconfiguration event occurs. When PowerHA SystemMirror processes a dynamic reconfiguration event, it updates the PowerHA SystemMirror configuration database object classes stored in the DCD on each cluster and replaces the PowerHA SystemMirror configuration database data stored in the ACD with the new PowerHA SystemMirror configuration database data in the DCD, in a coordinated, cluster-wide transition. It also refreshes the cluster daemons so that they reference the new configuration data.

After this processing, the cluster heartbeat is suspended briefly and the cluster is in an unstable state. The changed configuration becomes the active configuration. After cluster services are started on the newly added node, it is automatically integrated into the cluster.

The following figure illustrates the processing involved with adding a node to an active cluster using dynamic reconfiguration.

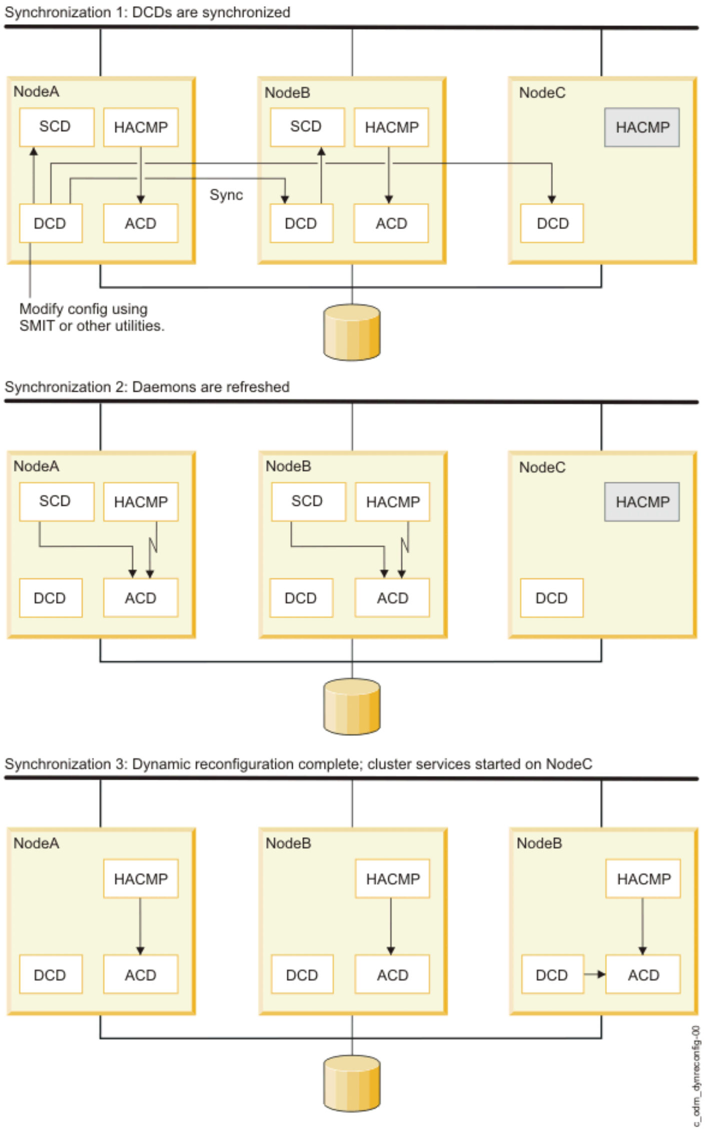


Figure 20. Dynamic reconfiguration processing

The node to be added is connected to a running cluster, but cluster services are inactive on this node. The configuration is redefined on NodeA. When the changes to the configuration are synchronized, the PowerHA SystemMirror configuration database data stored in the DCD on NodeA is copied to the DCDs on other cluster nodes and a dynamic reconfiguration event is triggered. PowerHA SystemMirror copies the new PowerHA SystemMirror configuration database data in the DCD into a temporary location on each node, called the Staging Configuration Directory (SCD). The location of the SCD is `/usr/es/sbin/cluster/etc/objrepos/stage`. By using this temporary location, PowerHA SystemMirror allows you to start making additional configuration changes while a dynamic reconfiguration is in progress. Before copying the new PowerHA SystemMirror configuration database data in the SCD over the current PowerHA SystemMirror configuration database data in the ACD, PowerHA SystemMirror verifies the new configuration.

Note: You can initiate a second reconfiguration while a dynamic reconfiguration is in progress, but you cannot synchronize it. The presence of an SCD on any cluster node acts as a lock, preventing the initiation of a new dynamic reconfiguration.

Resource group management

You can use the resource group management (clRGmove) utility to move resource groups to other cluster nodes and sites, or take them online or offline without stopping cluster services.

This gives you the flexibility for managing resource groups and their applications. You can also use this utility to free the node of any resource groups to perform system maintenance on a particular cluster node. The PowerHA SystemMirror resource group management utility (clRGmove) is significantly improved, which makes it easier for you to move the resource groups around for cluster management. In addition, you can more easily understand the consequences of manually moving resource groups. For example, you can predict whether the groups will stay on the nodes to which they were moved. PowerHA SystemMirror follows this principle: In all cases, when you move the resource groups, they stay on the nodes until you move them again. PowerHA SystemMirror moves them around when it needs to recover them.

If you want to check whether the resource group is currently hosted on the highest priority node that is now available, PowerHA SystemMirror presents intelligent pick list choices for nodes and sites. For example, if the group is currently hosted on one node, and PowerHA SystemMirror finds another node that has a higher priority, the SMIT pick list with destination nodes indicates which node has a higher priority. This way, you can always choose to move this group to this node. When you move groups to other nodes, these rules apply:

- For resource groups with a fallback policy of Never Fallback, moving a group will have no effect on the behavior of that group during future cluster events. The same is true for resource groups with the site fallback policy Online on Either Site.
- For resource groups with a fallback policy other than Never Fallback or Prefer Primary Site, moving a group results in a destination node becoming an acting highest priority node until you move it again. After you move the node, it becomes an acting highest priority node.

An important consideration for this behavior has to do with resource groups that have Fallback to Highest Priority Node policy or Prefer Primary Site policy. When you move such a resource group to a node other than its highest priority node, or to a primary site, the node to which it was moved becomes its temporarily preferred node while not being its highest priority node (as configured). Such groups stay on the nodes to which they were moved until you move them again. The groups also fall back to these nodes or sites.

Related concepts

PowerHA SystemMirror configuration process and facilities
These topics provide an overview of the PowerHA SystemMirror cluster configuration process and the administrative tools supplied with the PowerHA SystemMirror software.

Related information

Managing resource groups in a cluster

Planning resource groups

User-requested resource group management vs. automatic resource group management

In general, to keep applications highly available, PowerHA SystemMirror automatically manages (and sometimes moves) resource groups and applications included in them.

For instance, when it is necessary to recover a resource group, PowerHA SystemMirror might attempt to recover it automatically on another node during fallover or fallback operations. While moving a group, PowerHA SystemMirror adheres to the resource group policies that you specified, and other settings (for instance, rather than automatically recovering a failed resource group on another node, you can tell PowerHA SystemMirror to just notify you of the group's failure).

When you request PowerHA SystemMirror to perform resource group management, it uses the *clRGmove* utility, which moves resource groups by calling an *rg_move* event.

Note: When troubleshooting log files, it is important to distinguish between an *rg_move* event that in some cases is triggered automatically by PowerHA SystemMirror, and an *rg_move* event that occurs when you request PowerHA SystemMirror to manage resource groups for you. To identify the causes of operations performed on the resource groups in the cluster, look for the command output in SMIT and for information in the *hacmp.out* file.

Resource group management operations

You can use resource group management to perform a number of tasks.

Use resource group management to:

- Move a resource group from the node on one site to the node on another site.
- Move a resource group from one node to another.

  In a working cluster, temporarily move a nonconcurrent resource group from a node it currently resides on to any destination node. Resource groups that you move continue to behave consistently with the way you configured them, that is, they follow the startup, fallover and fallback policies specified for them. The SMIT user interface lets you clearly specify and predict the resource group's behavior, if you decide to move it to another node.

  If you use SMIT to move a resource group to another node, it remains on its new destination node until you manually move it again. PowerHA SystemMirror might need to move it during a fallover.
- Move the resource group back to the node that was originally its highest priority.

  The resource group might or might not have a fallback policy. If a resource group has a fallback policy of Fallback to Highest Priority Node, after you move it, the group assumes that the “new” node is now its preferred temporary location, and falls back to this node. To change this behavior, you can always move the group back to the node that was originally its highest priority node.

  Similarly, if you have a resource group that has a fallback policy Never Fallback, once you move this resource group, it will not move back to the node from which it was moved but will remain on its new destination node, until you move it again to another node. This way, you can be assured that the group always follows the Never Fallback policy that you specified for it.
- Bring a resource group online or offline on one or all nodes in the cluster.

Related information

Administering PowerHA SystemMirror

Cluster single point of control

With the Cluster single point of control (C-SPOC) utility, you can make changes to the whole cluster from a single cluster node.

Instead of performing administrative tasks on each cluster node, you can use the SMIT interface to issue a C-SPOC command once, on a single node, and the change is propagated across all cluster nodes.

Related concepts

PowerHA SystemMirror system management with C-SPOC

To facilitate management of a cluster, PowerHA SystemMirror provides a way to run commands from one node and then verify and synchronize the changes to all the other nodes. You can use the PowerHA SystemMirror System Management tool, the Cluster Single Point of Control (C-SPOC) to add users, files, and hardware automatically without stopping mission-critical jobs.

Dynamic adapter swap

The dynamic adapter swap functionality lets you swap the IP address of an active network interface card (NIC) with the IP address of a user-specified active, available "backup" network interface card on the same node and network.

Cluster services do not have to be stopped to perform the swap.

This feature can be used to move an IP address off a network interface card that is behaving erratically, to another NIC without shutting down the node. It can also be used if a hot pluggable NIC is being replaced on the node. Hot pluggable NICs can be physically removed and replaced without powering off the node. When the (hot pluggable) NIC that you want to replace is removed from the node, PowerHA SystemMirror makes the NIC unavailable as a backup.

You can configure adapter swap using SMIT. The service IP address is moved from its current NIC to a user-specified NIC. The service IP address then becomes an available "backup" address. When the new card is placed in the node, the NIC is incorporated into the cluster as an available "backup" again. You can then swap the IP address from the backup NIC to the original NIC.

Note: This type of dynamic adapter swap can only be performed within a single node. You cannot swap the IP address with the IP address on a different node with this functionality. To move a service IP address to another node, move its resource group using the Resource Group Management utility.

Automatic verification and synchronization

Automatic verification and synchronization minimizes downtime when you add a node to your cluster.

This process runs prior to starting cluster services and checks to make sure that nodes joining a cluster are synchronized appropriately. This process checks nodes entering either active or inactive configurations. Some examples of the typical configuration inconsistencies corrected by automatic verification and synchronization are:

- IP addresses are configured on the network interfaces that RSCT expects
- Shared volume groups are not set to be automatically varied on
- File Systems are not set to be automatically mounted.

If any additional configuration errors are found, cluster services are not started on the node, and detailed error messages enable you to resolve the inconsistencies.

Related information

Verifying and synchronizing a PowerHA SystemMirror cluster

Minimizing unscheduled downtime

Another important goal with PowerHA SystemMirror is to minimize unscheduled downtime in response to unplanned cluster component failures.

The PowerHA SystemMirror software provides the following features to minimize unscheduled downtime:

- Fast recovery to speed up the fallover in large clusters
- A delayed fallback timer to allow a custom resource group to fall back at a specified time
- IPAT via IP aliases to speed up the processing during recovery of service IP labels
- Automatic recovery of resource groups that are in the ERROR state, whenever a cluster node comes up. For more information, see the following section.

Recovering resource groups on node startup

The Cluster Manager tries to bring the resource groups that are currently in the ERROR state into the online (active) state on the joining node. This further increases the chances of bringing the applications back online. When a node starts up, if a resource group is in the ERROR state on any node in the cluster, this node attempts to acquire the resource group.

Note that the node must be included in the node list for the resource group.

The resource group recovery on node startup is different for nonconcurrent and concurrent resource groups:

- If the starting node fails to activate a nonconcurrent resource group that is in the ERROR state, the resource group continues to fall over to another node in the node list, if a node is available. The fallover action continues until all available nodes in the node list have been tried.
- If the starting node fails to activate a concurrent resource group that is in the ERROR state on the node, the concurrent resource group is abandoned in the ERROR state on that node. Note that the resource group might still remain online on other nodes.

Fast recovery

The PowerHA SystemMirror fast recovery feature speeds up fallover in large clusters.

Fast recovery allows you to select a file systems consistency check and a file systems recovery method:

- If you configure a file system to use a consistency check and a recovery method, it saves time by running logredo rather than fsck on each file system. If the subsequent mount fails, then it runs a full fsck .

If a file system suffers damage in a failure but can still be mounted, logredo might not succeed in fixing the damage, producing an error during data access.

- In addition, it saves time by acquiring, releasing, and falling over all resource groups and file systems in parallel, rather than serially.

Do not set the system to run these commands in parallel if you have shared, nested file systems. These must be recovered sequentially. (Note that the cluster verification utility does not report file system and fast recovery inconsistencies.) The varyonvg and varyoffvg commands always run on volume groups in parallel, regardless of the setting of the recovery method.

Delayed fallback timer for resource groups

The Delayed fallback timer allows a resource group to fall back to the higher priority node at a time that you specify.

The resource group that has a delayed fallback timer configured and that currently resides on a non-home node falls back to the higher priority node at the recurring time (daily, weekly, monthly or yearly), or on a specified date.

Related information

Planning PowerHA SystemMirror

Minimizing takeover time

In the case of a cluster failure, enhanced concurrent volume groups are taken over faster than in previous releases of PowerHA SystemMirror due to the improved disk takeover mechanism.

In the case of a cluster failure, enhanced concurrent volume groups are taken over faster than in previous releases of PowerHA SystemMirror due to the improved disk takeover mechanism. PowerHA SystemMirror automatically detects enhanced concurrent volume groups and ensures that the faster option for volume group takeover is launched in the event of a node failure, as long as you have included the enhanced concurrent mode volume groups (or convert the existing volume groups to enhanced concurrent volume groups) in your nonconcurrent resource groups.

This functionality is especially useful for fallover of volume groups made up of a large number of disks.

During fast disk takeover, PowerHA SystemMirror skips the extra processing needed to break the disk reserves, or update and synchronize the LVM information by running lazy update. As a result, the disk takeover mechanism of PowerHA SystemMirror used for enhanced concurrent volume groups is faster than disk takeover used for standard volume groups included in nonconcurrent resource groups

Maximizing disaster recovery

PowerHA SystemMirror can be an integral part of a comprehensive disaster recovery plan for your enterprise.

Several possible ways to distribute backup copies of data to different sites, for possible disaster recovery operations follow:

- PowerHA SystemMirror Enterprise Edition for Geographic Logical Volume Manager (GLVM)
- PowerHA SystemMirror Enterprise Edition for Metro Mirror (synchronous PPRC with ESS and DS systems)
- Cross-site LVM mirroring

Cross-site LVM mirroring

You can set up disks that are located at two different sites for remote LVM mirroring, using a Storage Area Network (SAN), for example. Cross-site LVM mirroring replicates data 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 (servers) within the distance supported by the Fibre Channel. Thus, two or more servers (nodes) that are located at different sites can access the same physical disks, which can be separated by some distance, through the common SAN. The disks can be combined into a volume group by using the AIX Logical Volume Manager, 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 still has the latest information, so 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. Automatic synchronization is not possible for all cases, but you can use C-SPOC to synchronize the data manually from the surviving mirrors to stagnant mirrors after a disk or site failure and subsequent reintegration.

Cluster events

This section describes how the PowerHA SystemMirror software responds to changes in a cluster to maintain high availability.

The PowerHA SystemMirror cluster software monitors all the components that make up the highly available application including disks, network interfaces, nodes and the applications themselves. The Cluster Manager uses different methods for monitoring different resources:

- RSCT subsystem is responsible for monitoring networks and nodes.
- The AIX LVM subsystem produces error notifications for volume group quorum loss.
- The Cluster Manager itself dispatches application monitors.

A PowerHA SystemMirror cluster environment is event-driven. An event is a change of status within a cluster that the Cluster Manager recognizes and processes. A cluster event can be triggered by a change affecting a network interface card, network, or node, or by the cluster reconfiguration process exceeding its time limit. When the Cluster Manager detects a change in cluster status, it executes a script designated to handle the event and its subevents.

Note: The logic of cluster events is described here for completeness. No manual intervention is required to ensure that PowerHA SystemMirror carries out cluster events correctly.

The following examples show some events the Cluster Manager recognizes:

- node_up and node_up_complete events (a node is joining the cluster)
- node_down and node_down_complete events (a node is leaving the cluster)
- local or global network_down event (a network has failed)
- global network_unstable event (a network is changing its state continuously)
- global network_stable event (a network is no longer changing its state continuously)
- network_up event (a network is connected)
- swap_adapter event (a network adapter has failed and is swapped with a new adapter)
- dynamic reconfiguration events
- site_up and site_up_complete events (a site is joining the cluster)
- site_down and site_down_complete events (a site is leaving the cluster)

When a cluster event occurs, the Cluster Manager runs the corresponding event script for that event. As the event script is being processed, a series of subevent scripts might be executed. The PowerHA SystemMirror software provides a script for each event and subevent. The default scripts are located in the `/usr/es/sbin/cluster/events` directory.

By default, the Cluster Manager calls the corresponding event script supplied with the PowerHA SystemMirror software for a specific event.

You can specify additional processing information to customize event handling for your site if needed.

Processing cluster events

The two primary cluster events that PowerHA SystemMirror software handles are fallover and reintegration.

- Fallover refers to the actions taken by the PowerHA SystemMirror software when a cluster component fails or a node leaves the cluster.
- Reintegration refers to the actions that occur within the cluster when a component that had previously abandoned the cluster returns to the cluster.

Event scripts control both types of actions. During event script processing, cluster-aware application programs see the state of the cluster as unstable.

Fallover

A fallover occurs when a resource group moves from its home node to another node because its home node leaves the cluster.

Nodes leave the cluster either by a planned transition (a node shutdown or stopping cluster services on a node), or by failure. In the former case, the Cluster Manager controls the release of resources held by the exiting node and the acquisition of these resources by nodes still active in the cluster. When necessary, you can override the release and acquisition of resources (for example, to perform system maintenance). You can also postpone the acquisition of the resources by integrating nodes (by setting the delayed fallback timer for custom resource groups).

Node failure begins when a node monitoring a neighboring node ceases to receive keepalive traffic for a defined period of time. If the other cluster nodes agree that the failure is a node failure, the failing node is removed from the cluster and its resources are taken over by the active nodes configured to do so.

If other components, such as a network interface card, fail, the Cluster Manager runs an event script to switch network traffic to a backup network interface card (if present).

Reintegration

A reintegration, or a fallback occurs when a resource group moves to a node that has just joined the cluster. When a node joins a running cluster, the cluster becomes temporarily unstable. The member nodes coordinate the beginning of the join process and then run event scripts to release any resources the joining node is configured to take over. The joining node then runs an event script to take over these resources. Finally, the joining node becomes a member of the cluster. At this point, the cluster is stable again.

Customizing event processing

The PowerHA SystemMirror software has an event customization facility that you can use to tailor event processing.

PowerHA SystemMirror events can be customized such that you can specify your own scripts to be run when the cluster events run.

You can customize PowerHA SystemMirror cluster events to include the following methods:

1. Notification methods are run when a PowerHA SystemMirror event runs.
2. Pre-events are methods that are called before the PowerHA SystemMirror event runs.
3. Post-events are methods that are called after the PowerHA SystemMirror event runs.

Related information

Configuring cluster events

Customizing event duration

PowerHA SystemMirror software issues a system warning each time a cluster event takes more time to complete than a specified timeout period.

Using the SMIT interface, you can customize the time period allowed for a cluster event to complete before PowerHA SystemMirror issues a system warning for it.

PowerHA SystemMirror cluster configurations

This chapter provides examples of the types of cluster configurations supported by the PowerHA SystemMirror software.

This list is by no means an exhaustive catalog of the possible configurations you can define using the PowerHA SystemMirror software. Rather, use them as a starting point for thinking about the cluster configuration best suited to your environment.

Standby configurations

Standby configurations are the traditional redundant hardware configurations where one or more standby nodes stand idle, waiting for a server node to leave the cluster.

Concurrent resource groups require all nodes to have simultaneous access to the resource group and cannot be used in a standby configuration.

Example of Standby configurations with online on home node only startup policy

This example shows resource groups with the online on home node only startup policy, fallover to next priority node in the list fallover policy, and fallback to higher priority node in the list fallback policy.

In the following standby configuration, the resource groups have these policies.

Startup policy

Online on Home Node Only.

Fallover policy
Fallover to Next Priority Node in the List.

Fallback policy
Fallback to Higher Priority Node in the List.

In the figure, a low number indicates a higher priority.

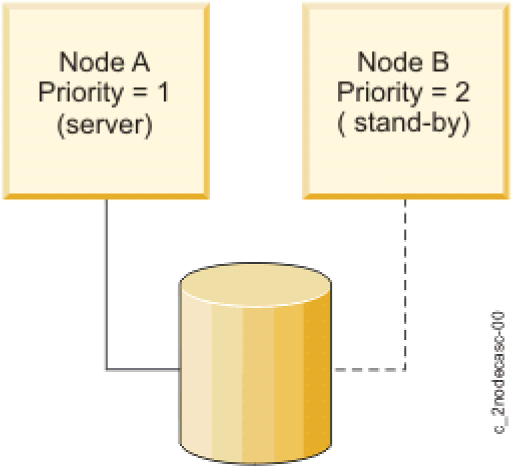


Figure 21. One-for-one standby configuration where IP label returns to the home node

In this setup, the cluster resources are defined as part of a single resource group. A node list is then defined as consisting of two nodes. The first node, Node A, is assigned a takeover that is ownership priority of 1. The second node, Node B, is assigned a takeover priority of 2.

At cluster startup, Node A, which has a priority of 1, assumes ownership of the resource group. Node A is the "server" node. Node B, which has a priority of 2, stands idle, ready should Node A fail or leave the cluster. Node B is, in effect, the "standby".

If the server node leaves the cluster, the standby node assumes control of the resource groups owned by the server, starts the highly available applications, and services clients. The standby node remains active until the node with the higher takeover priority rejoins the cluster. At that point, the standby node releases the resource groups it has taken over, and the server node reclaims them. The standby node then returns to an idle state.

Extending standby configurations

The standby configuration from the previously described example can be easily extended to larger clusters. The advantage of this configuration is that it makes better use of the hardware. The disadvantage is that the cluster can suffer severe performance degradation if more than one server node leaves the cluster.

The following figure illustrates a three-node standby configuration using the resource groups with these policies.

Startup policy
Online on Home Node Only.

Fallover policy
Fallover to Next Priority Node in the List.

Fallback policy
Fallback to Higher Priority Node in the List.

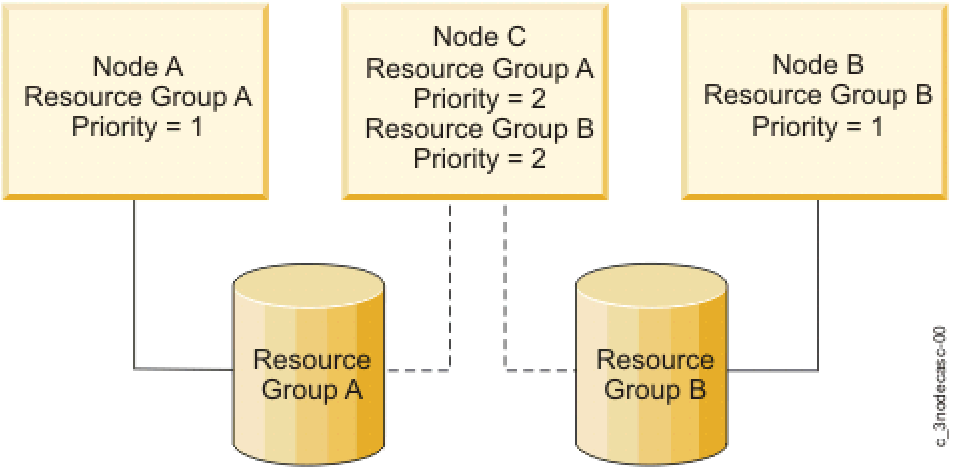


Figure 22. One-for-two standby configuration with three resource groups

In this configuration, two separate resource groups, A and B, and a separate node list for each resource group exist. The node list for Resource Group A consists of Node A and Node C. Node A has a takeover priority of 1, while Node C has a takeover priority of 2. The node list for Resource Group B consists of Node B and Node C. Node B has a takeover priority of 1; Node C again has a takeover priority of 2. A resource group can be owned by only a single node in a nonconcurrent configuration.

Since each resource group has a different node at the head of its node list, the cluster's workload is divided, or partitioned, between these two resource groups. Both resource groups, however, have the same node as the standby in their node lists. If either server node leaves the cluster, the standby node assumes control of that server node's resource group and functions as the departed node.

In this example, the standby node has three network interfaces and separate physical connections to each server node's external disk. Therefore, the standby node can, if necessary, take over for both server nodes concurrently. The cluster's performance, however, would most likely degrade while the standby node was functioning as both server nodes.

Example: Standby configurations with online using distribution policy startup

This example explains the resource group policies you must use with standby configurations.

In the following standby configuration, the resource groups have these policies:

- Startup policy: Online Using Distribution Policy (network-based or node-based)
- Fallover policy: Next Priority Node in the List
- Fallback policy: Never Fallback.

This configuration differs from a standby configuration in which the ownership priority of resource groups is not fixed. Rather, the resource group is associated with an IP address that can rotate among nodes. This makes the roles of server and standby fluid, changing over time.

The following figure illustrates the one-for-one standby configuration that is described in this section:

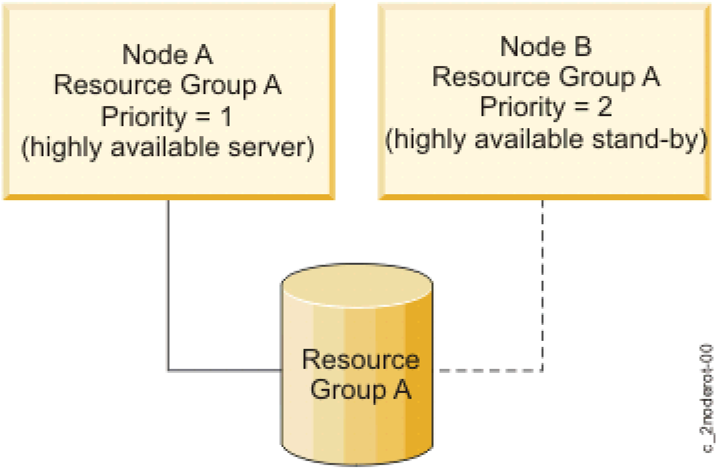


Figure 23. One-for-one standby configuration with resource groups where IP label rotates

At system startup, the resource group attaches to the node that claims the shared IP address. This node owns the resource group for as long as it remains in the cluster. If this node leaves the cluster, the peer node assumes the shared IP address and claims ownership of that resource group. Now, the peer node "owns" the resource group for as long as it remains in the cluster.

When the node that initially claimed the resource group rejoins the cluster, it does not take the resource group back. Rather, it remains idle for as long as the node currently bound to the shared IP address is active in the cluster. Only if the peer node leaves the cluster does the node that initially "owned" the resource group claim it once again. Thus, ownership of resources rotates between nodes.

Extending standby configurations

As with the first example of the standby configuration, configurations from this example can be easily extended to larger clusters. For example, in this one-for-two standby configuration from example the cluster could have two separate resource groups, each of which includes a distinct shared IP address.

At cluster startup, the first two nodes each claim a shared IP address and assume ownership of the resource group associated with that shared IP address. The third node remains idle. If an active node leaves the cluster, the idle node claims that shared IP address and takes control of that resource group.

Takeover configurations

In the takeover configurations, all cluster nodes do useful work, processing part of the cluster's workload. There are no standby nodes. Takeover configurations use hardware resources more efficiently than standby configurations since there is no idle processor. Performance can degrade after node detachment, however, since the load on remaining nodes increases.

One-sided takeover

This configuration has two nodes actively processing work, but only one node providing highly available services to cluster clients. That is, although there are two sets of resources within the cluster (for example, two server applications that handle client requests), only one set of resources needs to be highly available.

The following figure illustrates a two-node, one-sided takeover configuration. In the figure, a low number indicates a higher priority.

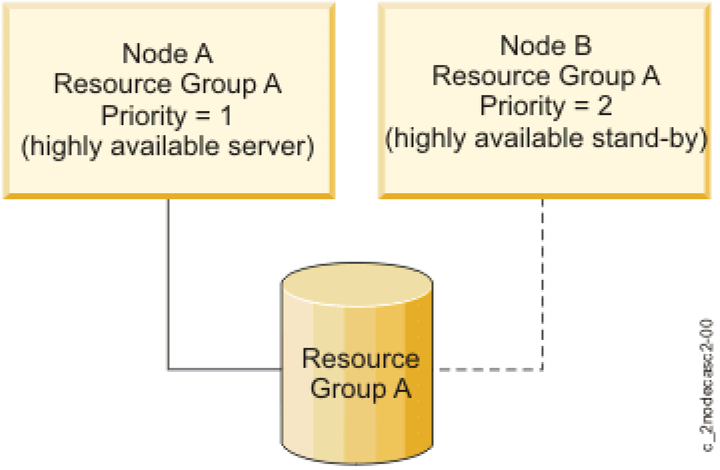


Figure 24. One-sided takeover configuration with resource groups in which IP label returns to the home node

This set of resources is defined as a PowerHA SystemMirror resource group and has a node list that includes both nodes. The second set of resources is not defined as a resource group and, therefore, is *not* highly available.

At cluster startup, Node A (which has a priority of 1) assumes ownership of Resource Group A. Node A, in effect, "owns" Resource Group A. Node B (which has a priority of 2 for Resource Group A) processes its own workload independently of this resource group.

If Node A leaves the cluster, Node B takes control of the shared resources. When Node A rejoins the cluster, Node B releases the shared resources.

If Node B leaves the cluster, however, Node A does not take over any of its resources, since Node B's resources are not defined as part of a highly available resource group in whose chain this node participates.

This configuration is appropriate when a single node is able to run all the critical applications that need to be highly available to cluster clients.

Mutual takeover

The mutual takeover for nonconcurrent access configuration has multiple nodes, each of which provides distinct highly available services to cluster clients. For example, each node might run its own instance of a database and access its own disk.

Furthermore, each node has takeover capacity. If a node leaves the cluster, a surviving node takes over the resource groups owned by the departed node.

The mutual takeover for nonconcurrent access configuration is appropriate when each node in the cluster is running critical applications that need to be highly available and when each processor is able to handle the load of more than one node.

The following figure illustrates a two-node mutual takeover configuration for nonconcurrent access. In the figure, a low number indicates a higher priority.

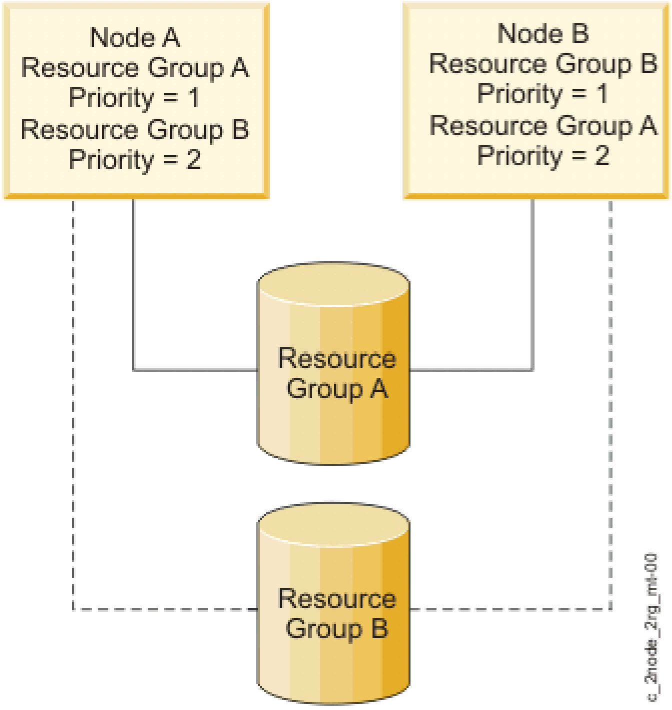


Figure 25. Mutual takeover configuration for nonconcurrent access

The key feature of this configuration is that the cluster's workload is divided, or partitioned, between the nodes. Two resource groups exist, in addition to a separate resource chain for each resource group. The nodes that participate in the resource chains are the same. It is the differing priorities within the chains that designate this configuration as mutual takeover.

The chains for both resource groups consist of Node A and Node B. For Resource Group A, Node A has a takeover priority of 1 and Node B has a takeover priority of 2. For Resource Group B, the takeover priorities are reversed. Here, Node B has a takeover priority of 1 and Node A has a takeover priority of 2.

At cluster startup, Node A assumes ownership of the Resource Group A, while Node B assumes ownership of Resource Group B.

If either node leaves the cluster, its peer node takes control of the departed node's resource group. When the "owner" node for that resource group rejoins the cluster, the takeover node relinquishes the associated resources; they are reacquired by the higher-priority, reintegrating node.

Two-node mutual takeover configuration

In this configuration, both nodes have simultaneous access to the shared disks and own the same disk resources. There is no takeover of shared disks if a node leaves the cluster, since the peer node already has the shared volume group varied on.

The following figure illustrates a two-node mutual takeover configuration for concurrent access:

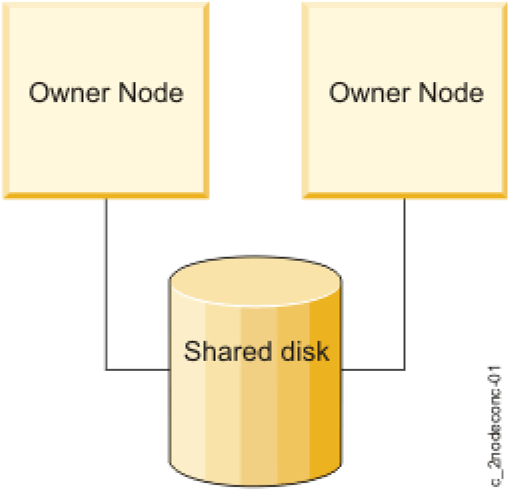


Figure 26. Two-node mutual takeover configuration for concurrent access

In this example, both nodes are running an instance of a server application that accesses the database on the shared disk. The application's proprietary locking model is used to arbitrate application requests for disk resources.

Running multiple instances of the same server application allows the cluster to distribute the processing load. As the load increases, additional nodes can be added to further distribute the load.

Eight-node mutual takeover configuration

In this configuration all nodes have simultaneous access to the shared disks and own the same disk resources.

The following figure illustrates an eight-node mutual takeover configuration for concurrent access:

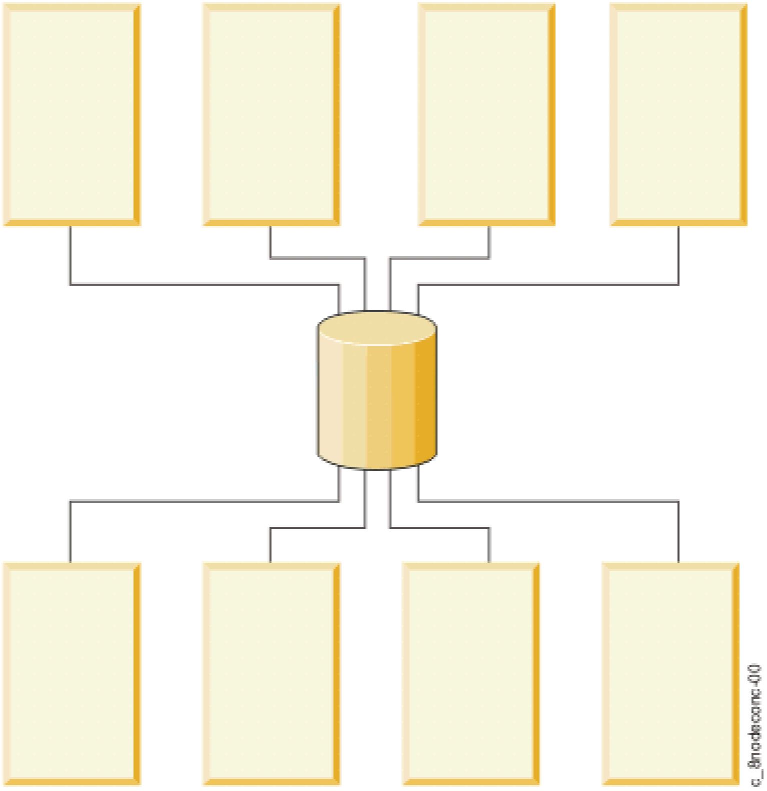


Figure 27. Eight-node mutual takeover

Each node is running a different server application. Clients query a specific application at a specific IP address. Therefore, each application controller and its associated IP address must be defined as part of a nonconcurrent resource group, and all nodes that are potential owners of that resource group must be included in a corresponding node list.

Concurrent access resource groups are supported in clusters with up to 16 nodes in PowerHA SystemMirror.

Cluster configurations with multitiered applications

A typical cluster configuration that could utilize parent and child dependent resource groups is the environment in which an application such as WebSphere depends on another application such as DB2.

In order to satisfy business requirements, a cluster-wide parent and child dependency must be defined between two or more resource groups.

The following figure illustrates the business scenario that utilizes dependencies between applications:

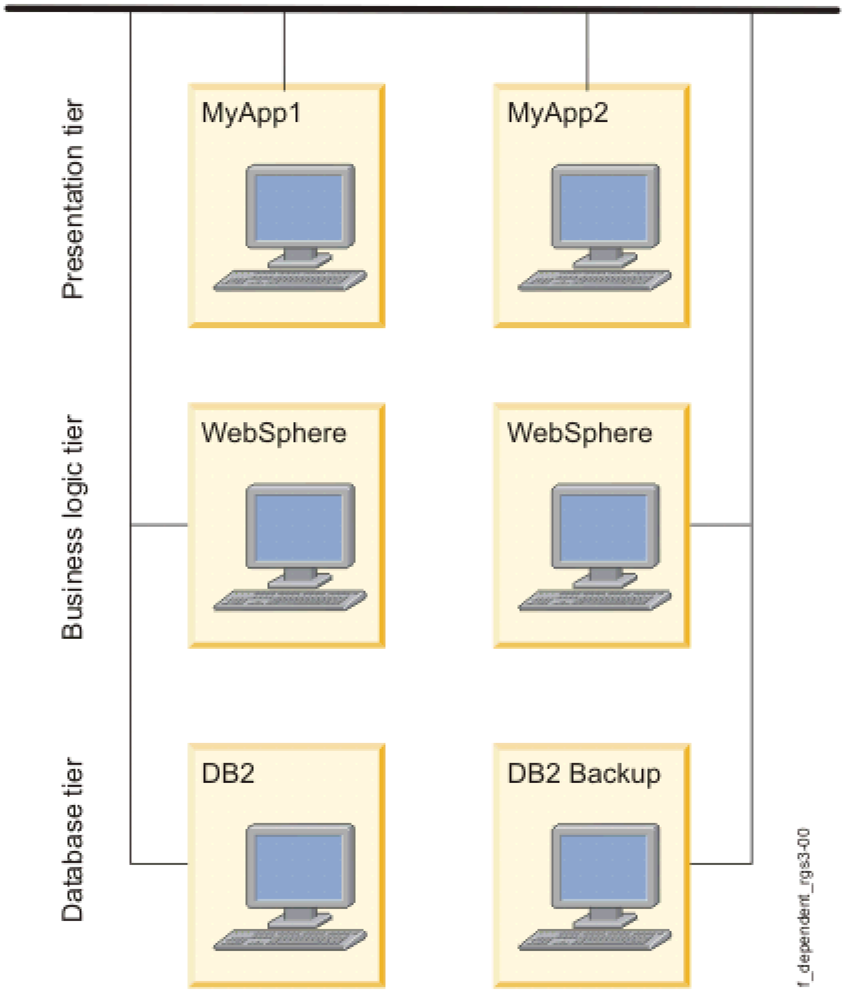


Figure 28. Typical multitiered cluster environment with dependencies between applications

Multitiered applications

Business configurations that use layered, or multitiered applications can also utilize dependent resource groups. For example, the back end 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 SAP require applications to be cycled (stopped and restarted) anytime a database fails. An environment like SAP provides many application services, and the individual application components often need to be controlled in a specific order.

Another area where establishing interdependencies between resource groups proves useful is when system services are required to support application environments. Services such as **cron** jobs for pruning log files or initiating backups need to move from node to node along with an application, but are typically not initiated until the application is established. These services can be built into application controller start and stop scripts. When greater granularity is needed, they can be controlled through pre- and post- event processing. Parent/child dependent resource groups allow an easier way to configure system services to be dependent upon applications they serve.

Related concepts

Resource group dependencies
PowerHA SystemMirror supports resource group ordering and customized serial processing of resources to accommodate cluster configurations where a dependency exists between applications residing in different resource groups.

Cluster configurations with resource group location dependencies

You can configure the cluster so that certain applications stay on the same node, or on different nodes not only at startup, but during fallover and fallback events. To do this, you configure the selected resource groups as part of a location dependency set.

Publishing model with same node and different nodes dependencies

The fictitious company, The XYZ Publishing, follows a business continuity model that involves separating the different platforms used to develop the web content. XYZ Publishinguses location dependency policies to keep some resource groups strictly on separate nodes and others together on the same node.

The Production database (PDB) and Production application (PApp) are hosted on the same node to facilitate maintenance (and perhaps the highest priority node for these resource groups has the most memory or faster processor). It also makes sense to set up a parent/child relation between them, since the application depends on the database. The database must be online for the application to function. The same conditions are true for the System Database (SDB) and the System application (Sapp) and for the QA Database (QADB) and the QA application (QAapp).

Since keeping the production database and application running is the highest priority, it makes sense to configure the cluster so that the three database resource groups stay on different nodes (make them an Online On Different Nodes dependency set), and assign the PDB resource group with the high priority. The SDB is the itermediate priority and the QADB is the low priority.

The databases and their related applications are each configured to belong to an Online On Same Node dependency set.

PowerHA SystemMirror handles these groups somewhat differently depending on how you configure startup, fallover, and fallback policies. It makes sense to have the participating node lists differ for each database and application set to facilitate keeping these resource groups on the preferred nodes.

The following figure shows the basic configuration of the three nodes and six resource groups.

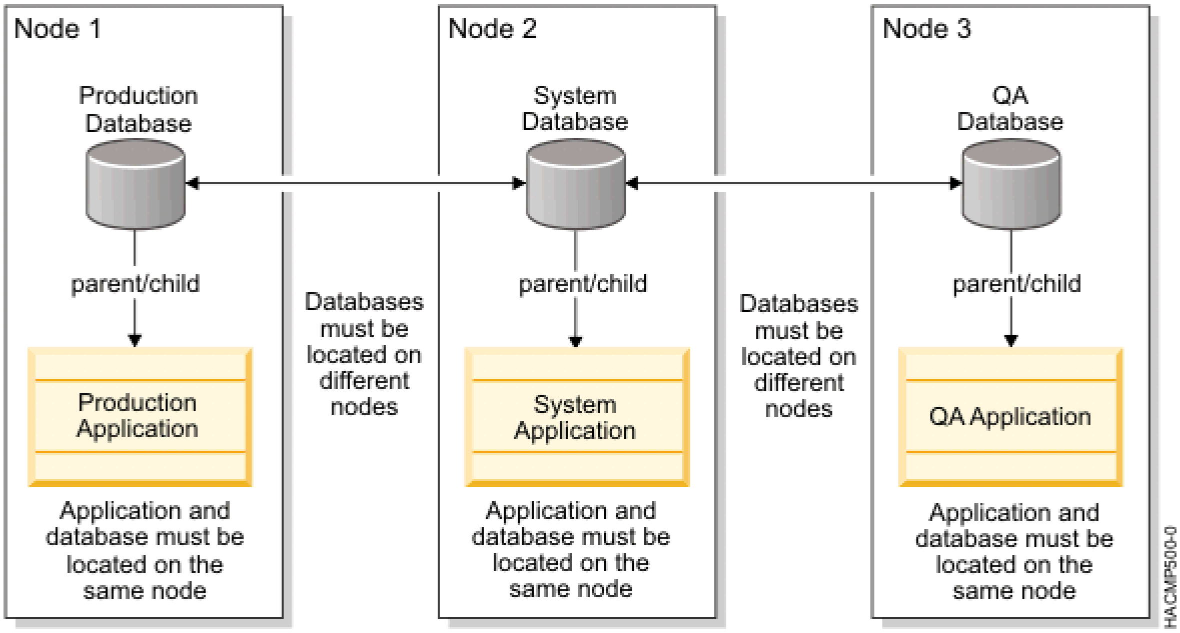

Node 1
Production Database
parent/child
Production Application
Application and database must be located on the same node
Databases must be located on different nodes
Node 2
System Database
parent/child
System Application
Application and database must be located on the same node
Databases must be located on different nodes
Node 3
QA Database
parent/child
QA Application
Application and database must be located on the same node
HACMP500-0

Figure 29. Publishing model with parent dependencies or child dependencies and location dependencies

Resource group policies

For the sake of illustration of this case, all six resource groups might have the following behavioral policies:

- Startup Policy: Online On First Available Node
- Fallover Policy: Fallover to Next Priority Node
- Fallback Policy: Never Fallback

Participating Nodes	Location Dependency	Parent/Child Dependency
• PApp: 1, 2, 3 • PDB: 1, 2, 3 • SApp: 2, 3 • SDB: 2, 3 • QAApp: 3 • QADB: 3	Online On The Same Node Dependent Groups: • PApp with PDB • SApp with SDB • QAApp with QADB Online On Different Nodes Dependent set: [PDB SDB QADB] Priority: PDB > SDB > QADB	• PApp (child) depends on PDB (parent) • SApp (child) depends on SDB (parent) • QAApp (child) depends on QADB (parent)

Related information

Resource group behavior during cluster events

Cross-site LVM mirror configurations for disaster recovery

You can set up disks that are located at two different sites for remote LVM mirroring, using a storage area network (SAN).

A *SAN* is a high-speed network that allows the establishment of direct connections between storage devices and processors (servers) within the distance supported by the Fibre Channel. Thus, two or more distantly separated servers (nodes) that are located at different sites can access the same physical disks, which might be distantly separated by using the common SAN. These remote disks can be combined into volume groups, using C-SPOC.

The logical volumes in a volume group can have up to three mirrors or copies, for example, one mirror at each site. Thus, the information stored on this logical volume might be kept highly available in case of a failure. For example, all nodes at one site (including the disk subsystem at that site) has a failure, the remote mirror at another site will still have the latest information and the operations can be continued on that site.

The primary intent of this feature is to support two-site clusters, where LVM mirroring through a SAN replicates data between the disk subsystem at each site for disaster recovery.

Another advantage of cross-site LVM mirroring is that after a site/disk failure and subsequent site reintegration,PowerHA SystemMirror attempts to synchronize the data from the surviving disks to the joining disks automatically. The synchronization occurs in the background and does not significantly impact the reintegration time.

The following figure illustrates a cross-site LVM mirroring configuration with a SAN:

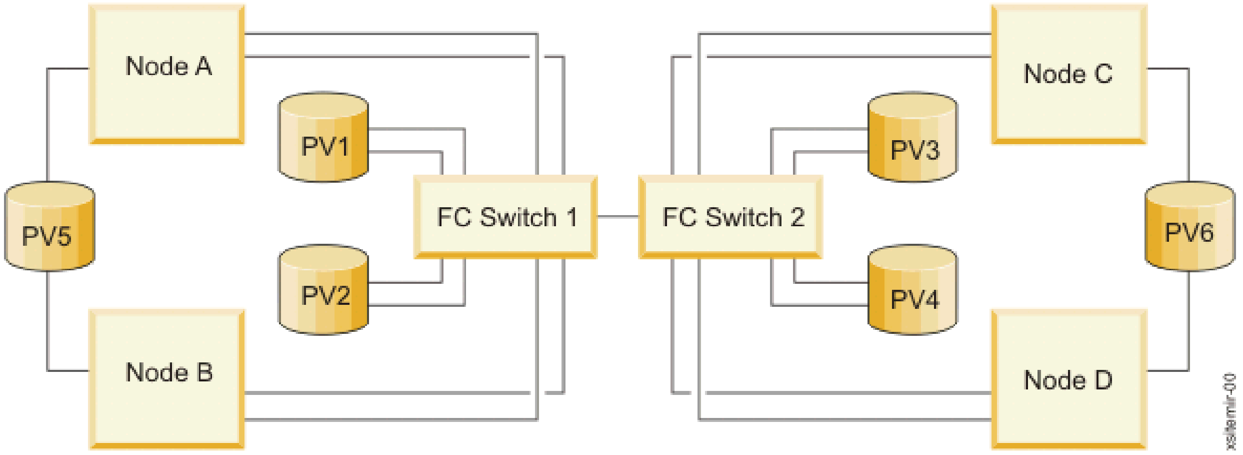


Figure 30. Cross-site LVM mirroring configuration with a SAN

The disks that are connected to at least one node at each of the two sites can be mirrored. In this example, PV4 is seen by Nodes A and B on Site 1 using the Fibre Channel Switch 1 that the Fibre Channel Switch 2, and is also seen on Node C using the Fibre Channel Switch 2. You could have a mirror of PV4 on Site 1. The disks that are connected to the nodes on one site only (PV5 and PV6) cannot be mirrored across sites.

The disk information is replicated from a local site to a remote site. The speed of this data transfer depends on the physical characteristics of the channel, the distance, and LVM mirroring performance.

Cluster configurations with dynamic LPARs

The advanced partitioning features of AIX provide the ability to dynamically allocate system CPU, memory, and I/O slot resources (*dynamic LPAR*).

Using PowerHA SystemMirror in combination with LPARs allows you to:

<Physical Adapter name>_Large_Send

Flag to indicate the large send option for this adapter

<Physical Adapter name>_Large_receive

Flag to indicate the large receive option for this adapter Note: When it is set and if the real adapter supports it, packets received by the real adapter is aggregated before they are passed to the next layer, resulting in better performance.

<Physical Adapter name>_DMA_Errors

The number of incoming packets dropped by the hardware due to the no resource error. Note: This error usually occurs because the receive buffers on the adapter were exhausted. Some adapters may have the size of the receive buffers as a configurable parameter.

<Physical Adapter name>_Speed
Indicates media speed attribute of this adapter

- Perform routine system upgrades through the dynamic allocation of system resources. When used with dynamic LPARs, PowerHA SystemMirror can reduce the amount of downtime for well-planned systems upgrades by automating the transition of your application workload from one logical partition to another, so that the first logical partition might be upgraded without risk to the application.
- Effectively redistribute CPU and memory resources to manage the workload. Combining PowerHA SystemMirror with dynamic LPAR lets you use customized application start and stop scripts to dynamically redistribute CPU and memory resources to logical partitions that are currently executing application workload, to further support application transition within a single frame. This way you maintain the processing power and resources necessary to support your applications, while minimal resources are devoted to upgrading, a less resource intensive task.

Note: Do not have all your cluster nodes configured as LPARs within the same physical server. This configuration could potentially be a significant single point of failure. The following example illustrates a cluster configuration that uses three LPARs:

- LPAR 1 is running a back end database (DB2 UDB)
- LPAR 2 is running WebSphere Application Server (WAS)
- LPAR 3 is running as a backup (standby) for both the DB2 and WAS LPARs. This LPAR contains only minimal CPU and memory resources.

When it is time to move either the DB2 or WAS application to the third LPAR (due to a planned upgrade or a resource failure in these LPARs, for instance), you can use customized application start and stop scripts in PowerHA SystemMirror to automate the dynamic reallocation of CPU and memory from the primary LPAR to the standby LPAR. This operation allows the third LPAR to acquire the CPU and memory resources necessary to meet business performance requirements. When PowerHA SystemMirror moves the resource group containing the application back to its home LPAR, the CPU and memory resources automatically move with it. Note: In general, dynamic LPARs allow dynamic allocation of CPU, memory and I/O slot resources. PowerHA SystemMirror and dynamic LPAR I/O slot resources are not compatible (although you can dynamically allocate I/O slot resources outside of PowerHA SystemMirror cluster).

The following figure illustrates this cluster environment:

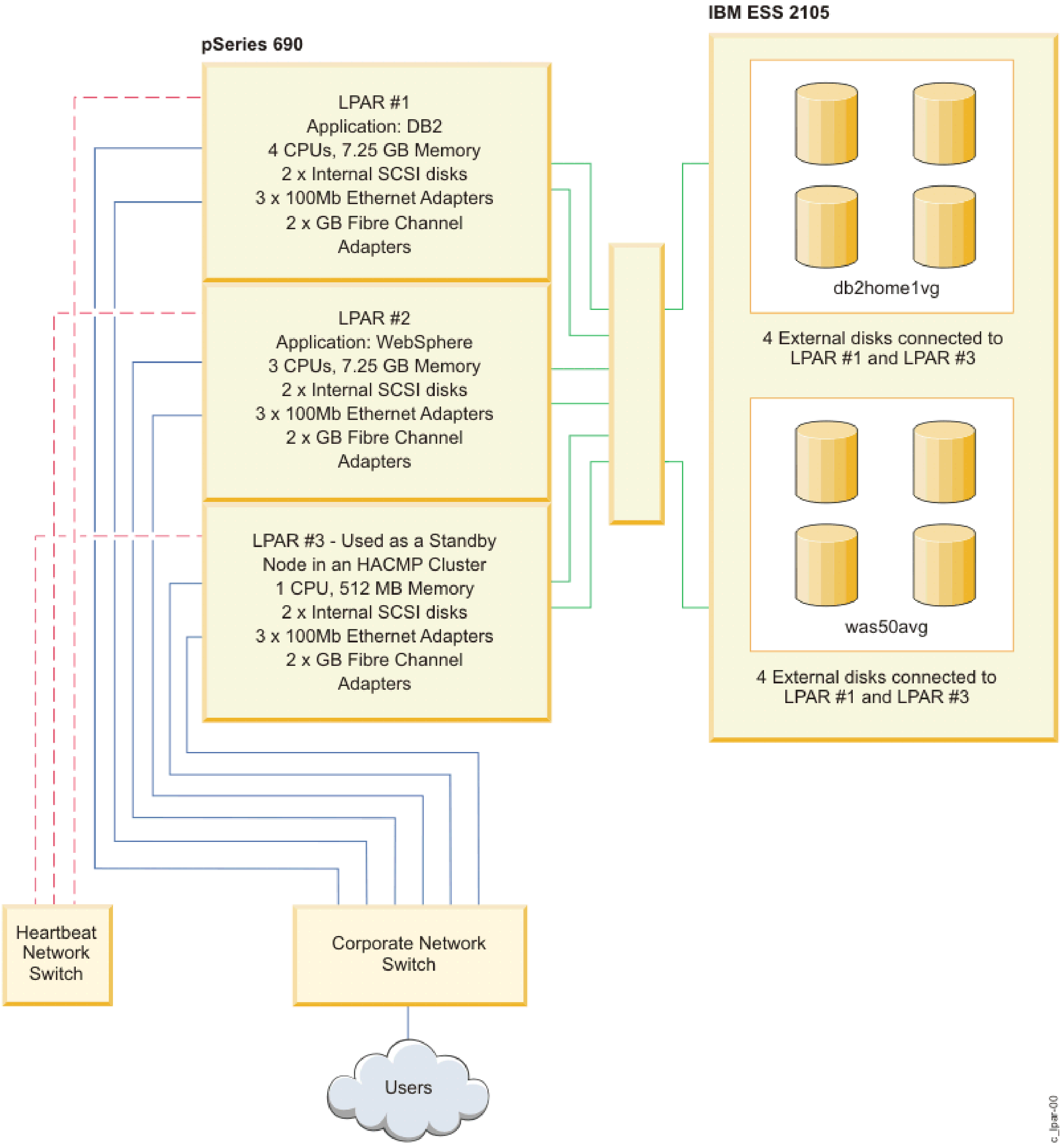

IBM ESS 2105
pSeries 690
LPAR #1
Application: DB2
4 CPUs, 7.25 GB Memory
2 x Internal SCSI disks
3 x 100Mb Ethernet Adapters
2 x GB Fibre Channel Adapters
LPAR #2
Application: WebSphere
3 CPUs, 7.25 GB Memory
2 x Internal SCSI disks
3 x 100Mb Ethernet Adapters
2 x GB Fibre Channel Adapters
LPAR #3 - Used as a Standby Node in an HACMP Cluster
1 CPU, 512 MB Memory
2 x Internal SCSI disks
3 x 100Mb Ethernet Adapters
2 x GB Fibre Channel Adapters
db2home1vg
4 External disks connected to LPAR #1 and LPAR #3
was50avg
4 External disks connected to LPAR #1 and LPAR #3
Heartbeat Network Switch
Corporate Network Switch
Users
c_lpar-00

Figure 31. Cluster with three LPARs

DLPARs and Capacity on Demand

PowerHA SystemMirror can move application resources between LPARs and can perform the necessary dynamic resource adjustments through the Resource Optimized High Availability (ROHA) function. The ROHA uses the features that are available with Power Systems to dynamically manage the following types of hardware resources:

- Capacity on Demand (CoD) functions (including On/Off CoD and Enterprise Pool CoD) manage memory and CPU resources at the frame (CEC) level.
- DLPAR functions manage memory and CPU resources at the logical partition level.

The CoD function can activate preinstalled resources (CPU and memory) that are inactive. ROHA uses the CoD function to activate these resources when the resource requirements for your environment change.

For example, during a takeover process your environment requires extra resources on the standby frame. In this example, the additional resources are dynamically provided and you do not have to add permanent hardware resources on the standby fame.

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. You can use the ROHA function to quickly and easily acquire extra resources to meet peek or unexpected workloads in your environment.

PowerHA SystemMirror configuration process and facilities

These topics provide an overview of the PowerHA SystemMirror cluster configuration process and the administrative tools supplied with the PowerHA SystemMirror software.

Information you provide to PowerHA SystemMirror

Prior to configuring a cluster, make sure the building blocks are planned and configured, and the initial communication path exists for PowerHA SystemMirror to reach each node. This section covers the basic tasks you need to perform to configure a cluster.

Information on physical configuration of a cluster

Physical configuration of a cluster consists of the several planning and configuration tasks.

These tasks include:

- Ensure the TCP/IP network support for the cluster.
- Configure the shared disk devices for the cluster.
- Configure the shared volume groups for the cluster.
- Consider the mission-critical applications for which you are using PowerHA SystemMirror. Also, consider application controller and what type of resource group management is best for each application.
- Examine issues relating to PowerHA SystemMirror clients.
- Ensure physical redundancy by using multiple circuits or power supplies that cannot be interrupted, redundant physical network interface cards, multiple networks to connect nodes and disk mirroring.

Related information

Planning PowerHA SystemMirror

AIX configuration information

Cluster components must be properly configured on the AIX level.

For this task, ensure that:

- Basic communication to cluster nodes exists.
- Volume groups, logical volumes, mirroring and file systems are configured and set up. To ensure logical redundancy, consider different types of resource groups, and plan how you will group your resources in resource groups.

Related information

Installing PowerHA SystemMirror

Information discovered by PowerHA SystemMirror

You can define the basic cluster components in just a few steps. To assist you in the cluster configuration, PowerHA SystemMirror can automatically retrieve the information necessary for configuration from each node.

Note: For easier and faster cluster configuration, you can also use a cluster configuration assistant. For more information, see Two-node cluster configuration assistant.

For the automatic discovery process to work, the following conditions should be met in PowerHA SystemMirror:

- You have previously configured the physical components and performed all the necessary AIX configurations.
- Working communications paths exist to each node. This information will be used to automatically configure the cluster TCP/IP topology when the standard configuration path is used.

Once these tasks are done, PowerHA SystemMirror automatically discovers predefined physical components within the cluster, and selects default behaviors. In addition, PowerHA SystemMirror performs discovery of cluster information if there are any changes made during the configuration process.

Running discovery retrieves current AIX configuration information from all cluster nodes. This information appears in pick lists to help you make accurate selections of existing components.

The PowerHA SystemMirror automatic discovery process is easy, fast, and does not place a "waiting" burden on you as the cluster administrator.

Cluster configuration options: Standard and extended

In this section, the configuration process is significantly simplified. While the details of the configuration process are covered in the *Administration Guide*, this section provides a brief overview of two ways to configure a PowerHA SystemMirror cluster.

Related information

Administering PowerHA SystemMirror

Configuring a PowerHA SystemMirror cluster using the standard configuration path

You can add the basic components of a cluster to the PowerHA SystemMirror configuration database in a few steps. The standard cluster configuration path simplifies and speeds up the configuration process, because PowerHA SystemMirror automatically launches discovery to collect the information and to select default behaviors.

If you use this path:

- Automatic discovery of cluster information runs by default. Before starting the PowerHA SystemMirror configuration process, you need to configure network interfaces or devices in AIX. In PowerHA SystemMirror, you establish initial communication paths to other nodes. Once this is done, PowerHA SystemMirror collects this information and automatically configures the cluster nodes and networks based on physical connectivity. All discovered networks are added to the cluster configuration.
- IP aliasing is used as the default mechanism for binding IP labels or addresses to network interfaces.
- You can configure the most common types of resources. However, customizing of resource group fallover and fallback behavior is limited.

Configuring a PowerHA SystemMirror cluster using the extended configuration path

In order to configure the less common cluster elements, or if connectivity to each of the cluster nodes is not established, you can manually enter the information in a way similar to previous releases of the PowerHA SystemMirror software.

When using the PowerHA SystemMirror extended configuration SMIT paths, if any components are on remote nodes, you must manually initiate the discovery of cluster information. That is, discovery is optional (rather than automatic, as it is when using the standard PowerHA SystemMirror configuration SMIT path).

Using the options under the extended configuration menu, you can add the basic components of a cluster to the PowerHA SystemMirror configuration database, as well as many additional types of resources. Use the extended configuration path to customize the cluster for all the components, policies, and options that are not included in the standard configuration menus.

Cluster security

All communication between nodes is sent through the Cluster Communications daemon, **clcomd**, which runs on each node.

The **clcomd** daemon manages the connection authentication between nodes and any message authentication or encryption configured. The PowerHA SystemMirror Cluster Communications daemon uses the trusted **/etc/cluster/rhosts** file, and removes reliance on an **/.rhosts** file. The daemon provides support for message authentication and encryption.

Installation, configuration, and management tools

PowerHA SystemMirror includes the tools described in these sections for installing, configuring, and managing clusters.

Smart Assists for integrating specific applications with PowerHA SystemMirror

The Smart Assist for a given application examines the configuration on the system to determine the resources PowerHA SystemMirror needs to monitor (Service IP label, volume groups). The Smart Assist then configures one or more resource groups to make applications and their resources highly available.

The Smart Assist takes the following actions:

- Discovers the installation of the application and if necessary the currently configured resources such as service IP address, file systems and volume groups
- Provides a SMIT interface for getting or changing configuration information from the user including a new service IP address
- Defines the application to PowerHA SystemMirror and supplies custom start and stop scripts for it
- Supplies an application monitor for the application
- Configures a resource group to contain:
- Configures resource group temporal and location dependencies, should the application solution require this
- Specifies files that need to be synchronized using the PowerHA SystemMirror File Collections feature
- Modifies previously configured applications as necessary
- Verifies the configuration
- Tests the application's cluster configuration.

Supported applications

PowerHA SystemMirror supplies Smart Assists for the following applications and configuration models:

- DB2
- DB2 - Hot Standby
- DB2 - Mutual Takeover
- WebSphere 6.0
- WebSphere Application Server 6.0
- WebSphere Cluster Transaction Log recovery
- Deployment Manager
- Tivoli® Directory Server
- IBM HTTP Server
- Oracle 10G

General application Smart Assist

The General Application Smart Assist helps users to configure installed applications that do not have their own Smart Assist.

The user supplies some basic information such as:

- Primary node - by default, the local node
- Takeover nodes - by default, all configured nodes except the local node
- Application Name
- Application Start Script
- Application Stop Script
- Service IP label

The General Smart Assist then completes the cluster configuration in much the same way as the Two-Node Cluster Configuration Assistant (but the configuration can have more than two nodes). The user can modify, test, or remove the application when using the General Application Smart Assist.

Smart Assist API

PowerHA SystemMirror includes a *Smart Assist developers guide* so that OEMs can develop Smart Assists to integrate their own applications with PowerHA SystemMirror.

Related information

Developing Smart Assist applications for PowerHA SystemMirror

Starting, stopping, and restarting cluster services

Once you install PowerHA SystemMirror and configure your cluster, you can start cluster services. In PowerHA SystemMirror, your options for starting, stopping and restarting cluster services have been streamlined and improved. PowerHA SystemMirror handles your requests to start and stop cluster services without disrupting your applications, allowing you to have full control.

In PowerHA SystemMirror, you can:

- *Start and restart cluster services*. When you start cluster services, or restart them after a shutdown, PowerHA SystemMirror by default automatically activates the resources according to how you defined them, taking into consideration application dependencies, application start and stop scripts, dynamic attributes and other parameters. That is, PowerHA SystemMirror automatically manages (and activates, if needed) resource groups and applications in them.

You can also start PowerHA SystemMirror cluster services and tell it not to start up any resource groups (and applications) automatically for you. If an application is already running, you no longer need to stop it before starting the cluster services.

Note: PowerHA SystemMirror relies on the application monitor and application startup script to verify whether it needs to start the application for you or the application is already running (PowerHA SystemMirror attempts not to start a second instance of the application). PowerHA SystemMirror relies on the configured application monitors to detect application failures. Application monitors must be configured for PowerHA SystemMirror to detect a running cluster during startup so that it does not start duplicate instances of the application. The alternative approach is to run scripts that ensure duplicate instances of the application controller are not started.

- *Shut down the cluster services*. During a PowerHA SystemMirror shutdown, you might select one of the following three actions for the resource groups:
  - Bring Offline
  - Move to other nodes
  - Place resource groups in an UNMANAGED state

The Cluster Manager remembers the state of all the nodes and responds appropriately when users attempt to restart the nodes.

Related information
Administering PowerHA SystemMirror

SMIT interface

You can use the SMIT panels supplied with the PowerHA SystemMirror software to perform the several tasks.

These tasks include:

- Configure clusters, nodes, networks, resources, and events.
- Capture and restore snapshots of cluster configurations.
- Read log files.
- Diagnose cluster problems.
- Manage a cluster using the C-SPOC utility.
- Perform resource group management tasks.
- Configure Automatic Error Notification.
- Perform dynamic adapter swap.
- Configure cluster performance tuning.
- Configure custom disk methods.

PowerHA SystemMirror system management with C-SPOC

To facilitate management of a cluster, PowerHA SystemMirror provides a way to run commands from one node and then verify and synchronize the changes to all the other nodes. You can use the PowerHA SystemMirror System Management tool, the Cluster Single Point of Control (C-SPOC) to add users, files, and hardware automatically without stopping mission-critical jobs.

You can perform the following tasks using C-SPOC:

- Start or stop PowerHA SystemMirror Services
- PowerHA SystemMirror Communication Interface Management
- PowerHA SystemMirror Resource Group and Application Management
- PowerHA SystemMirror File Collection Management
- PowerHA SystemMirror Log Viewing and Management

- PowerHA SystemMirror Security and Users Management
- PowerHA SystemMirror Logical Volume Management
- PowerHA SystemMirror Concurrent Logical Volume Management
- PowerHA SystemMirror Physical Volume Management
- Open a SMIT Session on a Node.

The C-SPOC utility simplifies maintenance of shared LVM components in clusters of up to 16 nodes. C-SPOC commands provide comparable functions in a cluster environment to the standard AIX commands that work on a single node. By automating repetitive tasks, C-SPOC eliminates a potential source of errors, and speeds up the process.

Without C-SPOC functionality, the system administrator must execute administrative tasks individually on each cluster node. For example, to add a user you usually must perform this task on each cluster node. Using the C-SPOC utility, a command executed on one node is also executed on other cluster nodes. Thus C-SPOC minimizes administrative overhead and reduces the possibility of inconsistent node states. Using C-SPOC, you issue a C-SPOC command once on a single node, and the user is added to all specified cluster nodes.

C-SPOC also makes managing logical volume components and controlling cluster services more efficient. You can use the C-SPOC utility to start or stop cluster services on nodes from a single node.

The following figure illustrates a two-node configuration and the interaction of commands, scripts, and nodes when starting cluster services from a single cluster node. Note the prefix *cl_* begins all C-SPOC commands.

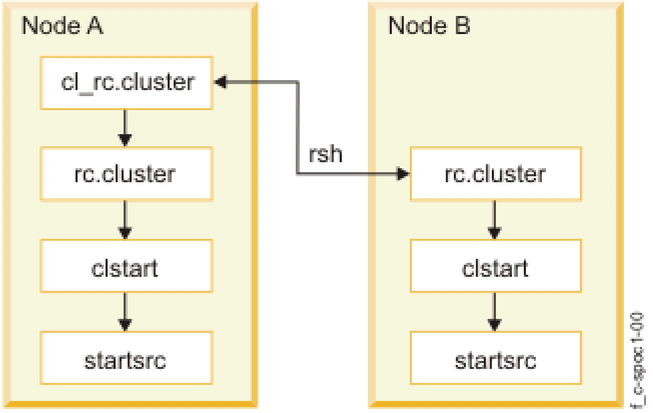


Figure 32. Flow of commands used at cluster startup by C-SPOC utility

C-SPOC provides this functionality through its own set of cluster administration commands, accessible through SMIT menus and panels. To use C-SPOC, select the *Cluster System Management* option from the PowerHA SystemMirror SMIT menu.

Cluster snapshot utility

The Cluster Snapshot utility allows you to save cluster configurations you would like to restore later.

You also can save additional system and cluster information that can be useful for diagnosing system or cluster configuration problems. You can create your own custom snapshot methods to store additional information about your cluster.

A cluster snapshot allows you to skip saving log files in the snapshot. Cluster snapshots are used for recording the cluster configuration information, whereas cluster logs only record the operation of the cluster and not the configuration information. By default, PowerHA SystemMirror no longer collects cluster log files when you create the cluster snapshot, although you can still specify collecting the logs in SMIT. Skipping the logs collection speeds up the running time of the snapshot utility and reduces the size of the snapshot.

Cross-cluster verification utility

In PowerHA SystemMirror Version 7.2.4, or later, you compare and check configuration of two different clusters by using the cross-cluster verification (CCV) utility.

The cluster verification utility **`clverify`** checks the PowerHA SystemMirror cluster configuration for accuracy and consistency. For example, if a service IP address is included in a resources group, the cluster verification utility, checks whether all nodes that are included in the resource group are connected to the network in which the service IP address is managed. The cluster verification utility checks the configuration within the domain of the PowerHA SystemMirror nodes that are defined in the cluster.

Even though the cluster verification utility checks the configuration, some errors might still occur that can lead to application downtime after the cluster is deployed in the production environment. To ensure high availability, you must configure a test cluster that mirrors the configuration of a production cluster. You must test and verify the test cluster before updating the production cluster.

In addition, multiple similar clusters might be deployed across applications in an enterprise. These clusters might be copied from an initial configuration and then modified to meet specific application or topology requirement. In such cases, the cluster verification utility checks the configuration only within the domain of the individual cluster. There is no facility available to verify or compare the configuration of one cluster to another cluster. Any comparison or verification must be performed manually, which can result in errors.

The CCV utility compares specific attributes of two different cluster configurations. It compares data that is collected from different clusters, cluster snapshots, active configuration directory, or the default configuration directory of a local cluster.

Related information

Verifying two different clusters by using the CCV utility

cl_ccv_collector command

cl_ccv_comparator command

clccv command

clmgr command

Resource group management utility

The resource group management utility, clRGmove, provides a means for managing resource groups in the cluster, and enhances failure recovery capabilities of PowerHA SystemMirror.

It allows you to move any type of resource group (along with its resources - IP addresses, applications, and disks) online, offline or to another node, without stopping cluster services. Resource group management helps you to manage your cluster more effectively, giving you better use of your cluster hardware resources.

Resource group management also allows you to perform selective maintenance without rebooting the cluster or disturbing operational nodes. For instance, you can use this utility to free the node of any resource groups to perform system maintenance on a particular cluster node.

Using the resource group management utility does not affect other resource groups currently owned by a node. The current node releases it, and the destination node acquires it just as it would during a node fallover. (If you have location dependencies configured between resource groups, PowerHA SystemMirror verifies and ensures that they are honored).

Use resource group management to complete the following tasks:

- Temporarily move a nonconcurrent resource group from one node to another (and from one site to another) in a working cluster.
- Bring a resource group online or offline on one or all nodes in the cluster.

When you move a group, it stays on the node to which it was moved, until you move it again. If you move a group that has Fallback to Highest Priority Node fallback policy, the group falls back or returns to its "new" temporary highest priority node (in cases when PowerHA SystemMirror has to recover it on other nodes during subsequent cluster events).

If you want to move the group again, PowerHA SystemMirror intelligently informs you (in the pick lists with destination nodes) if it finds that a node with a higher priority exists that can host a group. You can always choose to move the group to that node.

Customized event processing

You can define multiple pre-events and post-events to tailor your event processing for your site's unique needs.

Related information

Administering PowerHA SystemMirror

PowerHA SystemMirror file collection management

Like volume groups, certain files located on each cluster node need to be kept in sync in order for PowerHA SystemMirror (and other applications) to behave correctly. Such files include event scripts, application scripts, and some AIX and PowerHA SystemMirror configuration files. PowerHA SystemMirror File Collection management provides an easy way to request that a list of files be kept in sync across the cluster.

Using PowerHA SystemMirror file collection, you do not have to manually copy an updated file to every cluster node, verify that the file is properly copied, and confirm that each node has the same version of it.

Also, if one or more of these files is inadvertently deleted or damaged on one or more cluster nodes, it can take time and effort to determine the problem. Using PowerHA SystemMirror file collection, this scenario is mitigated. PowerHA SystemMirror detects when a file in a file collection is deleted or if the file size is changed to zero, and logs a message to inform the administrator. Two predefined PowerHA SystemMirror file collections are installed by default:

- **Configuration_Files**. A container for essential system files, such as **/etc/hosts** and **/etc/services**.
- **PowerHA SystemMirror_Files**. A container for all the user-configurable files in the PowerHA SystemMirror configuration. This is a special file collection that the underlying file collection propagation utility uses to reference all the user-configurable files in the PowerHA SystemMirror configuration database (ODM) classes.

Related information

Administering PowerHA SystemMirror

Installing PowerHA SystemMirror

Monitoring tools

PowerHA SystemMirror supplies several different tools for monitoring.

Many of the utilities described here use the **clhosts** file to enable communication among PowerHA SystemMirror cluster nodes.

Related concepts

Understanding the clhosts file
Many of the monitors described in this topic, including Clinfo, HAView, and clstat utility rely on the use of a `clhosts` file. The `clhosts` file contains IP address information that helps enable communications among PowerHA SystemMirror cluster nodes. The `clhosts` file resides on all PowerHA SystemMirror cluster servers and clients.

Related information

Administering PowerHA SystemMirror

Cluster manager

The Cluster Manager provides SNMP information and traps for SNMP clients.

It gathers cluster information relative to cluster state changes of nodes and interfaces. Cluster information can be retrieved using SNMP commands or by SNMP-based client programs.

Related concepts

Cluster manager and SNMP monitoring programs
A PowerHA SystemMirror cluster is dynamic and can undergo various transitions in its state over time.

Cluster information program

The Cluster Information Program (Clinfo) gathers cluster information from SNMP and enables clients communicating with this program to be aware of changes in a cluster state.

Related concepts

Cluster information program
The Cluster Information Program (Clinfo), the **clinfo** daemon, is an SNMP-based monitor.

Application monitoring

Application monitoring enables you to configure multiple monitors for an application controller to monitor specific applications and processes; and define action to take upon detection of an unexpected termination of a process or other application failures.

Related concepts

Eliminating applications as a single point of failure
The primary reason to create PowerHA SystemMirror clusters is to provide a highly available environment for mission-critical applications.

clam_nfsv4 application monitor

The clam_nfsv4 application monitor is automatically added by PowerHA SystemMirror Version 7.1 when Network File System version 4 (NFSV4) resources are specified within a resource group.

The clam_nfsv4 application monitor script runs once per minute by default. The monitors restart count is zero. The clam_nfsv4 application monitor has a failure action of fallover. When a fallover occurs, any error logged by the monitor results in the NFSV4 application server moving to an offline state, and the resource groups on the application server move to the next highest priority node.

You cannot remove the clam_nfsv4 application monitor, but you can disable it by changing the failure action to notify.

The clam_nfsv4 application monitor verifies that nodes with cross-mount configurations are correctly mounted. The monitor also verifies that the NFS server can respond to client request made through cross-mount configurations.

The clam_nfsv4 application monitor checks that the following subsystem functions are running:

- nfsrgyd
- nfsd
- rpc.mountd
- rpc.statd
- rpc.lockd

If any of the subsystem functions are not running, the clam_nfsv4 application monitor identifies the NFSV4 application server as offline.

You must register the NFS nodes that contain resource groups. You must also configure NFS to correctly export resource groups.

Related information

Using NFS with PowerHA SystemMirror

NFS cross-mounting in PowerHA SystemMirror

Troubleshooting clam_nfsv4 application monitor

Show cluster applications SMIT option

The Show Cluster Applications SMIT option provides an application-centric view of the cluster configuration.

This utility displays existing interfaces and information in an "application down" type of view.

Cluster status utility (clstat)

The Cluster Status utility, **/usr/es/sbin/cluster/clstat**, monitors cluster status. The utility reports the status of key cluster components: the cluster itself, the nodes in the cluster, the network interfaces connected to the nodes, and the resource groups on each node.

It reports whether the cluster is up, down, or unstable. It also reports whether a node is up, down, joining, leaving, or reconfiguring, and the number of nodes in the cluster. The **clstat** utility provides ASCII, Motif, X Window System, and HTML interfaces. You can run **clstat** from ASCII SMIT.

For the cluster as a whole, **clstat** indicates the cluster state and the number of cluster nodes. For each node, **clstat** displays the IP label and address of each service network interface attached to the node, and whether that interface is up or down. **clstat** also displays resource group state.

You can view cluster status information in ASCII or X Window System display mode or through a web browser.

If an IP version 6 loopback address is specified in the `/usr/es/sbin/cluster/etc/clhosts` file and IP version 6 addresses are not used in the cluster, the cluster status information that is displayed by **clstat** might continuously change. If the state of **clstat** repeatedly changes from UP to DOWN and DOWN to UP, remove the `::1` entry from the `/usr/es/sbin/cluster/etc/clhosts` file.

If you want to keep the `::1` entry in the `/usr/es/sbin/cluster/etc/clhosts` file, complete the following steps:

1. Verify that the `::1` entry is in the `/usr/es/sbin/cluster/etc/clhosts` file.
2. Configure Simple Network Management Protocol (SNMP) version 3 to support IP version 6 address by adding the following line to the `/etc/snmpdv3.conf` file:

   ```
   COMMUNITY public public noAuthNoPriv :: 0 -
   ```

3. Restart the **snmpd** daemon, the **`clinfoES`** daemon, and any dependent daemons.

Note: The **clstat** utility uses the Clinfo API to retrieve information about the cluster. Therefore, ensure Clinfo is running on the client system to view the **clstat** display.

Application availability analysis tool

The Application Availability Analysis tool measures uptime statistics for applications with application controllers defined to PowerHA SystemMirror.

The PowerHA SystemMirror software collects, timestamps, and logs extensive information about the applications you choose to monitor with this tool. Using SMIT, you can select a time period and the tool displays uptime and downtime statistics for a given application during that period.

Persistent node IP labels

A *persistent node IP label* is a useful administrative tool that lets you contact a node even if the PowerHA SystemMirror cluster services are down on that node.

When you define persistent node IP labels PowerHA SystemMirror attempts to put an IP address on the node. Assigning a persistent node IP label to a network on a node allows you to have a node-bound IP address on a cluster network that you can use for administrative purposes to access a specific node in the cluster. A persistent node IP label is an IP alias that can be assigned to a specific node on a cluster network and that:

- Always stays on the same node (is *node-bound*)
- Co-exists on a network interface card that already has a service IP label defined
- Does *not* require installing an additional physical network interface card on that node
- Is *not* part of any *resource group* .

There can be one persistent node IP label per network per node.

PowerHA SystemMirror verification and synchronization

The PowerHA SystemMirror verification and synchronization process verifies that specific PowerHA SystemMirror modifications to AIX system files are correct, the cluster and its resources are configured correctly, security (if set up) is configured correctly, all nodes agree on the cluster topology, network configuration, and the ownership and takeover of PowerHA SystemMirror resources.

Verification also indicates whether custom cluster snapshot methods exist and whether they are executable on each cluster node.

Whenever you have configured, reconfigured, or updated a cluster, you should then run the cluster verification procedure. If the verification succeeds, the configuration is automatically synchronized. Synchronization takes effect immediately on an active cluster.

The verification utility keeps a detailed record of the information in the PowerHA SystemMirror configuration database on each of the nodes after it runs. Subdirectories for each node contain information for the last successful verification (pass), the next-to-last successful verification (pass.prev), and the last unsuccessful verification (fail).

Messages output by the utility indicate where the error occurred (for example, the node, device, command, and so forth).

Verification with automatic cluster configuration monitoring

PowerHA SystemMirror provides automatic cluster configuration monitoring. By default, PowerHA SystemMirror automatically runs *verification* on the node that is first in alphabetical order once every 24 hours at midnight.

The cluster administrator is notified if the cluster configuration has become invalid. When cluster verification completes on the selected cluster node, this node notifies the other cluster nodes. Every node stores the information about the date, time, which node performed the verification, and the results of the verification in the **/var/hacmp/log/clutils.log** file. If the selected node becomes unavailable or cannot complete cluster verification, you can detect this by the lack of a report in the /**var/hacmp/log/clutils.log** file. If cluster verification completes and detects some configuration errors, you are notified about the potential problems:

- The exit status of verification is published across the cluster along with the information about cluster verification process completion.
- Broadcast messages are sent across the cluster and displayed on *stdout*. These messages inform you about detected configuration errors.

Verification with corrective actions

Cluster verification consists of a series of checks performed against various user-configured PowerHA SystemMirror server components. Each check attempts to detect either a cluster consistency issue or a configuration error.

Some error conditions result when information important to the operation of PowerHA SystemMirror, but not part of the PowerHA SystemMirror software itself, is not propagated properly to all cluster nodes.

By default, verification runs with the automatic corrective actions mode enabled for both the Standard and Extended configuration. This is the recommended mode for running verification. If necessary, the automatic corrective actions mode can be disabled for the Extended configuration. However, note that running verification in automatic corrective action mode enables you to automate many configuration tasks, such as creating a client-based **clhosts** file, which is used by many of the monitors

When verification detects any of the following conditions, you can authorize a corrective action before error checking continues:

- PowerHA SystemMirror shared volume group time stamps do not match on all nodes.
- The **/etc/hosts** file on a node does not contain all IP labels and IP address managed by PowerHA SystemMirror.
- A file system is not created on a node that is part of the resource group, although disks are available.
- Disks are available, but a volume group has not been imported to a node.
- Required **/etc/services** entries are missing on a node.
- Required PowerHA SystemMirror **snmpd** entries are missing on a node.

If an error found during verification triggers any corrective actions, then the utility runs all checks again after it finishes the first pass. If the same check fails again and the original problem is an error, the error is logged and verification fails. If the original condition is a warning, verification succeeds.

Related concepts

Understanding the clhosts file
Many of the monitors described in this topic, including Clinfo, HAView, and clstat utility rely on the use of a `clhosts` file. The `clhosts` file contains IP address information that helps enable communications among PowerHA SystemMirror cluster nodes. The `clhosts` file resides on all PowerHA SystemMirror cluster servers and clients.

Custom verification methods

Through SMIT you also can add, change, or remove custom-defined verification methods that perform specific checks on your cluster configuration. You can perform verification from the command line or through the SMIT interface to issue a customized remote notification method in response to a cluster event.

Understanding the clhosts file

Many of the monitors described in this topic, including Clinfo, HAView, and clstat utility rely on the use of a `clhosts` file. The `clhosts` file contains IP address information that helps enable communications among PowerHA SystemMirror cluster nodes. The `clhosts` file resides on all PowerHA SystemMirror cluster servers and clients.

There are differences for the `clhosts` file, depending on where the file resides, as summarized in the following table:

Table 5. clhosts file

clhosts file	Description
server-based file	This file is in the `/usr/es/sbin/cluster/etc/` directory on all PowerHA SystemMirror server nodes. During the installation of PowerHA SystemMirror, the default IP version 4 (IPv4) loopback address of 127.0.0.1 is automatically added to the file. If there are any IP version 6 (IPv6) addresses configured at the time of the installation, the default IPv6 loopback address of ::1 is also added. The name loopback and the alias local host that this IP address usually defines are not required.
client-based file	This file is in the `/usr/es/sbin/cluster/etc/` directory on all PowerHA SystemMirror client nodes. When you run verification with the automatic corrections function, this file is automatically generated on the server and is named **clhosts.client**. You must copy this file to the client nodes manually and rename it to `clhosts`. This file contains all known IP addresses and must never contain 127.0.0.1, ::1, loopback, or local host addresses.

When a monitor daemon starts up, it reads in the local `/usr/es/sbin/cluster/etc/clhosts` file to determine which nodes are available for communication as follows:

- For daemons running on a PowerHA SystemMirror *server node*, the local server-based `clhosts` file only requires the loopback address `(127.0.0.1 and ::1)`, that is automatically added to the server-based `clhosts` file when the server portion of PowerHA SystemMirror is installed.
- For daemons running on a PowerHA SystemMirror *client node*, the local client-based `clhosts` file should contain a list of the IP addresses for the PowerHA SystemMirror server nodes. In this way, if a particular PowerHA SystemMirror server node is unavailable (for example, powered off), then the daemon on the client node still can communicate with other PowerHA SystemMirror server nodes.

The PowerHA SystemMirror verification utility assists in populating the client-based `clhosts` file in the following manner by finding all available PowerHA SystemMirror server nodes, creating a `/usr/es/sbin/cluster/etc/clhosts.client` file on the server nodes, and populating the file with the IP addresses of those PowerHA SystemMirror server nodes.

After you finish verifying and synchronizing PowerHA SystemMirror on your cluster, you must manually copy this **clhosts.client** file to each client node as `/usr/es/sbin/cluster/etc/clhosts` (rename it by removing the **.client** extension).

Related concepts

PowerHA SystemMirror verification and synchronization
The PowerHA SystemMirror verification and synchronization process verifies that specific PowerHA SystemMirror modifications to AIX system files are correct, the cluster and its resources are configured correctly, security (if set up) is configured correctly, all nodes agree on the cluster topology, network configuration, and the ownership and takeover of PowerHA SystemMirror resources.

Troubleshooting tools

Typically, a functioning PowerHA SystemMirror cluster requires minimal intervention. If a problem occurs, however, diagnostic and recovery skills are essential. Thus, troubleshooting requires that you identify the problem quickly and apply your understanding of the PowerHA SystemMirror software to restore the cluster to full operation.

Related concepts

Cluster test tool
The Cluster Test Tool is a utility that lets you test a PowerHA SystemMirror cluster configuration to evaluate how a cluster behaves under a set of specified circumstances, such as when a node becomes inaccessible, a network becomes inaccessible, a resource group moves from one node to another, and so forth.

Log files

The PowerHA SystemMirror software writes the messages it generates to the system console and to several log files. Because each log file contains a different level of detail, system administrators can focus on different aspects of PowerHA SystemMirror processing by viewing different log files.

The main log files include:

- The **/var/hacmp/adm/cluster.log** file tracks cluster events.
- The **/var/hacmp/log/hacmp.out** file records the output generated by configuration scripts as they execute. Event summaries appear after the verbose output for events initiated by the Cluster Manager, making it easier to scan the **hacmp.out** file for important information. In addition, event summaries provide HTML links to the corresponding events within the **hacmp.out** file.
- The **/var/hacmp/adm/history/cluster.mmddyyyy** log file logs the daily cluster history.
- The /**var/hacmp/clverify/clverify.log** file contains the verbose messages output during verification. Cluster verification consists of a series of checks performed against various PowerHA SystemMirror configurations. Each check attempts to detect either a cluster consistency issue or an error. The messages output by the verification utility indicate where the error occurred (for example, the node, device, command, and so forth).

PowerHA SystemMirror allows you to view, redirect, save and change parameters of the log files, so you can tailor them to your particular needs.

Related information

Administering PowerHA SystemMirror

Resetting PowerHA SystemMirror tunable values

While configuring and testing a cluster, you might change a value for one of the PowerHA SystemMirror tunable values that affects the cluster performance. Or, you might want to reset tunable values to their default settings without changing any other aspects of the configuration.

A third-party cluster administrator or a consultant might be asked to take over the administration of a cluster that they did not configure and might need to reset the tunable values to their defaults. You can reset cluster tunable values using the SMIT interface. PowerHA SystemMirror takes a cluster snapshot, prior to resetting. After the values have been reset to defaults, if you want to return to customized cluster settings, you can apply the cluster snapshot. Resetting the cluster tunable values resets information in the cluster configuration database. The information that is reset or removed comprises the following categories:

- Information supplied by the users (for example, pre- and post-event scripts and network parameters, such as netmasks). Note that resetting cluster tunable values does not remove the pre- and post-event scripts that you already have configured. However, if you reset the tunable values, PowerHA SystemMirror's knowledge of pre- and post-event scripts is removed from the configuration, and these scripts are no longer used by PowerHA SystemMirror to manage resources in your cluster. You can reconfigure PowerHA SystemMirror to use these scripts again, if needed.
- Information automatically generated by PowerHA SystemMirror during configuration and synchronization. This includes node and network IDs, and information discovered from the operating system, such as netmasks. Typically, users cannot see generated information.

Related information

Administering PowerHA SystemMirror

Installing PowerHA SystemMirror

Cluster status information file

When you use the PowerHA SystemMirror Cluster Snapshot utility to save a record of a cluster configuration (as seen from each cluster node), you optionally cause the utility to run many standard AIX commands and PowerHA SystemMirror commands to obtain status information about the cluster. This information is stored in a file, identified by the **.info** extension, in the snapshots directory.

The snapshots directory is defined by the value of the SNAPSHOTPATH environment variable. By default, the cluster snapshot utility includes the output from the commands, such as **cllssif**, **cllsnw**, **df**, **ls**, and **netstat**. You can create custom snapshot methods to specify additional information you would like stored in the **.info** file.

A cluster snapshot allows you to skip saving log files in the snapshot. Cluster snapshots are used for recording the cluster configuration information, whereas cluster logs only record the operation of the cluster and *not* the configuration information. By default, PowerHA SystemMirror no longer collects cluster log files when you create the cluster snapshot, although you can still specify collecting the logs in SMIT. Skipping the logs collection reduces the size of the snapshot and speeds up running the snapshot utility. The size of the cluster snapshot depends on the configuration. For instance, a basic two-node configuration requires roughly 40KB.

Automatic error notification

You can use the AIX Error Notification facility to detect events not specifically monitored by the PowerHA SystemMirror software. For example, a disk adapter failure occurs, and you can specify a response to take place if the event occurs.

Normally, you define error notification methods manually, one by one. PowerHA SystemMirror provides a set of pre-specified notification methods for important errors that you can automatically "turn on" in one step through the SMIT interface, saving considerable time and effort by not having to define each notification method manually.

Custom remote notification

You can define a notification method through the SMIT interface to issue a customized notification method in response to a cluster event. You can also send text messaging notification to any address including a cell phone, or mail to an email address.

After configuring a remote notification method, you can send a test message to confirm that the configuration is correct.

You can configure any number of notification methods, for different events and with different text 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.

Event preambles and summaries

Details of cluster events are recorded in the **hacmp.out** file. The verbose output of this file contains many lines of event information; you see a concise summary at the end of each event's details. For a quick and efficient check of what has happened in the cluster lately, you can view a compilation of only the event summary portions of current and previous **hacmp.out** files, by using the **Display Event Summaries** panel in SMIT.

You can also select to save the compiled event summaries to a file of your choice. Optionally, event summaries provide HTML links to the corresponding events in the **hacmp.out** file. The Cluster Manager also prints out a preamble that tells you which resource groups are enqueued for processing for each event; you can see the processing order that will be followed.

Related information

Troubleshooting PowerHA SystemMirror

Cluster test tool

The Cluster Test Tool is a utility that lets you test a PowerHA SystemMirror cluster configuration to evaluate how a cluster behaves under a set of specified circumstances, such as when a node becomes inaccessible, a network becomes inaccessible, a resource group moves from one node to another, and so forth.

You can start the test, let it run unattended, and return later to evaluate the results of your testing.

If you want to run an automated suite of basic cluster tests for topology and resource group management, you can run the automated test suite from SMIT. If you are an experienced PowerHA SystemMirror administrator and want to tailor cluster testing to your environment, you can also create custom tests that can be run from SMIT.

It is recommended to run the tool after you initially configure PowerHA SystemMirror and before you put your cluster into a production environment; after you make cluster configuration changes while the cluster is out of service; or at regular intervals even though the cluster appears to be functioning well.

Note: The Cluster Test Tool is not fully supported by the latest PowerHA releases.

Ansible integration with PowerHA SystemMirror

The Ansible® collections for PowerHA SystemMirror filesets are located in the `/usr/es/sbin/cluster/samples` directory in the form of a tar file (compressed format). You can extract the Ansible collections from the tar file and integrate the ansible collections with the centralised node after installing PowerHA SystemMirror 7.2.8 or later.

Note: A centralized node is the server that stores and manages the Ansible playbook and roles. The centralized node simplifies the Ansible management by providing a single location for storing and updating the Ansible content.

To configure the PowerHA SystemMirror cluster and resources, you can run the Ansible playbooks from the centralized node.

Note: A playbook is a set of instructions that defines how to automate a task or a process.

The Ansible framework helps you with the following.

1. Simple installation.
2. Ensure meeting the prerequisites before creating a cluster. For example, updating the `/etc/hosts` and `/etc/cluster/rhosts` files.
3. Configure various resources like resource group, volume group, file system, service IP network, interfaces, applications, WPAR, NFSv2v3, and NFSv4.

Prerequisites

Before you get started with the Ansible integration, ensure that your system meets the following requirements.

1. Ensure that the centralized node is running on the IBM AIX version 7.2 or 7.3, or later for better compatibility with the IBM AIX file sets.
2. Install the PowerHA SystemMirror version 7.2.8, or later, on the centralized node.
3. Install Python on both the centralized and the remote servers.

   Note: It is mandatory to install the Python. Python version 3.7.9 is compatible with AIX 7.2 and AIX 7.3. If you are using higher version of AIX, check python compatible version and use it. For more information see, Ansible Releases and maintenance.
4. Install the Ansible filesets on the centralized node. To download the ansible software package, see https://www.ibm.com/support/pages/aix-toolbox-linux-applications-downloads-alpha.

   Verify that the Ansible is installed on the centralized node by running the following command.

   ```
   ansible --version
   ```

   The output displays the Ansible is installed on your system.
5. Establish password-less Secure socket shell (SSH) connections between the centralized node and the host nodes to communicate with the remote servers without requiring a password. You can check password-less connectivity by running the command: `ssh <remote node ip>`.
6. Ensure Remote Shell Communication (RSH) is enabled on all nodes. Check the `/etc/inetd.conf` file to verify they the RSH is enabled on all nodes by uncommenting RSH-related fields and then update the `/.rhosts` file with double plus sign (++). For more details, see .rhosts File Format for TCP/IP
7. Extract the Ansible filesets from the tarball provided in the PowerHA SystemMirror installable files. The tarball is located in the centralized node in the `/usr/es/sbin/cluster/samples/ansible/` path.

   To the extract the Ansible filesets, complete the following steps:

   a. Login as a root or as a privileged user in the centralized node.
   b. Copy the tarball from path "`/usr/es/sbin/cluster/samples/ansible/`" to path "`/`".
   c. Switch to the home (#cd) or to the root (home) directory where the tarball needs to be extracted.

d. Extract the following tarball.

```
# gunzip /usr/es/sbin/cluster/samples/ansible/ansible_powerha_tarball.tar.gz
```

e. Extract the contents of the tarball by running the following command.

```
# tar -xvf /usr/es/sbin/cluster/samples/ansible/ansible_powerha_tarball.tar
```

After extraction, two new folders with the following contents are created.

power_ha folder
: This folder contains Ansible roles, playbooks, and a template file. To access power_ha folder, enter the following path:

```
/.Ansible/collections/ansible_collections/ibm/power_ha
```

Ansible folder
: This folder contains the `ansible.cfg` host file, template file, and external variable file. To access Ansible folder, enter the following path:

```
/etc/ansible
```

8. To locate and run the Ansible binary file, update the path variable. The Ansible binary files are generally installed in the `/opt/freeware/bin` directory. To update the path variable, run the following command.

```
# export PATH=$PATH:/opt/freeware/bin
```

9. To verify the installed version of ansible, run the following command.

```
# ansible --version
```

Ansible driven deployment

To install and configure various resources by using the Ansible playbook, you must complete the following steps.

1. Go to the `/etc/ansible` directory and open the host file with any available editors. The application has been tested with `vi` editor.

```
# vi /etc/ansible/hosts
Example /etc/ansible/hosts:
```

Update the file with the IP addresses of the target nodes, appropriate username, and python path for the nodes.

```
######### POWERHA Ansible STARTS ############
[powerha_remote_servers]
# provide PowerHA servers ip addresses
# Example:
x.x.x.x
x.x.x.x
[powerha_remote_servers:vars]
ansible_user='root'
ansible_python_interpreter='/usr/bin/python3'
######### POWERHA Ansible ENDS ############
```

Note: Step 1 is essential for the Ansible to identify and connect to the remote servers.

2. Open the **`external_vars.yml`** file located in the `/etc/ansible` directory. Update the file with the values of **NODE_DETAILS** and **NODES** and **Test case** variable details specific to your environment. This file customizes variables that are used in your Ansible playbooks. To update variables in `vi` editor, run the following command.

```
# vi /etc/ansible/external_var.yml
```

3. Once steps 1 and 2 are completed and the necessary variables are updated, you can now run Ansible playbook. Go to the path `/.ansible/collections/ansible_collections/ibm/power_ha`, and run the Ansible commands along with the necessary tags.

   Example: To change the working directory and run a playbook with specific tags, use the following command.

   ```
   # cd /.ansible/collections/ansible_collections/ibm/power_ha/playbooks # ansible-playbook
   demo_playbook.yml --tags <tag1>,<tag2>.
   ```

   This step selectively runs tasks or roles within the playbook based on the provided tags.

Ansible Playbooks

You can use the following commands from the playbooks to install, uninstall, configure, and unconfigure PowerHA SystemMirror cluster and resources.

1. To create map hosts, enter the following command:

   ```
    ansible-playbook demo_map_hosts.yml
   ```

2. To create a standard cluster, enter the following command:

   ```
   ansible-playbook demo_cluster.yml --tags standard
   ```

3. To create a linked cluster, enter the following command:

   ```
   ansible-playbook demo_cluster.yml --tags linked
   ```

4. To create a stretched cluster, enter the following command:

   ```
   ansible-playbook demo_cluster.yml --tags stretched
   ```

5. To remove a cluster, enter the following command:

   ```
   ansible-playbook demo_cluster.yml --tags delete
   ```

6. To create a resource group, enter the following command:

   ```
   ansible-playbook demo_resource_group.yml –tags create
   ```

7. To remove a resource group, enter the following command:

   ```
   ansible-playbook demo_resource_group.yml –tags delete
   ```

8. To add a network, enter the following command:

   ```
   ansible-playbook demo_network.yml –tags create
   ```

9. To remove a network, enter the following command:

   ```
   ansible-playbook demo_network.yml –tags delete
   ```

10. To add an interface, enter the following command:

    ```
    ansible-playbook demo_interface.yml –tags create
    ```

11. To remove an interface, enter the following command:

    ```
    ansible-playbook demo_interface.yml –tags delete
    ```

12. To add a file system, enter the following command:

    ```
    ansible-playbook demo_file_system.yml –tags create
    ```

13. To remove a file system, enter the following command:

```
ansible-playbook demo_file_system.yml –tags delete
```

14. To add an application, enter the following command:

```
ansible-playbook demo_applications.yml –tags create
```

15. To remove an application, enter the following command:

```
ansible-playbook demo_applications.yml –tags delete
```

16. To add a volume group, enter the following command:

```
ansible-playbook demo_volume_groups.yml –tags create
```

17. To remove a volume group, enter the following command:

```
ansible-playbook demo_volume_groups.yml –tags delete
```

18. To add a Service IP address, enter the following command:

```
ansible-playbook demo_service_ip.yml –tags create
```

19. To remove a Service IP address, enter the following command:

```
ansible-playbook demo_service_ip.yml –tags delete
```

20. To start Cluster Services, enter the following command:

```
ansible-playbook demo_start_stop_services.yml –tags start
```

21. To stop Cluster Services, enter the following command:

```
ansible-playbook demo_start_stop_services.yml –tags stop
```

22. To add AIX Workload Partitions (WPAR), enter the following command:

```
ansible-playbook demo_wpar.yml –tags create
```

23. To remove WPAR, enter the following command:

```
ansible-playbook demo_wpar.yml –tags delete
```

24. To add Network File System v2v3 (NFS), enter the following command:

```
ansible-playbook demo_nfsv2v3.yml –tags create
```

25. To remove NFSv2v3, enter the following command:

```
ansible-playbook demo_nfsv2v3.yml –tags delete
```

26. To add NFSv4, enter the following command:

```
ansible-playbook demo_nfsv4.yml –tags create
```

27. To remove NFSv4, enter the following command:

```
ansible-playbook demo_nfsv4.yml –tags delete
```

28. To install PowerHA SystemMirror, enter the following command:

```
ansible-playbook demo_PowerHA.yml –tags install
```

29. To uninstall PowerHA SystemMirror, enter the following command:

```
ansible-playbook demo_PowerHA.yml –tags uninstall
```

30. To install the Cluster health, enter the following command:

    ```
    ansible-playbook demo_cluster_health.yml
    ```

31. To move the Resource group, enter the following command:

    ```

    ansible-playbook demo_move_resource_group.yml
    ```

Notes:

1. All the field input values are mentioned in the comments in the `/etc/ansible/external_var.yml` file.
2. For running **start** and **stop** cluster services playbook, you do not update any value in the `/etc/ansible/external_var.yml` file.
3. **NODES** and **NODE_DETAILS** variables in the `/etc/ansible/external_var.yml` file are mandatory for working with any playbook.
4. You can refer to the `Sample_external_var.yml` located in the `/etc/ansible` for updating `external_var.yml` file.
5. Make sure to check the status of the cluster as a prerequisite to run the Ansible playbook.

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.

© Copyright IBM Corp. 2017, 2023

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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

IBM®