FUJITSU

FUJITSU Software PRIMECLUSTER

Concepts Guide 4.6

Oracle Solaris / Linux

J2UL-2501-04ENZ0(00)
May 2022

Preface

This manual is a conceptual overview of the PRIMECLUSTER suite of products. PRIMECLUSTER is a fourth-generation clustering solution that provides high availability and scalability, independent of the operating system and hardware platform. Its modular software architecture consists of a basic set of modules deployed on all computers (nodes) in a cluster, plus optional modules that support specific types of applications. The modular architecture allows flexible clustering solutions for a wide range of customers. The solutions can be adapted to virtually any current or future platform.

Note

This guide describes all the components of PRIMECLUSTER. All of these components might not be available for all releases. "PRIMECLUSTER Software Release Guide" and "PRIMECLUSTER Installation Guide" should be checked to verify which features are available for a specific platform.

Target Readers

This manual is intended for end users, system administrators, and support personnel. The purpose of this manual is to provide conceptual information only. It is not designed as a guide for administration, configuration, or installation (for more information, refer to the section "Related documentation").

About this manual

This manual is organized as follows.

Chapter title	Description
Chapter 1 Clustering technology overview	The chapter describes the concepts and benefits of clustering, including the main components of PRIMECLUSTER.
Chapter 2 PRIMECLUSTER architecture	The chapter describes the PRIMECLUSTER architecture and discusses the key features that PRIMECLUSTER provides.
Chapter 3 Cluster interconnect details	The chapter describes the concepts, requirements, and design considerations of the cluster interconnect.
Chapter 4 Reliant Monitor Services (RMS)	The chapter describes the basic concepts, components, and benefits of RMS.
Chapter 5 RMS wizard	The chapter describes the concepts of the RMS Wizard Tools.
Appendix A Release Information	The appendix lists the main changes in this manual.

Related documentation

Refer to the following manuals as necessary when setting up the cluster:

- PRIMECLUSTER Installation and Administration Guide
- PRIMECLUSTER Installation and Administration Guide Cloud Services
- PRIMECLUSTER Web-Based Admin View Operation Guide
- PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide
- PRIMECLUSTER Reliant Monitor Services (RMS) with Wizard Tools Configuration and Administration Guide
- PRIMECLUSTER Global Disk Services Configuration and Administration Guide
- PRIMECLUSTER Global File Services Configuration and Administration Guide
- PRIMECLUSTER Global Link Services Configuration and Administration Guide: Redundant Line Control Function
- PRIMECLUSTER Global Link Services Configuration and Administration Guide: Redundant Line Control Function for Virtual NIC Mode

- PRIMECLUSTER Global Link Services Configuration and Administration Guide: Multipath Function
- PRIMECLUSTER DR/PCI Hot Plug User's Guide
- PRIMECLUSTER Messages
- PRIMECLUSTE Easy Design and Configuration Guide
- FJQSS (Information Collection Tool) User's Guide

Note

The PRIMECLUSTER documentation includes the following documentation in addition to those listed above:

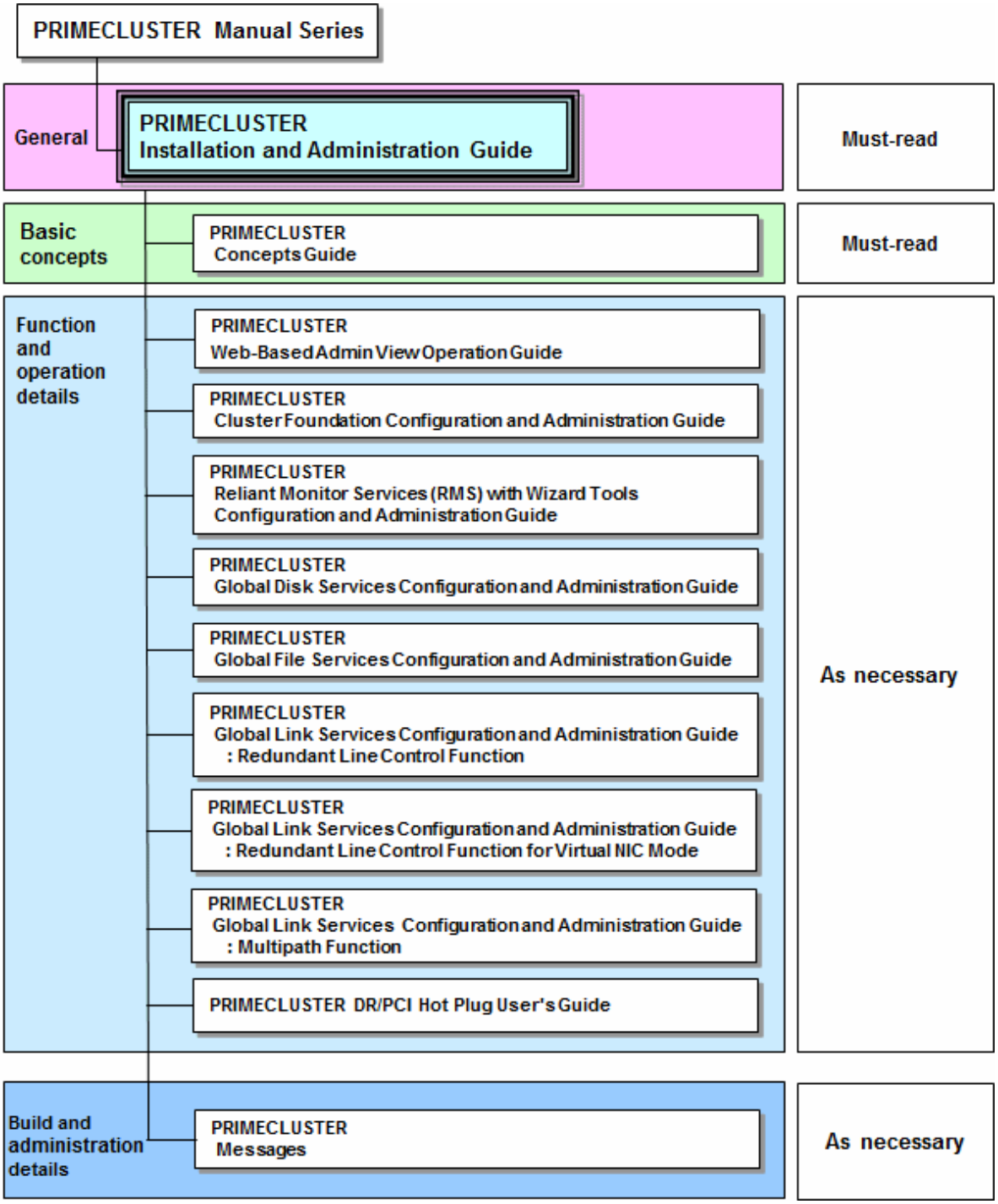
- PRIMECLUSTER Software Release Guide and Installation Guide

  This Software Release Guide and Installation Guide are provided with each PRIMECLUSTER product package.

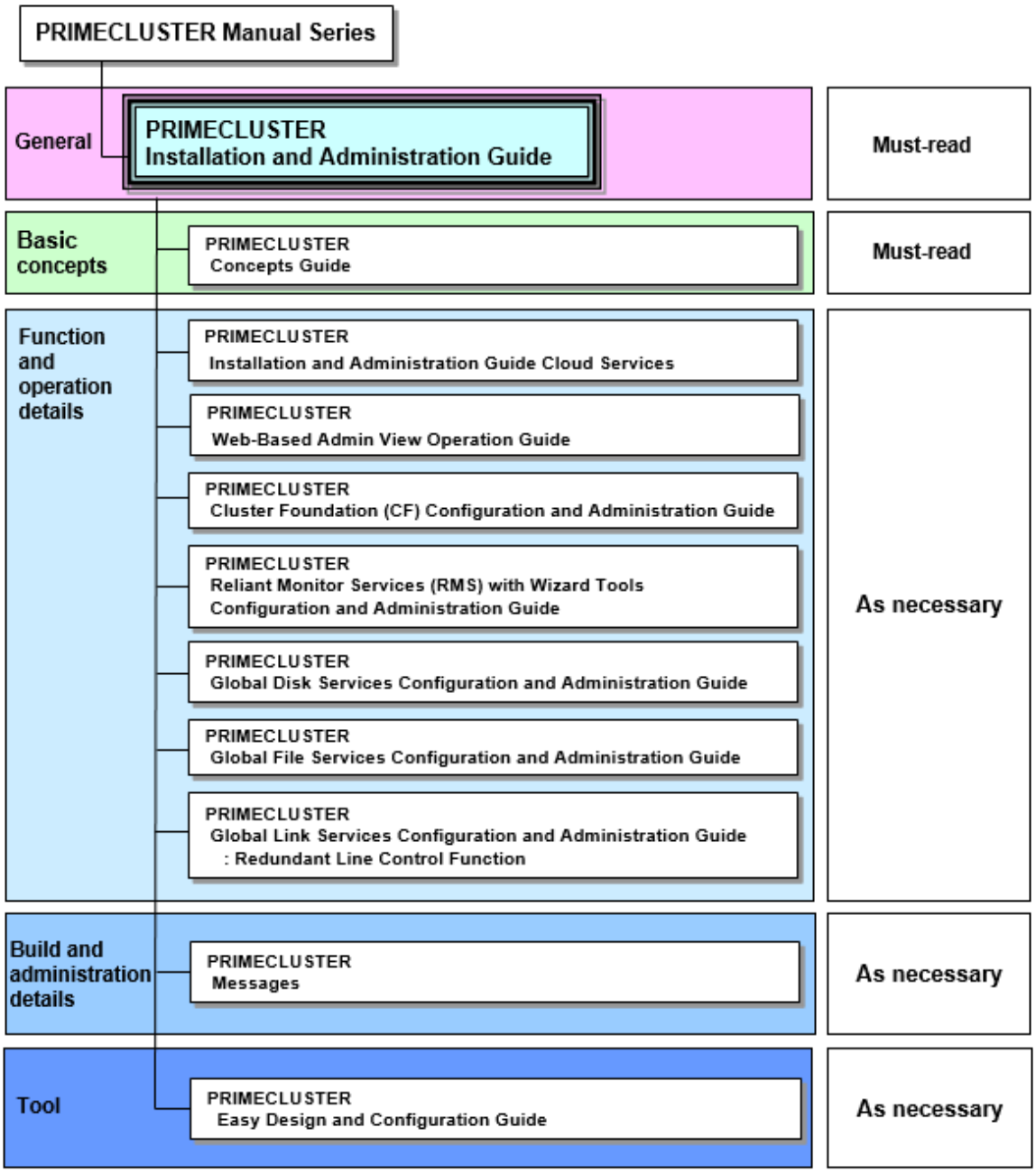
  The data is stored on "DVD" of each package. For details on the file names, see the documentation.

Manual Series

Solaris



Linux (RHEL)



Manual Printing

If you want to print a manual, use the PDF file found on the DVD for the PRIMECLUSTER product. The correspondences between the PDF file names and manuals are described in the Software Release Guide for PRIMECLUSTER that comes with the product.

Adobe Reader is required to read and print this PDF file. To get Adobe Reader, see Adobe Systems Incorporated's website.

Conventions

Notation

Prompts

Command line examples that require system administrator (or root) rights to execute are preceded by the system administrator prompt, the hash sign (#). Entries that do not require system administrator rights are preceded by a dollar sign ($).

In some examples, the notation node# indicates a root prompt on the specified node. For example, a command preceded by fuji2# would mean that the command was run as user *root* on the node named fuji2.

Manual page section numbers

In manuals, helps, and messages of PRIMECLUSTER, a section number in a manual page is shown in parentheses after a command name or a file name. Example: cp(1)

For Linux, or Oracle Solaris 11.4 or later, replace the section numbers as follows:
- "(1M)" to "(8)"
- "(4)" to "(5)"
- "(5)" to "(7)"
- "(7)" to "(4)"

The keyboard

Keystrokes that represent nonprintable characters are displayed as key icons such as [Enter] or [F1]. For example, [Enter] means press the key labeled *Enter*; [Ctrl-b] means hold down the key labeled *Ctrl* or *Control* and then press the [B] key.

Typefaces

The following typefaces highlight specific elements in this manual.

Typeface	Usage
Constant Width	Computer output and program listings; commands, file names, manual page names and other literal programming elements in the main body of text.
Italic	Variables in a command line that you must replace with an actual value. May be enclosed in angle brackets to emphasize the difference from adjacent text, e.g., <*nodename*>RMS; unless directed otherwise, you should not enter the angle brackets. The name of an item in a character-based or graphical user interface. This may refer to a menu item, a radio button, a checkbox, a text input box, a panel, or a window title.
Bold	Items in a command line that you must type exactly as shown.

Typeface conventions are shown in the following examples.

Example 1

Several entries from an /etc/passwd file are shown below:

```
root:x:0:1:0000-Admin(0000):/:/sbin/ksh
sysadm:x:0:0:System Admin.:/usr/admin:/usr/sbin/sysadm
setup:x:0:0:System Setup:/usr/admin:/usr/sbin/setup
daemon:x:1:1:0000-Admin(0000):/:
```

Example 2

To use the cat(1) command to display the contents of a file, enter the following command line:

```
$ cat file
```

Command syntax

The command syntax observes the following conventions.

Symbol	Name	Meaning
[ ]	Brackets	Enclose an optional item.
{ }	Braces	Enclose two or more items of which only one is used. The items are separated from each other by a vertical bar (|).

Symbol	Name	Meaning
|	Vertical bar	When enclosed in braces, it separates items of which only one is used. When not enclosed in braces, it is a literal element indicating that the output of one program is piped to the input of another.
( )	Parentheses	Enclose items that must be grouped together when repeated.
...	Ellipsis	Signifies an item that may be repeated. If a group of items can be repeated, the group is enclosed in parentheses.

Notation symbols

Material of particular interest is preceded by the following symbols in this manual:

Point

Contains important information about the subject at hand.

Note

Describes an item to be noted.

Example

Describes operation using an example.

Information

Describes reference information.

See

Provides the names of manuals to be referenced.

Abbreviations

Oracle Solaris might be described as Solaris, Solaris Operating System, or Solaris OS.

If "Solaris X" is indicated in the reference manual name of the Oracle Solaris manual, replace "Solaris X" with "Oracle Solaris 10 (Solaris 10)," or "Oracle Solaris 11 (Solaris 11)."

Red Hat Enterprise Linux is abbreviated as RHEL.

RHEL is described as Linux.

Red Hat OpenStack Platform is abbreviated as RHOSP.

PRIMEQUEST 3000/2000 Series are abbreviated as PRIMEQUEST.

FUJITSU Hybrid IT Service FJcloud-O is abbreviated as FJcloud-O.

FUJITSU Hybrid IT Service FJcloud-V is abbreviated as FJcloud-V.

FUJITSU Hybrid IT Service FJcloud-Baremetal is abbreviated as FJcloud-Baremetal.

FUJITSU Hybrid IT Service for Microsoft Azure is abbreviated as "for Azure."

FUJITSU Hybrid IT Service for AWS is abbreviated as "for AWS."

FJcloud-V sold by FUJITSU LIMITED and NIFCLOUD sold by FUJITSU CLOUD TECHNOLOGIES LIMITED are abbreviated as "NIFCLOUD" in this manual.

"for Azure" sold by FUJITSU LIMITED and Microsoft Azure sold by Microsoft Corporation in the United States are abbreviated as "Azure" in this manual.

"for AWS" sold by FUJITSU LIMITED and AWS (Amazon Web Services) sold by Amazon.com, Inc. are abbreviated as "AWS" in this manual.

Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

Trademarks

Requests

- No part of this documentation may be reproduced or copied without permission of FUJITSU LIMITED.
- The contents of this documentation may be revised without prior notice.

Date of publication and edition

December 2019, First edition
May 2022, Third edition

Copyright notice

Contents

Chapter 1 Clustering technology overview

This chapter introduces the basic concepts and benefits of clustering, including the main components of PRIMECLUSTER.

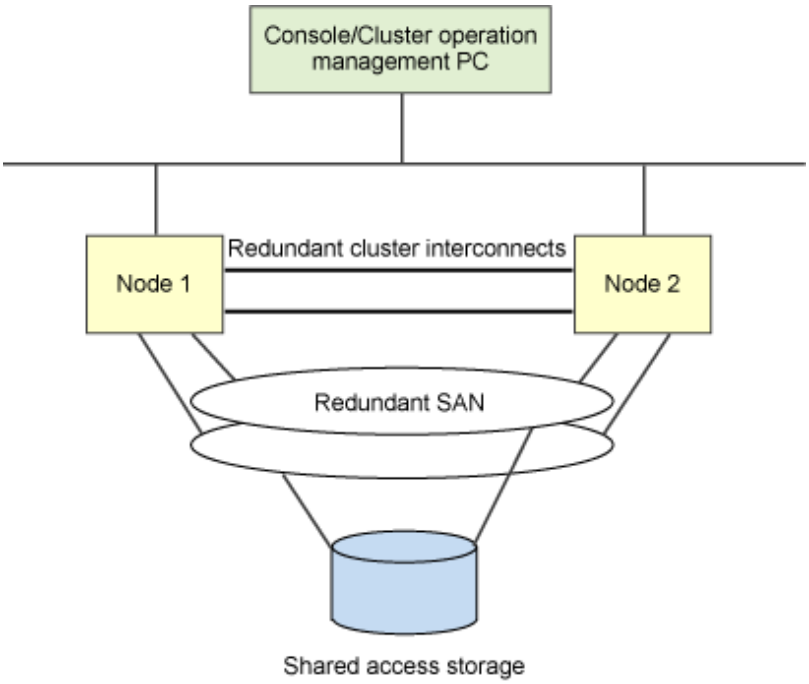
1.1 Introduction

Clustering is imprecisely defined in distributed computing. In general, a cluster is a combination of computers or partitions of a computer (nodes) that act cooperatively to provide one or more of the following:

- High availability (HA)
  Increasing service availability by using redundant components
- Scalability
  Supplied by replicating application resources

This document focuses on cluster servers that provide HA and scalability as provided by the PRIMECLUSTER software suite. It does not discuss other kinds of clustering such as administrative clusters or scientific computing clusters.

Figure 1.1 Typical two-node cluster



PRIMECLUSTER also supports the single-node cluster configuration with one node. Application status is monitored in a single-node cluster. If an error is detected, availability can be improved by automatically restarting the application and trying recovery.

Furthermore, PRIMECLUSTER supports the following virtualization environments and cloud environments:

Virtualization environments

- KVM environment
- RHOSP environment
- VMware environment
- Oracle VM Server for SPARC environment
- Oracle Solaris Zones environment
  - Kernel Zone
  - Non-global Zone

Cloud environments

- FJcloud-O environment
- NIFCLOUD environment
- FJcloud-Baremetal environment
- AWS environment
- Azure environment

1.2 High availability

HA clusters use redundant components to compensate for a failure. Each node in the cluster must know if the other nodes in the cluster are operational. They do this by sending heartbeats over the cluster interconnects.

1.2.1 Cluster interconnects

The cluster interconnect is a dedicated network connection PRIMECLUSTER uses for communication processing between nodes and it is the most basic building block of a cluster. Cluster interconnect redundancy is highly recommended to prevent cluster partition due to failure of cluster interconnect.

In addition to heartbeat requests, the cluster interconnects carry messages between nodes such as notification of events, communications between processes, and cluster file access. Additional details are discussed in the chapter "Chapter 3 Cluster interconnect details" later in this document.

1.2.2 HA manager

PRIMECLUSTER HA manager is the Reliant Monitor Services (RMS) which ensures high availability of applications within the cluster. It monitors the state of resources for applications and the resources used by those applications. It conducts provision of wizard to enable the recovery of user's operation and assuring integrity as a user's asset.

1.2.2.1 Protecting data integrity

HA manager protects data integrity by performing the following tasks:

- Monitoring applications
- Handling cluster partition
- Starting applications automatically only when all cluster nodes are in a known state (except when otherwise controlled due to settings of the *HV_AUTOSTARTUP_IGNORE* or *PARTIAL_CLUSTER* environment variables)

Monitoring applications

RMS is configured with rules specific to the applications and the configuration of the cluster. When a detector reports a failure, RMS takes the appropriate actions to recover the resources that are needed to provide continued availability of the application. The recovery actions are defined for every application and resource.

RMS recovery actions are as follows:

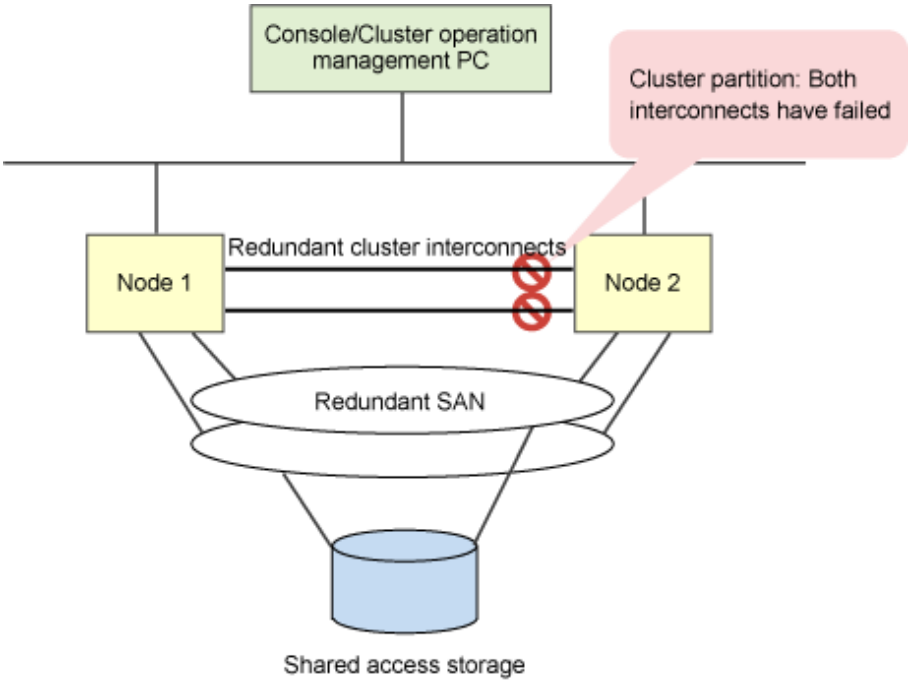
- Local recovery
  The application's resources are recovered, and the application is restarted on the same cluster node.
- Remote recovery
  The application's resources are recovered, and the application is restarted on another cluster node.

Cluster partition

A cluster partition is the result of multiple failures in the cluster interconnects. Even though cluster interconnects fail, some or all of the nodes continue to operate, but the communication between some cluster nodes remain stopped (this is sometimes called split-brain syndrome). Redundant cluster interconnects are effective but not enough for preventing split-brain syndrome.

The figure below shows an example of two breaks in the redundant cluster interconnects in which Node 1 and Node 2 can no longer communicate with each other. However, both nodes still have access to the SAN. Therefore, if recovery actions were taken independently on each node, two instances of an application could be running unaware on the two nodes of the cluster. If these instances were to make uncoordinated updates to their data, then data corruption would occur. Clearly, this condition cannot be allowed.

Figure 1.2 Cluster partition in two-node cluster



To prevent data corruption, PRIMECLUSTER provides a system to ensure the integrity between nodes as follows:

1. When a heartbeat occurs and each node cannot communicate with a target node (if it is not clear that the nodes still operate or they have been stopped), set the target node as *LEFTCLUSTER* state.
2. Cancel *LEFTCLUSTER* state.
3. PRIMECLUSTER checks that nodes within the cluster system are the following states before starting a recovery processing at each node.

   - All the nodes are either *UP* or *DOWN*.
     (no node is *LEFTCLUSTER* state)
   - Operating nodes can communicate every other operating node.

   In PRIMECLUSTER, if the integrity between nodes is ensured as above, it is called "cluster consistent state (quarum)."

The terms consistent state and quorum are used interchangeably in PRIMECLUSTER documents. The cluster is in a consistent state when every node of the cluster is in a known state (*UP* or *DOWN*) and each node that is *UP* can communicate with every other node that is *UP*. The applications in the cluster should ensure that the cluster is in a consistent state before starting any operations that will alter shared data. For example, RMS ensures that the cluster is in a consistent state before it will start any applications in the cluster.

PRIMECLUSTER performs elimination through a variety of methods, depending on the architecture of the nodes. When PRIMECLUSTER determines that a node becomes the *LEFTCLUSTER* state, it eliminates that node; thereby, recovering the application and guaranteeing data integrity.

Note

The term *quorum* has been used in various ways in the literature to describe the handling of cluster partitions. Usually, this implies that when (n + 1)/2 nodes can see each other, they have a quorum and the other nodes will not perform any I/O operations. Because PRIMECLUSTER methods differ from the conventional meaning of quorum, the term *cluster integrity* was adopted. Some PRIMECLUSTER commands use the *quorum* term heritage, but in these cases *cluster integrity* is implied.

Cluster Integrity Monitor

The purpose of the Cluster Integrity Monitor (CIM) is to allow applications to determine when it is safe to perform operations on shared resources. It is safe to perform operations on shared resources when a node is a member of a cluster that is in a consistent state.

A consistent state is when all the nodes of a cluster that are members of the CIM set are in a known and safe state. The nodes that are members of the CIM set are specified in the CIM configuration. Only these nodes are considered when the CIM determines the state of the cluster.

When a node first joins or forms a cluster, the CIM indicates that the cluster is consistent only if it can determine the following:

- The status of the other nodes that make up the CIM set
- Whether the nodes of the CIM set are in a safe state

The method used to determine the state of the members of the cluster is sometimes called the CIM method. The CIM can use several different CIM methods; however, the following are available by default and are discussed here:

- NSM
  The Node State Monitor (NSM) monitors the node states at fixed cycles, and it tracks the state of the nodes that are currently, or have been, members of the cluster. This is also known as the NULL or default CIM method. NSM is an integrated part of the PRIMECLUSTER CF.
- RCI
  The RCI (Remote Cabinet Interface) is a special SPARC Enterprise M series environmental control and state network that can asynchronously both report on the state of the systems and control the systems on Solaris systems. (For more information, refer to "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide.")
- XSCF SNMP
  The XSCF SNMP (eXtended System Control Facility Simple Network Management Protocol) is a special SPARC M10, M12 environmental control and state network that can asynchronously both report on the state of the systems and control the systems on Solaris systems. (For more information, refer to "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide.")
- MMB
  The MMB (Management Board) is a special PRIMEQUEST environmental control and state network that can asynchronously both report the status of the systems and control the systems on Linux systems. (For more information, refer to "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide.")

PRIMECLUSTER allows you to register and use multiple CIM methods. When multiple CIM methods are registered, CIM uses the lower priority method to check the state of a node only if the higher priority method cannot determine the node state. For example, if RCI and NSM are registered as CIM methods and RCI has the higher priority, then CIM uses the CIM method that uses RCI to check the node status.

If the target is a node or a partition, the RCI CIM method returns *UP* or *DOWN*, and then processing ends. However, if the node being checked by the RCI method is not connected to the RCI or if the RCI is not operating properly, then the RCI method will fail. CIM then uses the NSM-based CIM method to check the node state.

Similarly if MMB and NSM are registered as CIM methods and MMB has the higher priority, then CIM uses the CIM method that uses MMB to check the node status. In this case, if the target is a PRIMEQUEST node, the MMB CIM method returns *UP* or *DOWN*, and then processing ends. However, if the node being checked by the MMB method is not connected to the MMB or if the MMB is not operating properly, then the MMB method will fail. CIM then uses the NSM-based CIM method to check the node state.

The CIM reports on whether a node state in a cluster is consistent (true), or a node state is not consistent (false) for the cluster. True and false are defined as follows:

- *TRUE*
  A known state for all CIM nodes
- *FALSE*
  An unknown state for any cluster CIM node

Shutdown Facility

The CIM allows applications to determine when a cluster is in a consistent state, but it does not take action to resolve inconsistent clusters. Many different methods to ensure consistent clusters have been used in the high-availability field, but there is only one method that has proven completely effective and does not require cooperation between the nodes. PRIMECLUSTER uses this method known as the Shutdown Facility (SF) to return to a consistent cluster state when something occurs to disrupt that state. In the cluster partition example

shown in Figure 1.2 Cluster partition in two-node cluster, both nodes will report the other node as having the state *LEFTCLUSTER*. The CIM will return a *FALSE* status. To get the cluster into a consistent state, SF forces one of the nodes into a safe state by either forcing a panic or shutting off the power.

The SF can be configured to eliminate nodes through a variety of methods. When the SF receives a request to eliminate a node, it tries to shut down the node by using the methods in the order that were specified. Once a method has successfully eliminated the node, the node's state is changed to *DOWN* by the SF.

The transition from *LEFTCLUSTER* to *DOWN* is the signal used by the various cluster services to start recovery actions. Note that different systems will support different shutdown methods. For example, the cluster console is available for Solaris, but is not available for Linux.

If all of the configured SF methods fail to return a positive acknowledgement that the requested node has been eliminated, then no further action is taken. This leaves the cluster in an inconsistent state and requires operator intervention to proceed.

This approach ensures that damage to user data could not occur by inadvertently allowing an application to run in two parts of a partitioned cluster. This also protects from the situation where a node fails to respond to heartbeats (for example, an extreme system load) and then comes back to life later. In this case, the application on the node that returns to life may continue to run even though the other node has taken action to start that application.

PRIMECLUSTER allows you to use various hardware-specific methods to set definition that reset nodes on which Solaris or Linux operate. For more information, refer to "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide."

Monitoring Agents (MA)

PRIMECLUSTER provides a mechanism where hardware monitors can be used to quickly detect a system state change and inform the cluster membership functions. Without this monitoring capability, only the cluster heartbeat timeout will detect that a node has panicked; this will take up to 10 seconds with the default heartbeat interval. When a Monitoring Agent (MA) is used, it can detect a node panic very quickly. For example, with PRIMEPOWER hardware and the RCI, the MA takes less than 1 second to detect a system panic. MAs are implemented as plug-ins that interfaces with the Shutdown Facility.

The MA technology allows PRIMECLUSTER to recover from monitored node failures very quickly. For non-cluster aware applications the time from when a node panic occurs to the time that the application recovery begins can be as short as 2.5 seconds under optimal conditions. The time the application takes to start up and become ready to operate varies from one application to another. For cluster-aware applications, such as Oracle RAC, the time from a system panic to the time Oracle has started recovery and is processing queries on the surviving nodes can be as short as 6.5 seconds. At this point, Oracle may still be performing some recovery actions that might impact performance, but it is able to respond to user queries.

If a node fails, PRIMECLUSTER does the following:

1. Detects a node failure
2. Notifies of the failure
3. Confirms the node state
4. Eliminates the node

The MA notifies SF of a node failure on detecting it. SF seeks a redundant confirmation regarding the node state to assess the reliability of the failure notification. This verification procedure is required to prevent the node that is normally running from being shut down.

SF confirms the node state as follows:

- Collects the node state information from all registered MAs again.
- Checks if the response to the CF heartbeat request is returned.

SF prompts the MA to eliminate the failed node when all the MAs notify SF of the node failure, and CF notifies SF of the failure in responding to the heartbeat request. When the node elimination is done, this brings the other node *DOWN*.

I/O Fencing function

Uses an exclusive control function by SCSI-3 Persistent Reservation in the cluster configuration connected to the shared disk device and prevents simultaneous access from both of the nodes.

This function can be only used in the following virtualization environments.

- VMware environment
- Oracle VM Server for SPARC environment

1.2.2.2 RMS configuration tools

To properly recover an application, RMS must know about the resources an application requires for proper operation. The configuration of the resources and the relationship between the resources can be very complex. RMS Wizard Tools performs configuration definition to specify these information to the RMS. These information can be configured through GUI by using userApplication Configuration Wizard.

The configuration tool foundations (RMS Wizard Tools) capture generic information about the cluster and common application services.

1.2.3 Patrol diagnosis facility (Solaris)

The patrol diagnosis facility periodically diagnoses the following hardware units that are connected to the standby node.

- Shared disk units

The function diagnoses whether a shared disk unit has become unusable because the power is switched off, a cable is disconnected (adapter side or device side) or because of some other reason.

If the diagnosis results indicate that an error was detected in a shared disk unit, an error message is output.

- Network interface cards

The function diagnoses whether any network interface card cannot communicate because a cable is disconnected or because of some other reason.

If the diagnosis results indicate that an error of a network interface card was detected, an error message is output and switching to the standby node (failover) becomes disabled.

1.3 Scalability

Scalability is another benefit of PRIMECLUSTER. Scalability is provided by the cluster's ability to grow in computing capacity when demand increases. There are two basic types of applications relative to scalability. These types of applications can be divided as follows:

- Applications that interact closely with the cluster software and are designed for a distributed environment
- Applications that are not aware of the cluster

Applications that interact with cluster software

An example of a scalable application that interacts with cluster software is Oracle RAC. Oracle RAC starts an instance of the database server on some or all the nodes of the cluster.

Applications that are not aware of the cluster

Applications that do not have special provisions for distributing their work in the cluster can be replicated on different nodes. If the application that access the same file is to be run simultaneously, share the files between cluster nodes by using Global Files Services (hereinafter GFS) function. It means that distributing the applications on multiple nodes increases the effect of load balancing.

For details on GFS, see the section "2.3 PRIMECLUSTER components" and the manuals for "PRIMECLUSTER Global File Services Configuration and Administration Guide."

1.4 Single-node cluster

Single-node cluster is a cluster system with one node, and it is possible to monitor and control services on the node.

If an error is detected, availability can be improved by automatically restarting the application and trying recovery.

You can also use this mode as a development environment for creating and testing cluster applications.

However, services will be stopped if a hardware error occurs. Moreover, no failover occurs in the single-node cluster operation.

1.5 Taking over data between nodes

In PRIMECLUSTER, the following methods can be selected for taking over data between cluster nodes:

- A shared disk method

  With this method, data is stored in a shared disk unit.

  You can use the following shared disk units:

  - A shared disk unit connecting to multiple servers through SAN (Storage Area Network)

    This is suitable when data accessibility performance or data availability is especially important, or data capacity or data scalability is required.

  - NAS (Network Attached Storage) unit

    You can build a cluster system at a relatively low cost.

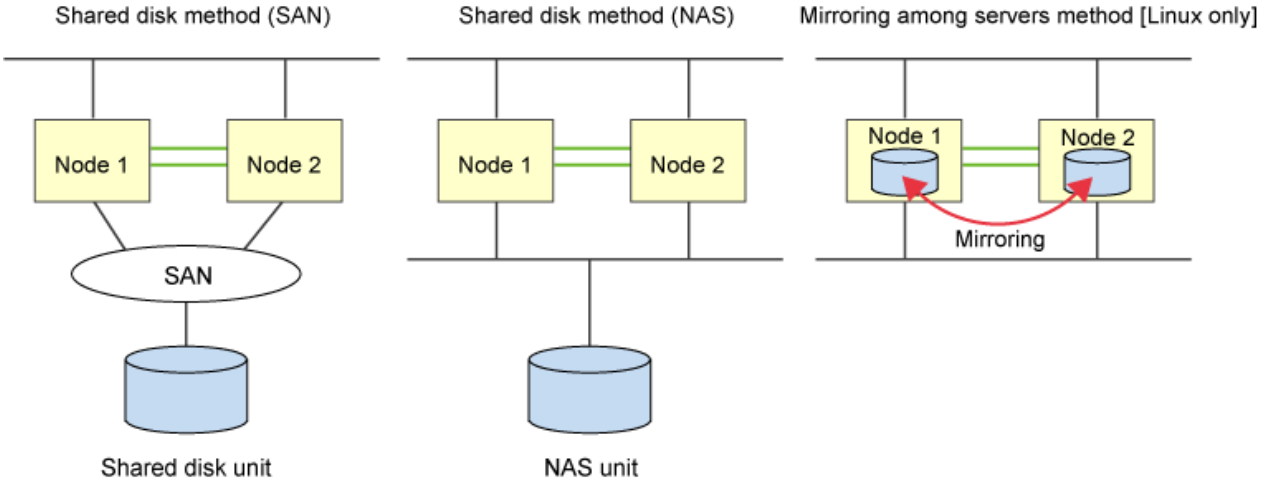
- Mirroring among servers methods (Linux)

  With this method, data is stored in local disks on each node and those local disks are mirrored through a network with the mirroring among servers of Global Disk Services (hereinafter GDS).

  You can build a cluster system at a low cost because an expensive external storage is not required.

  This is suitable for a small-scale system that keeps and updates less data.

Figure 1.3 Taking over data between nodes



1.6 Virtualization support

PRIMECLUSTER allows a redundant configuration in the following virtualization environments and also enables high reliability of the integrated system:

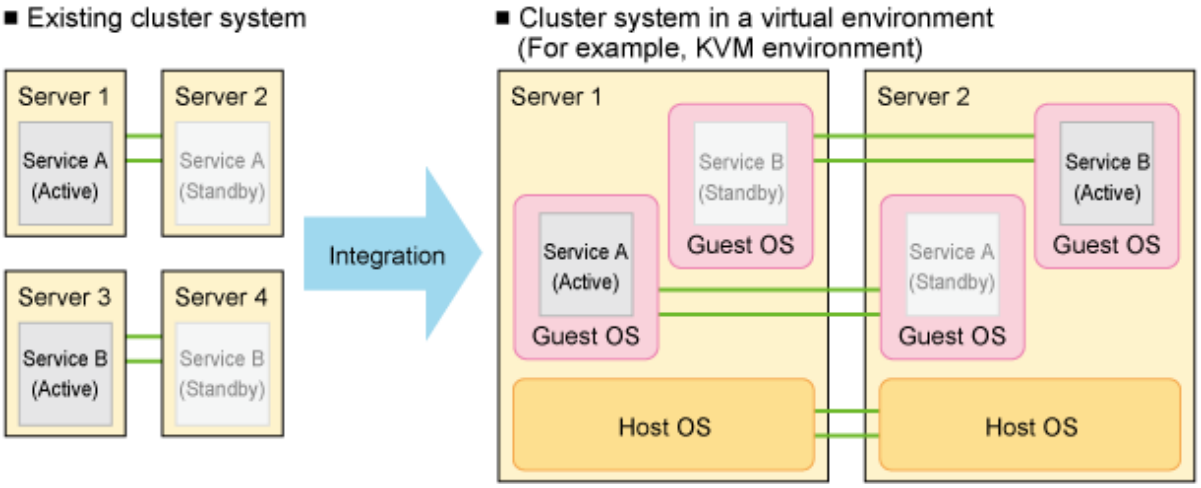
- KVM environment
- RHOSP environment
- VMware environment
- Oracle VM Server for SPARC environment
- Oracle Solaris Zones environment
  - Kernel Zone

- Non-global Zone

Previously, the cluster system has been built and operated with different servers for each service.
However, it is possible to integrate multiple services to a single server by supporting virtualization environments.

Within a server, CPU resources can be allocated and operated for each service.

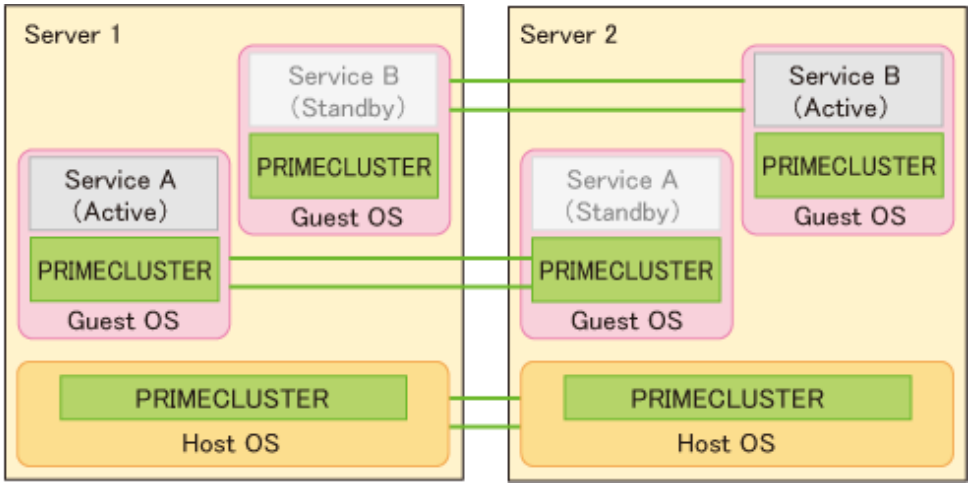
Figure 1.4 Existing cluster system and cluster system in a virtualization environment (For example, KVM environment)



KVM environment

By installing PRIMECLUSTER into each host OS and guest OS, you can immediately switch servers in the event of an operating system error as well as an application error.

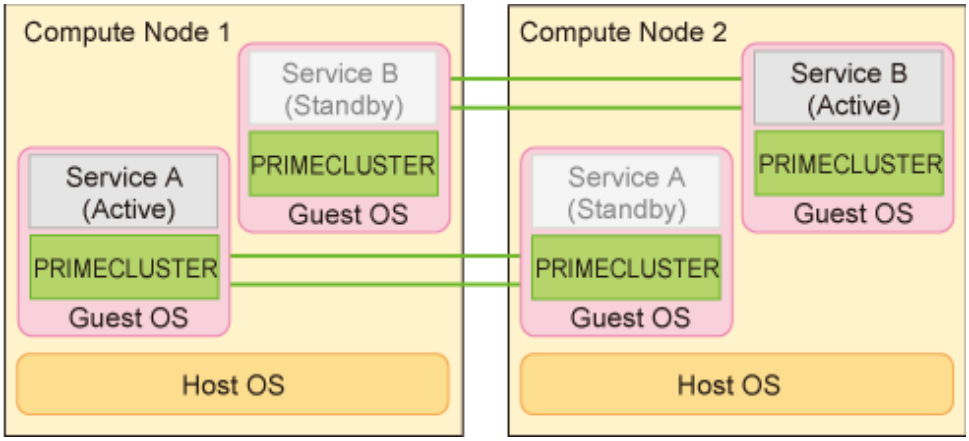
Figure 1.5 Cluster system in a virtualization environment (For example, KVM environment)



RHOSP environment

By installing PRIMECLUSTER into the guest OS, you can immediately switch the guest OS when an error occurs in applications or in OS.

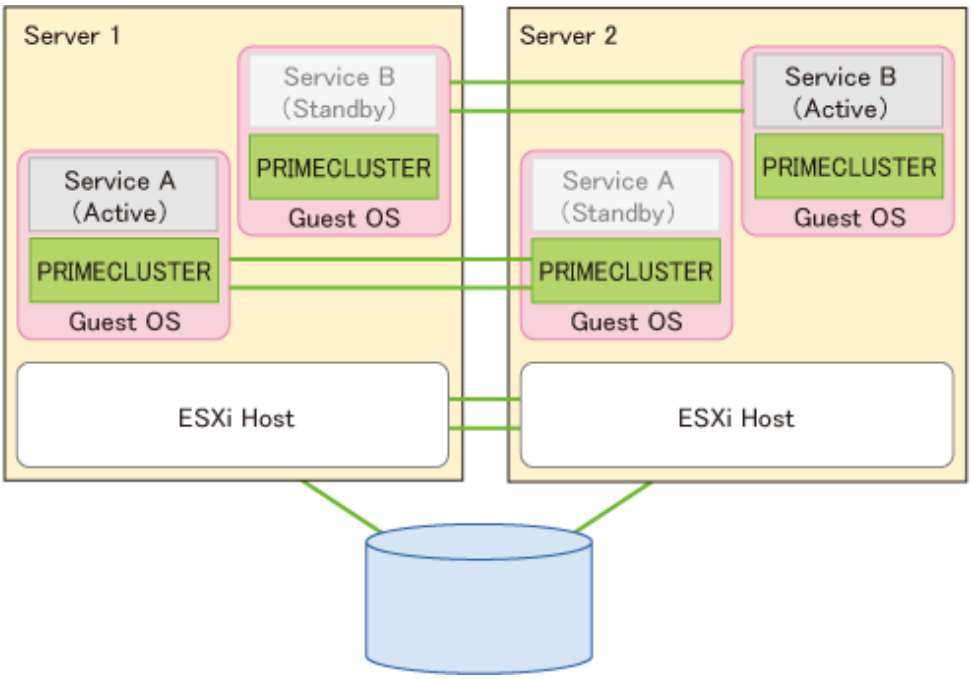
Figure 1.6 Cluster system in a virtualization environment (For example, RHOSP environment)



VMware environment

Install PRIMECLUSTER into guest OSes only.
When an error of applications or operating systems occurs, you can safely and securely switch servers if VMware vCenter Server functional cooperation or the I/O fencing function using shared disks is used.

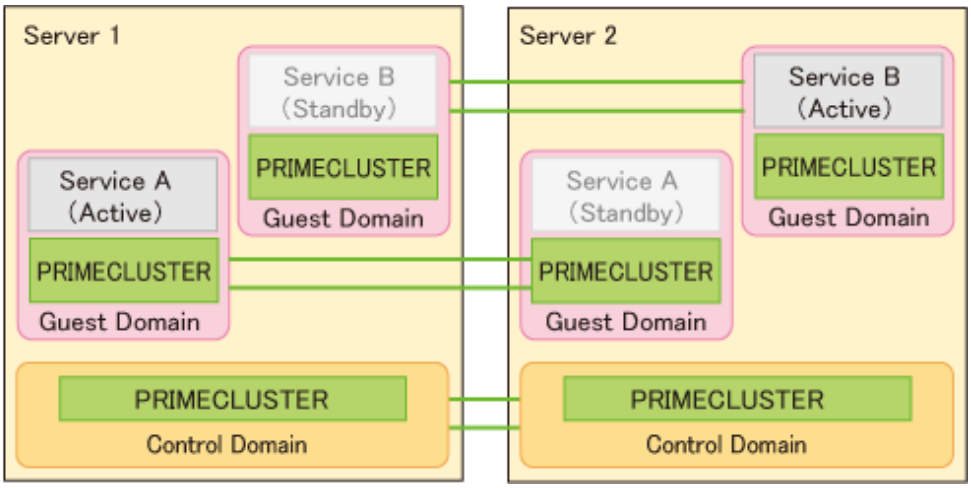
Figure 1.7 Cluster system in a virtualization environment (For example, VMware environment)



Oracle VM Server for SPARC environment

By installing PRIMECLUSTER into control domain and guest domains, you can immediately switch servers in the event of a guest domain error as well as a partition error.

Figure 1.8 Cluster system in a virtualization environment (For example, Oracle VM Server for SPARC environment)



Oracle Solaris Kernel Zones environment

By installing PRIMECLUSTER into a control domain or a guest domain after creating Kernel Zones in a control domain or a guest domain, you can immediately switch servers in the event of a domain error as well as a partition error.

Figure 1.9 Cluster system in a virtualization environment (For example, Oracle Solaris Kernel Zones environment on Control Domain)

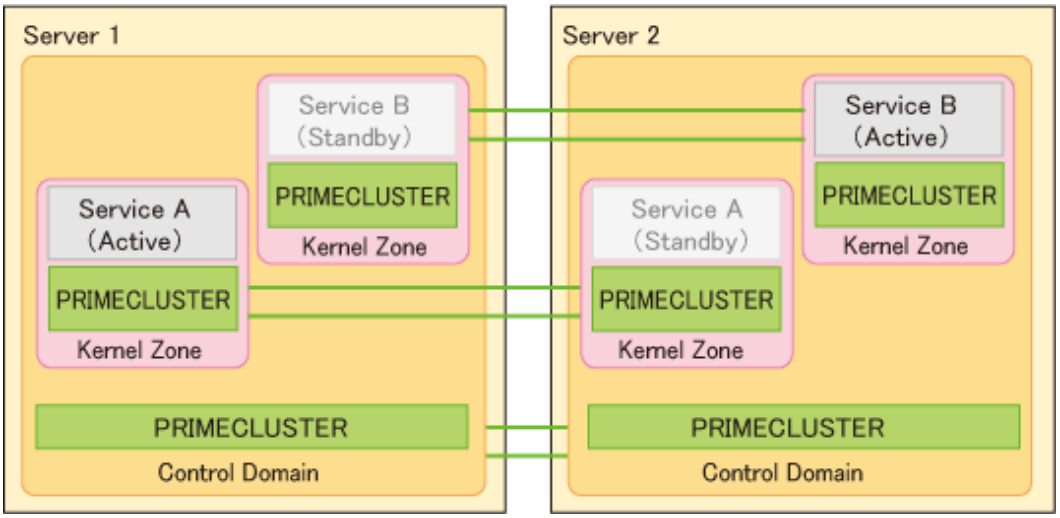
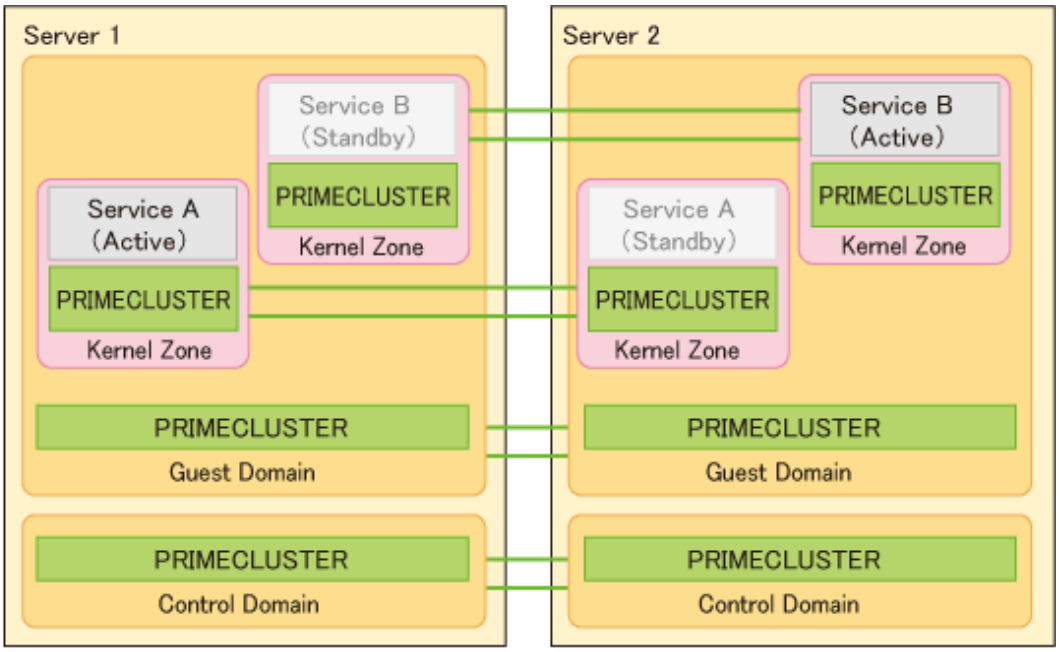


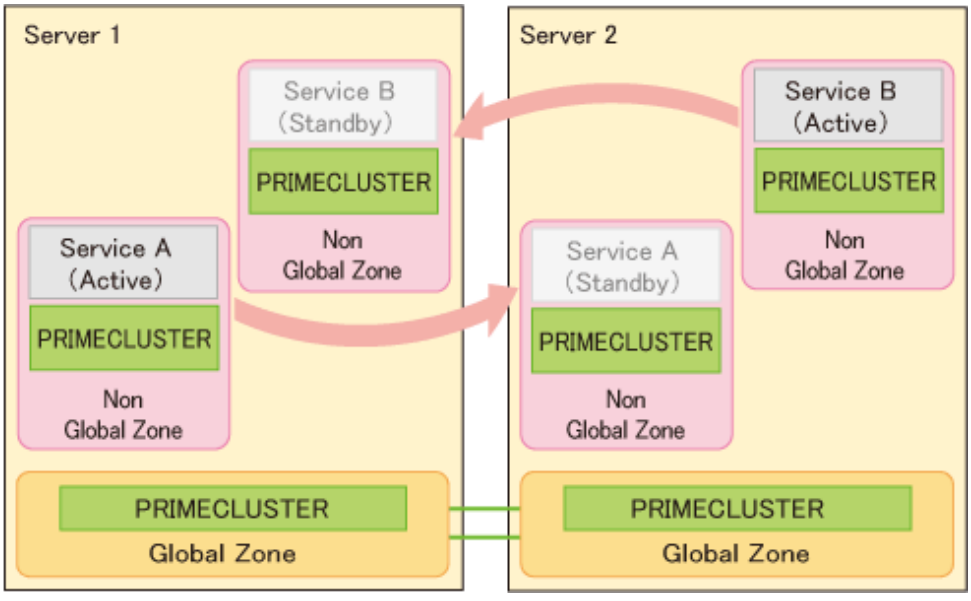
Figure 1.10 Cluster system in a virtualization environment (For example, Oracle Solaris Kernel Zones environment on Guest Domain)



Oracle Solaris Non-global Zones environment

By installing PRIMECLUSTER into the global zone, you can switch each server when an error occurs in the global zone.
Moreover, by installing PRIMECLUSTER into non-global zones, you can switch each non-global zone when an error occurs in applications.

Figure 1.11 Cluster system in a virtualization environment (For example, Oracle Solaris Non-global Zones environment)



1.7 Cloud support

PRIMECLUSTER allows a redundant configuration in the following cloud environments and also enables high reliability of the integrated system:

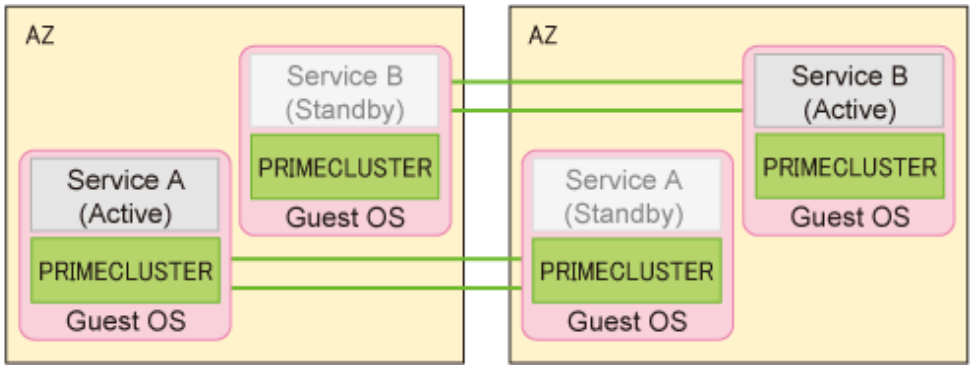
- FJcloud-O environment

- NIFCLOUD environment
- FJcloud-Baremetal environment
- AWS environment
- Azure environment

By installing PRIMECLUSTER into the guest OS, you can immediately switch the guest OS when an error occurs in applications or in OS.

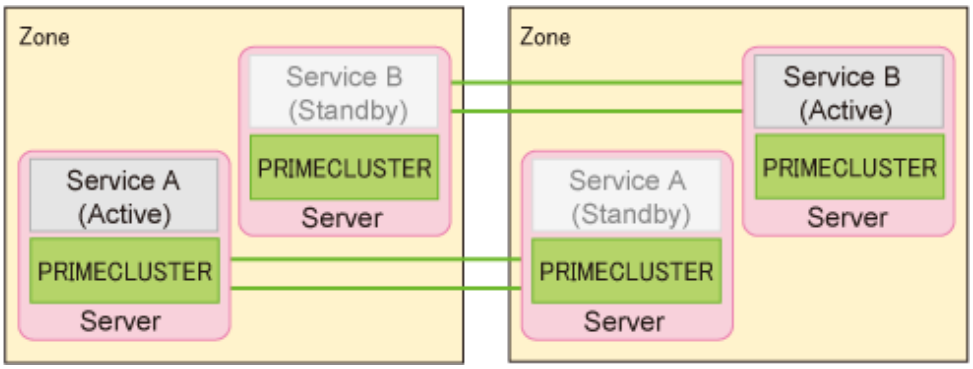
FJcloud-O environment

Figure 1.12 Cluster system in an FJcloud-O environment



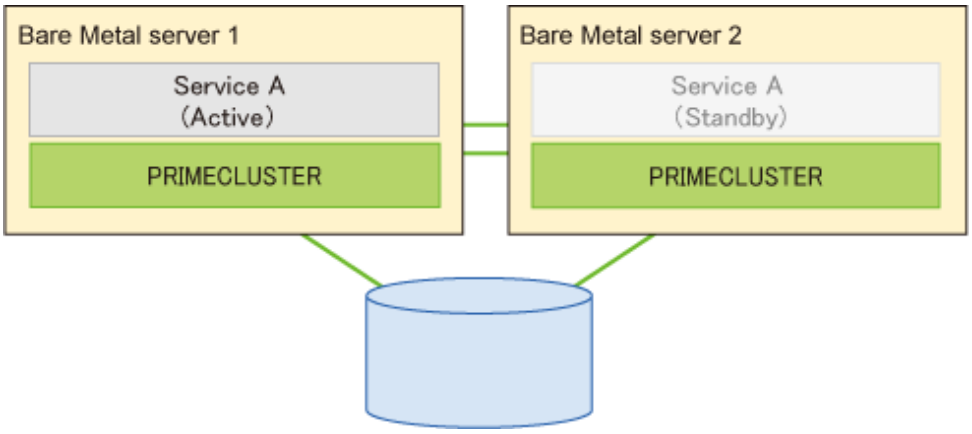
NIFCLOUD environment

Figure 1.13 Cluster system in a NIFCLOUD environment



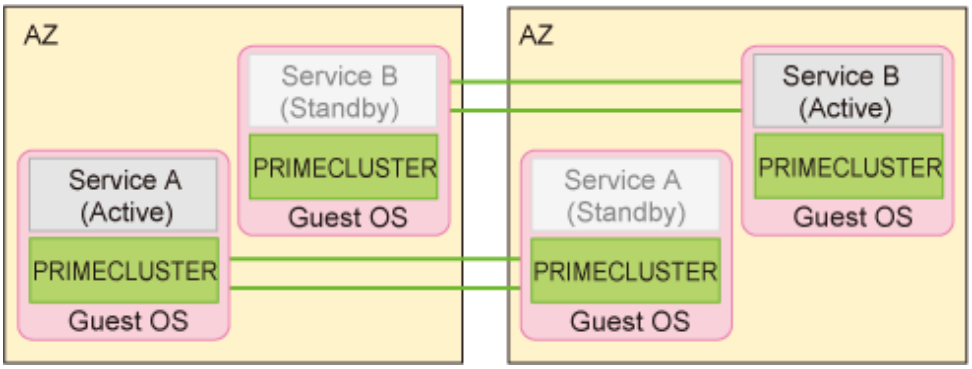
FJcloud-Baremetal environment

Figure 1.14 Cluster system in an FJcloud-Baremetal environment (using RHEL)



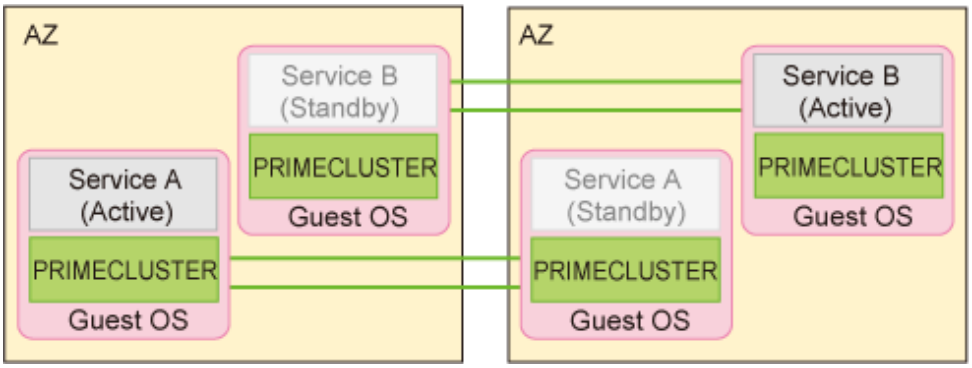
AWS environment

Figure 1.15 Cluster system in an AWS environment



Azure environment

Figure 1.16 Cluster system in an Azure environment



1.8 Switching behavior performed when an error occurs

This section describes the switching behavior according to each cluster system configuration when an error occurs.

1.8.1 Linux

This section describes the availability of cluster system in the following environments in Linux.

- Cluster system in the physical environment
- Cluster system in the virtual environment
- Cluster system in the cloud environment

1.8.1.1 Physical environment and virtual environment

This section describes the availability of cluster system in the following environments in Linux.

- Cluster system in the physical environment
- Cluster system between guest OSes with the Host OS failover function (KVM)
- Cluster system between guest OSes on multiple host OSes (KVM)
- Cluster system between guest OSes on one host OS (KVM)
- Cluster system between guest OSes on multiple compute nodes (RHOSP)
- Cluster system between guest OSes on one compute node (RHOSP)
- Cluster system between guest OSes on multiple ESXi hosts (VMware)

- Cluster system between guest OSes on one ESXi host (VMware)

The table below summarizes the availability of error detection in each monitored cluster system.

Table 1.1 Availability according to each cluster system configuration (in a physical environment and virtual environment)

Monitoring target	Physical server	KVM			RHOSP		VMware	
		Cluster system between guest OSes with the Host OS failover function	Cluster system between guest OSes on multiple host OSes	Cluster system between guest OSes on one host OS	Cluster system between guest OSes on multiple compute nodes	Cluster system between guest OSes on one compute node	Cluster system between guest OSes on multiple ESXi hosts	Cluster system between guest OSes on one ESXi host
1. Unit	Y	Y	N	N	Y*1	N	Y*2	N
2. Shared disk and path of disk access	Y	Y	Y	N	Y	N	Y	N
3. Public LAN	Y	Y	Y	N	Y	N	Y	N
4. OS (physical, host OS/ ESXi host)	Y	Y	N	N	Y*1	N	Y*2	N
5. OS (guest OS)	-	Y	Y	Y	Y	Y	Y*3	Y*4
6. Service (cluster application)	Y	Y	Y	Y	Y	Y	Y	Y

Service continuity when an error occurs Y: Available, N: Unavailable, - : Excluded

*1 The service can be continued by configuring high availability for compute instances.
For more information on configuring high availability for compute instances, refer to "High Availability for Compute Instances" in "Red Hat OpenStack Platform."

*2 Only when the I/O fencing function is used or VMware vCenter Server functional cooperation and VMware vSphere HA are used, if a hang-up is detected in a guest OS and the guest OS cannot be switched to the standby system automatically, the guest OS will be changed to LEFTCLUSTER state.

*3 When the guest OS cannot be switched to the standby system automatically, the guest OS becomes the LEFTCLUSTER state.

*4 Only when VMware vCenter Server functional cooperation is used, the guest OS can be switched automatically.

Figure 1.17 Physical environment

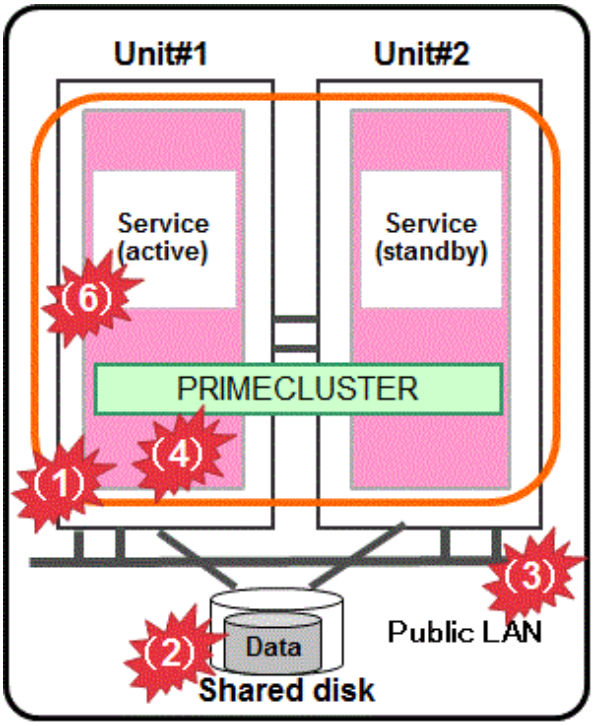
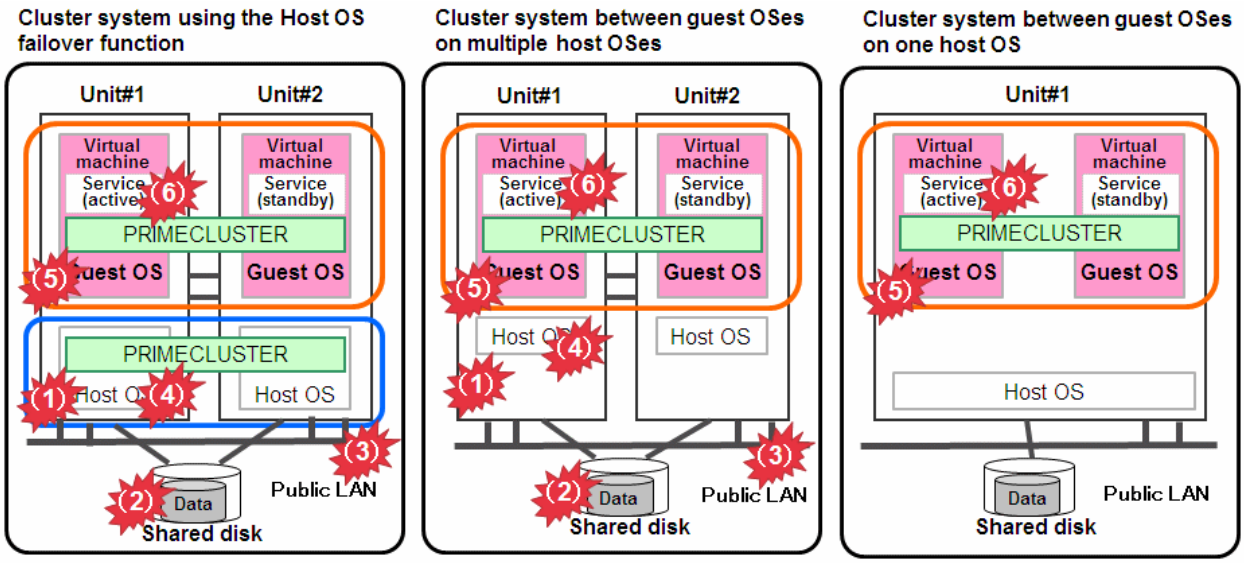


Figure 1.18 Virtual environment



For the RHOSP environment, read "host OS" as "compute node". For the VMware environment, read "host OS" as "ESXi host."

How to detect an error in the following targets to be monitored

1. Unit

   For PRIMEQUEST 2000, the asynchronous monitoring linked with Management Board (MMB), and for PRIMEQUEST3000, the asynchronous monitoring linked with iRMC/MMB, immediately detects a panic or a reset triggered by an error in CPU, memory, or others, and the service is switched to the standby system. For PRIMERGY and virtual environments, an error is detected by the heartbeat monitoring, and the service is switched to the standby system. *1

2. Shared disk and path of disk access

   Combining with the volume management function (GDS), the system detects a failure of a disk access or disk access path (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed or a failure of the whole system of the disk access path occurs.

3. Public LAN

   Combining with the network multiplexing function (Global Link Services, hereinafter referred to as GLS), the system detects a failure of a network adapter or a route in the public LAN (monitored by the Gls resource), and the service is switched to the standby system when a failure of the whole system of the network occurs.

4. OS (physical and host OS/ESXi host)

   An error is detected by the heartbeat monitoring, and the service is switched to the standby system. *1

5. OS (guest OS)

   An error is detected by the heartbeat monitoring, and the service is switched to the standby system.

6. Service (cluster application)

   When a resource error of the cluster application occurs, the service is switched to the standby system.

*1 For the cluster system between guest OSes (RHOSP, VMware) on different host OSes, the status becomes LEFTCLUSTER. After the guest OS is restarted by high availability configuration for compute instances (RHOSP) or the vSphere HA function (VMware), LEFTCLUSTER state of the guest OS is automatically cleared and the service is switched to the standby system.

1.8.1.2 Cloud environment

This section describes the availability of cluster systems in the following environments in Linux.

- Cluster system between guest OSes (FJcloud-O)
- Cluster system in multiple zones (NIFCLOUD)
- Cluster system in a single zone (NIFCLOUD)
- Cluster system between Bare Metal servers (FJcloud-Baremetal)
- Cluster system in multiple Availability Zones (Multi-AZ) (AWS)
- Cluster system in a single Availability Zone (Single-AZ) (AWS)
- Cluster system in multiple Availability Zones (Azure)
- Cluster system in a single Availability Zone (Azure)

The table below summarizes the availability of error detection in each monitored cluster system.

Table 1.2 Availability according to each cluster system configuration (in a cloud environment)

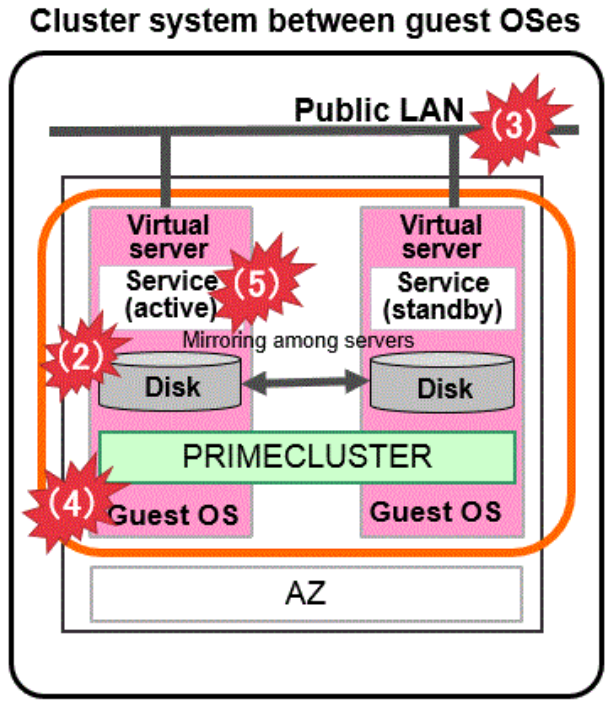
Monitoring target	FJcloud-O	NIFCLOUD		FJcloud-Baremetal	AWS		Azure	
	Cluster system between guest OSes	Cluster system in multiple zones	Cluster system in a single zone	Cluster system between Bare Metal servers	Cluster system in multiple Availability Zones (Multi-AZ)	Cluster system in a single Availability Zone (Single-AZ)	Cluster system in multiple Availability Zones	Cluster system in a single Availability Zone
1. AZ/Zone	N	Y *1	N	- *2	Y	N	Y *1	N
2. Disk	Y	Y	Y	Y	Y	Y	Y	Y
3. Public LAN	Y	Y	Y	Y	Y	Y	Y	Y
4. OS (guest OS)	Y	Y	Y	Y	Y	Y	Y	Y
5. Service (cluster application)	Y	Y	Y	Y	Y	Y	Y	Y
6. Bare Metal server	-	-	-	Y	-	-	-	-

Service continuity when an error occurs Y: Available, N: Unavailable, - : Excluded

*1 An error is detected in AZ (Azure) or a zone (NIFCLOUD), and the node becomes LEFTCLUSTER. Continue the operation by recovering the LEFTCLUSTER state. For how to recover from the LEFTCLUSTER state, refer to "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide."

*2 There is no AZ in East Japan region 3 and West Japan region 3, where an FJcloud-Baremetal environment is provided.

Figure 1.19 FJcloud-O environment



How to detect an error in the following targets to be monitored

1. AZ

   AZ is not a target to be monitored.

2. Disk

   Combining with the volume management function (GDS), the system detects an error of a disk access (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed.

3. Public LAN

   Combining with the network multiplexing function (GLS), the system detects a failure of a network adapter or a route in the public LAN (monitored by the Gls resource), and the service is switched to the standby system when a failure of the whole system of the network occurs.

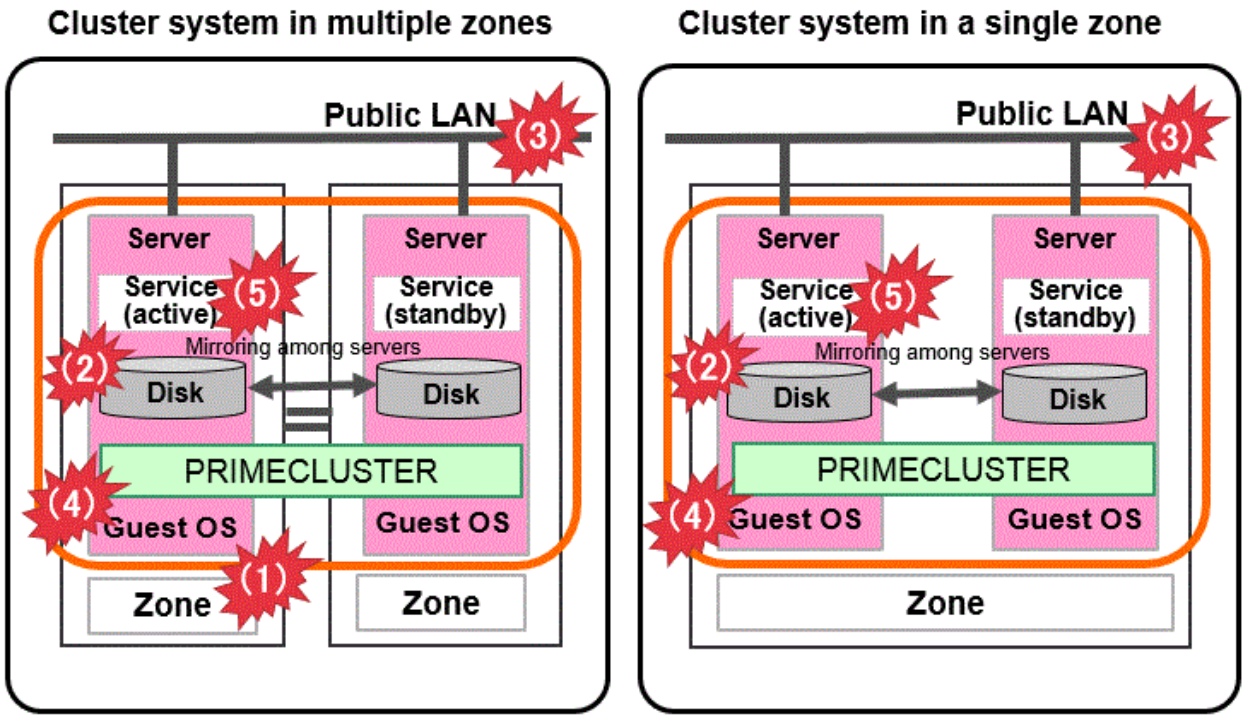
4. OS (guest OS)

   An error is detected by the heartbeat monitoring, and the service is switched to the standby system.

5. Service (cluster application)

   When a resource error of the cluster application occurs, the service is switched to the standby system.

Figure 1.20 NIFCLOUD environment



How to detect an error in the following targets to be monitored

1. Zone

   The cyclic monitoring of the cluster interconnect detects an error of a zone, and the node becomes LEFTCLUSTER.

2. Disk

   GDS monitors I/O to a disk, and when an error of the disk access occurs, the disk is detached and the service continues.

   If an I/O error occurs in all slices in a mirror, the service is automatically switched to the standby system.

3. Public LAN

   The network monitoring using ICMP detects a route failure, and the service is automatically switched to the standby system.

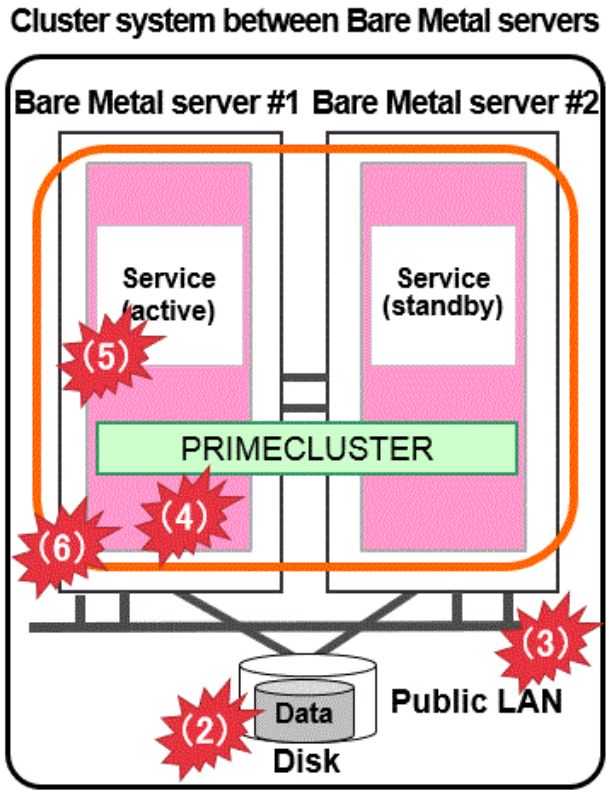
4. OS (guest OS)

   The cyclic monitoring of the cluster interconnect detects an error of the guest OS, and the service is automatically switched to the standby system.

5. Service (cluster application)

   When a resource error of the cluster application occurs, the service is automatically switched to the standby system.

Figure 1.21 FJcloud-Baremetal environment



How to detect an error in the following targets to be monitored

2. Disk

Combining with the volume management function (GDS), the system detects an error of a disk access (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed.

3. Public LAN

Combining with the network multiplexing function (GLS), the system detects a failure of a network adapter or a route in the public LAN (monitored by the Gls resource), and the service is switched to the standby system when a failure of the whole system of the network occurs.

4. OS (guest OS)

An error is detected by the heartbeat monitoring, and the service is switched to the standby system.

5. Service (cluster application)

When a resource error of the cluster application occurs, the service is switched to the standby system.

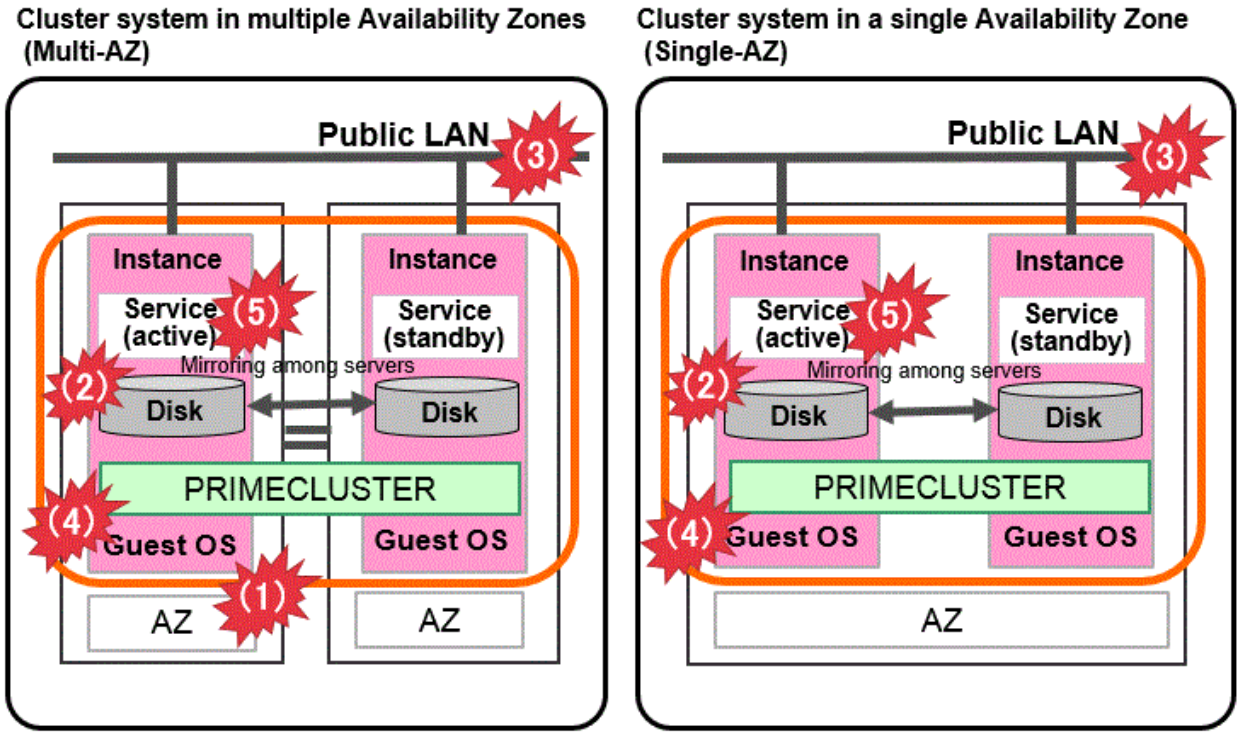
6. Bare Metal server

An error is detected by the heartbeat monitoring, and the service is switched to the standby system.

See

When using VMware, refer to "1.8.1.1 Physical environment and virtual environment."

Figure 1.22 AWS environment



How to detect an error in the following targets to be monitored

1. AZ

   An error is detected by the heartbeat monitoring, and the service is automatically switched.

2. Disk

   Combining with the volume management function (GDS), the system detects an error of a disk access (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed.

3. Public LAN

   By registering scripts for control to the Cmdline resource, the system detects a route failure, and the service is switched to the standby system in the event of a network failure.

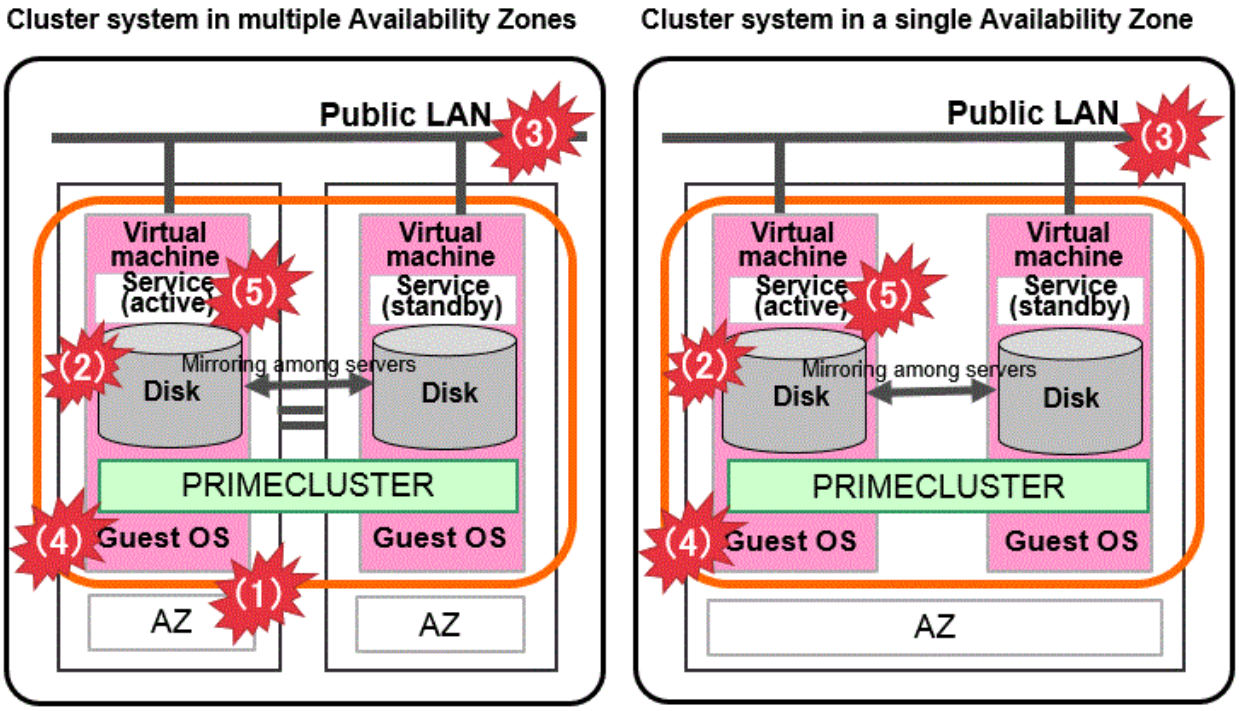
4. OS (guest OS)

   An error is detected by the heartbeat monitoring, and the service is switched to the standby system.

5. Service (cluster application)

   When a resource error of the cluster application occurs, the service is switched to the standby system.

Figure 1.23 Azure environment



How to detect an error in the following targets to be monitored

1. AZ

   An error is detected by the heartbeat monitoring, and the node becomes LEFTCLUSTER.

2. Disk

   Combining with the volume management function (GDS), the system detects an error of a disk access (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed.

3. Public LAN

   By registering scripts for control to the Cmdline resource, the system detects a route failure, and the service is switched to the standby system in the event of a network failure.

4. OS (guest OS)

   An error is detected by the heartbeat monitoring, and the service is switched to the standby system.

5. Service (cluster application)

   When a resource error of the cluster application occurs, the service is switched to the standby system.

1.8.2 Oracle Solaris

This section describes the availability of cluster system in the following environments in Oracle Solaris.

- Cluster system in the physical environment
- Cluster system in Oracle VM Server for SPARC environment
- Cluster system in Oracle Solaris Zones environment
  - Cluster system in Oracle Solaris Kernel Zones environment
  - Cluster system in Oracle Solaris Non-global Zones environment

1.8.2.1 Oracle Solaris (Physical environment and Oracle VM Server for SPARC environment)

This section describes the availability of cluster system in the following physical environment and Oracle VM Server for SPARC environment in Oracle Solaris.

- Cluster system in the physical environment
- Cluster system between guest domains among different physical partitions in an Oracle VM Server for SPARC environment
- Cluster system between guest domains within the same physical partition in an Oracle VM Server for SPARC environment
- Cluster system between control domains in an Oracle VM Server for SPARC environment

The table below summarizes the availability of error detection in each monitored cluster system.

Table 1.3 Availability according to each cluster system configuration

Monitoring target	Physical environment	Oracle VM Server for SPARC environment		
		Cluster system between guest domains among different physical partitions	Cluster system between guest domains within the same physical partition	Cluster system between control domains
1. Physical partition	Y	Y	N	Y
2. Shared disk and path of disk access	Y	Y	N	Y
3. Public LAN	Y	Y	N	Y
4. OS (physical and control domains)	Y	Y	Y*1	Y
5. OS (guest domain)	-	Y	Y	Y*2
6. Service (cluster application)	Y	Y	Y	Y*3

Service continuity when an error occurs Y: Available, N: Unavailable, - : Excluded

*1 The service can be continued because the OS in the guest domain is available even when an OS error in the control domain occurs.

*2 The OS in the guest domain cannot be monitored. When the state of the guest domain (state displayed in the ldm list-domain) is in error, PRIMECLUSTER in the control domain monitors the state of the guest domain so that the service can be continued by switching the OS in the guest domain to the standby system.

*3 The service (cluster application) on the control domain can be monitored but the service on the guest domain cannot be monitored.

Figure 1.24 Physical environment

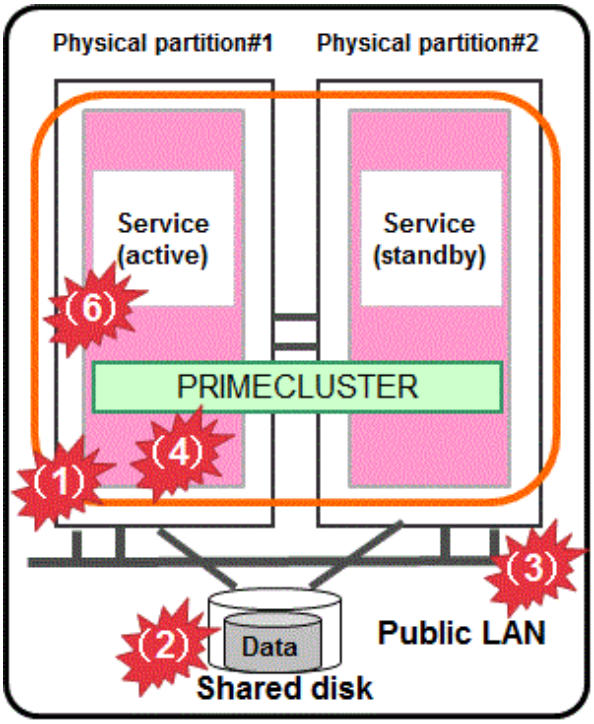
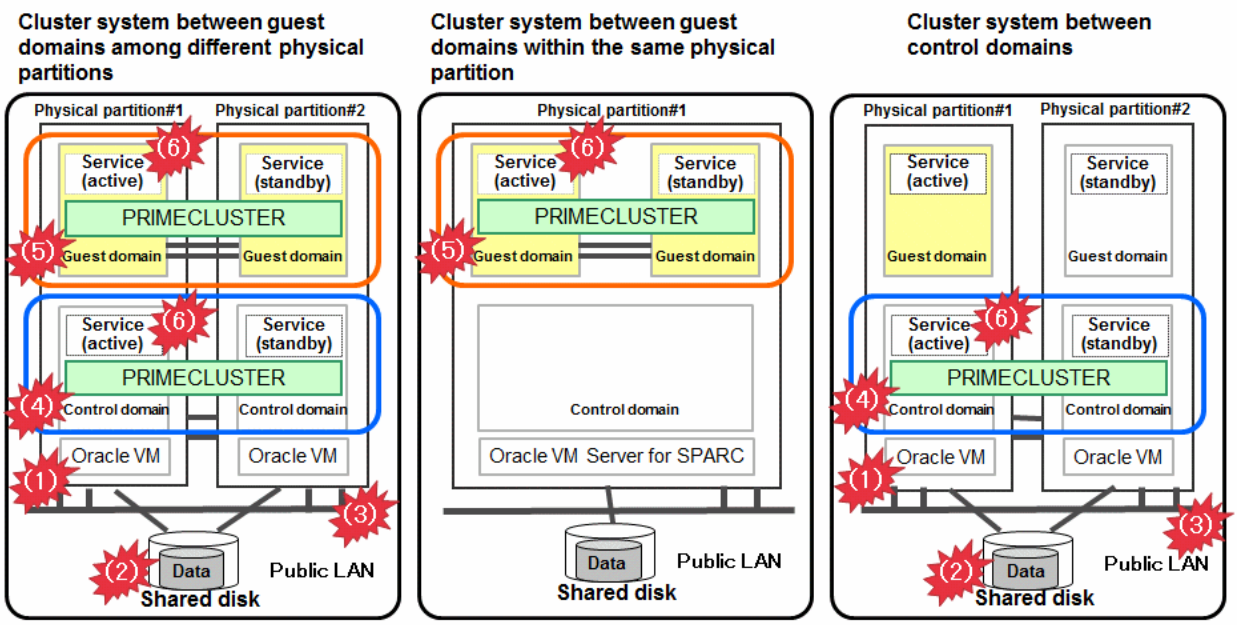


Figure 1.25 Oracle VM Server for SPARC environment



How to detect an error in the following targets to be monitored

1. Physical partition

   The asynchronous monitoring linked with the system monitoring function of server immediately detects a panic or a reset triggered by an error in CPU, memory, or others, and the service is switched to the standby system.

2. Shared disk and path of disk access

   Combining with the volume management function (GDS), the system detects a failure of a disk access or disk access path (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed or a failure of the whole system of the disk access path occurs.

3. Public LAN

   Combining with the network multiplexing function (GLS), the system detects a failure of a network adapter or a route in the public LAN (monitored by the Gls resource), and the service is switched to the standby system when a failure of the whole system of the network occurs.

4. OS (physical and control domains)

   A panic or a reset of the OS is immediately detected by the asynchronous monitoring, and the service is switched to the standby system. A hang-up of the OS in the control domain is detected by the cyclic monitoring of cluster interconnect (LAN) and the service is switched to the standby system.

   For the cluster system between guest domains within the same physical partition, an OS error in the control domain cannot be detected because it is a single domain.

5. OS (guest domain)

   A panic or a reset of the OS is immediately detected by the asynchronous monitoring, and the service is switched to the standby system. A hang-up of the OS in the guest domain is detected by the cyclic monitoring of cluster interconnect (LAN) and the service is switched to the standby system.

   For the cluster system between control domains, an error of the service in a guest domain cannot be detected.

6. Service (cluster application)

   When a resource error of the cluster application occurs, the service is switched to the standby system.

1.8.2.2 Oracle Solaris (Oracle Solaris Kernel Zones environment)

This section describes the availability of following cluster systems in Oracle Solaris Kernel Zones.

- Cluster system between Kernel Zones among different physical partitions (Control domain)
- Cluster system between Kernel Zones among different physical partitions (Guest domain)
- Cluster system between Kernel Zones within the same physical partition (Control domain)
- Cluster system between Kernel Zones within the same physical partition (Guest domain)

The table below summarizes the availability of error detection in each monitored cluster system.

Table 1.4 Availability according to each cluster system configuration

Monitoring target	Oracle Solaris Kernel Zones environment			
	Cluster system between Kernel Zones among different physical partitions (Control domain)	Cluster system between Kernel Zones among different physical partitions (Guest domain)	Cluster system between Kernel Zones within the same physical partition (Control domain)	Cluster system between Kernel Zones within the same physical partition (Guest domain)
1. Physical partition	Y	Y	N	N
2. Shared disk and path of disk access	Y	Y	N	N
3. Public LAN	Y	Y	N	N
4. OS (physical and control domains)	Y	Y*1	N	Y*1
5. OS (guest domain)	-	Y	-	Y*2
6. OS (Kernel Zones)	Y	Y	Y	Y
7. Service (cluster application)	Y	Y	Y	Y

Service continuity when an error occurs Y: Available, N: Unavailable, - : Excluded

*1 The service can be continued because the OS in the guest domain is available even when an OS error in the control domain occurs.

*2 The service cannot be continued in the cluster system between Kernel Zones within the same guest domain.

Figure 1.26 Oracle Solaris Kernel Zones environment (among different physical partitions)

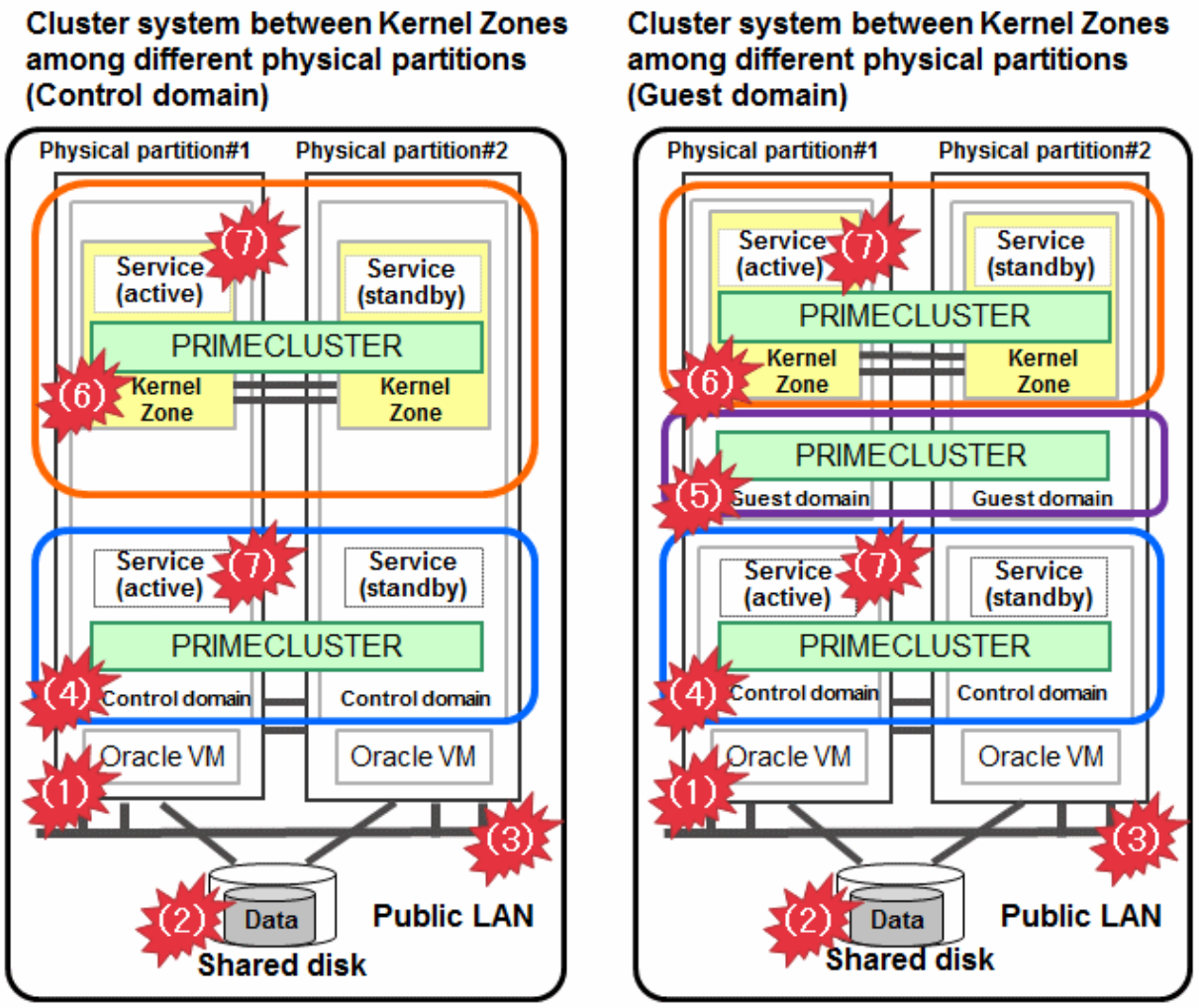
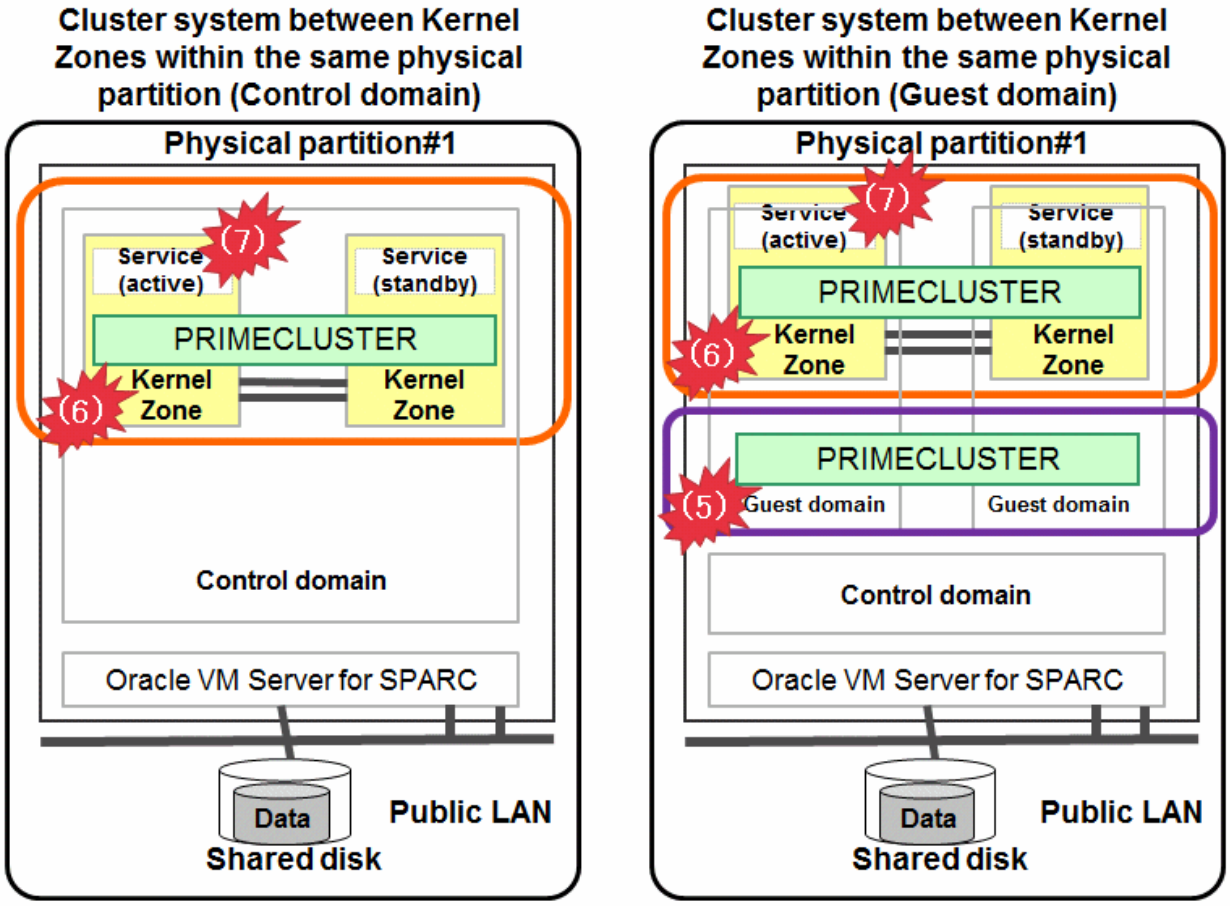


Figure 1.27 Oracle Solaris Kernel Zones environment (within the same physical partition)



How to detect an error in the following targets to be monitored

1. Physical partition

   The asynchronous monitoring linked with the system monitoring function of server immediately detects a panic or a reset triggered by an error in CPU, memory, or others, and the service is switched to the standby system.

2. Shared disk and path of disk access

   Combining with the volume management function (GDS), the system detects a failure of a disk access or disk access path (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed or a failure of the whole system of the disk access path occurs.

3. Public LAN

   Combining with the network multiplexing function (GLS), the system detects a failure of a network adapter or a route in the public LAN (monitored by the Gls resource), and the service is switched to the standby system when a failure of the whole system of the network occurs.

4. OS (physical and control domains)

   A panic or a reset of the OS is immediately detected by the asynchronous monitoring, and the service is switched to the standby system. Additionally, a hang-up of the OS is detected by the cyclic monitoring of cluster interconnect (LAN), and the service is switched to the standby system.

   For the cluster system between Kernel Zones within the same physical partition, an OS error in the control domain cannot be detected because it is a single domain.

5. OS (guest domain)

   A panic or a reset of the OS is immediately detected by the asynchronous monitoring, and the service is switched to the standby system. Additionally, a hang-up of the OS is detected by the cyclic monitoring of cluster interconnect (LAN), and the service is switched to the standby system.

   For the cluster system between Kernel Zones within the same guest domain, an OS error in the guest domain cannot be detected because it is a single domain.

6. OS (Kernel Zones)

   A panic, a reset, or a hang-up of the OS is detected by the cyclic monitoring of cluster interconnect (LAN), and the service is switched to the standby system.

7. Service (cluster application)

   When a resource error of the cluster application occurs, the service is switched to the standby system.

1.8.2.3 Oracle Solaris (Oracle Solaris Non-global Zones environment)

This section describes the availability of cluster system in the following environments in Oracle Solaris Non-global Zones.

- Cold-standby environment (non-global zone on the standby system is inactive [service is also inactive on the standby system])
- Warm-standby environment (non-global zone on the standby system is active [service is inactive on the standby system])
- Single node cluster environment

The table below summarizes the availability of error detection in each monitored cluster system.

Table 1.5 Availability according to each cluster system configuration

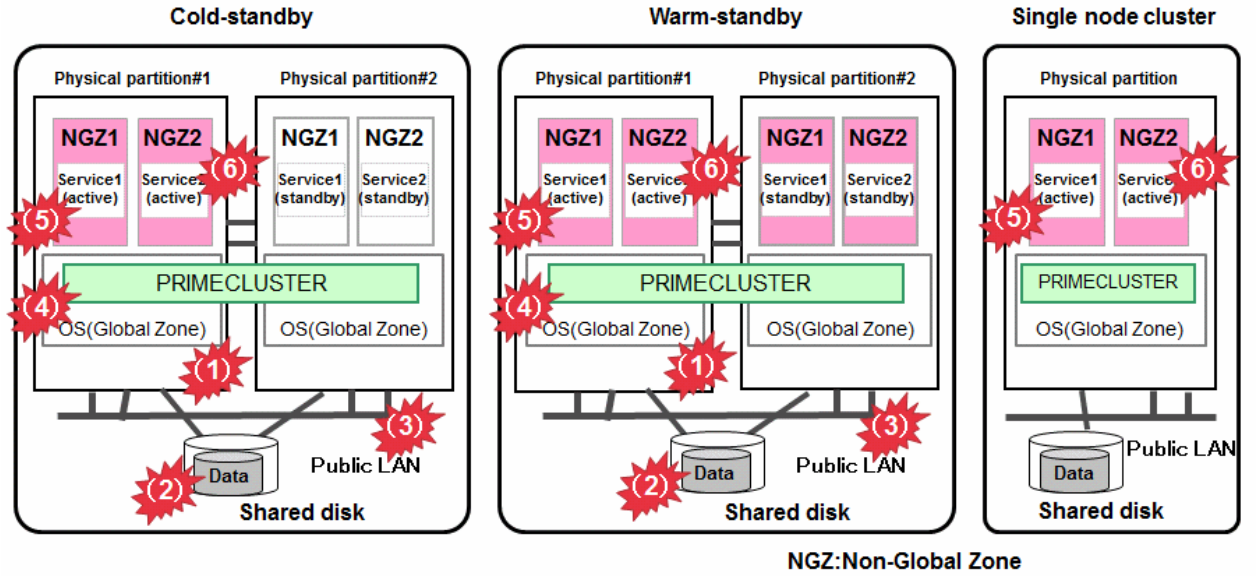
Monitoring target	Cold-standby	Warm-standby	Single node cluster
1. Physical partition	Y	Y	-
2. Shared disk and path of disk access	Y	Y	-
3. Public LAN	Y	Y	-
4. OS (global zone)	Y	Y	-
5. OS (non-global zone)	Y	Y	Y*1
6. Service (cluster application)	Y	Y	Y*2

Service continuity when an error occurs Y: Available, N: Unavailable, - : Excluded

*1 When an error is detected, the service can be continued by restarting the non-global zone.

*2 When an error is detected, the service can be continued by restarting the service (cluster application).

Figure 1.28 Oracle Solaris Zones environment



How to detect an error in the following targets to be monitored

1. Physical partition

   The asynchronous monitoring linked with the system monitoring function of server immediately detects a panic or a reset triggered by an error in CPU, memory, or others, and the service is switched to the standby system.

2. Shared disk and path of disk access

   Combining with the volume management function (GDS), the system detects a failure of a disk access or disk access path (monitored by the Gds resource), and the service is switched to the standby system when the disk cannot be accessed or a failure of the whole system of the disk access path occurs.

3. Public LAN

   Combining with the network multiplexing function (GLS), the system detects a failure of a network adapter or a route in the public LAN (monitored by the Gls resource), and the service is switched to the standby system when a failure of the whole system of the network occurs.

4. OS (global zone)

   A panic or a reset of the OS is immediately detected by the asynchronous monitoring, and the service is switched to the standby system. A hang-up of the OS in the global zone is detected by the cyclic monitoring of cluster interconnect (LAN) and the service is switched to the standby system.

5. OS (non-global zone)

   - Check for errors (login (zlogin command) is impossible) of the non-global zone, and the service is switched to the standby system.
   - For a single node cluster, the global zone of the PRIMECLUSTER restarts the non-global zone.

6. Service (cluster application)

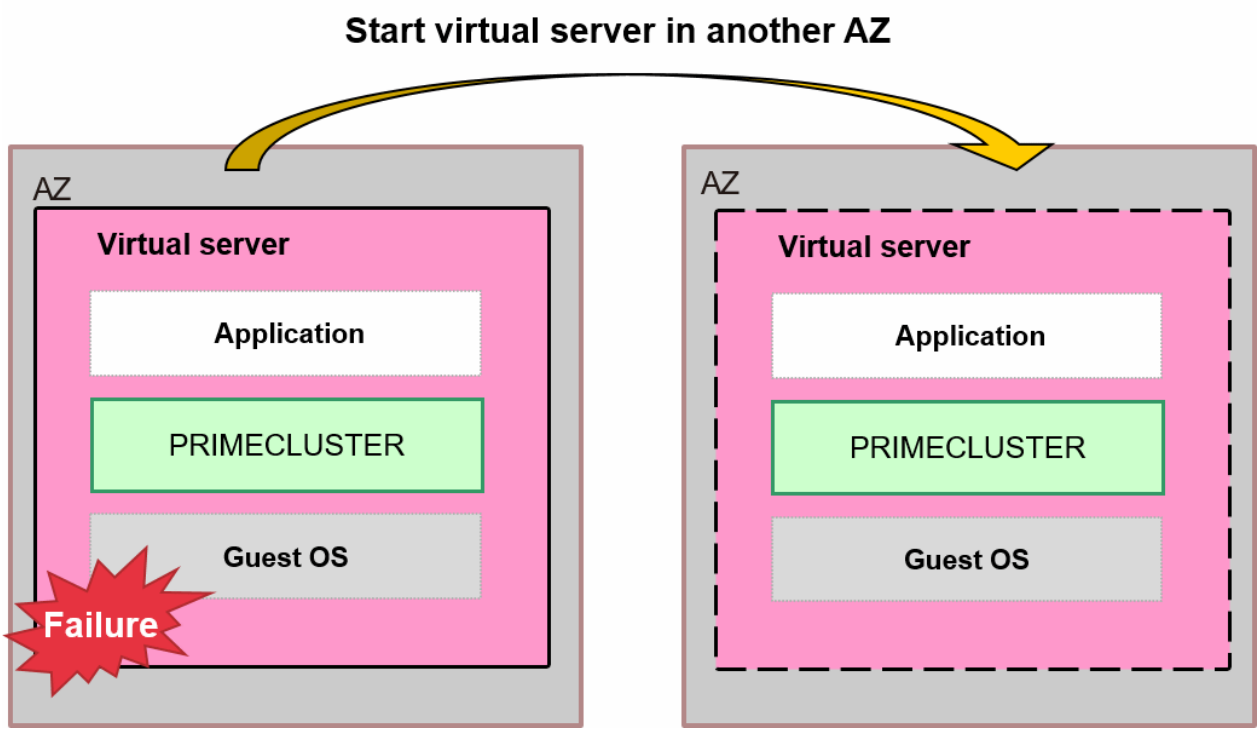
   When a resource error of the cluster application occurs, the service is switched to the standby system.

   For a single node cluster, restart the non-global zone.

1.9 Smart workload recovery

Smart workload recovery provided by PRIMECLUSTER Cloud Edition is a standby-less switching feature designed to provide low-cost operation and high availability to public cloud users. Detects an error in a virtual server deployed in a public cloud and starts the virtual server in a different availability zone.

By linking RMS, which is a module of PRIMECLUSTER, with the services provided by the public cloud, such as serverless infrastructure and resource monitor, the virtual server can be switched not only in the case of an abnormality of the virtual server that can be switched by the HA function provided by the public cloud as standard, but also in the case of an abnormal application or availability zone failure, and the availability of the application can be improved.



1.9.1 Error that can be detected or switched

This function detects the following errors and switches the virtual server.

- Application error
  - The PRIMECLUSTER (RMS) detects application error.
- Virtual server error
  - Abnormalities in the cloud infrastructure running virtual servers are detected by the functions provided by the public cloud (for example, CloudWatch).

When the above error occurs, the virtual server is started in a normal availability zone different from the availability zone in which the error occurred. This allows you to build fault-tolerant systems against public cloud infrastructure failures (Example: Power down per server rack, per VM host, etc.).

Note

If the cloud API used by PRIMECLUSTER becomes unavailable due to a public cloud failure, metadata (* 1) of the switching source virtual server may not be inherited by the switching destination virtual server at the time of switching. When this event occurs, immediately reset the metadata of the virtual server in order to restart the application on the virtual server to be switched to. Refer to "Chapter 8 Smart Workload Recovery Messages" in the "FUJITSU Software PRIMECLUSTER Messages" for details on the messages and actions to be taken.

*1 : For AWS, this includes instance metadata such as security groups and tags.

1.9.1.1 If an error occurs in the application

The following describes the behavior when an error occurs in an application.

1. Abnormalities occurred in application.

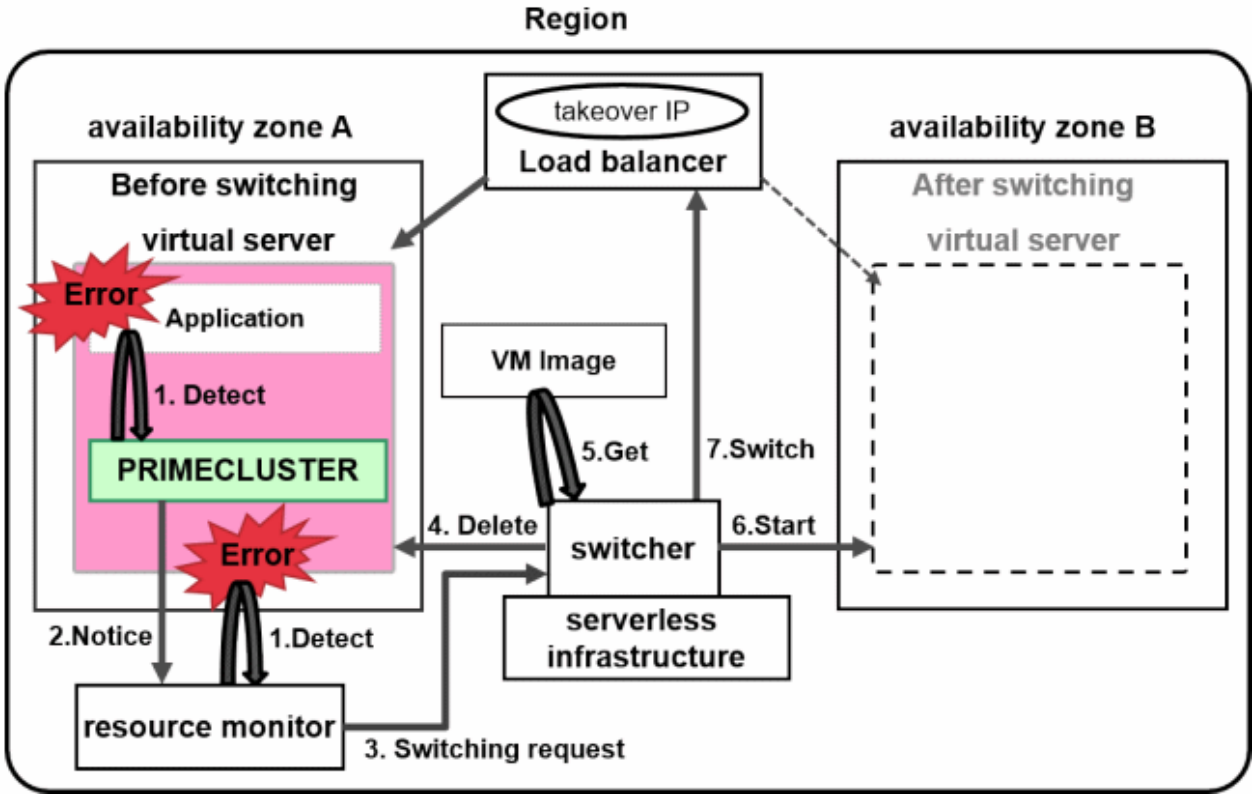
2. RMS detects an application error.
3. RMS notifies public cloud resource monitor when virtual server stops.
4. When the resource monitor of the public cloud receives an error, it requests the switcher on the serverless base to switch.
5. The switcher destroys the virtual server from which it was switched and starts the virtual server in any availability zone different from the one from which it was switched.

1.9.1.2 When an error occurs in the virtual server

The following describes the behavior when an error occurs in the virtual server.

1. An error occurred in the virtual server.
2. Abnormality detected by public cloud resource monitor.
3. Public cloud resource monitor requires switcher on serverless infrastructure.
4. The switcher destroys the virtual server from which it was switched and starts the virtual server in any availability zone different from the one from which it was switched.

Figure 1.29 Abnormal behavior



Chapter 2 PRIMECLUSTER architecture

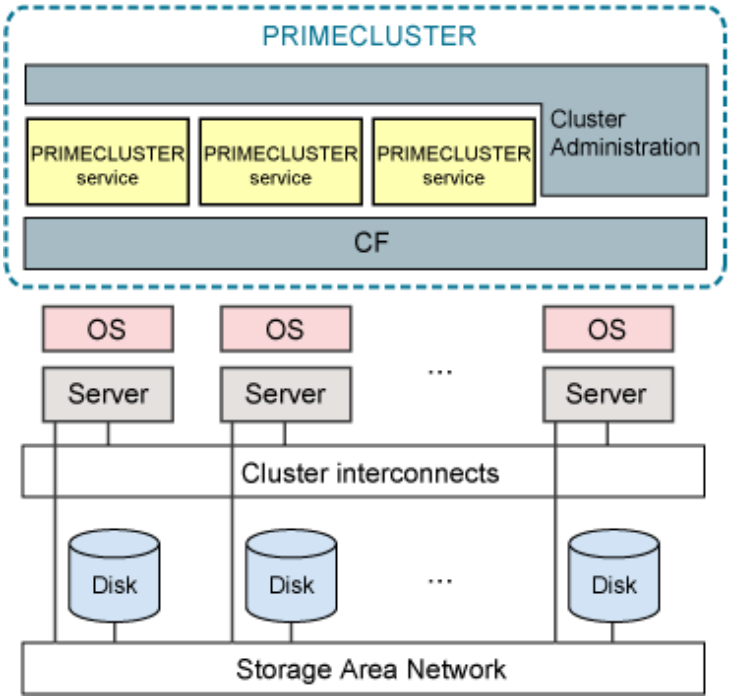
This chapter explains the PRIMECLUSTER architecture and discusses the key features that PRIMECLUSTER provides.

2.1 Architectural overview

The PRIMECLUSTER design is based on a long history of clustering and high availability (HA) software and hardware experience. The figure below is a conceptual model of a typical cluster design that illustrates PRIMECLUSTER's position as a middleware solution. Features of the PRIMECLUSTER solution include:

- PRIMECLUSTER is easily ported to new hardware platforms, operating systems, and cluster interconnects.
- PRIMECLUSTER provides services only for the management or use of the cluster.
- PRIMECLUSTER supplies interfaces so that other applications, such as enterprise management software, can interact with or call on services provided by PRIMECLUSTER.

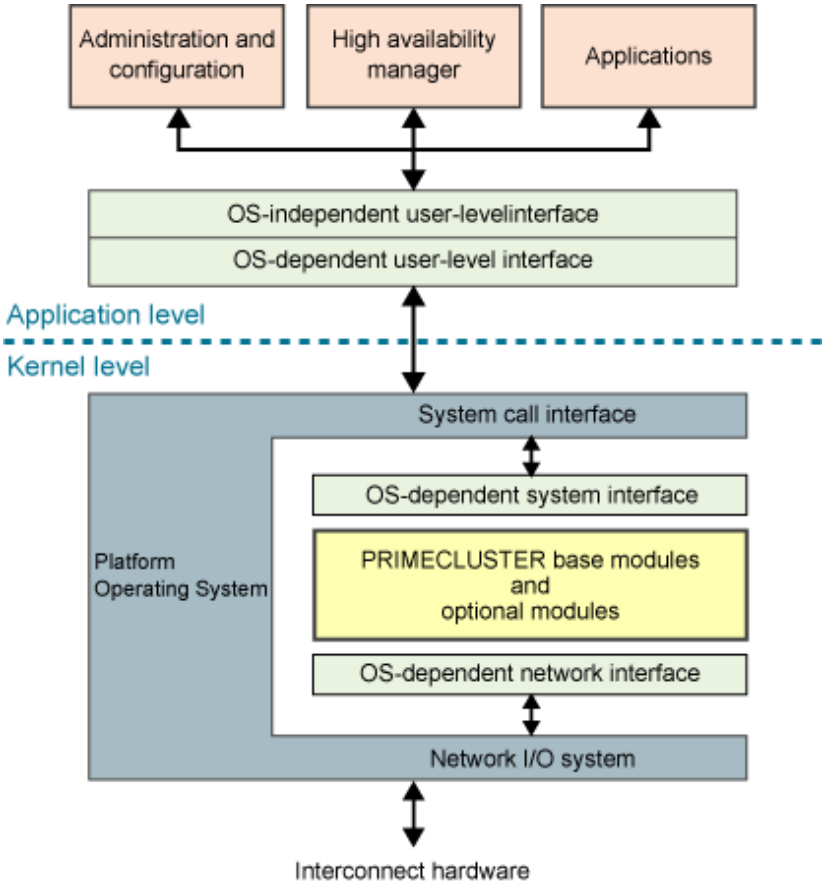
Figure 2.1 Diagram of a typical PRIMECLUSTER setup



The figure below shows a conceptual overview of the PRIMECLUSTER software architecture and how it interfaces with a server's native operating system. All of the PRIMECLUSTER software modules use operating-system-independent interfaces with an operating-system-dependant (OSD) layer to communicate between themselves and to access the base operating system services. Some examples of the services that the OSD provides are as follows:

- Memory allocations
- Synchronizations
- Device and network access

Figure 2.2 PRIMECLUSTER framework overview



2.2 PRIMECLUSTER key design features

The PRIMECLUSTER clustering software has been designed to satisfy the following goals:

- Modularity
- Platform independence
- Scalability
- Availability
- Guaranteed data integrity

2.2.1 Modularity

PRIMECLUSTER is composed of a core set of modules called the Cluster Foundation (CF) that provide the basic clustering functions. In addition, PRIMECLUSTER has optional modules such as the Parallel Application Services (PAS) module and the Reliant Monitor Services (RMS) module.

2.2.2 Platform independence

PRIMECLUSTER is independent of the operating system or hardware platform. Its modules are designed and coded based on an abstraction of an operating system's kernel functions. This is done with an OSD adaptation layer specific to different operating systems and network interconnects. This approach allows PRIMECLUSTER to plug seamlessly into the supported operating system without any modification to that operating system.

2.2.3 Scalability

PRIMECLUSTER provides not only high availability, but scalability as well. PRIMECLUSTER allows multiple nodes to act together to provide a common service. For example, with PAS, you can run a parallel database across multiple nodes. GFS provides you with a cluster wide file system so cooperating processes on multiple nodes can access the same data.

PRIMECLUSTER's scalability is important for customers who find that an application's demand for resources (especially CPU resources) has exceeded the capacity of a single machine. By clustering the nodes, more capacity is available for these types of applications.

2.2.4 Availability

PRIMECLUSTER implements a symmetric software architecture: All cluster information is fully replicated across nodes, thus avoiding single software points of failure. PRIMECLUSTER also supports multiple redundant interconnects, avoiding single hardware points of failure. RMS, the PRIMECLUSTER high availability manager, ensures the availability of applications by restarting or recovering applications on other nodes if there is a node failure. PRIMECLUSTER also includes an optional network load-balancing module (GLS) that can improve network availability.

If two or more points fail at the same time, a failover of the user application may not occur to protect data integrity.

2.2.5 Guaranteed data integrity

PRIMECLUSTER's algorithms guarantee that network partitions (or split-brain syndrome) even during multiple hardware interconnect failures do not result in data inconsistency problems. The quorum algorithms ensure that only one partial cluster can operate during a network partition.

2.3 PRIMECLUSTER components

The PRIMECLUSTER core component, Cluster Foundation (CF), provides the basic cluster services on which all other components are built. CF includes:

- Cluster Admin
  Supplies the interface for cluster configuration, administration, operation, and diagnostics services.
- Web-Based Admin View
  Provides a framework under which all the PRIMECLUSTER GUIs, including Cluster Admin, run.
- Cluster Resource Management (CRM)
  CRM conducts management of resource database that is synchronous in between each cluster nodes. Cluster resource database is a database exclusive for PRIMECLUSTER products.
- PRIMECLUSTER SF
  Provides a function to guarantee the shutdown of other nodes.

The optional components and products are as follows:

- Reliant Monitor Services (RMS)
  Manages high-availability switchover (failover) of application processes and their resources.
- RMS Wizard Tools
  RMS configuration tool foundations.

RMS Wizard Tools is mutually exclusive with respect to installation and operation.

- Parallel Application Services (PAS)
  Provides a high-performance, low-latency communication facility that can be used by applications.
- GDS
  Provides the volume management component that improves the availability and manageability of disk-stored data. Without using GDS, failover to the standby system is performed when the resource of the cluster application has an error due to the disk access error, not when the disk access error occurs. For this reason, failover to the standby system takes much time. Specifying the error location also requires much time.

- GFS
  Provides a file system that can be accessed from two or more nodes to which a shared disk unit is connected. (Only in Oracle Solaris 10 environment)
- GLS
  Enables high reliability communications through the use of multiple network interface cards to create multiple redundant paths to a local system. Without introducing GLS, the following functions are disabled: configuring the redundant network, using VLAN, and monitoring the standby node.

2.3.1 CF

The Cluster Foundation is the base on which all the other modules are built. It provides the fundamental services, such as the OSD, that all other PRIMECLUSTER components use as well as the base cluster services.

CF has the following features:

- Contains a loadable pseudo device driver that automatically loads when the system starts
- Supplies the CF driver that contains the CF kernel-level OSD and generic modules

Some of the functions that CF provides are detailed in the sections that follow.

CF uses the cluster interconnect to perform the control of existence monitoring of nodes and communication between nodes. For details on cluster interconnect, see "Chapter 3 Cluster interconnect details."

2.3.2 Cluster Admin

The Cluster Admin provides the following features:

- Construction
- Administration
- Operations and diagnostics services
- Using Cluster Admin, construction, administration and operation of cluster system can be done from any node within the cluster system. GUI serves as the administrative interface; a conventional command-line interface is also available on a node. Diverse, clear-reporting metrics and event logs provide concise and timely information on the state of the cluster.

2.3.3 Web-Based Admin View

Web-Based Admin View is a GUI framework used by the PRIMECLUSTER products. The features of Web-Based Admin View are as follows:

- Common framework for multiple GUIs

  In addition to the Cluster Admin GUI, which controls CF, RMS, and SF, PRIMECLUSTER contains GUIs for other services such as GDS and GFS. In the Web-Based Admin View, all of these GUIs operate as a common framework.
- A single login for multiple GUIs.
- Password encryption. Passwords sent from the client to the management server are encrypted.
- Logging of all GUI commands dealing with configuration or administration.
- The ability to off load the management overhead onto the management servers outside the cluster.

See

For additional information about Web-Based Admin View features, see "PRIMECLUSTER Web-Based Admin View Operation Guide."

2.3.4 Cluster Resource Management (CRM)

CRM conducts management of resource database that is synchronous in between each cluster nodes. Cluster resource database is a database exclusive for PRIMECLUSTER products. It is not a generic type of database that can be used on other applications.

CRM conducts matching of resource database in between nodes using CIP, and manages the resource database which is used by the component with in the PRIMECLUSTER to be identical on all nodes.

2.3.5 PRIMECLUSTER SF

The PRIMECLUSTER Shutdown Facility (SF) provides a function to guarantee that other nodes are shut down during error processing such as when contention for user resources occurs in a cluster system.

When CF confirms that a cluster node has restarted and can guarantee that the node was shut down before, PRIMECLUSTER SF does not shut down the node.

PRIMECLUSTER SF is made up of the following major components:

- Shutdown Daemon (SD)
  The SD monitors the state of cluster machines and provides an interface for gathering status and requesting manual machine shutdown.
- One or more Shutdown Agents (SA)
  The SA's role is to guarantee the shutdown of a remote cluster node.
- MA (asynchronous monitoring)
  In addition to the SA, the MA monitors the state of remote cluster nodes and immediately detects failures in those nodes.
  The route for forcibly stopping the cluster node is checked regularly (every 10 minutes).

Shutdown Agents (SA)

The SA guarantees a reliable suspension of the remote cluster node. The SA varies depending on the architecture of each cluster node.

The SA provides the following functions:

- Forcibly shutting down a failure node

  The SA guarantees to shut down a failure node.

- Checking a connection (Shutdown Agent testing)

  The SA periodically (every ten minutes) checks if it can properly connect to the optional hardware or the virtual machine function, each of which is used to forcibly shut down a node.

The PRIMECLUSTER Shutdown Facility provides the following Shutdown Agents:

- RCI (SA_pprcip, SA_pprcir): Remote Cabinet Interface

  This SA uses the RCI, which is one of the hardware units installed in SPARC Enterprise M-series, to stop other nodes with certainty by intentionally triggering a panic or reset in those nodes.

- XSCF (SA_xscfp, SA_xscfr, SA_rccu, SA_rccux): eXtended System Control Facility

  The SA uses the XSCF, which is one of the hardware units installed in SPARC Enterprise M-series, to stop other nodes with certainty by intentionally triggering a panic or reset in those nodes.
  If the XSCF is being used in the console, the Shutdown Facility stops other nodes with certainty by sending the break signal to those nodes.

- XSCF SNMP (SA_xscfsnmpg0p, SA_xscfsnmpg1p, SA_xscfsnmpg0r, SA_xscfsnmpg1r, SA_xscfsnmp0r, SA_xscfsnmp1r)
  eXtended System Control Facility Simple Network Management Protocol

  The SA uses the XSCF, which is one of the hardware units installed in SPARC M10, M12 to stop other nodes with certainty by intentionally triggering a panic or reset in those nodes.

- ALOM (SA_sunF): Advanced Lights Out Management

  The SA uses ALOM of SPARC Enterprise T1000, T2000 to stop other nodes with certainty by sending the break signal to those nodes.

- ILOM (SA_ilomp, SA_ilomr): Integrated Lights Out Manager

  The SA uses ILOM of SPARC Enterprise T5120, T5220, T5140, T5240, T5440, SPARC T3, T4, T5, T7, S7 series to stop other nodes with certainty by intentionally triggering a panic or reset in those nodes.

- KZONE (SA_kzonep, SA_kzoner, SA_kzchkhost)

  Oracle Solaris Kernel Zones

  If Oracle Solaris Kernel Zones are used with SPARC M10, M12 and SPARC T4, T5, T7, S7 series, the node can be completely stopped by intentionally panicking or resetting another node (Kernel Zone).

  The status of the global zone host is also checked so that when the global zone host is stopped, another node (Kernel Zone) is determined to be stopped. The global zone host is not forcibly stopped.

- BLADE (SA_blade)

  This SA, which can be used in the PRIMERGY blade server, uses the SNMP command to stop other nodes with certainty by shutting them down.

- IPMI (SA_ipmi): Intelligent Platform Management Interface

  This SA uses the IPMI to operate iRMC (integrated Remote Management Controller), which is one of the hardware modules installed in PRIMERGY, and stop other nodes with certainty by shutting them down.

- kdump (SA_lkcd)

  The SA uses kdump in PRIMERGY or the PRIMERGY blade server to stop other nodes with certainty by intentionally triggering a panic.

- MMB (SA_mmbp, SA_mmbr): Management Board

  This SA uses the MMB, which is one of the hardware units installed in PRIMEQUEST 2000, to forcibly stop other nodes with certainty by intentionally triggering a panic or reset in those nodes.

- iRMC (SA_irmcp, SA_irmcr, SA_irmcf)

  This SA uses iRMC / MMB, which are the hardware units installed in PRIMEQUEST 3000, to forcibly stop other nodes with certainty by intentionally triggering a panic, reset or shutting off the power in those nodes.

Note

This SA is not available in PRIMERGY iRMC.

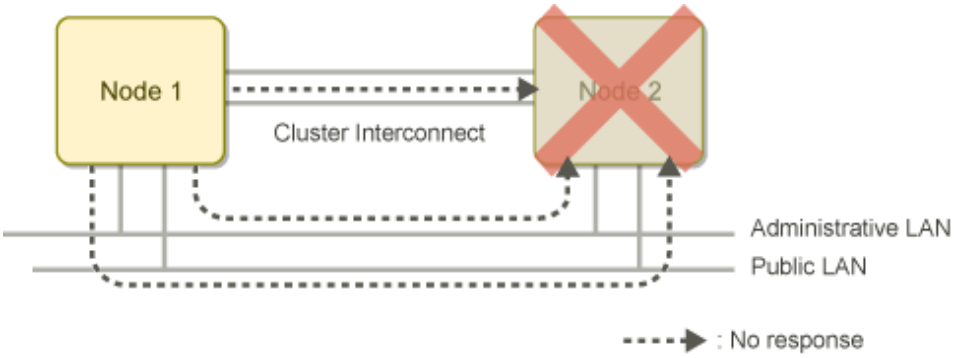
- ICMP (SA_icmp)

  The SA uses the network path to check the state of other nodes. If no response is received from other nodes, the SA determines that nodes are shut down.
  Other nodes are not forcibly shut down.
  If all network interfaces that are used to check the state of other nodes are stopped, the state of those nodes cannot be checked. Therefore, SA_icmp assumes that those nodes are running.

  The figure below shows an example of state confirmation by SA_icmp if one node (Node 2) goes down in a cluster system with two nodes.

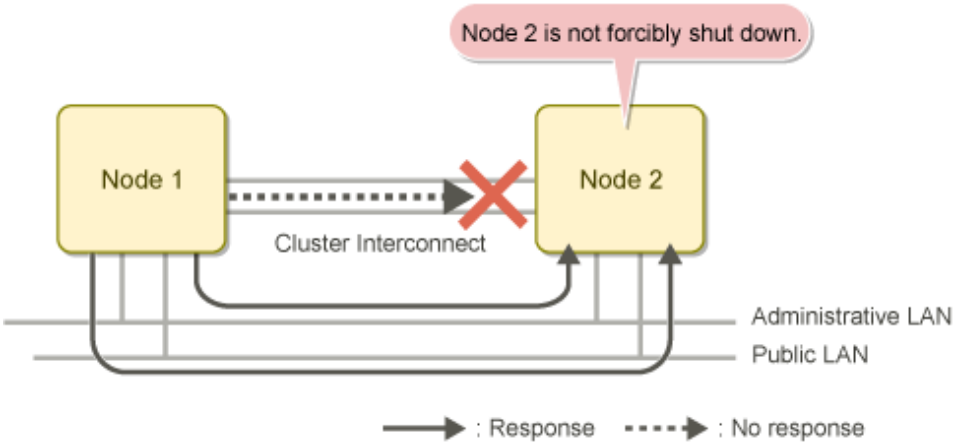
If no response is received from Node 2 through all specified network paths, SA_icmp determines that Node 2 is shut down.

Figure 2.3 State confirmation by SA_icmp if the other node goes down



The figure below shows an example of state confirmation by SA_icmp if the cluster interconnect fails in a cluster system with two nodes.
If Node 1 receives a response from Node 2 on any of specified network path, SA_icmp determines that Node 2 is running.
In this case, Node 2 is not forcibly shut down by SA_icmp.

Figure 2.4 State confirmation by SA_icmp if the cluster interconnect fails



- VMCHKHOST (SA_vmchkhost)

  When the cluster system is installed in the host OS with the KVM virtual machine function, the SA checks the status of guest OSes together with the cluster system of the host OS.
  Other nodes are not forcibly shut down.

- libvirt (SA_libvirtgp, SA_libvirtgr)

  When using the KVM virtual machine function in PRIMERGY, the PRIMERGY blade server, and PRIMEQUEST 3000/2000 series, the SA stops other nodes with certainty by intentionally triggering a panic or reset in those nodes.

- VMware vCenter Server functional cooperation (SA_vwvmr)

  Cooperating with VMware vCenter Server, the SA stops other nodes (guest OSes) with certainty by intentionally powering off.

- FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API (SA_vmk5r)

  This shutdown agent enables reliable node shutdown by intentionally shutting down or powering off the remote node (virtual server/ Bare Metal server) using the FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API.

- OpenStack API (SA_vmosr)

  This shutdown agent enables reliable node shutdown by intentionally restarting the remote node (instance) using the OpenStack API.

- AWS CLI (SA_vmawsAsyncReset)

  This shutdown agent enables reliable node shutdown by intentionally shutting down the remote node (instance) using the AWS CLI.

- Azure CLI (SA_vmazureReset)

  This shutdown agent enables reliable node shutdown by intentionally powering off or restarting the remote node (virtual machine) using the Azure CLI.

- NIFCLOUD API (SA_vmnifclAsyncReset)

  This shutdown agent enables reliable node shutdown by intentionally powering off the remote node (server) using the NIFCLOUD API.

MA (Monitoring Agent)

The Monitoring Agent (MA) has the capability to monitor the state of a system and promptly detect a failure such as system panic and shutdown. This function is provided by taking advantage of the hardware features that detect the state transition and inform the upper-level modules.

Without the MA, the cluster heartbeat time-out detects only a communication failure during periodic intervals. The MA allows the PRIMECLUSTER system to quickly detect a node failure.

The MA provides the following functions:

- Monitoring a node state

  The MA monitors the state of the remote node that uses the hardware features. It also notifies the Shutdown Facility (SF) of a failure in the event of an unexpected system panic and shutoff. Even when a request of responding to heartbeat is temporarily disconnected between cluster nodes because of an overloaded system, the MA recognizes the correct node state.

- Forcibly shutting down a failure node

  The MA provides a function to forcibly shut down the node as Shutdown Agent (SA).

- Checking a connection (Shutdown Agent testing)

  The MA provides a function as the SA (Shutdown Agent). It periodically (every ten minutes) checks if it can properly connect to the optional hardware that is used to monitor a node state or forcibly shut down a node.

PRIMECLUSTER SF provides the following Monitoring Agents:

RCI Monitoring Agents (SPARC Enterprise M Series)

The MA monitors the node state and detects a node failure by using the SCF/RCI mounted on SPARC Enterprise M-series. The System Control Facility (SCF), which is implemented on a hardware platform, monitors the hardware state and notifies the upper-level modules. The MA assures node elimination and prevents access to the shared disk.

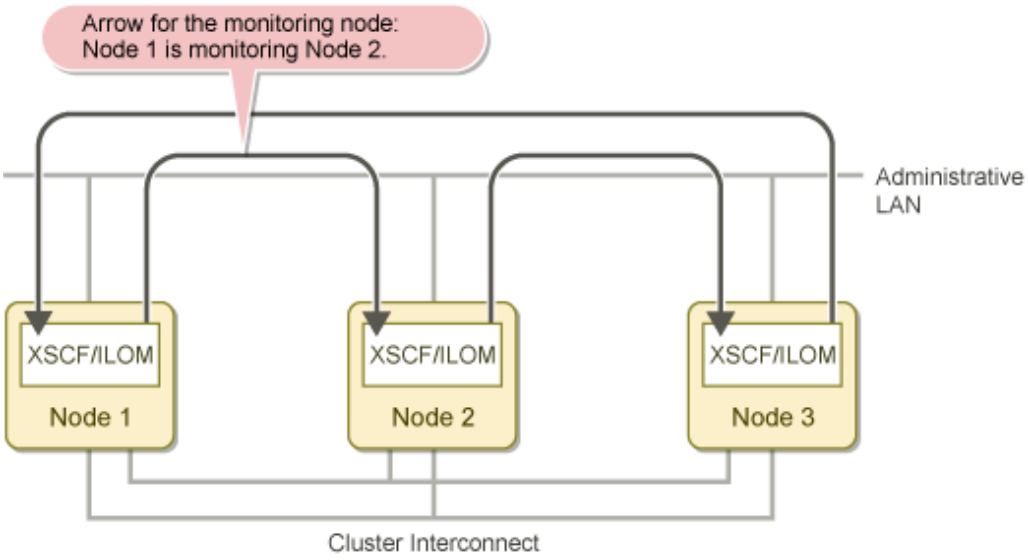
Console Monitoring Agents (Available server models are limited to SPARC Enterprise M-series and most of SPARC Enterprise T-series.)

The console monitoring agent monitors message output to the console of each node using XSCF/ILOM. If an error message of a node failure is output to one node, the other node detects the message and notifies SF of a node failure. Normally, the console monitoring agent creates a loop, monitoring another node, for example, A controls B, B controls C, and C controls A. If one node goes down because of a failure, another node takes over the monitoring role instead of this failed node.

The console monitoring agent also ensures node elimination by sending a break signal to the failed node.

The figure below shows how the monitoring feature is taken over in a cluster system with three nodes if one node goes down. The arrow indicates that a node monitors another node.

Figure 2.5 MA normal operation



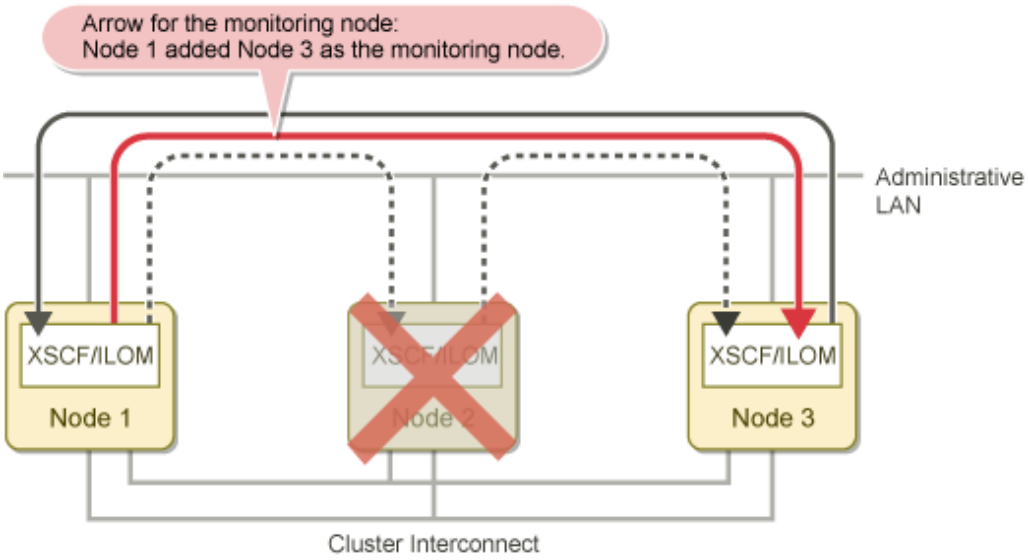
When a failure occurs, and Node 2 is *DOWN*, the following actions occur:

- Node 1 begins to monitor Node 3.
- The following message is output to the */var/adm/messages* file of Node 1:

  ```
  FJSVcluster: INFO: DEV: 3044: The console monitoring agent took over monitoring (node: targetnode)
  ```

The figure below shows how Node 1 added Node 3 as the monitored node when Node 2 went down.

Figure 2.6 MA operation in the event of node failure



Note

If monitoring function is taken over while the console monitoring agent is stopped, the stopped console monitoring agent is resumed.

When Node 2 recovers from the failure and starts, the following actions occur:

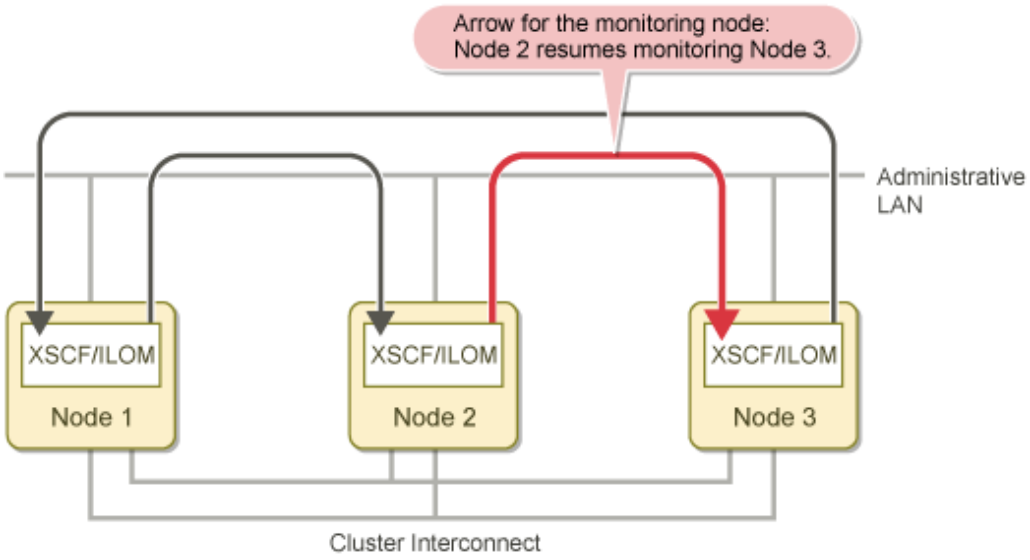
- The original monitoring mode is restored.

- The following message is output to the */var/adm/messages* file of Node 1:

  ```
  FJSVcluster: INFO: DEV: 3045: The console monitoring agent cancelled to monitor (node:
  targetnode)
  ```

The figure below shows how Node 2 returns to monitoring Node 3 once it has been restored to the cluster.

Figure 2.7 Node recovery



The following are possible messages that might be found in the */var/adm/messages* file:

- FJSVcluster: INFO: DEV: 3042: The RCI monitoring agent has been started
  Indicates that the RCI monitoring agent is enabled.
- FJSVcluster: INFO: DEV: 3043: The RCI monitoring agent has been stopped.
  Indicates that the monitoring feature is disabled.
- FJSVcluster: INFO: DEV: 3040: The console monitoring agent has been started (node:*monitored node name*)
  Indicates that the monitoring feature of the console monitoring agent is enabled.
- FJSVcluster: INFO: DEV: 3041: The console monitoring agent has been stopped (node:*monitored node name*)
  Indicates that the monitoring feature of the console monitoring agent is disabled. When the monitoring feature is not enabled, the other feature that forcibly brings the node *DOWN* might not work.

Note

The console monitoring agent monitors the console message of the remote node. So it cannot recognize the node state in the event of an unexpected shutdown. In such a case, the node goes into the *LEFTCLUSTER* state, and you need to mark the remote node *DOWN*. For how to mark a node with DOWN, see "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide."

SNMP asynchronous monitoring (SPARC M10, M12)

This function monitors the node state by using the eXtended System Control Facility (XSCF) installed in the SPARC M10, M12.

The function can ascertain node failures by having the XSCF report the node state to the software using SNMP (Simple Network Management Protocol).

This function can intentionally trigger a panic or a reset in other nodes to forcibly stop those nodes with certainty and prevent contention over user resources.

MMB asynchronous monitoring (PRIMEQUEST 2000)

This function uses the MMB, which is one of the hardware units installed in PRIMEQUEST 2000, to monitor nodes. The function can ascertain node failures by having the MMB, which is one of the standard units installed in the hardware, report the node state to the software.

This function can intentionally trigger a panic or a reset in other nodes to forcibly stop those nodes with certainty and prevent contention over user resources.

iRMC asynchronous monitoring (PRIMEQUEST 3000)

This function uses the iRMC and MMB, which are the hardware units installed in PRIMEQUEST 3000, to monitor nodes. The function can ascertain node failures by having the iRMC and MMB, the standard units installed in the hardware, report the node state to the software. This function can intentionally trigger a panic, reset or shutting off the power in other nodes to forcibly stop those nodes with certainty and prevent contention over user resources.

This SA is not available in PRIMERGY iRMC.

Node state monitoring of the RCI asynchronous monitoring function operates from when message (a) shown below is output until message (b) is output.
The messages for the console asynchronous monitoring function are messages (c) and (d).
The messages for the SNMP asynchronous monitoring function are messages (e) and (f).
The messages for the MMB asynchronous monitoring function are messages (g) and (h).
The messages for the iRMC asynchronous monitoring function are messages (i) and (j).
When node state monitoring is disabled, the function that forcibly stops nodes may not operate normally.

```
(a) FJSVcluster: INFO: DEV: 3042: The RCI monitoring agent has been started.

(b) FJSVcluster: INFO: DEV: 3043: The RCI monitoring agent has been stopped.

(c) FJSVcluster: INFO: DEV: 3040: The console monitoring agent has been started (node:monitored node name).

(d) FJSVcluster: INFO: DEV: 3041: The console monitoring agent has been stopped (node:monitored node name).

(e) FJSVcluster: INFO: DEV: 3110: The SNMP monitoring agent has been started.

(f) FJSVcluster: INFO: DEV: 3111: The SNMP monitoring agent has been stopped.

(g) FJSVcluster: INFO: DEV: 3080: The MMB monitoring agent has been started.

(h) FJSVcluster: INFO: DEV: 3081: The MMB monitoring agent has been stopped.

(i) FJSVcluster: INFO: DEV: 3120: The iRMC asynchronous monitoring agent has been started.

(j) FJSVcluster: INFO: DEV: 3121: The iRMC asynchronous monitoring agent has been stopped.
```

2.3.6 RMS

RMS is an application availability manager that ensures the availability of both hardware and software resources in a cluster with configuration of 2 or more nodes. This is accomplished through redundancy and through the ability to fail over monitored resources to surviving nodes.

For example, monitored resources can be almost any system component, such as the following:

- File system
- Volume (disk)
- Application
- Network interface

- Entire node

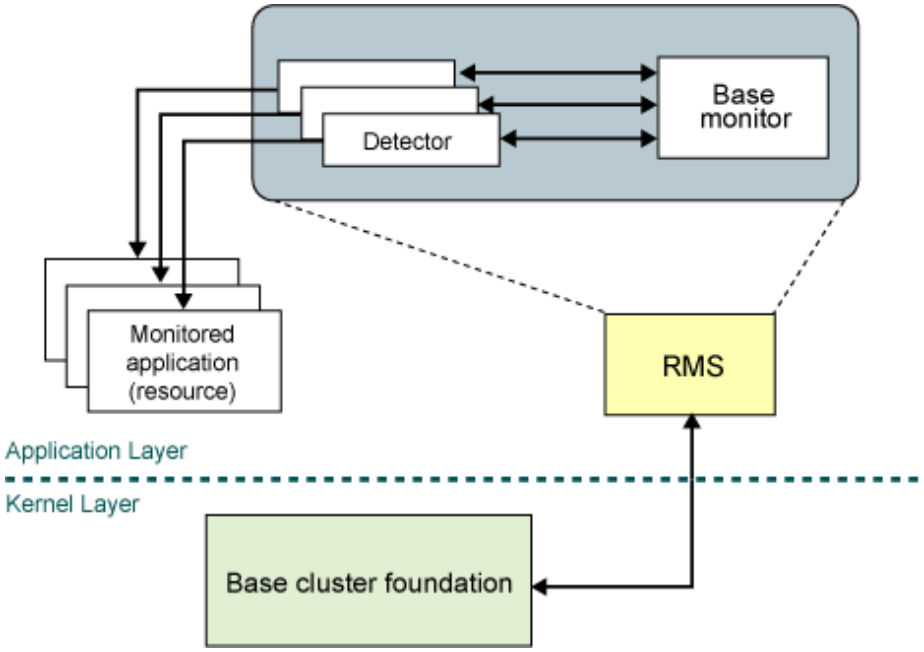
For redundancy, RMS uses multiple nodes in the cluster. Each node is configured to assume the resource load from any other node. In addition, RAID hardware and/or RAID software replicate data stored on secondary storage devices.

For application availability, RMS monitors resources with detector programs. When a resource fails, RMS triggers a user-defined response. Normally, the response is to make the resource available on other nodes.

Resources that are mutually dependent can be combined into logical groups such that the failure of any single resource in the group triggers a response for the entire group. During switchover, RMS ensures that all of a group's resources on the original node (before the failure) are brought offline prior to any resources being brought online on the new node. This prevents any possibility of data corruption by two or more nodes attempting to access a resource simultaneously.

The figure below shows how RMS uses detectors to monitor resources. A detector reports any changes in the state of a resource to the RMS base monitor, which then determines if any action is required.

Figure 2.8 RMS resource monitoring



2.3.6.1 RMS configuration tools

RMS configuration tools facilitate the creation of RMS configurations.

- RMS Wizard Tools
- This contains configuration tool foundations and interface with the RMS base. They simplify the configuration setup and work with RMS in the HA (high availability) cluster environment.

2.3.7 PAS

Parallel database applications were one of the first commercial applications to use cluster technology. By partitioning the workload and data across multiple nodes in a cluster, parallel database applications can achieve performance beyond the limits of a single, large multiprocessor server.

2.3.8 GDS

GDS is a volume management software that improves the availability and manageability of disk-stored data. GDS protects data from hardware failures and operational mistakes, and supports the management of disk units.

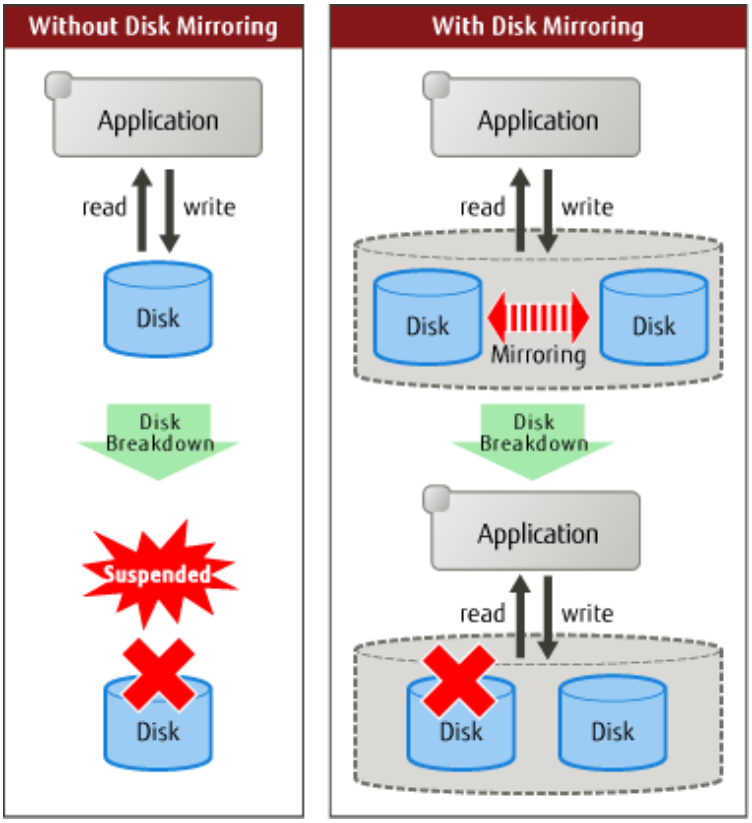
GDS has following functions, which are closely related:

- To improve availability of disk data

- To improve manageability of disk data

GDS's mirroring function protects data from hardware failures by maintaining replicas of disk data on multiple disks. This allows users to continue to access the disk data without stopping the application in the event of an unexpected trouble.

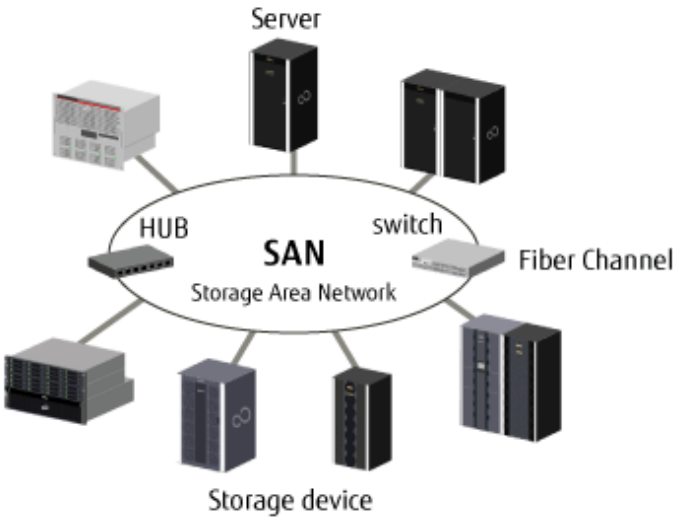
Figure 2.9 Disk Mirroring



The GDS management functions reduce the system administrator's workloads of disk management. The user-friendly functions simplify management, and at the same time, prevent data corruption by operational mistakes.

In a SAN (Storage Area Network) environment, multiple servers can be connected to multiple disk units (see the figure below.). Disk-stored data can be accessed from those servers. This allows simultaneous access to a file system or database and improves the efficiency of data duplication between the servers and backup procedures. On the other hand, it also carries the risk of data damage, as multiple servers will compete to access the shared disk. Therefore, volume management functions suitable for the SAN environment are essential.

Figure 2.10 SAN (Storage Area Network)



GDS provides volume management functions suitable for SAN environments. GDS allows users to integrate management of all disk units connected to all servers, including local disk units that are connected to specific servers as well as disk units that are shared by multiple servers in a SAN environment.

Note

Except for a VMware environment, GDS cannot manage system disks in PRIMERGY.

GDS's main functions include:

- Mirroring of system disks (Solaris server, PRIMEQUEST, and VMware environment)
- Mirroring of shared disks
- Mirroring of disk array units
- Mirroring among servers which mirrors local disks of an operating node and a standby node through a network (Linux server)
- Hot spare that automatically recovers mirroring in the event of a disk failure
- Hot swap that replaces a failed disk without stopping applications
- JRM (Just Resynchronization Mechanism) that pursues high-speed recovery of mirroring in the event of an unexpected system down or cluster failover
- Integrated management and access control of disks in a SAN environment
- Automatic configuration that recognizes physical connections to servers and disk units, registers the configuration information, and checks the connections (Solaris server only)
- Concatenation that enables creation of large volumes of data
- Striping that distributes load of access to disks
- Logical partitioning that enables flexible use of disks
- Snap-shot that supports backup minimizing the effect on core services

See

See "PRIMECLUSTER Global Disk Services Configuration and Administration Guide" for further details.

2.3.9 GFS

GFS provides the GFS Shared File System which can be simultaneously shared with multiple nodes.

Solaris version can be used with Solaris 10.

2.3.9.1 GFS Shared File System

The GFS Shared File System is a highly reliable file system, assuring simultaneous access from multiple nodes to which a shared disk unit is connected (up to 2 nodes).

The GFS Shared File System provides the following features:

- Simultaneous shared access to a file or file system from two or more nodes
- Data consistency and integrity when file data is referenced or updated by two or more nodes
- When a node is down, file operation can be continued on the other node while maintaining the consistency of the file system
- File system high-speed recovery function
- High-speed input/output (I/O) processing with sequential block assignment of file area
- File access using file cache of each node

- Multi-volume supports the distribution of input/output processing load and the use of large-scale file systems
- Addition of online volume (area extension) is possible without disrupting or reconfiguring the file system
- Creation, deletion, expansion, and operation of the file system can be performed using GUI

Simultaneous shared access while maintaining consistency

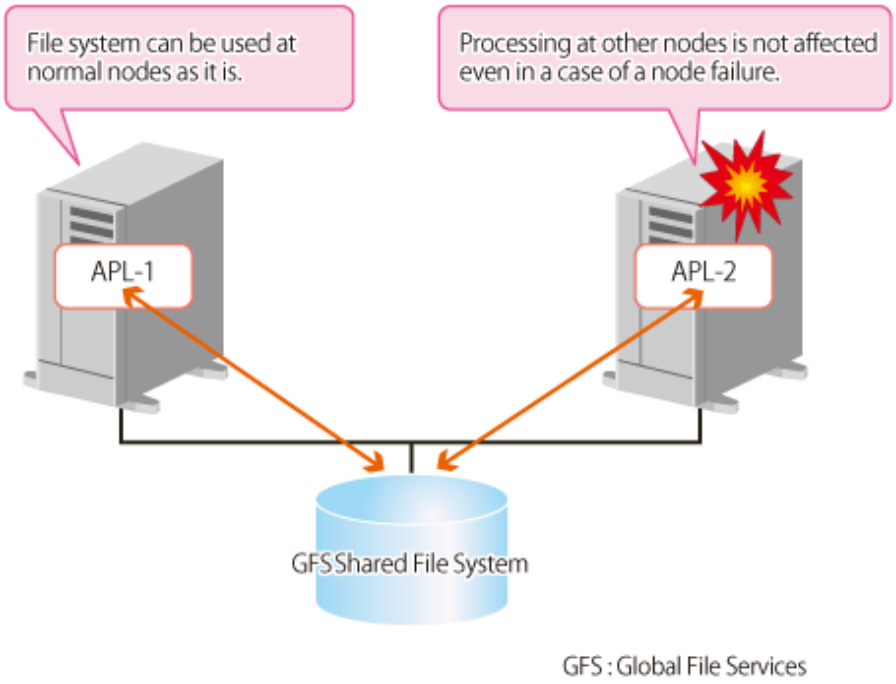
The GFS Shared File System assures integrity of data when data is updated from multiple nodes. The file-lock function is enabled on across nodes using a conventional API of UNIX file system. When a distributed application is executed on multiple nodes, the conventional API of UNIX file system is used ensuring application data transfer.

High availability

When the GFS Shared File System is used from multiple nodes, file access can be continued from the other node even if a node is down. The file system information retained by the downed node automatically restores the consistency within the GFS on the other nodes. That is, the application program running on the other nodes can continue processing without causing an error in the file system operation.

The operations required to change file system structure (such as file creation or deletion) are recorded in the area called the update log with the GFS Shared File System. By using the information stored in this area, a system failure can be recovered in seconds without having to make a full check of the file system structure.

Figure 2.11 Operation continuation when a node is down



GFS : Global File Services

Data accessibility performance

GFS enables a file system on a shared disk unit to be accessed from two or more nodes. With a conventional distributed file system, data is transferred to the client that requested access from the server on which file system data is managed by means of network communication over a LAN. The requested node directly accesses the disk unit. This reduces the network load from NFS and speeds up the response time required to read or write the request.

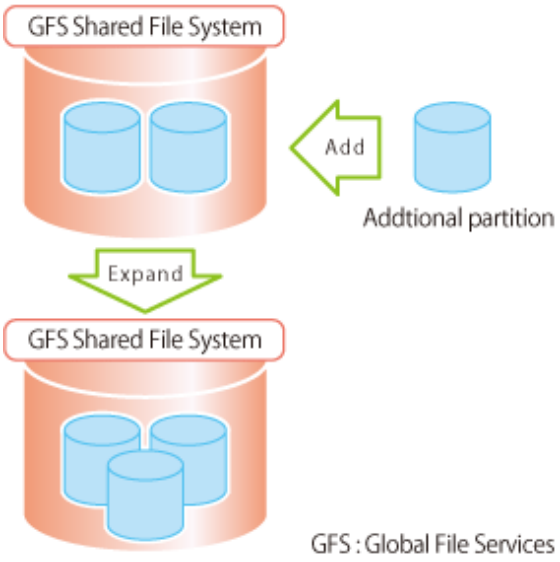
By allocating sequential blocks to the file data, the GFS Shared File System enables collective I/O processing to improve the file system performance.

The GFS provides a feature that integrates multiple partitions into one file system. In the case a configuration with multiple partitions, the round-robin allocation method is used, so the file data area can be used from different partitions for each file. Therefore, the I/O load can be distributed into multiple disk units and the file system performance is improved. This function makes it easy to add the data partition described afterwards.

Scalability

With the GFS Shared File System, the file system can be easily expanded by specifying an empty disk partition. Thus, a shortage of free space in a file system can be solved in a short time.

Figure 2.12 Additional partition



2.3.9.2 Benefits

GFS has the following benefits:

- The use of file cache on each system and high speed data access not using the LAN improve access performance.
- CPU load distribution of application is enabled via files on two or more systems assuring data integrity.
- A high availability system is provided by continuing file access on other node when the current node goes down.
- Area management on extent base and multi-volume support provides high-speed file access.
- Multi-volume supports the use of large-scale file systems and addition of online volume (area extension) is enabled without disrupting or reconfiguring the file system, and it makes the resource management easy.
- Operation of the file system using GUI eases environment configuration and management.

2.3.10 GLS

GLS is a software product that enables high reliability communications through the use of multiple network interface cards (NICs) to create redundant multiple transfer routes to a local system. Global Link Services provides network solutions that are suitable for systems in which communications continuity is important.

The benefits of Global Link Services are as follows:

- Multiple NICs can provide redundant transfer routes to offer high availability and path failure protection.
- The use of GLS means that applications can continue to run even in the event of changes in the LAN configuration, the redundant transfer route, or any network fault that may occur in a transfer route.

GLS provides the following two functions:

- Redundant Line Control Function
  - Fast switching mode
    Multiplexing transfer routes between servers on the same network.
  - NIC switching mode
    Configuring redundant transfer routes between servers and switches/HUBs on the same network.

- Virtual NIC mode (Solaris)
  Configuring redundant transfer routes between servers and switches/HUBs on the same network. By using the redundant route in the virtualization environment (guest domain or non-global zone), the virtual server is effectively aggregated.
- Virtual NIC mode (Linux)
  Configuring redundant transfer routes between servers and switches/HUBs on the same network. By using the redundant route in the virtualization environment (guest OS of the KVM virtual machine function), the virtual server is effectively aggregated.
- GS/SURE linkage mode (Solaris) and GS linkage mode (Linux)
  Multiplexing transfer routes among the server on the same network, global server/SURE SYSTEM, and ExINCA.

- Multipath function
  - Multipath mode (Solaris)
    Multiplexing transfer routes between servers and switches on the same network.
  - Multilink ethernet mode (Solaris)
    Distributing transfer data load on the multiplexed transfer route between servers and switches on the same network.

See

For details on the functions of Global Link Services, see "PRIMECLUSTER Global Link Services Configuration and Administration Guide: Redundant Line Control Function," "PRIMECLUSTER Global Link Services Configuration and Administration Guide: Redundant Line Control Function for Virtual NIC Mode," and "PRIMECLUSTER Global Link Services Configuration and Administration Guide: Multipath Function." The manual of virtual NIC mode and the manual of the multipath function are for Solaris only.

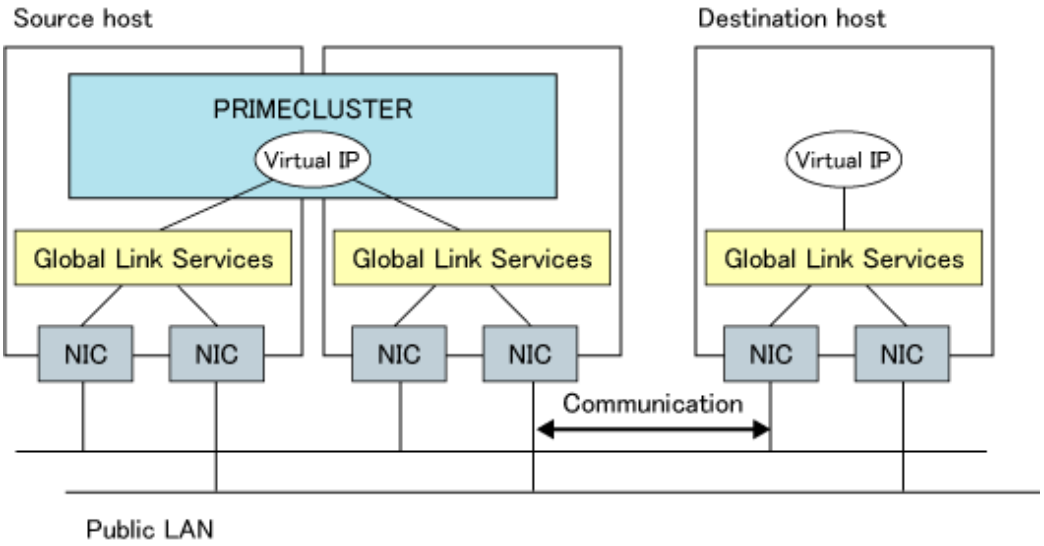
2.3.10.1 Fast switching mode

A redundant transmission route between Solaris servers or Linux servers in the same network is used so that the total amount of data transferred can be increased, and that the data communication can be continued even if the transmission route fails.

It also enables higher levels of throughput through redundant transmission routes. GLS performs early failure detection, so when one transmission route fails, the failed route will be cut off then the system will be operated on a reduced scale. The compatible hosts are SPARC Servers, PRIMEPOWER, GP7000F, Fujitsu S series, GP-S, PRIMERGY, and PRIMEQUEST.

Note that fast switching mode cannot be used to communicate with hosts on the other networks beyond the router.

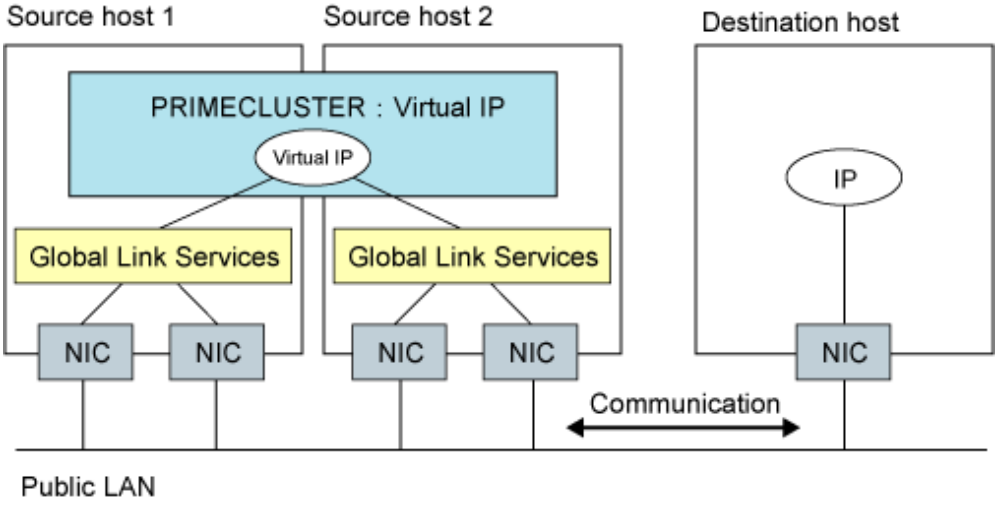
Figure 2.13 Fast switching mode



2.3.10.2 NIC switching mode

Redundant NICs (LAN cards) are connected to each other on the same network and used exclusively to control the switch of transmission route. There are no restrictions on remote systems to communicate with via router.

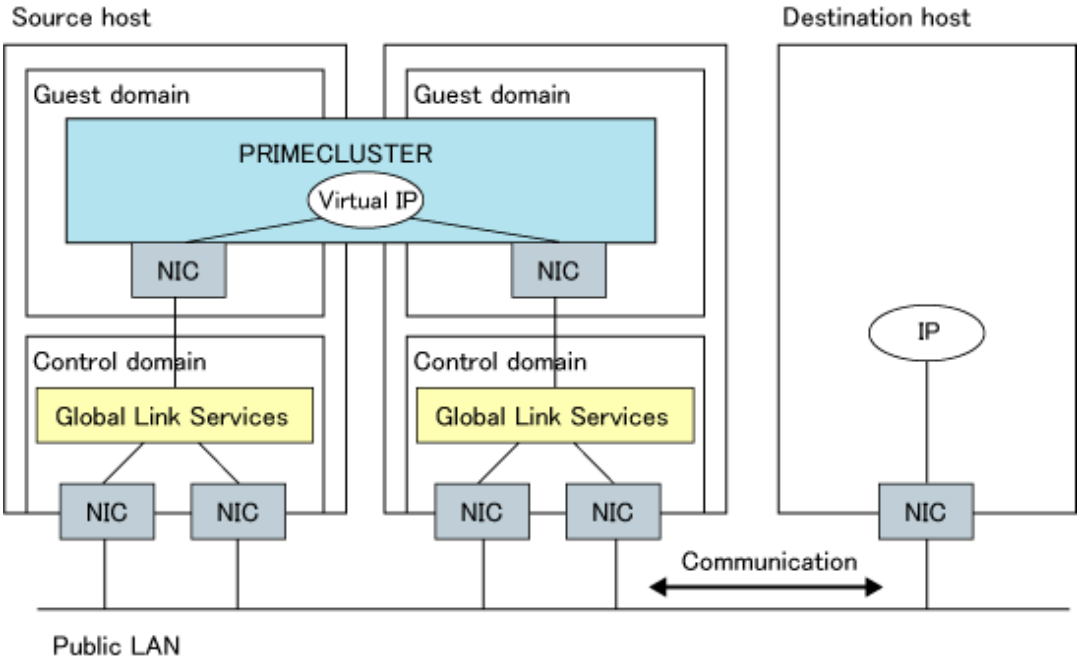
Figure 2.14 NIC switching mode



2.3.10.3 Virtual NIC mode (Solaris)

Virtual NIC mode (Solaris) enables communication with multiple physical NICs (LAN cards) that are connected on the same network, in order to show a single logical generating virtual interface. There are no restrictions on the devices that can be connected, and it is possible to communicate with hosts on other networks via the router. For Oracle VM environment, using the virtual interface created on the control domain can be used to communicate with the guest domain.

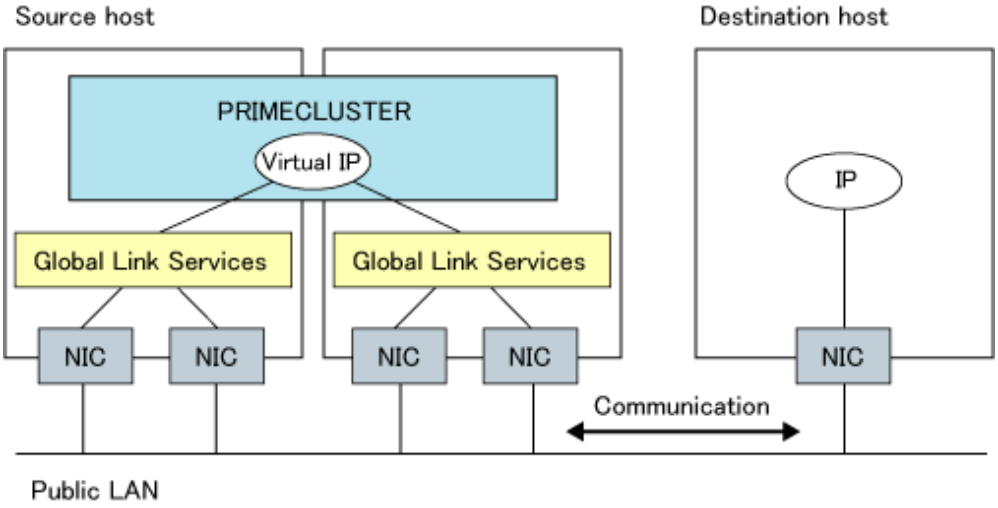
Figure 2.15 Virtual NIC mode (Solaris)



2.3.10.4 Virtual NIC mode Linux

Virtual NIC mode (Linux) enables communication with multiple physical NICs (LAN cards) that are connected on the same network, in order to show a single logical generating virtual interface. In this mode, switching of transfer route is controlled using redundant NIC. There are no restrictions on the devices that can be connected, and it is possible to communicate with hosts on other networks via the router.

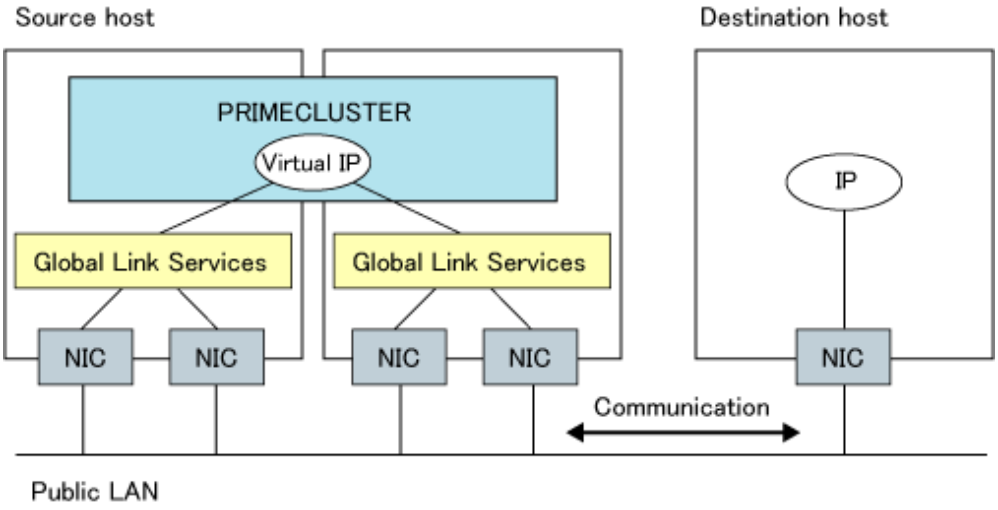
Figure 2.16 Virtual NIC mode (Linux)



2.3.10.5 GS/SURE linkage mode (Solaris), GS linkage mode (Linux)

Enables the system to control transfer route by using a Fujitsu method for high-reliability communication between the system and Global Server. In this mode, redundant lines are used simultaneously. During normal operation, transfer routes are automatically assigned to each TCP connection for communication. In the event of an error, the system disconnects the faulty transfer route and operates on a reduced scale by moving the TCP connection to the normal transfer route.

Figure 2.17 GS/SURE linkage mode (Solaris), GS linkage mode (Linux)



Chapter 3 Cluster interconnect details

This chapter discusses the differences between a cluster interconnect and a network as well as the requirements of the PRIMECLUSTER cluster interconnects.

3.1 Overview

The cluster interconnect is the most fundamental part of the cluster. All of the cluster's services rely on the cluster interconnects to transport messages between nodes and to determine the state of a node by its ability to respond to heartbeat requests.

3.1.1 A cluster interconnect is different from a network

The function of a cluster interconnect is different than the traditional use of a network in several ways; therefore, it is useful to think of them as separate technologies. The highest priority use of the cluster interconnects is to carry heartbeat requests and responses.

Heartbeat messages are used to determine the state of the nodes and the cluster interconnects. When a message fails to get through, the cluster software assumes that a failure has occurred and takes action to recover. However, no network is 100 percent reliable, and the PRIMECLUSTER ICF protocol tolerates errors such as lost packets or out of order delivery.

In the cluster interconnects, the physical connections are redundant so that if one fails, one or more remain to carry the messages. However, a sustained outage of all cluster interconnects results in the cluster management software taking action to recover as if a node had failed.

3.1.2 Interconnect protocol

The ICF (Internode Communication Facility) protocol used by PRIMECLUSTER is designed specifically for cluster communications. It is a low-latency protocol that guarantees ordered delivery of messages. Overhead of ICF will be less than TCP/IP. ICF uses the Ethernet protocol or is a service over IP (CF/IP).

Note

- Devices that only accept TCP/IP cannot route the ICF protocol when it is configured directly on the Ethernet. In this instance, if you need to use a router, it must be a level-two router.
- ICF can only be used in the internal components of CF and cannot be used at the user lever resources. Cluster interconnect protocol (CIP) is used for applications that access the cluster interconnect. CIP provides a standard TCP/IP protocol on the ICF.

3.2 Cluster interconnect requirements

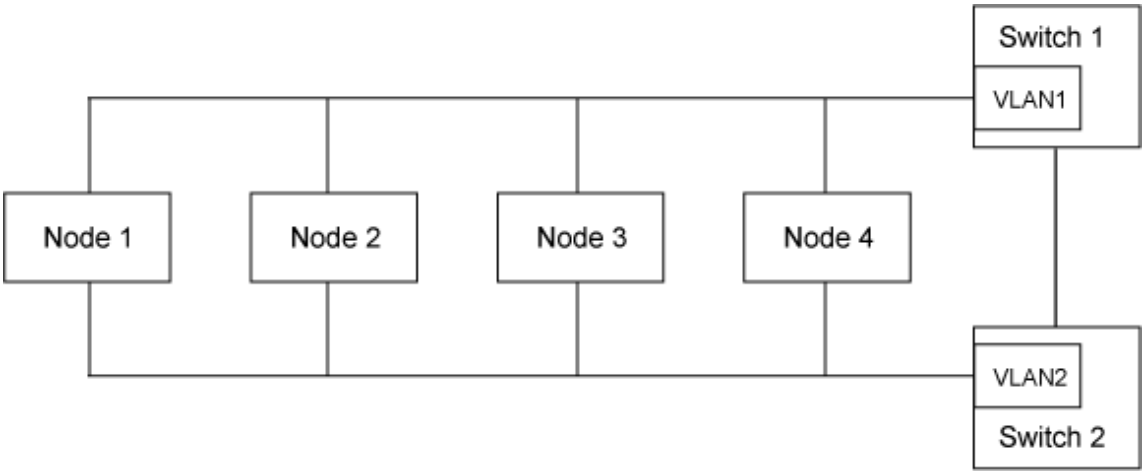
PRIMECLUSTER uses all significant forms of Ethernet and most devices that support TCP/IP, including mixtures of different technologies. The cluster interconnects must be redundant to assure reliable operation of the cluster; that is, there must be two or more independent connections between all of the nodes in the cluster.

When connecting multiple switches and routers by using cluster interconnects, use VLAN to detach each switch or router logically.

Example:

When connecting multiple switches as the following figure, use VLAN to detach each switch logically.

Figure 3.1 Connecting multiple switches



Each interconnect must support all of the nodes in the cluster.

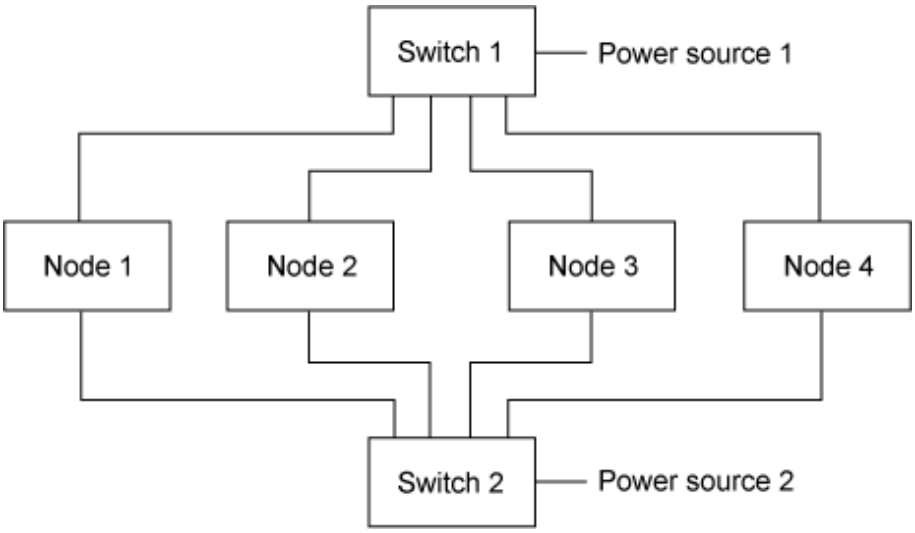
3.2.1 Redundancy

To be redundant, the cluster interconnect must use two or more independent connections and data paths. An example of a redundant cluster interconnects is as follows:

- Only one port is used for each Ethernet board. If more than one port of the same board is used as a cluster interconnect, these ports would be a common point of failure.
- Connections to nodes must be independent.
- No two cluster interconnects can share the same hub, switch, or routers.

The figure below shows a typical four-node cluster. In this diagram, there are two switches. Each of these switches has their own power connection on different circuits (when connecting two switches, it is necessary to use VLAN to detach each switch logically). If the switches were connected to the same power strip in a rack, for example, the power strip would be a single point of failure for the cluster interconnects.

Figure 3.2 Typical four-node cluster



3.2.2 Routes

Since redundant cluster interconnects are required for reliability, ICF is designed to use all of the available bandwidth. Each connection between two nodes in the cluster is called a route. When ICF has a message to send to another node, it chooses a route so that the message traffic is balanced between all the available routes. In the four-node cluster shown in Figure 3.2 Typical four-node cluster, each node has two routes to each of the other nodes.

3.2.2.1 Heartbeats

When everything is functioning properly in the cluster, all of the routes are in the *UP* state (available for use). Every *UP* route participates in carrying message traffic if it functions at the same speed (mixed speed cluster interconnects are discussed below.) In addition, every route is always used for heartbeat requests. (A heartbeat is a message indicating that a node is functional.) If a certain number of heartbeat requests fail on a route, then that route is marked as *DOWN*. A *DOWN* route is never used for message traffic. However, heartbeat requests are still attempted on the *DOWN* routes, and if one succeeds, the route is returned to the *UP* state. This last behavior is sometimes called self-healing routes. The use of all the *UP* routes for message traffic is called port aggregation or trunking.

Failures of heartbeat requests can come from several causes. When a network component fails, that route is not usable and the behavior is the same as described in the previous paragraph. When a node is *DOWN* to the only remaining route to another node, PRIMECLUSTER does not mark this route as *DOWN*, rather it marks the node as *LEFTCLUSTER*.

LEFTCLUSTER is a node state that indicates that the node cannot communicate with other nodes in the cluster. That is, the node has left the cluster. Once a node is marked as *LEFTCLUSTER* or *DOWN*, no more heartbeat attempts are made on any of the routes to the node until the node rejoins the cluster.

Note

The last route to a node is never marked *DOWN*, even if the node is in the *LEFTCLUSTER* or *DOWN* state.

Sometimes, events happen on a node that prevents the node from responding to heartbeat requests for some reason other than a failure. For example, a Fibre Channel controller can prevent other network device drivers from executing while attempting a link recovery. This can result in a false detection of node failure. To allow for some flexibility in responding to this condition, the number of missed heartbeat requests on the final route to a node before declaring the node as *LEFTCLUSTER* is tunable.

You can only adjust the number of failed-heartbeat requests that occur before declaring the node *LEFTCLUSTER*. The other parameters for ICF are not tunable. It is important that the route-detection algorithm marks routes as down as soon as possible, so that messages are switched to alternate routes without a noticeable delay. If a route is momentarily not available, for whatever reason, the self-healing mechanism will quickly reactivate the route after it becomes functional. "PRIMECLUSTER Cluster Foundation (CF) Configuration and Administration Guide" has details on tuning the ICF parameters.

PRIMECLUSTER also supports the use of non-symmetric cluster interconnects, for example, one cluster interconnect could use gigabit Ethernet and a second fast-Ethernet. When making decisions about routing, the cluster interconnect speed is also considered. In the previous example, the gigabit Ethernet would always be used in preference to the fast-Ethernet whenever it is available. If there are multiple cluster interconnects at the same speed, the port aggregation is done across all the devices with the same speed. Heartbeat requests are always sent on every cluster interconnect, independent of the cluster interconnect speed.

3.2.3 Consideration of items during design

When designing cluster interconnects, it is important for users to consider the following:

- Bandwidth
- Latency
- Reliability
- Device interface
- Security

3.2.3.1 Bandwidth

PRIMECLUSTER does not require much bandwidth for its own use. PRIMECLUSTER requires less than 0.002 Mbps on each of the cluster interconnects.
For this reason, there is no need to consider the bandwidth for the following conditions:

- When the cluster interconnect is not sharing one bandwidth with the public LAN or the administrative LAN
- When the user application does not conduct communication using the cluster interconnect

Refer to the table below as an example of bandwidth use. Suppose that in the configuration shown in Figure 3.2 Typical four-node cluster that there are two 100 Mbps Ethernets configured for the cluster interconnect. Assume that the available bandwidth for each cluster interconnect is 80 Mbps, and assume that the end-user application needs 36 Mbps on each node for the cluster file system and other activities. (This is an example. The actual bandwidth used by an application varies, depending on the application.)

Table 3.1 Example of cluster interconnects with two 100 Mbps Ethernet boards

Item	Bandwidth	Total bandwidth
100 Mbps Ethernet x 2	80 Mbps	160 Mbps (= 2 Interconnects x 80 Mbps)
PRIMECLUSTER requirements	0.002 Mbps	0.016 Mbps (= 4 Nodes x 2 Interconnects x 0.002 Mbps)
User application requirements	36 Mbps	144 Mbps (= 36 Mbps x 4 Nodes)

```
Total use = (PRIMECLUSTER requirements + User application requirements) / Total bandwidth of 100 Mbps
Ethernet boards x 100
= (0.016 + 144) / 160 x 100 = 90%
```

For this example, two fast-Ethernet interconnects use over 90 percent of the bandwidth.

Note

It is recommended that an initial installation has at least 30 percent available bandwidth capacity because the latency of the cluster interconnect increases and it may cause a false detection of heartbeat failure when total use nears 100 percent.

For this example, workload and configuration, one additional fast-Ethernet interconnect should be added to provide the excess capacity. The table below shows the same calculation with this addition.

Table 3.2 Example of cluster interconnects with three 100 Mbps Ethernet boards

Item	Bandwidth	Total bandwidth
100 Mbps Ethernet x 3	80 Mbps	240 Mbps (= 3 Interconnects x 80 Mbps)
PRIMECLUSTER requirements	0.002 Mbps	0.024 Mbps (= 4 Nodes x 3 Interconnects x 0.002 Mbps)
User application requirements	36 Mbps	144 Mbps (= 36 Mbps x 4 Nodes)

```
Total use = (PRIMECLUSTER requirements + User application requirements) / Total bandwidth of 100 Mbps
Ethernet boards x 100
= (0.024 + 144) / 240 x 100 = 60%
```

This new configuration gives a comfortable 40 percent available bandwidth margin, which means that required margins (30 percent or more) are secured. PRIMECLUSTER supports a maximum of four cluster interconnect devices. In the example above, triple redundant cluster interconnects are used.

3.2.3.2 Latency

As previously stated, PRIMECLUSTER relies on heartbeat requests and responses to determine that nodes or other resources are functional. When a heartbeat is not received in a preset interval, PRIMECLUSTER starts recovery actions. The Cluster Foundation (CF) software on each node sends a heartbeat request every 200 ms on each interconnect to every other node in the cluster. A heartbeat request is sent 50 times in every 200 ms until the timeout period (default: 10 seconds). If there is no response from the target node, CF will mark that node as *LEFTCLUSTER*.

The 200 ms interval is a reasonable design for a maximum latency in the cluster interconnects. This interval is long enough so that a small message and response can span transcontinental distances. This interval is also fixed and cannot be changed.

3.2.3.3 Reliability

Ethernet as an interconnect technology has not shown any problems with PRIMECLUSTER. The communications protocol used by PRIMECLUSTER is ICF. ICF guarantees that messages are delivered correctly and in order to its clients. However, ICF was designed with fairly reliable communications in mind. When the cluster interconnect is reliable, ICF has very low overhead, but when it is unreliable, the overhead of ICF increases. This is similar to other protocols like TCP/IP; errors in the cluster interconnect will result in messages being resent.

Note

- Resending messages consumes the bandwidth while it also affects the length of response wait time. For these reasons, in order to avoid resending messages, the use of high reliable cluster interconnect is important.
- An Ethernet error rate greater than 1 error per 1,000,000 bytes indicates that there is some problem with the Ethernet layer that should be investigated. (Use the command *netstat*(1) or *ip*(1)to find the error rate.)

3.2.3.4 Device interface (Solaris)

PRIMECLUSTER depends on the DLPI (Data Link Provider Interface) for devices in Solaris. If a device does not support a DLPI interface, PRIMECLUSTER does not recognize the device as eligible for use as a cluster interconnect. In addition, the device must appear to be an Ethernet device. Some devices support TCP/IP, but are not Ethernet-type devices. Keep in mind that PRIMECLUSTER does not use TCP/IP for its cluster interconnects; rather it uses the Ethernet protocols.

3.2.3.5 Security

With the PRIMECLUSTER family of products, it is assumed that the cluster interconnects are private networks; however, it is possible to use public networks as cluster interconnects because ICF does not interfere with other protocols running on the physical media. The security model for running PRIMECLUSTER depends on physical separation of the cluster interconnect networks.

Note

For reasons of security, it is strongly recommended not to use public networks for the cluster interconnects.

The use of public networks for the cluster interconnects allows any machine on that public network to join the cluster (assuming that it is installed with the PRIMECLUSTER products). Once joined, an unauthorized user, through the node, would have full access to all cluster services.

Chapter 4 Reliant Monitor Services (RMS)

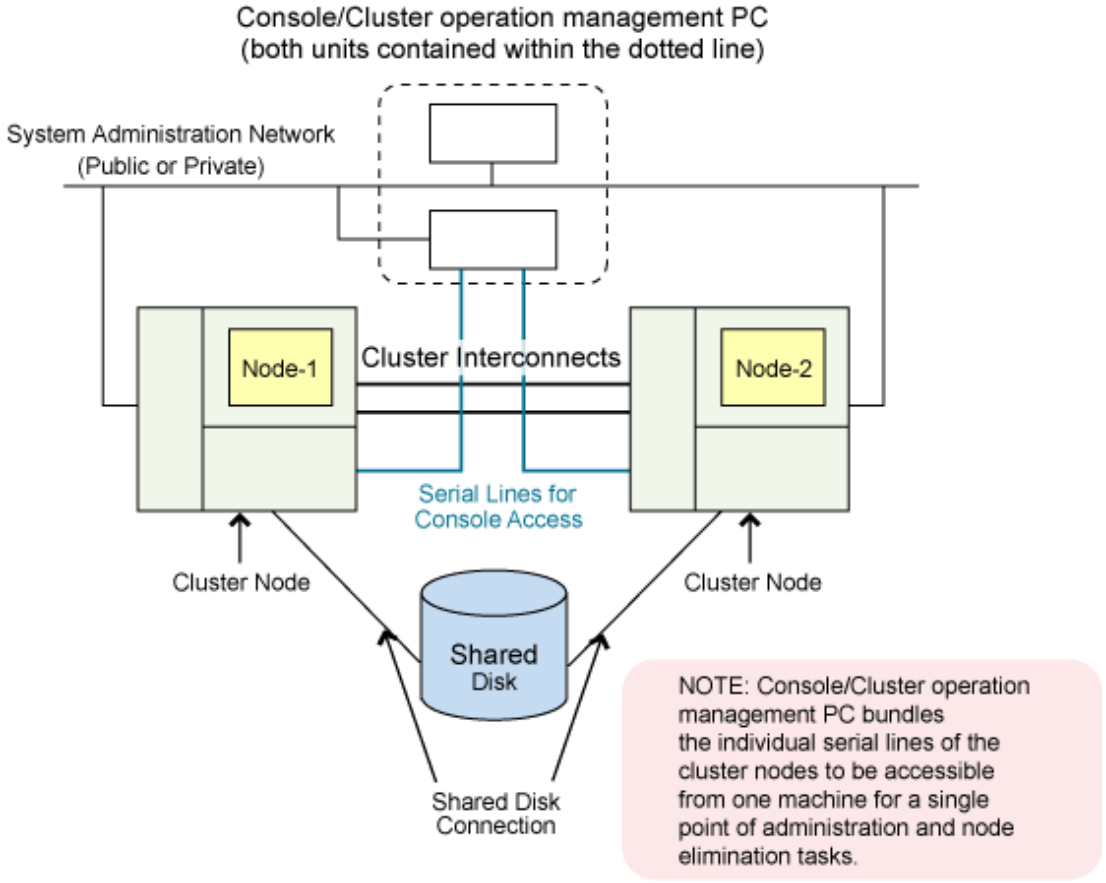
This chapter introduces the basic concepts of RMS. It begins with an overview of the basic terms and concepts used to provide high availability by means of RMS, and then it goes into more detail on how RMS works.

4.1 RMS overview

RMS is a software monitor designed to provide system-level High Availability (HA) to applications. A minimum of two nodes with shared access to common storage are administered from a console or cluster operation management PC. As the PRIMECLUSTER HA monitor, RMS uses detectors to monitor the state of components and resources for applications. If a resource or application fails, the event is notified by the detector and reported to RMS. RMS then takes the appropriate action as described in the sections that follow.

The figure below shows the components of a basic RMS cluster on Solaris.

Figure 4.1 RMS cluster on Solaris



For RMS, high availability means *maximum availability of applications* rather than *uninterrupted availability of individual nodes*.

RMS accomplishes high availability in two ways: redundancy and switchover.

4.1.1 Redundancy

To provide high availability of RMS resources, RMS takes advantage of the following redundancy scheme:

- Multiple nodes, each configured to assume the resource load of any other RMS node
- Duplication of stored data using mirror disks, hardware RAID, and remote mirroring
- Multiple-path access to storage media
- Multiple communications channels dedicated to internode communications

Multiple nodes

An RMS configuration is made up of multiple nodes, each containing identical operating systems and RMS software. The maximum number of nodes in an RMS configuration is theoretically unlimited.

See

Refer to "PRIMECLUSTER Software Release Guide" and "PRIMECLUSTER Installation Guide" for the appropriate number of nodes.

Shared storage

All the nodes in the RMS configuration must have access to whatever data is shared. Typically, all the nodes have the ability to access shared disks over a SAN. However, other access methods, such as Network Attached Storage are possible.

RMS network

RMS uses TCP/IP protocols for communication between the nodes in the configuration. RMS uses Cluster Interconnect Protocol (CIP) on the redundant cluster interconnect that configures PRIMECLUSTER so that RMS can monitor RMS on other nodes in the RMS configuration.

See

See "3.1.2 Interconnect protocol" for more information.

4.1.2 Application switchover

RMS operates on an object-oriented basis. The objects can be almost any system component, such as virtual disks, file system mount points, processes, and so on. These objects are defined as *resources*. A resource is categorized into a grouping called a *object type*. An object type has specific properties, or attributes, which limit and define what monitoring or action can occur in relation to that resource. Resources are monitored by programs called *detectors*. This very general object-oriented design permits a high degree of flexibility in the type and level of monitoring.

Resources which are dependent on each other can be grouped together to form logical applications. Failure of any resource in such a group generally triggers a reaction by the entire application.

4.1.2.1 Automatic switchover

Any failure of a resource triggers a user-defined reaction. In most cases, the most significant reaction to failures is switchover.

A *switchover*, sometimes known as failover, consists of first bringing an application into a well-defined *Offline* state and then restarting the application under the control of RMS on another node. RMS supports symmetrical switchover, which means that every RMS node is able to take on resources from any other RMS node. For example, if the node that is running an application fails, RMS automatically shuts down the node where the failure occurred and redistributes its application load to another operational RMS node.

The details of performing automatic switchover are defined in user-specified scripts and configuration files that RMS accesses when a failure is recognized. The RMS wizards are generally used to create these files.

4.1.2.2 Manual switchover

Resources can be switched manually for such purposes as hardware maintenance. For example, in a two-node RMS configuration, all applications can be temporarily moved to one node while maintenance is performed on the other. Then all applications can be switched back to the operational node while maintenance is performed on the second node. The only impact to users might be a momentary interruption in service while the applications are being switched, and perhaps a slowdown in response time while all applications are operating on a single node.

4.1.2.3 IP aliasing

RMS uses IP aliasing to allow switchover of the communication link. It is possible for several IP addresses (aliases) to be allocated to one physical network interface. With IP aliasing, the user is able to continue communicating with the same IP address, even though the application is now running on another node.

4.1.2.4 Data integrity

RMS ensures that all resources of an application are offline on the current node before initiating a switchover. These resources are then switched online on another operational node. This behavior ensures that multiple nodes do not access the same resource, thus protecting against data loss.

In the rare event of simultaneous multiple faults, RMS protects against data corruption by preventing a switchover. This design means that in certain circumstances, switchover may be prevented entirely.

Although high availability is the goal of RMS, data integrity takes priority over high availability.

4.2 RMS monitoring and switchover

The RMS software is composed of the following processes, scripts, files, and parameters that work together to maintain high availability of the RMS resources:

- Base monitor
- Configuration files
- Configuration scripts
- Detectors
- RMS environment variables

Configuration files, scripts, and environment variables can be customized, allowing switchover scenarios to be tailored for site-specific needs.

See

For information on customization, see "4.4 Customization options."

4.2.1 Base monitor

The base monitor is the central process that monitors a node from the RMS cluster. The base monitor performs the following functions:

- Controls and coordinates all state changes in RMS
- Ensures that the appropriate action is taken if there are any problems with the monitored resources
- Obtains information concerning resources and the action to be taken from an RMS configuration file, which the system administrator creates, according to the requirements for use by RMS

The base monitor is started by the Cluster Admin GUI, or with the *hvcm* command, and runs as a process under the name *bm* (base monitor) on every node in the cluster.

4.2.2 Configuration file

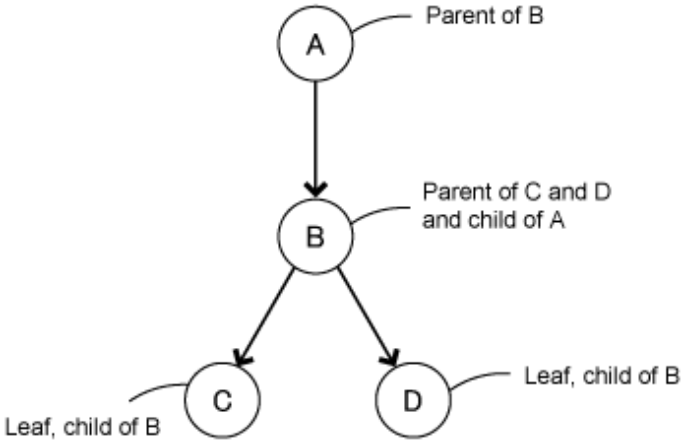
An RMS *configuration file* is a text file that is usually generated by the RMS Wizard Tools. It provides input to the base monitor that consists of definitions of the resources that are to be monitored by RMS, including their interdependencies.

4.2.2.1 Interdependencies

In RMS, the terms *parent* and *child* are used to represent dependent relationships between objects and their resource objects. The term *leaf object* is used to indicate that an object in a system graph has no children. In the configuration file, the leaf object definition is at the beginning of the file. A leaf object cannot have children, while a child can be both a parent and a child.

The figure below helps illustrate the parent/child relationships between objects.

Figure 4.2 Configuration file interdependencies



The figure above shows the following relationships:

- Object A requires the resource B to function properly, making A a parent of B and B a child of A.
- Object B requires the resources C and D to function properly, making B a parent of C and a parent of D.
- Objects C and D are leaf objects because they do not have any children. They are also children of B.

4.2.2.2 Object types

A *object type* is a collection of similar resources monitored as a group (for example, all object types in a group using the same disk drives). Each object type has certain properties, or *attributes*, which limit or define what monitoring can occur for that type of resource. The attributes associated with a particular object type define how the base monitor acts and reacts for a resource of that object type during normal operations. Attributes commonly specify a device name or a script, and can be specified in any order in the object definition for that resource.

Some attributes are mandatory; others are optional. An example of a mandatory attribute is a device name for a *vdisk* object type. Typical optional attributes are scripts which are executed under certain conditions. Most attributes can be used for most object definitions.

Some attributes are valid only for particular object types, while some object types require that specific attributes be included in their object definitions.

4.2.2.3 Object definitions

A *object definition* is a statement in the configuration file, beginning with the keyword *object*, that describes a particular resource in terms that RMS understands. Specifics of an object definition include the following:

- Object type
- Resource name
- Attributes
- Child resources that the particular resource depends upon

After RMS has been installed and verified, the configuration file must be created before RMS can begin monitoring resources. If a resource does not have an object definition in the configuration file, the resource is *not* recognized by RMS.

4.2.3 Scripts

An RMS configuration *script* is a shell program or executable that reacts to and/or invokes a change in the state of an RMS resource.

The following states are possible for all RMS resources:

- *Faulted*
- *Offline*

- *OfflineFault*
- *Online*
- *Unknown*
- *Wait*
- *Deact*
- *Inconsistent*
- *Standby*
- *Warning*

All RMS actions and reactions are executed as scripts. Without scripts, RMS is only capable of monitoring resources, not activating any changes. Scripts are identified as attributes in the object definition for the particular resource, and are run by the base monitor as needed in response to detector-reported state changes. For example, if the state of an RMS network resource changes from *Online* to *Faulted*, the base monitor responds by initiating the fault script listed in the object definition for that resource.

RMS distinguishes between scripts which are designed to change a state (request-triggered scripts) and scripts which represent a reaction to a specific state (state-triggered scripts).

Request-triggered scripts include:

- *PreOnlineScript*
- *PreOfflineScript*
- *PreCheckScript*
- *OnlineScript*
- *OfflineScript*
- *OfflineDoneScript*

State-triggered scripts include:

- *PostOnlineScript*
- *PostOfflineScript*
- *FaultScript*

Your RMS may include additional scripts.

 See

For further information on resource states and scripts, see "PRIMECLUSTER Reliant Monitor Services (RMS) with Wizard Tools Configuration and Administration Guide."

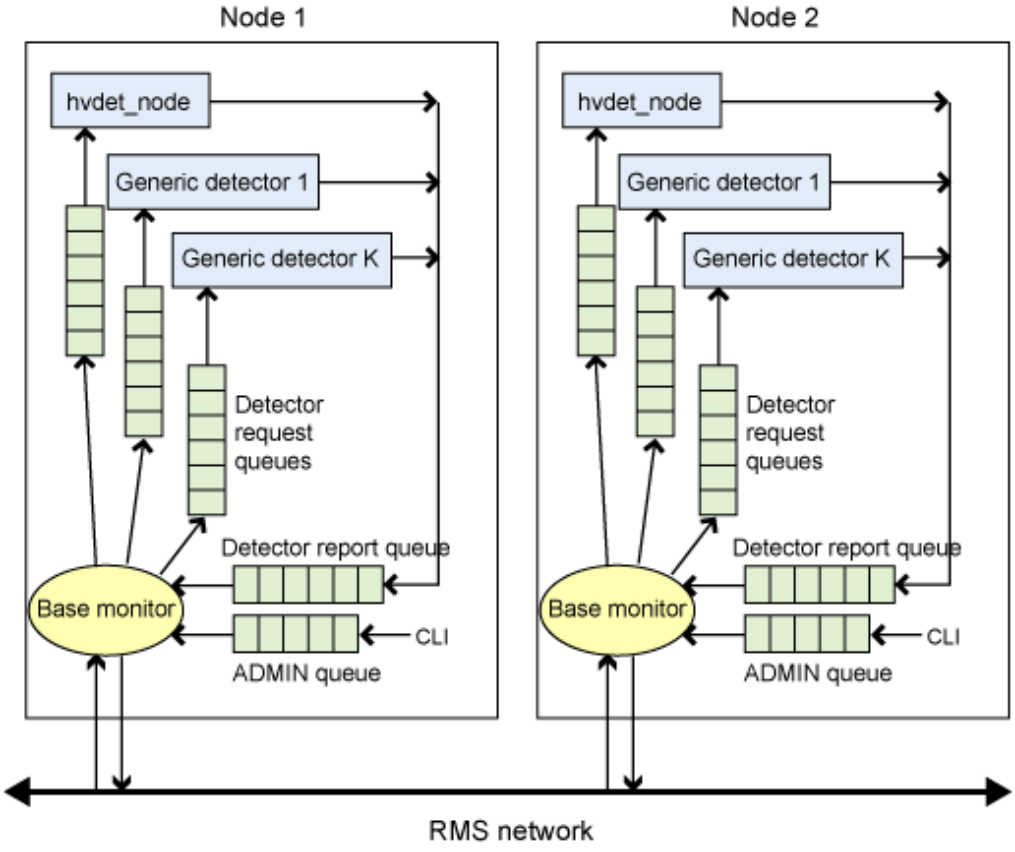
4.2.4 Detectors

A *detector* is a process that monitors the state of a certain type of resource, such as mirrored disks. A detector is started by the base monitor when RMS is started.

Each detector uses an internal table to track information about one or more resources. The detector scans each resource listed in its internal table, obtains the current state of each resource, and then compares the current state to the previous state. If the current state is not the same as the previous state, the detector reports the current state to the base monitor through the detector report queue for that type of resource. Based on the information from the configuration file, the base monitor then determines, what, if any, action is necessary for the particular situation involving that resource.

The figure below illustrates the interactions between the base monitor and the RMS detectors.

Figure 4.3 Base monitor interactions



4.2.5 RMS environment variables

Environment variables specify values for the base monitor during startup system events. RMS comes with a number of high availability environment variables, such as HV_AUTOSTART_WAIT, RELIANT_PATH, and so on. The default settings of these RMS environment variables can be adjusted as needed for individual clusters and applications.

4.3 RMS administration

You can administer RMS from the command-line interface (CLI) or from the graphical user interface (GUI). The preferred method for RMS administration is the GUI, which is called Cluster Admin.

4.4 Customization options

Fault detection and recovery schemes can be customized to fit the needs of each site by modifying configuration files, detectors, and scripts for the resources that are to be monitored by RMS. These modifications should be planned in advance and implemented after RMS installation is completed, and before RMS is declared operational.

4.4.1 Generic types and detectors

RMS provides a generic resource type for defining special resources that cannot use the system-level supplied resource types. The generic detector interface allows the definition of up to 64 custom resource types and their detectors.

Chapter 5 RMS wizard

This chapter introduces the basic concepts of the RMS configuration tools.
After a brief overview, it discusses the RMS wizard product and some of the functions it provides.

5.1 RMS wizard overview

Creating an RMS configuration file (see the section "Configuration file") is a complex task that requires detailed knowledge of both the user applications and the RMS environment.

RMS wizard simplifies both the configuration and the operation of RMS. RMS configuration tools can manage the system, RMS, and applications.

Note

RMS wizard manages the system, RMS, and applications. However, RMS and RMS Wizard Tools should be treated as a different component.

5.2 RMS wizard architecture

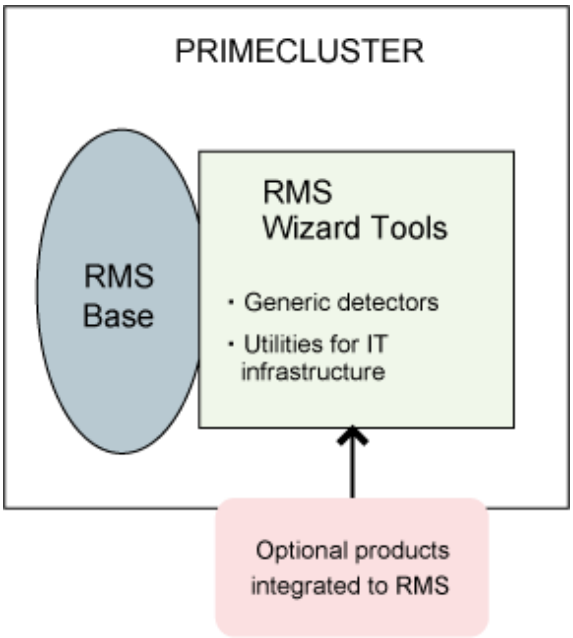
The RMS wizard is provided with the following product:

- RMS Wizard Tools

  This contains configuration tool foundations and interface with the RMS base. It simplifies the configuration setup and work with RMS in the HA (high availability) cluster environment.

The figure below indicates the relevance of RMS Wizard Tools and with RMS.

Figure 5.1 RMS Wizard architecture



5.3 RMS Wizard Tools

The RMS Wizard Tools, which is an optional product, creates and manages the RMS configuration with high availability RMS configuration files.

- Starting a user operation that is configured on the node in the cluster

- Present standard resources in a logical sequence to generate correct configuration structures
- Copying the whole configuration
- Preventing or allowing configuration changes

To support additional HA configurations, there are some utility wizards offered as part of the configuration tool foundation that manage some system standard resources and applications not covered by a specific application wizard, such as the following:

- Mounting/unmounting file systems
- Installing/uninstalling IP aliases
- Configuring/deconfiguring virtual disks

5.3.1 Shared-storage applications

Shared-storage application wizards configure RMS for different types of disk storage. For example, you can configure software-based solutions such as GDS or Veritas VxVM, or you can configure hardware solutions like Dell EMC. These wizards can also configure multiple SAN (Storage Area Network) modules, allowing you the capacity to stack modules.

Appendix A Release Information

This appendix lists the main changes in this manual.

No.	Edition	Location	Description
1	Second edition	1.1 Introduction 1.7 Cloud support 1.8.1.2 Cloud environment 2.3.5 PRIMECLUSTER SF	Added the descriptions of the following cloud environments. - NIFCLOUD environment - FJcloud-Baremetal environment - AWS environment - Azure environment
2	Second edition	1.1 Introduction 1.7 Cloud support 1.8.1.2 Cloud environment 2.3.5 PRIMECLUSTER SF	Changed the name of "FUJITSU Cloud Service for OSS" to "FUJITSU Hybrid IT Service FJcloud-O."
3	Second edition	1.6 Virtualization support	Changed the figure for a VMware environment.
4	Second edition	2.3.5 PRIMECLUSTER SF	Changed the following. - Description of the Shutdown Agent - Description of ICMP (SA_icmp) - Description of the Monitoring Agent
5	Third edition	1.7 Cloud support	Changed the description of an FJcloud-Baremetal environment.
6	Third edition	1.7 Cloud support 1.8.1.2 Cloud environment	Changed the descriptions of a NIFCLOUD environment.
7	Third edition	1.9 Smart workload recovery	Added smart workload recovery description.
8	Third edition	2.3.5 PRIMECLUSTER SF	Changed the description of the following shutdown agent. - FUJITSU Hybrid IT Service FJcloud-O/FJcloud-Baremetal API (SA_vmk5r)
9	Third edition	2.3.5 PRIMECLUSTER SF	Added the description of the following shutdown agent. - NIFCLOUD API (SA_vmnifclAsyncReset)

Glossary

AC (Access Client)

See *Access Client*.

Access Client

GFS kernel module on each node that communicates with the Meta Data Server and provides simultaneous access to a shared file system.

administrative LAN

In PRIMECLUSTER configurations, an administrative LAN is a private local area network (LAN) on which machines such as the System Console and Cluster operation management PC reside. Because normal users do not have access to the administrative LAN, it provides an extra level of security. The use of an administrative LAN is optional.

See also *public LAN*.

API

See *Application Program Interface*.

application (RMS)

A resource categorized as a *userApplication* used to group resources into a logical collection.

Application Program Interface

A shared boundary between a service provider and the application that uses that service.

attribute (RMS)

The part of an object definition that specifies how the base monitor acts and reacts for a particular object type during normal operations.

automatic power control

This function is provided by the Enhanced Support Facility (ESF), and it automatically switches the server power on and off.

automatic switchover (RMS)

The procedure by which RMS automatically switches control of a *userApplication* over to another node after specified conditions are detected.

See also *directed switchover (RMS), failover (RMS, SIS), switchover (RMS), symmetrical switchover (RMS)*.

availability

Availability describes the need of most enterprises to operate applications via the Internet 24 hours a day, 7 days a week. The relationship of the actual to the planned usage time determines the availability of a system.

base cluster foundation (CF)

This PRIMECLUSTER module resides on top of the basic OS and provides internal interfaces for the CF (Cluster Foundation) functions that the PRIMECLUSTER services use in the layer above.

See also *Cluster Foundation (CF)*.

base monitor (RMS)

The RMS module that maintains the availability of resources. The base monitor is supported by daemons and detectors. Each node being monitored has its own copy of the base monitor.

bm

base monitor

BMC (Baseboard Management Controller)

A dedicated processor for monitoring and diagnosis of environmental factors (e.g. temperature, voltage) and parts and units.

Cache Fusion

The improved interprocess communication interface in Oracle 9i that allows logical disk blocks (buffers) to be cached in the local memory of each node. Thus, instead of having to flush a block to disk when an update is required, the block can be copied to another node by passing a message on the cluster interconnect, thereby removing the physical I/O overhead.

CCBR (only in Solaris)

See *Cluster Configuration Backup and Restore*.

ccbr.conf (only in Solaris)

The environment configuration file that is used for backup and restore operations, and is placed in the "/opt/SMAW/ccbr" directory. This file is used in the *$CCBRHOME* variable setting. For details, see the manual pages for the *cfbackup*(1M) and *cfrestore*(1M) commands and the comments in the *ccbr.conf* file.

ccbr.gen (only in Solaris)

The file that stores the generation number and is placed in the "/opt/SMAW/ccbr" directory. A value of 0 or higher is stored in this file. For details, see the manual pages for the *cfbackup*(1M) and *cfrestore*(1M) commands.

CCBRHOME variable (only in Solaris)

The variable that identifies the directory in which backup data is stored. The initial value is the "/var/spool/pcl4.1/ccbr" directory. This variable can be set only in the *ccbr.conf* file.

CF

See *Cluster Foundation (CF)*.

CDL

Configuration Definition Language

child (RMS)

A resource defined in the configuration file that has at least one parent. A child can have multiple parents, and can either have children itself (making it also a parent) or no children (making it a leaf object).

See also *resource (RMS), object (RMS), parent (RMS)*.

CIM

Cluster Integrity Monitor

CIP

Cluster Interconnect Protocol

CLI

command-line interface

CLM

Cluster Manager

cluster

A set of computers that work together as a single computing source. Specifically, a cluster performs a distributed form of parallel computing.

See also *RMS configuration*.

Cluster Admin

A Java-based, OS-independent management tool for PRIMECLUSTER products such as CF, RMS, and SIS. Cluster Admin is available from the Web-Based Admin View interface.

See also *Cluster Foundation (CF), Reliant Monitor Services (RMS), Scalable Internet Services (SIS), Web-Based Admin View*.

Cluster Configuration Backup and Restore (Only in Solaris)

CCBR provides a simple method to save the current PRIMECLUSTER configuration information of a cluster node. It also provides a method to restore the configuration information.

Cluster Foundation (CF)

The set of PRIMECLUSTER modules that provides basic clustering communication services.

See also *base cluster foundation (CF)*.

cluster interconnect (CF)

The set of private network connections used exclusively for PRIMECLUSTER communications.

Cluster Join Services (CF)

This PRIMECLUSTER module handles the forming of a new cluster and the addition of nodes.

cluster partition (CF)

This condition exists when two or more nodes in a cluster cannot communicate over the cluster interconnect; however, with applications still running, the nodes can continue to read and write to a shared device, compromising data integrity.

See also *split-brain syndrome*.

cold-standby

The operation which does not allow the preliminary operation needed to establish the operating state immediately on the standby node.

configuration file (RMS)

The RMS configuration file that defines the monitored resources and establishes the interdependencies between them. The default name of this file is *config.us*.

CRM

Cluster Resource Management

custom detector (RMS)

See *detector (RMS)*.

custom type (RMS)

See *generic type (RMS)*.

daemon

A continuous process that performs a specific function repeatedly.

database node (SIS)

Nodes that maintain the configuration, dynamic data, and statistics in a SIS configuration.

See also *gateway node (SIS), service node (SIS), Scalable Internet Services (SIS)*.

detector (RMS)

A process that monitors the state of a specific object type and reports a change in the resource state to the base monitor.

directed switchover (RMS)

The RMS procedure by which an administrator switches control of a *userApplication* over to another node.

See also *automatic switchover (RMS), failover (RMS, SIS), switchover (RMS), symmetrical switchover (RMS)*.

DLPI

Data Link Provider Interface

DOWN (CF)

A node state that indicates that the node is unavailable (marked as down). A *LEFTCLUSTER* node must be marked as *DOWN* before it can rejoin a cluster.

See also *UP (CF), LEFTCLUSTER (CF), node state (CF)*.

ENS (CF)

See *Event Notification Services (CF)*.

environment variables

Variables or parameters that are defined globally.

error detection (RMS)

The process of detecting an error. For RMS, this includes initiating a log entry, sending a message to a log file, or making an appropriate recovery response.

Ethernet

LAN standard that is standardized by IEEE 802.3. Currently, except for special uses, nearly all LANs are Ethernets. Originally the expression Ethernet was a LAN standard name for a 10 megabyte per second type LAN, but now it is also used as a general term that includes high-speed Ethernets and gigabyte Ethernets.

Event Notification Services (CF)

This PRIMECLUSTER module provides an atomic-broadcast facility for events.

Fast switching mode

One of the LAN duplexing modes presented by GLS.

This mode uses a multiplexed LAN simultaneously to provide enhanced communication scalability between servers and high speed switchover if a LAN failure occurs.

failover (RMS, SIS)

With SIS, this process switches a failed node to a backup node. With RMS, this process is known as switchover.

See also *automatic switchover (RMS), directed switchover (RMS), switchover (RMS), symmetrical switchover (RMS)*.

gateway node (SIS)

Gateway nodes have an external network interface. All incoming packets are received by this node and forwarded to the selected service node, depending on the scheduling algorithm for the service.

See also *service node (SIS), database node (SIS), Scalable Internet Services (SIS)*.

GFS Shared File System

The GFS Shared File System is a shared file system that allows simultaneous access with consistency/integrity maintained from the multiple nodes connected to the shared disk device. It also features continued process of other nodes even if some nodes have failed.

Global Disk Services

This optional product provides volume management that improves the availability and manageability of information stored on the disk unit of the Storage Area Network (SAN).

Global File Services

This optional product provides direct, simultaneous accessing of the file system on the shared storage unit from two or more nodes within a cluster.

Global Link Services

This PRIMECLUSTER optional module provides network high availability solutions by multiplying a network route.

generic type (RMS)

An object type which has generic properties. A generic type is used to customize RMS for monitoring resources that cannot be assigned to one of the supplied object types.

See also *object type (RMS)*.

graph (RMS)

See *system graph (RMS)*.

graphical user interface

A computer interface with windows, icons, toolbars, and pull-down menus that is designed to be simpler to use than the command-line interface.

GUI

See *graphical user interface*.

HA

high availability

high availability

A system design philosophy in which redundant resources are employed to avoid single points of failure.

See also *Reliant Monitor Services (RMS)*.

hot-standby

The operation which enables preliminary operation so that the operating state can be established immediately on the standby node.

ICF

Internode Communication Facility

interconnect (CF)

See *cluster interconnect (CF)*.

Internet Protocol address

A numeric address that can be assigned to computers or applications.

See also *IP aliasing*.

Internode Communications facility

This module is the network transport layer for all PRIMECLUSTER internode communications. It interfaces by means of OS-dependent code to the network I/O subsystem and guarantees delivery of messages queued for transmission to the destination node in the same sequential order unless the destination node fails.

I/O

input/output

IP address

See *Internet Protocol address*.

IP aliasing

This enables several IP addresses (aliases) to be allocated to one physical network interface. With IP aliasing, the user can continue communicating with the same IP address, even though the application is now running on another node.

See also *Internet Protocol address*.

iRMC (integrated Remote Management Controller)

Abbreviation for integrated Remote Management Controller which is one of the hardware mounted in PRIMEQUEST/PRIMERGY.

JOIN (CF)

See *Cluster Join Services (CF)*.

LAN

local area network

keyword

A word that has special meaning in a programming language. For example, in the configuration file, the keyword *object* identifies the kind of definition that follows.

leaf object (RMS)

A bottom object in a system graph. In the configuration file, this object definition is at the beginning of the file. A leaf object does not have children.

LEFTCLUSTER (CF)

A node state that indicates that the node cannot communicate with other nodes in the cluster. That is, the node has left the cluster. The reason for the intermediate *LEFTCLUSTER* state is to avoid the network partition problem.

See also *UP (CF), DOWN (CF), cluster partition (CF), node state (CF)*.

link (RMS)

Designates a child or parent relationship between specific resources.

local area network

See *public LAN*.

local node

The node from which a command or process is initiated.

See also *remote node, node*.

log file

The file that contains a record of significant system events or messages. The base monitor, wizards, and detectors can have their own log files.

MDS

See *Meta Data Server*.

message

A set of data transmitted from one software process to another process, device, or file.

message queue

A designated memory area which acts as a holding place for messages.

Meta Data Server

GFS daemon that centrally manages the control information of a file system (meta-data).

MIB

Management Information Base

MIPC

Mesh Interprocessor Communication

mixed model cluster

A cluster system that is built from different SPARC Enterprise models. For example, one node is SPARC Enterprise M3000, and another node is SPARC Enterprise M4000. The models are divided into the following groups: SPARC M12-2/M12-2S, SPARC M10-1/M10-4/M10-4S, SPARC S7-2/S7-2L, SPARC T7-1/T7-2/T7-4, SPARC T5-2/T5-4/T5-8, SPARC T4-1/T4-2/T4-4, SPARC T3-1/T3-2/T3-4, SPARC Enterprise T1000/T2000, SPARC Enterprise T5120/T5220/T5140/T5240/T5440, and SPARC Enterprise M3000/M4000/M5000/M8000/M9000.

MMB

Abbreviation for Management Board, which is one of the hardware units installed in PRIMEQUEST.

mount point

The point in the directory tree where a file system is attached.

multihosting

Multiple controllers simultaneously accessing a set of disk drives.

native operating system

The part of an operating system that is always active and translates system calls into activities.

NIC

network interface card

NIC switching mode

One of the LAN duplexing modes presented by GLS. The duplexed NIC is used exclusively, and LAN monitoring between the server and the switching HUB, and switchover if an error is detected are implemented.

node

A host which is a member of a cluster. A computer node is the same as a computer.

node state (CF)

Every node in a cluster maintains a local state for every other node in that cluster. The node state of every node in the cluster must be either *UP, DOWN,* or *LEFTCLUSTER*.

See also *UP (CF), DOWN (CF), LEFTCLUSTER (CF)*.

NSM

Node State Monitor

object (RMS)

In the configuration file or a system graph, this is a representation of a physical or virtual resource.

See also *leaf object (RMS), object definition (RMS), object type (RMS)*.

object definition (RMS)

An entry in the configuration file that identifies a resource to be monitored by RMS. Attributes included in the definition specify properties of the corresponding resource. The keyword associated with an object definition is *object*.

See also *attribute (RMS), object type (RMS)*.

object type (RMS)

A category of similar resources monitored as a group, such as disk drives. Each object type has specific properties, or attributes, which limit or define what monitoring or action can occur. When a resource is associated with a particular object type, attributes associated with that object type are applied to the resource.

See also *generic type (RMS)*.

online maintenance

The capability of adding, removing, replacing, or recovering devices without shutting or powering off the node.

operating system dependent (CF)

This module provides an interface between the native operating system and the abstract, OS-independent interface that all PRIMECLUSTER modules depend upon.

Oracle Real Application Clusters (RAC)

Oracle RAC allows access to all data in a database to users and applications in a clustered or MPP (massively parallel processing) platform. Formerly known as Oracle Parallel Server (OPS).

OSD (CF)

See *operating system dependent (CF)*.

parent (RMS)

An object in the configuration file or system graph that has at least one child.

See also *child (RMS), configuration file (RMS), system graph (RMS)*.

PAS

Parallel Application Services

physical IP address

IP address that is assigned directly to the interface (for example, hme0) of a network interface card.

primary node (RMS)

The default node on which a user application comes online when RMS is started. This is always the nodename of the first child listed in the *userApplication* object definition.

PRIMECLUSTER services (CF)

Service modules that provide services and internal interfaces for clustered applications.

private network addresses

Private network addresses are a reserved range of IP addresses specified by the Internet Assigned Numbers Authority. They may be used internally by any organization but, because different organizations can use the same addresses, they should never be made visible to the public internet.

private resource (RMS)

A resource accessible only by a single node and not accessible to other RMS nodes.

See also *resource (RMS), shared resource*.

public LAN

The local area network (LAN) by which normal users access a machine.

See also *administrative LAN*.

queue

See *message queue*.

RCCU

Abbreviation for Remote Console Connection Unit.

See also *remote console connection unit*.

RCI

Remote Cabinet Interface

redundancy

This is the capability of one object to assume the resource load of any other object in a cluster, and the capability of RAID hardware and/or RAID software to replicate data stored on secondary storage devices.

Reliant Monitor Services (RMS)

The package that maintains high availability of user-specified resources by providing monitoring and switchover capabilities.

remote console connection unit

Device that converts an RS232C interface and a LAN interface. This device allows another device (personal computer) that is connected to the LAN to use the TTY console functions through the Telnet function.

remote node

A node that is accessed through a LAN or telecommunications line.

See also *local node, node*.

reporting message (RMS)

A message that a detector uses to report the state of a particular resource to the base monitor.

resource (RMS)

A hardware or software element (private or shared) that provides a function, such as a mirrored disk, mirrored disk pieces, or a database server. A local resource is monitored only by the local node.

See also *private resource (RMS), shared resource*.

resource definition (RMS)

See *object definition (RMS)*.

resource label (RMS)

The name of the resource as displayed in a system graph.

resource state (RMS)

Current state of a resource.

RMS

See *Reliant Monitor Services (RMS)*.

RMS commands

Commands that enable RMS resources to be administered from the command line.

RMS configuration

A configuration made up of two or more nodes connected to shared resources. Each node has its own copy of operating system and RMS software, as well as its own applications.

RMS Wizard Tools

A software package composed of various configuration and administration tools used to create and manage applications in an RMS configuration. Provides interface between foundation of RMS Wizard and BM (Base Monitor).

RMS Wizard

A software tool for creating specified configuration in order for RMS to operate.

See also *RMS Wizard Tools*

SA

Shutdown Agent

SAN

See *Storage Area Network*.

Scalable Internet Services (SIS)

Scalable Internet Services is a TCP connection load balancer, and dynamically balances network access loads across cluster nodes while maintaining normal client/server sessions for each connection.

scalability

The ability of a computing system to dynamically handle any increase in work load. Scalability is especially important for Internet-based applications where growth caused by Internet usage presents a scalable challenge.

script (RMS)

A shell program executed by the base monitor in response to a state transition in a resource. The script may cause the state of a resource to change.

SD

Shutdown Daemon

service node (SIS)

Service nodes provide one or more TCP services (such as FTP, Telnet, and HTTP) and receive client requests forwarded by the gateway nodes.

See also *database node (SIS), gateway node (SIS), Scalable Internet Services (SIS)*.

SF

Shutdown Facility

shared resource

A resource, such as a disk drive, that is accessible to more than one node.

See also *private resource (RMS), resource (RMS)*.

Shutdown Facility

A facility that forcibly stops a node in which a failure has occurred. When PRIMECLUSTER decides that system has reached a state in which the quorum is not maintained, it uses the Shutdown Facility (SF) to return the cluster system to the quorum state.

Single node cluster

An operation mode of a cluster system consisting of one node.

SIS

See *Scalable Internet Services (SIS)*.

split-brain syndrome

See *cluster partition (CF)*.

state

See *resource state (RMS)*.

Storage Area Network

The high-speed network that connects multiple, external storage units and storage units with multiple computers. The connections are generally fiber channels.

switching mode

LAN duplexing mode presented by GLS.
There are a total of six switching mode types: fast switching mode, NIC switching mode, GS/SURE linkage mode, virtual NIC mode, multipath mode, and Multi-link Ethernet mode.

switchover (RMS)

The process by which RMS switches control of a *userApplication* over from one monitored node to another.

See also *automatic switchover (RMS), directed switchover (RMS), failover (RMS, SIS), symmetrical switchover (RMS)*.

symmetrical switchover (RMS)

This means that every RMS node is able to take on resources from any other RMS node.

See also *automatic switchover (RMS), directed switchover (RMS), failover (RMS, SIS), switchover (RMS)*.

synchronized power control

When the power of one node is turned in the cluster system, this function turns on all other powered-off nodes and disk array unit that are connected to nodes through RCI cables.

system disk (GDS)

Disk on which the active operating system is installed. System disk refers to the entire disk that contains the slices that are currently operating as one of the following file systems (or the swap area):

For Solaris: */*, */usr*, */var*, or swap area
For Linux: */*, */usr*, */var*, */boot*, */boot/efi*, or swap area

system graph (RMS)

A visual representation (a map) of monitored resources used to develop or interpret the configuration file.

See also *configuration file (RMS)*.

type

See *object type (RMS)*.

UP (CF)

A node state that indicates that the node can communicate with other nodes in the cluster.

See also *DOWN (CF)*, *LEFTCLUSTER (CF)*, *node state (CF)*.

VIP

Virtual Interface Provider

warm-standby

In Oracle Solaris Zones environments, with the non-global zones started up on both the operating server and standby server as is, this operation switches over only the applications operating within the non-global zone, and takes over services. Since the standby system's non-global zone OS becomes a startup status, a faster switchover than the cold-standby is possible.

Web-Based Admin View

A Java-based, OS-independent interface to PRIMECLUSTER management components.

See also *Cluster Admin*.

wizard (RMS)

An interactive software tool that creates a specific type of application using pretested object definitions. An enabler is a type of wizard.

Wizard Tools (RMS)

See *RMS Wizard Tools*.

XSCF

eXtended System Control Facility

Index