Oracle® Solaris Cluster Concepts Guide

Part No: E37723
March 2013, E37723–01

ORACLE

130801@25097

Contents

Preface

The *Oracle Solaris Cluster Concepts Guide* contains conceptual information about the Oracle Solaris Cluster product on both SPARC and x86 based systems.

Note – This Oracle Solaris Cluster release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, and Intel 64. In this document, x86 refers to the larger family of 64-bit x86 compatible products. Information in this document pertains to all platforms unless otherwise specified.

Who Should Use This Book

This document is intended for the following audiences:

- Service providers who install and service cluster hardware
- System administrators who install, configure, and administer Oracle Solaris Cluster software
- Application developers who develop failover and scalable services for applications that are not currently included with the Oracle Solaris Cluster product

To understand the concepts that are described in this book, you should be familiar with the Oracle Solaris operating system and have expertise with the volume manager software that you can use with the Oracle Solaris Cluster product.

You should determine your system requirements and purchase the required equipment and software. The *Oracle Solaris Cluster Data Services Planning and Administration Guide* contains information about how to plan, install, set up, and use the Oracle Solaris Cluster software.

How This Book Is Organized

The *Oracle Solaris Cluster Concepts Guide* contains the following chapters:

Chapter 1, "Introduction and Overview," provides an overview of the overall concepts that you need to know about Oracle Solaris Cluster.

Chapter 2, "Key Concepts for Hardware Service Providers," describes the concepts that hardware service providers should understand. These concepts can help service providers understand the relationships between hardware components. These concepts can also help service providers and cluster administrators better understand how to install, configure, and administer cluster software and hardware.

Chapter 3, "Key Concepts for System Administrators and Application Developers," describes the concepts system administrators and developers who will use the Oracle Solaris Cluster application programming interface (API) should know. Developers can use this API to turn a standard user application, such as a web browser or database, into a highly available data service that can run in the Oracle Solaris Cluster environment.

Related Documentation

Information about related Oracle Solaris Cluster topics is available in the documentation that is listed in the following table. All Oracle Solaris Cluster documentation is available at `http://www.oracle.com/technetwork/indexes/documentation/index.html#sys_sw`.

Topic	Documentation
Concepts	*Oracle Solaris Cluster Concepts Guide*
Hardware installation and administration	*Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual* and individual hardware administration guides
Software installation	*Oracle Solaris Cluster Software Installation Guide*
Data service installation and administration	*Oracle Solaris Cluster Data Services Planning and Administration Guide* and individual data service guides
Data service development	*Oracle Solaris Cluster Data Services Developer's Guide*
System administration	*Oracle Solaris Cluster System Administration Guide* *Oracle Solaris Cluster Quick Reference*
Software upgrade	*Oracle Solaris Cluster Upgrade Guide*
Error messages	*Oracle Solaris Cluster Error Messages Guide*

Topic	Documentation
Command and function references	*Oracle Solaris Cluster Reference Manual* *Oracle Solaris Cluster Data Services Reference Manual*

Getting Help

If you have problems installing or using the Oracle Solaris Cluster software, contact your service provider and provide the following information:

- Your name and email address
- Your company name, address, and phone number
- The model and serial numbers of your systems
- The release number of the operating system (for example, the Solaris 10 OS)
- The release number of Oracle Solaris Cluster software (for example, 3.3)

Use the commands in the following table to gather information about your systems for your service provider.

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices
`psrinfo -v`	Displays information about processors
`showrev -p`	Reports which patches are installed
SPARC: `prtdiag -v`	Displays system diagnostic information
`/usr/cluster/bin/clnode show-rev -v`	Displays Oracle Solaris Cluster release and package version information

You should also have available the contents of the `/var/adm/messages` file.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Description	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

◆ ◆ ◆ CHAPTER 1

Introduction and Overview

The Oracle Solaris Cluster product is an integrated hardware and software solution that you use to create highly available and scalable services. The *Oracle Solaris Cluster Concepts Guide* provides the conceptual information that you need to gain a more complete picture of the Oracle Solaris Cluster product. Use this book with the entire Oracle Solaris Cluster documentation set to provide a complete view of the Oracle Solaris Cluster software.

This chapter provides an overview of the general concepts that underlie the Oracle Solaris Cluster product. It includes the following information:

- Provides an introduction and high-level overview of the Oracle Solaris Cluster software
- Describes several views of the Oracle Solaris Cluster software by audiences
- Identifies key concepts that you need to understand before you use the Oracle Solaris Cluster software
- Maps key concepts to procedures and related information in the Oracle Solaris Cluster documentation
- Maps cluster-related tasks to the related procedures in the documentation

This chapter contains the following sections:

Introduction to the Oracle Solaris Cluster Environment

The Oracle Solaris Cluster environment extends the Oracle Solaris operating system into a cluster operating system. A *cluster* is a collection of one or more nodes that belong exclusively to that collection.

A cluster offers several advantages over traditional single-server systems. These advantages include support for failover and scalable services, capacity for modular growth, the ability to set load limits on nodes, and a low entry price compared to traditional hardware fault-tolerant systems.

Additional benefits of the Oracle Solaris Cluster software include the following:

- Reduces or eliminates system downtime because of software or hardware failure
- Ensures availability of data and applications to end users regardless of the kind of failure that would normally take down a single-server system
- Increases application throughput by enabling services to scale to additional processors by adding nodes to the cluster and balancing load
- Provides enhanced availability of the system by enabling you to perform maintenance without shutting down the entire cluster

In a cluster that runs on the Oracle Solaris OS, a global cluster and a zone cluster are types of clusters.

- A *global cluster* consists of a set of Oracle Solaris global zones. A global cluster is composed of one or more global-cluster voting nodes and optionally, zero or more global-cluster non-voting nodes.

  A global-cluster voting node is a `native` brand global zone in a global cluster that contributes votes to the total number of quorum votes, that is, membership votes in the cluster. This total determines whether the cluster has sufficient votes to continue operating. A global-cluster non-voting node is a `native` brand non-global zone in a global cluster that does *not* contribute votes to the total number of quorum votes.
- A *zone cluster* consists of a set of non-global zones (one per global-cluster node), that are configured to behave as a separate "virtual" cluster.

For more information about global clusters and zone clusters, see "Overview of Administering Oracle Solaris Cluster" in *Oracle Solaris Cluster System Administration Guide* and "Working With a Zone Cluster" in *Oracle Solaris Cluster System Administration Guide*.

Three Views of the Oracle Solaris Cluster Software

This section describes three different views of the Oracle Solaris Cluster software by different audiences and the key concepts and documentation relevant to each view.

These views are typical for the following professionals:

- Hardware installation and service personnel
- System administrators
- Application developers

Hardware Installation and Service View

To hardware service professionals, the Oracle Solaris Cluster software looks like a collection of off-the-shelf hardware that includes servers, network devices and equipment, and storage. These components are all cabled together so that every component has a backup and no single point of failure exists.

Key Concepts – Hardware

Hardware service professionals need to understand the following cluster concepts:

- Cluster hardware configurations and cabling
- Installing and servicing (adding, removing, replacing):
  - Network interface components (adapters, junctions, cables)
  - Host bus adapter
  - Disk arrays
  - Disk drives
  - The administrative console and the console access device

See the following sections for more information:

- "Cluster Nodes" on page 18
- "Software Components for Cluster Hardware Members" on page 20
- "Multihost Devices" on page 21
- "Local Disks" on page 22
- "Removable Media" on page 22
- "Cluster Interconnect" on page 22
- "Public Network Interfaces" on page 23
- "Logging Into the Cluster Remotely" on page 24
- "Administrative Console" on page 24
- "SPARC: Oracle Solaris Cluster Topologies" on page 25
- "x86: Oracle Solaris Cluster Topologies" on page 33

Oracle Solaris Cluster Documentation for Hardware Professionals

The *Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual* includes procedures and information associated with hardware service concepts.

System Administrator View

To the system administrator, the Oracle Solaris Cluster product is a set of cluster nodes that share storage devices.

The system administrator sees software that performs specific tasks:

- Specialized cluster software that is integrated with the Oracle Solaris OS, which forms the high availability framework that monitors the health of the cluster nodes
- Specialized software that monitors the health of user application programs that are running on the cluster nodes
- Optional volume management software that sets up and administers disks
- Specialized cluster software that enables all cluster nodes to access all storage devices, even those nodes that are not directly connected to disks
- Specialized cluster software that enables files to appear on every cluster node as though they were locally attached to that node

Key Concepts – System Administration

System administrators need to understand the following concepts and processes:

- The interaction between the hardware and software components
- The general flow of how to install and configure the cluster including:
  - Installing the Oracle Solaris OS
  - Installing and configuring Oracle Solaris Cluster software
  - Installing and configuring a volume manager (optional)
  - Installing and configuring application software to be cluster ready
  - Installing and configuring Oracle Solaris Cluster data service software
- Cluster administrative procedures for adding, removing, replacing, and servicing cluster hardware and software components
- Configuration modifications to improve performance

The following sections contain material relevant to these key concepts:

- "Administrative Interfaces" on page 38
- "Cluster Time" on page 38
- "High-Availability Framework" on page 39
- "Campus Clusters" on page 39
- "Global Devices" on page 40
- "Device Groups" on page 43
- "Global Namespace" on page 46
- "Cluster File Systems" on page 47
- "Disk Path Monitoring" on page 49
- "Quorum and Quorum Devices" on page 52
- "Load Limits" on page 58"Load Limits" on page 58
- "Data Services" on page 59
- "Using the Cluster Interconnect for Data Service Traffic" on page 68
- "Resources, Resource Groups, and Resource Types" on page 69
- "Support for Oracle Solaris Zones" on page 72

- "Service Management Facility" on page 75
- "System Resource Usage" on page 76

Oracle Solaris Cluster Documentation for System Administrators

The following Oracle Solaris Cluster documents include procedures and information associated with system administration concepts:

- *Oracle Solaris Cluster Software Installation Guide*
- *Oracle Solaris Cluster System Administration Guide*
- *Oracle Solaris Cluster Error Messages Guide*
- *Oracle Solaris Cluster 3.3 3/13 Release Notes*

Application Developer View

The Oracle Solaris Cluster software provides *data services* for web and database applications. Data services are created by configuring off-the-shelf applications to run under control of the Oracle Solaris Cluster software. The Oracle Solaris Cluster software provides configuration files and management methods that start, stop, and monitor the applications. It also provides two kinds of highly available services for applications: failover services and scalable services. For more information, see "Key Concepts – Application Development" on page 15.

If you need to create a new failover or scalable service, you can use the Oracle Solaris Cluster Application Programming Interface (API) and the Data Service Enabling Technologies API (DSET API) to develop the necessary configuration files and management methods that enable the service's application to run as a data service on the cluster. For more information on failover and scalable applications, see the *Oracle Solaris Cluster System Administration Guide*.

Key Concepts – Application Development

Application developers should understand the following concepts:

- How the characteristics of their application help determine whether it can be made to run as a failover or scalable data service.
- The Oracle Solaris Cluster API, DSET API, and the "generic" data service. Developers need to determine which tool is most suitable for them to use to write programs or scripts to configure their application for the cluster environment.
- The relationship between failover and scalable applications and nodes:
  - In a failover application, an application runs on one node at a time. If that node fails, the application fails over to another node in the same cluster.
  - In a scalable application, an application runs on several nodes to create a single, logical service. If a node that is running a scalable application fails, failover does not occur. The application continues to run on the other nodes. For more information, see "Failover Data Services" on page 62 and "Scalable Data Services" on page 63.

The following sections contain material relevant to these key concepts:

- "Data Services" on page 59
- "Resources, Resource Groups, and Resource Types" on page 69

Oracle Solaris Cluster Documentation for Application Developers

The following Oracle Solaris Cluster documents include procedures and information associated with the application developer concepts:

- *Oracle Solaris Cluster Data Services Developer's Guide*
- *Oracle Solaris Cluster Data Services Planning and Administration Guide*

Oracle Solaris Cluster Software Tasks

All Oracle Solaris Cluster software tasks require some conceptual background. The following table provides a high-level view of the tasks and the documentation that describes task steps. The concepts sections in this book describe how the concepts map to these tasks.

TABLE 1–1 Task Map: Mapping User Tasks to Documentation

Task	Instructions
Install cluster hardware	*Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual*
Install Oracle Solaris software on the cluster	*Oracle Solaris Cluster Software Installation Guide*
Install and configure Oracle Solaris Cluster software	*Oracle Solaris Cluster Software Installation Guide*
Install and configure volume management software	*Oracle Solaris Cluster Software Installation Guide* and your volume management documentation
Install and configure Oracle Solaris Cluster data services	*Oracle Solaris Cluster Data Services Planning and Administration Guide*
Service cluster hardware	*Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual*
Administer Oracle Solaris Cluster software	*Oracle Solaris Cluster System Administration Guide*
Administer volume management software	*Oracle Solaris Cluster System Administration Guide* and your volume management documentation
Administer application software	Your application documentation
Problem identification and suggested user actions	*Oracle Solaris Cluster Error Messages Guide*
Create a new data service	*Oracle Solaris Cluster Data Services Developer's Guide*

◆ ◆ ◆ CHAPTER 2

Key Concepts for Hardware Service Providers

This chapter describes the key concepts that are related to the hardware components of an Oracle Solaris Cluster configuration.

This chapter covers the following topics:

Oracle Solaris Cluster System Hardware and Software Components

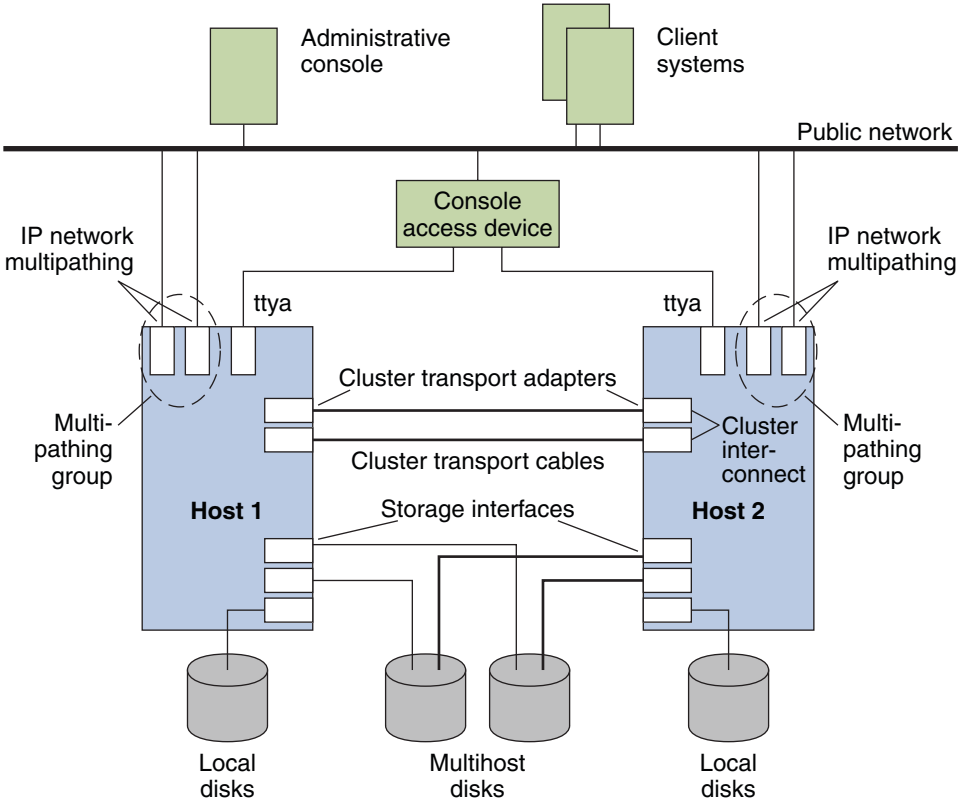
This information is directed primarily to hardware service providers. These concepts can help service providers understand the relationships between hardware components before they install, configure, or service cluster hardware. Cluster system administrators might also find this information useful as background information before installing, configuring, and administering cluster software.

A cluster is composed of several hardware components, including the following:

- Cluster nodes with local disks (unshared)
- Multihost storage (disks/LUNs are shared between cluster nodes)
- Removable media (tapes, CD-ROMs, and DVDs)
- Cluster interconnect
- Public network interfaces
- Administrative console
- Console access devices

The following figure illustrates how the hardware components work with each other.

FIGURE 2–1 Oracle Solaris Cluster Hardware Components



Administrative console and console access devices are used to reach the cluster nodes or the terminal concentrator as needed. The Oracle Solaris Cluster software enables you to combine the hardware components into a variety of configurations. The following sections describe these configurations:

- "SPARC: Oracle Solaris Cluster Topologies" on page 25
- "x86: Oracle Solaris Cluster Topologies" on page 33

Cluster Nodes

An Oracle Solaris *host* (or simply *cluster node*) is one of the following hardware or software configurations that runs the Oracle Solaris OS and its own processes:

- A physical machine that is not configured with a virtual machine or as a hardware domain
- Oracle VM Server for SPARC guest domain
- Oracle VM Server for SPARC I/O domain

- A hardware domain

Depending on your platform, Oracle Solaris Cluster software supports the following configurations:

- SPARC: Oracle Solaris Cluster software supports from 1–16 cluster nodes in a cluster. Different hardware configurations impose additional limits on the maximum number of nodes that you can configure in a cluster composed of SPARC based systems. See "SPARC: Oracle Solaris Cluster Topologies" on page 25 for the supported configurations.
- x86: Oracle Solaris Cluster software supports from 1–8 cluster nodes in a cluster. Different hardware configurations impose additional limits on the maximum number of nodes that you can configure in a cluster composed of x86 based systems. See "x86: Oracle Solaris Cluster Topologies" on page 33 for the supported configurations.

Cluster nodes are generally attached to one or more multihost storage devices. Nodes that are not attached to multihost devices can use a cluster file system to access the data on multihost devices. For example, one scalable services configuration enables nodes to service requests without being directly attached to multihost devices.

In addition, nodes in parallel database configurations share concurrent access to all the disks.

- See "Multihost Devices" on page 21 for information about concurrent access to disks.
- See "SPARC: Clustered Pair Topology" on page 25 and "x86: Clustered Pair Topology" on page 33 for more information about parallel database configurations and scalable topology.

Public network adapters attach nodes to the public networks, providing client access to the cluster.

Cluster members communicate with the other nodes in the cluster through one or more physically independent networks. This set of physically independent networks is referred to as the *cluster interconnect*.

Every node in the cluster is aware when another node joins or leaves the cluster. Additionally, every node in the cluster is aware of the resources that are running locally as well as the resources that are running on the other cluster nodes.

Nodes in the same cluster should have the same OS and architecture, as well as similar processing, memory, and I/O capability to enable failover to occur without significant degradation in performance. Because of the possibility of failover, every node must have enough excess capacity to support the workload of all nodes for which they are a backup or secondary.

Software Components for Cluster Hardware Members

To function as a cluster member, a cluster node must have the following software installed:

- Oracle Solaris OS
- Oracle Solaris Cluster software
- Data service applications
- Optional: Volume management (for example, Solaris Volume Manager)

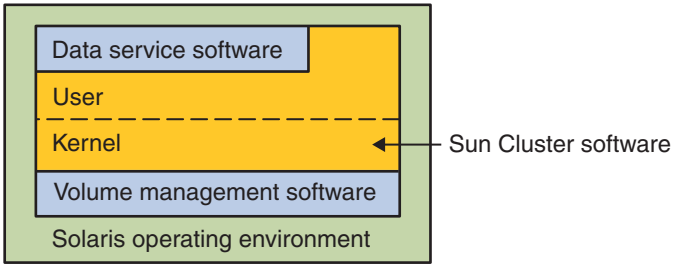
  An exception is a configuration that uses hardware redundant array of independent disks (RAID). This configuration might not require a software volume manager such as Solaris Volume Manager.

For more information, see the following:

- The *Oracle Solaris Cluster Software Installation Guide* for information about how to install the Oracle Solaris OS, Oracle Solaris Cluster, and volume management software.
- The *Oracle Solaris Cluster Data Services Planning and Administration Guide* for information about how to install and configure data services.
- Chapter 3, "Key Concepts for System Administrators and Application Developers," for conceptual information about these software components.

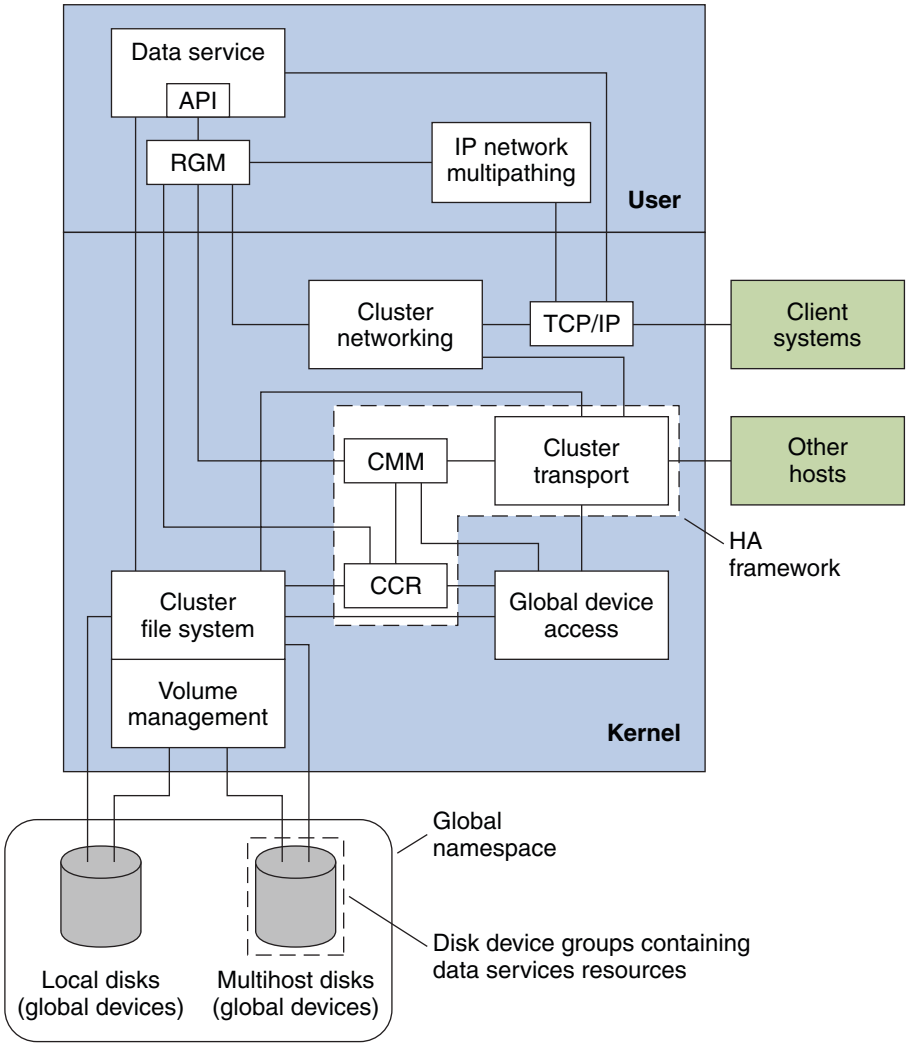
The following figure provides a high-level view of the software components that work together to create the Oracle Solaris Cluster environment.

FIGURE 2–2 High-Level Relationship of Oracle Solaris Cluster Components



The following figure shows a high-level view of the software components that work together to create the Oracle Solaris Cluster software environment.

FIGURE 2–3 Oracle Solaris Cluster Software Architecture



Multihost Devices

LUNs that can be connected to more than one cluster node at a time are multihost devices. A cluster with more than two nodes does not require quorum devices. A *quorum device* is a shared storage device or quorum server that is shared by two or more nodes and that contributes votes that are used to establish a quorum. The cluster can operate only when a quorum of votes is available. For more information about quorum, see "Quorum and Quorum Devices" on page 52.

Multihost devices have the following characteristics:

- Ability to store application data, application binaries, and configuration files.
- Protection against host failures. If clients request the data through one node and the node fails, the I/O requests are handled by the surviving node.

A volume manager can provide software RAID protection for the data residing on the multihost devices.

Combining multihost devices with disk mirroring protects against individual disk failure.

Local Disks

Local disks are the disks that are connected only to a single cluster node. Local disks are therefore not protected against node failure (they are not highly available). However, all disks, including local disks, are included in the global namespace and are configured as *global devices*. Therefore, the disks themselves are visible from all cluster nodes.

See "Global Devices" on page 40 for more information about global devices.

Removable Media

Removable media, such as tape drives and CD-ROM drives, are supported in a cluster. You install, configure, and service these devices in the same way as in a nonclustered environment. Refer to *Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual* for information about installing and configuring removable media.

See the "Global Devices" on page 40 section for more information about global devices.

Cluster Interconnect

The *cluster interconnect* is the physical configuration of devices that is used to transfer cluster-private communications and data service communications between cluster nodes in the cluster.

Only nodes in the cluster can be connected to the cluster interconnect. The Oracle Solaris Cluster security model assumes that only cluster nodes have physical access to the cluster interconnect.

You can set up from one to six cluster interconnects in a cluster. While a single cluster interconnect reduces the number of adapter ports that are used for the private interconnect, it provides no redundancy and less availability. If a single interconnect fails, moreover, the cluster is at a higher risk of having to perform automatic recovery. Whenever possible, install two or more cluster interconnects to provide redundancy and scalability, and therefore higher availability, by avoiding a single point of failure.

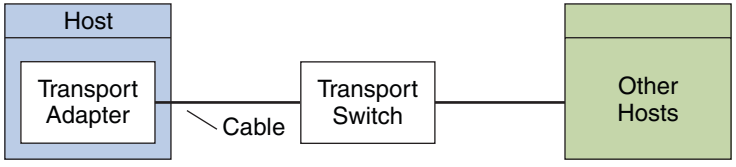
The cluster interconnect consists of three hardware components: adapters, junctions, and cables. The following list describes each of these hardware components.

- Adapters – The network interface cards that are located in each cluster node. Their names are constructed from a device name immediately followed by a physical-unit number, for example, qfe2. Some adapters have only one physical network connection, but others, like the qfe card, have multiple physical connections. Some adapters combine both the functions of a NIC and an HBA.

  A network adapter with multiple interfaces could become a single point of failure if the entire adapter fails. For maximum availability, plan your cluster so that the paths between two nodes do not depend on a single network adapter.
- Junctions – The switches that are located outside of the cluster nodes. In a two-node cluster, junctions are not mandatory. In that case, the nodes can be connected to each other through back-to-back network cable connections. Configurations of more than two nodes generally require junctions.
- Cables – The physical connections that you install either between two network adapters or between an adapter and a junction.

Figure 2–4 shows how the two nodes are connected by a transport adapter, cables, and a transport switch.

FIGURE 2–4 Cluster Interconnect



Public Network Interfaces

Clients connect to the cluster through the public network interfaces.

You can set up cluster nodes in the cluster to include multiple public network interface cards that perform the following functions:

- Enable a cluster node to be connected to multiple subnets
- Provide public network availability by having interfaces acting as backups for one another (through IPMP)

If one of the adapters fails, IP network multipathing software is called to fail over the defective interface to another adapter in the group. For more information about IPMP, see Chapter 27, "Introducing IPMP (Overview)," in *Oracle Solaris Administration: IP Services*.

No special hardware considerations relate to clustering for the public network interfaces.

Logging Into the Cluster Remotely

You must have console access to all nodes in the cluster.

To gain console access, use one of the following methods:

- The `cconsole` utility can be used from the command line to log into the cluster remotely. For more information, see the `cconsole(1M)` man page.
- The terminal concentrator that you purchased with your cluster hardware.
- The system controller on Oracle servers, such as Sun Fire servers (also for SPARC based clusters).
- Another device that can access `ttya` on each node.

Only one supported terminal concentrator is available from Oracle and use of the supported Sun terminal concentrator is optional. The terminal concentrator enables access to `/dev/console` on each node by using a TCP/IP network. The result is console-level access for each node from a remote machine anywhere on the network.

Other console access methods include other terminal concentrators, `tip` serial port access from another node, and dumb terminals.

Caution – You can attach a keyboard or monitor to a cluster node provided that the keyboard and monitor are supported by the base server platform. However, you cannot use that keyboard or monitor as a console device. You must redirect the console to a serial port and Remote System Control (RSC) by setting the appropriate OpenBoot PROM parameter.

Administrative Console

You can use a dedicated workstation or administrative console to reach the cluster nodes or the terminal concentrator as needed to administer the active cluster. Usually, you install and run administrative tool software, such as the Cluster Control Panel (CCP), on the administrative console. Using `cconsole` under the CCP enables you to connect to more than one node console at a time. For more information about how to use the CCP, see the Chapter 1, "Introduction to Administering Oracle Solaris Cluster," in *Oracle Solaris Cluster System Administration Guide*.

You use the administrative console for remote access to the cluster nodes, either over the public network or, optionally, through a network-based terminal concentrator.

Oracle Solaris Cluster does not require a dedicated administrative console, but using one provides these benefits:

- Enables centralized cluster management by grouping console and management tools on the same machine
- Provides potentially quicker problem resolution by your hardware service provider

SPARC: Oracle Solaris Cluster Topologies

A topology is the connection scheme that connects the Oracle Solaris nodes in the cluster to the storage platforms that are used in an Oracle Solaris Cluster environment. Oracle Solaris Cluster software supports any topology that adheres to the following guidelines:

- An Oracle Solaris Cluster environment that is composed of SPARC based systems supports from 1–16 cluster nodes in a cluster. Different hardware configurations impose additional limits on the maximum number of nodes that you can configure in a cluster composed of SPARC based systems.
- A shared storage device can connect to as many nodes as the storage device supports.
- Shared storage devices do not need to connect to all nodes of the cluster. However, these storage devices must connect to at least two nodes.

You can configure Oracle VM Server for SPARC software guest domains and I/O domains as cluster nodes. In other words, you can create a clustered pair, pair+N, N+1, and N*N cluster that consists of any combination of physical machines, I/O domains, and guest domains. You can also create clusters that consist of only guest domains and I/O domains.

Oracle Solaris Cluster software does not require you to configure a cluster by using specific topologies. The following topologies are described to provide the vocabulary to discuss a cluster's connection scheme:

- "SPARC: Clustered Pair Topology" on page 25
- "SPARC: Pair+N Topology" on page 26
- "SPARC: N+1 (Star) Topology" on page 27
- "SPARC: N*N (Scalable) Topology" on page 28
- "SPARC: Oracle VM Server for SPARC Software Guest Domains: Cluster in a Box Topology" on page 29
- "SPARC: Oracle VM Server for SPARC Software Guest Domains: Clusters Span Two Different Hosts Topology" on page 30
- "SPARC: Oracle VM Server for SPARC Software Guest Domains: Redundant I/O Domains" on page 32

The following sections include sample diagrams of each topology.

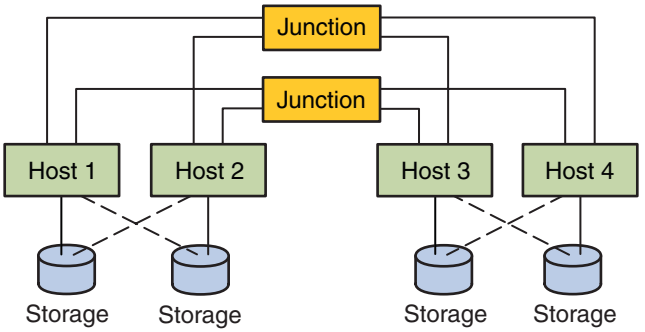
SPARC: Clustered Pair Topology

A clustered pair topology is two or more pairs of Oracle Solaris nodes that operate under a single cluster administrative framework. In this configuration, failover occurs only between a pair. However, all nodes are connected by the cluster interconnect and operate under Oracle Solaris Cluster software control. You might use this topology to run a parallel database application on one pair and a failover or scalable application on another pair.

Using the cluster file system, you could also have a two-pair configuration. More than two nodes can run a scalable service or parallel database, even though all the nodes are not directly connected to the disks that store the application data.

The following figure illustrates a clustered pair configuration.

FIGURE 2–5 SPARC: Clustered Pair Topology



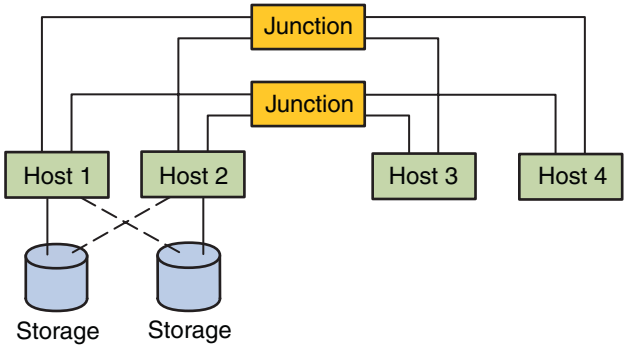
SPARC: Pair+N Topology

The pair+N topology includes a pair of cluster nodes that are directly connected to the following:

- Shared storage
- An additional set of nodes that use the cluster interconnect to access shared storage (they have no direct connection themselves)

The following figure illustrates a pair+N topology where two of the four nodes (Host 3 and Host 4) use the cluster interconnect to access the storage. This configuration can be expanded to include additional nodes that do not have direct access to the shared storage.

FIGURE 2–6 Pair+N Topology



SPARC: N+1 (Star) Topology

An N+1 topology includes some number of primary cluster nodes and one secondary node. You do not have to configure the primary nodes and secondary node identically. The primary nodes actively provide application services. The secondary node need not be idle while waiting for a primary node to fail.

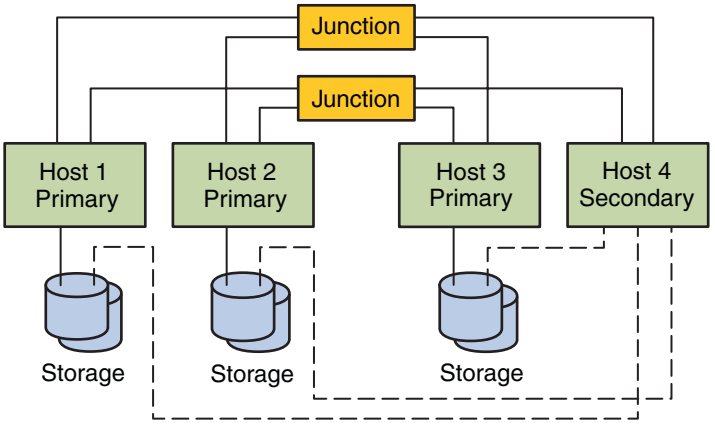
The secondary node is the only node in the configuration that is physically connected to all the multihost storage.

If a failure occurs on a primary node, Oracle Solaris Cluster fails over the resources to the secondary node. The secondary node is where the resources function until they are switched back (either automatically or manually) to the primary node.

The secondary node must always have enough excess CPU capacity to handle the load if one of the primary nodes fails.

The following figure illustrates an N+1 configuration.

FIGURE 2–7 SPARC: N+1 Topology

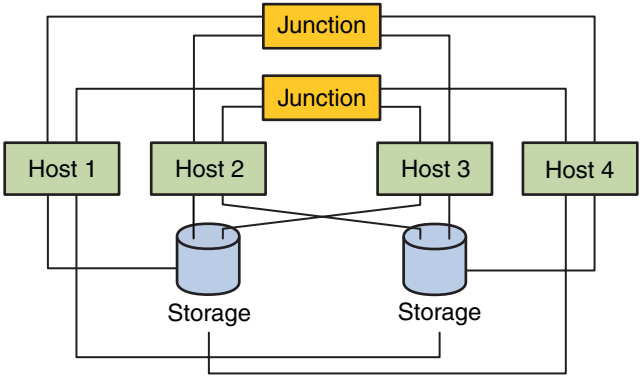


SPARC: N*N (Scalable) Topology

An N*N topology enables every shared storage device in the cluster to connect to every cluster node in the cluster. This topology enables highly available applications to fail over from one node to another without service degradation. When failover occurs, the new node can access the storage device by using a local path instead of the private interconnect.

The following figure illustrates an N*N configuration.

FIGURE 2–8 SPARC: N*N Topology



SPARC: Oracle VM Server for SPARC Software Guest Domains: Cluster in a Box Topology

In this Oracle VM Server for SPARC guest domain topology, a cluster and every node within that cluster are located on the same Oracle Solaris host. Each guest domain acts the same as a node in a cluster. This configuration includes three nodes rather than only two.

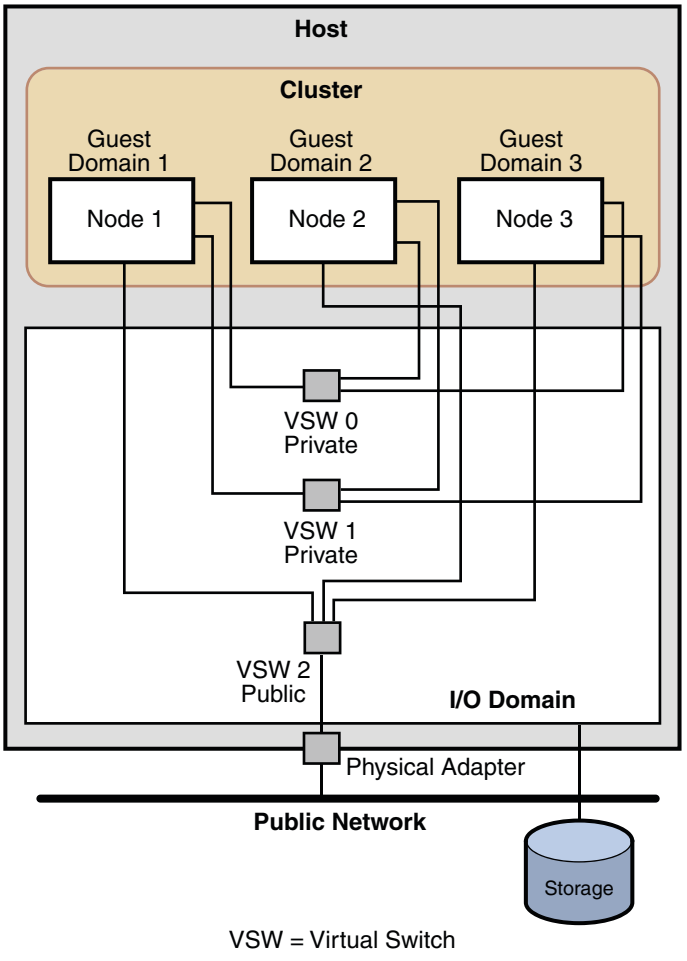
In this topology, you do not need to connect each virtual switch (VSW) for the private network to a physical network because they need only communicate with each other. In this topology, cluster nodes can also share the same storage device because all cluster nodes are located on the same host or box. To learn more about guidelines for using and installing guest domains or I/O domains in a cluster, see "How to Install Oracle VM Server for SPARC Software and Create Domains" in *Oracle Solaris Cluster Software Installation Guide*.

Caution – The common host or box in this topology represents a single point of failure.

All nodes in the cluster are located on the same host or box. Developers and administrators might find this topology useful for testing and other non-production tasks. This topology is also called a "cluster in a box." Multiple clusters can share the same physical host or box.

The following figure illustrates a cluster in a box configuration.

FIGURE 2–9 SPARC: Cluster in a Box Topology



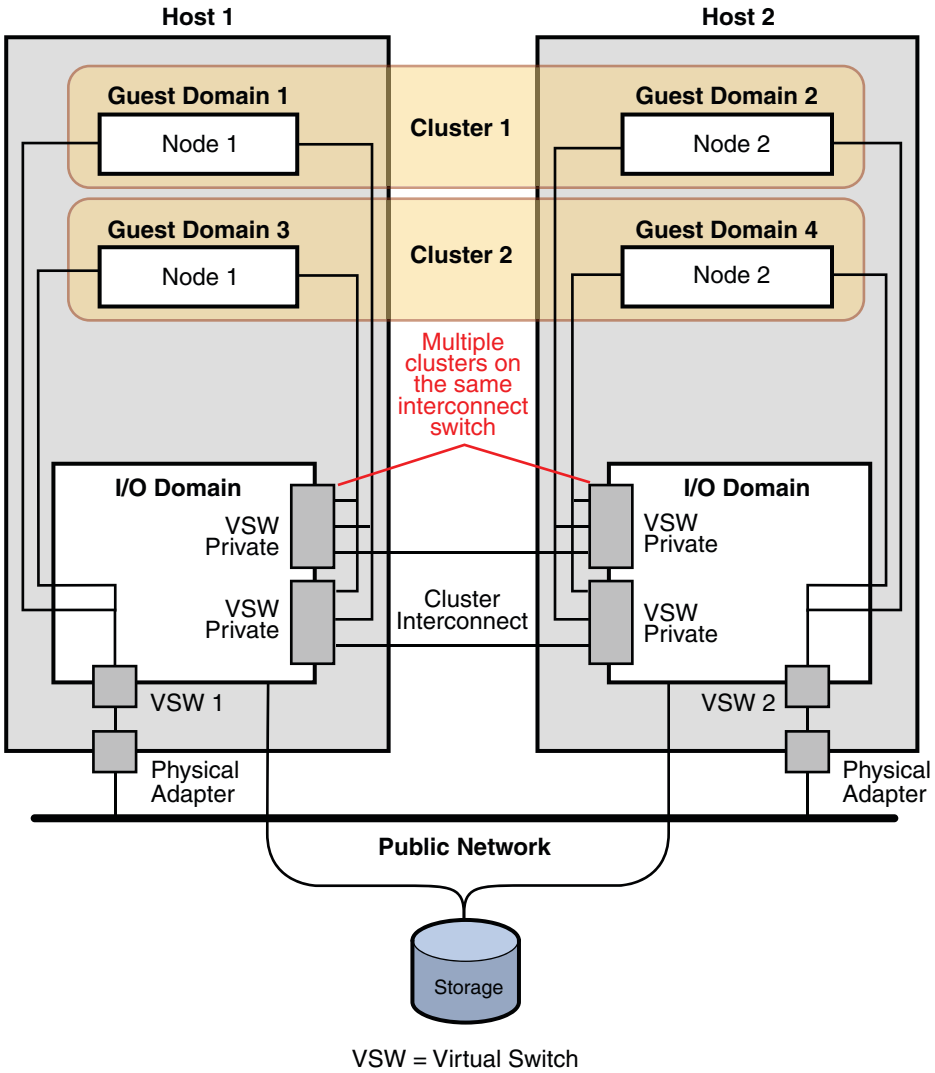
SPARC: Oracle VM Server for SPARC Software Guest Domains: Clusters Span Two Different Hosts Topology

In this Oracle VM Server for SPARC software guest domain topology, each cluster spans two different hosts and each cluster has one host. Each guest domain acts the same as a host in a cluster. In this configuration, because both clusters share the same interconnect switch, you must specify a different private network address on each cluster. If you specify the same private network address on clusters that share an interconnect switch, the configuration fails.

To learn more about guidelines for using and installing Oracle VM Server for SPARC software guest domains or I/O domains in a cluster, see "How to Install Oracle VM Server for SPARC Software and Create Domains" in *Oracle Solaris Cluster Software Installation Guide*.

The following figure illustrates a configuration in which more than a single cluster spans two different hosts.

FIGURE 2–10 SPARC: Clusters Span Two Different Hosts



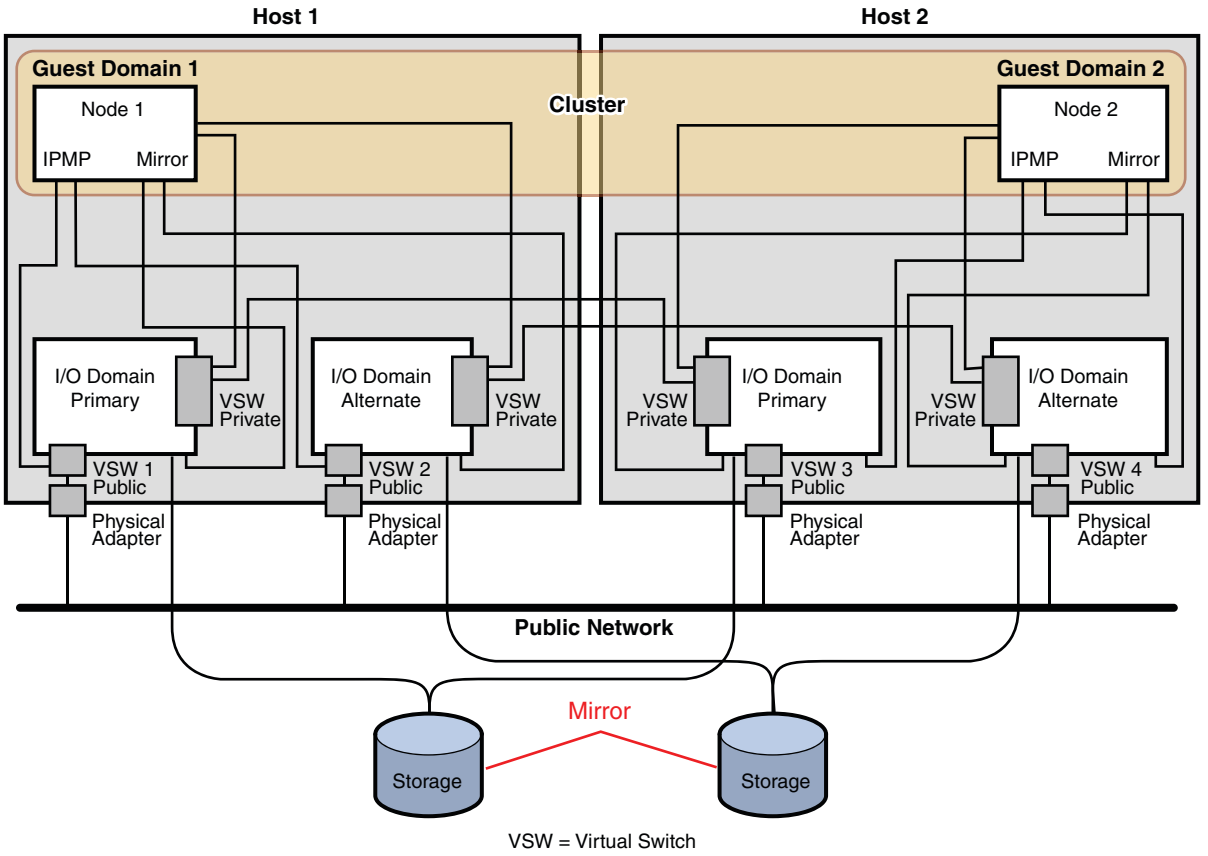
SPARC: Oracle VM Server for SPARC Software Guest Domains: Redundant I/O Domains

In this Oracle VM Server for SPARC software guest domain topology, multiple I/O domains ensure that guest domains, which are configured as cluster nodes, continue to operate if an I/O domain fails. Each guest domain node acts the same as a cluster node in a cluster.

To learn more about guidelines for using and installing guest domains or I/O domains in a cluster, see "How to Install Oracle VM Server for SPARC Software and Create Domains" in *Oracle Solaris Cluster Software Installation Guide*.

The following figure illustrates a configuration in which redundant I/O domains ensure that nodes within the cluster continue to operate if an I/O domain fails.

FIGURE 2–11 SPARC: Redundant I/O Domains



x86: Oracle Solaris Cluster Topologies

A topology is the connection scheme that connects the cluster nodes to the storage platforms that are used in the cluster. Oracle Solaris Cluster supports any topology that adheres to the following guidelines:

- Oracle Solaris Cluster software supports from one to eight cluster nodes in a cluster. Different hardware configurations impose additional limits on the maximum number of nodes that you can configure in a cluster composed of x86 based systems. The next sections describe the supported node configurations.
- Shared storage devices must connect to nodes.

Oracle Solaris Cluster does not require you to configure a cluster by using specific topologies. The following clustered pair topology, which is a topology for clusters that are composed of x86 based nodes, is described to provide the vocabulary to discuss a cluster's connection scheme. This topology is a typical connection scheme.

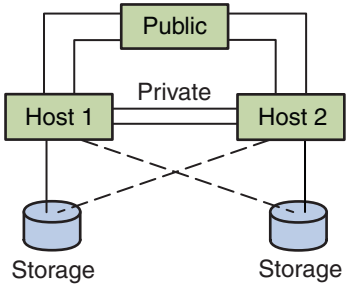
The following section includes a sample diagram of the topology.

x86: Clustered Pair Topology

A clustered pair topology consists of two cluster nodes that operate under a single cluster administrative framework. In this configuration, failover occurs only between a pair. However, all nodes are connected by the cluster interconnect and operate under Oracle Solaris Cluster software control. You might use this topology to run a parallel database or a failover or scalable application on the pair.

The following figure illustrates a clustered pair configuration.

FIGURE 2–12 x86: Clustered Pair Topology



x86: N+1 (Star) Topology

An N+1 topology includes some number of primary cluster nodes and one secondary node. You do not have to configure the primary nodes and secondary node identically. The primary nodes actively provide application services. The secondary node need not be idle while waiting for a primary node to fail.

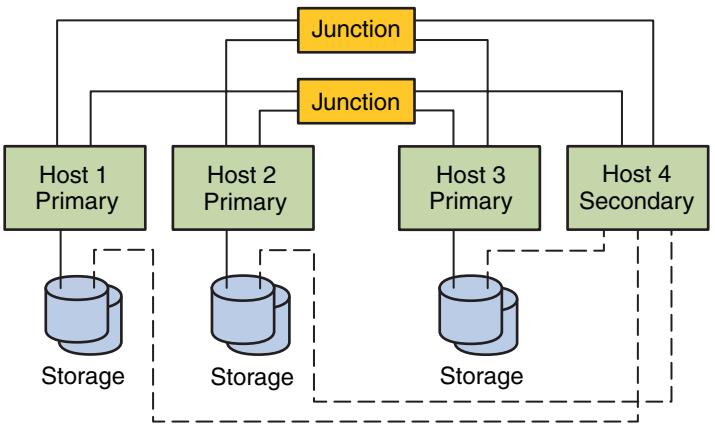
The secondary node is the only node in the configuration that is physically connected to all the multihost storage.

If a failure occurs on a primary node, Oracle Solaris Cluster fails over the resources to the secondary node. The secondary node is where the resources function until they are switched back (either automatically or manually) to the primary node.

The secondary node must always have enough excess CPU capacity to handle the load if one of the primary nodes fails.

The following figure illustrates an N+1 configuration.

FIGURE 2–13 x86: N+1 Topology

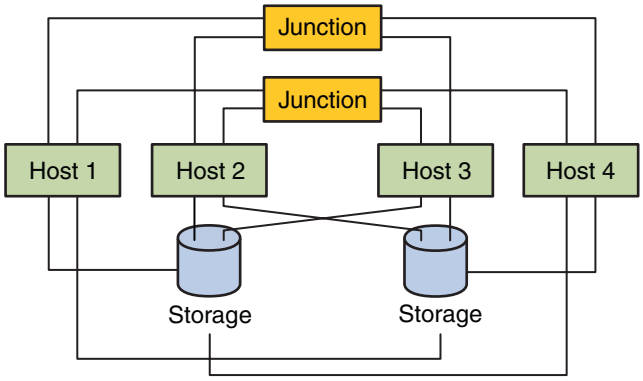


N*N (Scalable Topology)

An N*N topology enables every shared storage device in the cluster to connect to every cluster node in the cluster. This topology enables highly available applications to fail over from one node to another without service degradation. When failover occurs, the new node can access the storage device by using a local path instead of the private interconnect.

The following figure illustrates an N*N configuration.

FIGURE 2–14 N*N Topology



◆ ◆ ◆ CHAPTER 3

Key Concepts for System Administrators and Application Developers

This chapter describes the key concepts that are related to the software components of the Oracle Solaris Cluster environment. The information in this chapter is directed to system administrators and application developers who use the Oracle Solaris Cluster API. Cluster administrators can use this information in preparation for installing, configuring, and administering cluster software. Application developers can use the information to understand the cluster environment in which they work.

This chapter covers the following topics:

Administrative Interfaces

You can choose how you install, configure, and administer the Oracle Solaris Cluster software from several user interfaces. You can perform system administration tasks either through the Oracle Solaris Cluster Manager graphical user interface (GUI) or through the command-line interface. Some utilities on top of the command-line interface, such as `scinstall` and `clsetup`, simplify selected installation and configuration tasks. Refer to "Administration Tools" in *Oracle Solaris Cluster System Administration Guide* for more information about the administrative interfaces.

Cluster Time

Time between all Oracle Solaris nodes in a cluster must be synchronized. Whether you synchronize the cluster nodes with any outside time source is not important to cluster operation. The Oracle Solaris Cluster software employs the Network Time Protocol (NTP) to synchronize the clocks between nodes.

A change in the system clock of a fraction of a second generally causes no problems. However, if you run `date` or `rdate` on an active cluster, you can force a time change much larger than a fraction of a second to synchronize the system clock to the time source. This forced change might cause problems with file modification timestamps or confuse the NTP service.

When you install the Oracle Solaris OS on each cluster node, you have an opportunity to change the default time and date setting for the node. You can accept the factory default.

When you install Oracle Solaris Cluster software by using the `scinstall` command, one step in the process is to configure NTP for the cluster. Oracle Solaris Cluster software supplies two template files, `etc/inet/ntp.conf` and `etc/inet/ntp.conf.sc`, that establish a peer relationship between all cluster nodes. One node is designated the "preferred" node. Nodes are identified by their private host names and time synchronization occurs across the cluster interconnect. For instructions about how to configure the cluster for NTP, see Chapter 2, "Installing Software on Global-Cluster Nodes," in *Oracle Solaris Cluster Software Installation Guide*.

Alternately, you can set up one or more NTP servers outside the cluster and change the `ntp.conf` file to reflect that configuration.

In normal operation, you should never need to adjust the time on the cluster. However, if the time was set incorrectly when you installed the Oracle Solaris Operating System and you want to change it, the procedure for doing so is included in Chapter 9, "Administering the Cluster," in *Oracle Solaris Cluster System Administration Guide*.

Campus Clusters

Standard Oracle Solaris Cluster systems provide high availability and reliability from a single location. If your application must remain available after unpredictable disasters such as an earthquake, flood, or power outage, you can configure your cluster as a campus cluster.

Campus clusters enable you to locate cluster components, such as cluster nodes and shared storage, in separate rooms that are several kilometers apart. You can separate your nodes and shared storage and locate them in different facilities around your corporate campus or elsewhere within several kilometers. When a disaster strikes one location, the surviving nodes can take over service for the failed node. This enables applications and data to remain available for your users. For additional information about campus cluster configurations, see the *Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual*.

High-Availability Framework

The Oracle Solaris Cluster software makes all components on the "path" between users and data highly available, including network interfaces, the applications themselves, the file system, and the multihost devices. A cluster component is generally highly available if it survives any single (software or hardware) failure in the system. Failures that are caused by data corruption within the application itself are excluded.

The following table shows types of Oracle Solaris Cluster component failures (both hardware and software) and the kinds of recovery that are built into the high-availability framework.

TABLE 3–1 Levels of Oracle Solaris Cluster Failure Detection and Recovery

Failed Cluster Component	Software Recovery	Hardware Recovery
Data service	HA API, HA framework	Not applicable
Public network adapter	IP network multipathing	Multiple public network adapter cards
Cluster file system	Primary and secondary replicas	Multihost devices
Mirrored multihost device	Volume management (Solaris Volume Manager)	Hardware RAID-5
Global device	Primary and secondary replicas	Multiple paths to the device, cluster transport junctions
Private network	HA transport software	Multiple private hardware-independent networks
Node	CMM, failfast driver	Multiple nodes
Zone	HA API, HA framework	Not applicable

The Oracle Solaris Cluster software's high-availability framework detects a node failure quickly and migrates the framework resources on a remaining node in the cluster. At no time are all framework resources unavailable. Framework resources on a failed node are fully available during recovery. Furthermore, framework resources of the failed node become available as soon as they are recovered. A recovered framework resource does not have to wait for all other framework resources to complete their recovery.

Highly available framework resources are recovered transparently to most of the applications (data services) that are using the resource. The semantics of framework resource access are fully preserved across node failure. The applications cannot detect that the framework resource server has been moved to another node. Failure of a single node is completely transparent to programs on remaining nodes because they use the files, devices, and disk volumes that are available to the recovery node. This transparency exists if an alternative hardware path exists to the disks from another node. An example is the use of multihost devices that have ports to multiple nodes.

Global Devices

The Oracle Solaris Cluster software uses *global devices* to provide cluster-wide, highly available access to any device in a cluster from any node. If a node fails while providing access to a global device, the Oracle Solaris Cluster software automatically uses another path to the device. The Oracle Solaris Cluster software then redirects the access to that path. For more information, see "Device IDs and DID Pseudo Driver" on page 40. Oracle Solaris Cluster global devices include disks, CD-ROMs, and tapes. However, the only multiported global devices that Oracle Solaris Cluster software supports are disks. Consequently, CD-ROM and tape devices are not currently highly available devices. The local disks on each server are also not multiported, and thus are not highly available devices.

The cluster automatically assigns unique IDs to each disk, CD-ROM, and tape device in the cluster. This assignment enables consistent access to each device from any node in the cluster. The global device namespace is held in the `/dev/global` directory. See "Global Namespace" on page 46 for more information.

Multiported global devices provide more than one path to a device. Because multihost disks are part of a device group that is hosted by more than one cluster node, the multihost disks are made highly available.

Device IDs and DID Pseudo Driver

The Oracle Solaris Cluster software manages shared devices through a construct known as the Device ID (DID) pseudo driver. This driver is used to automatically assign unique IDs to every device in the cluster, including multihost disks, tape drives, and CD-ROMs.

The DID pseudo driver is an integral part of the shared device access feature of the cluster. The DID driver probes all nodes of the cluster and builds a list of unique devices, assigning to each

device a unique major and a minor number that are consistent on all nodes of the cluster. Access to shared devices is performed by using the normalized DID logical name instead of the traditional Oracle Solaris logical name, such as `c0t0d0` for a disk.

This approach ensures that any application that accesses disks (such as a volume manager or applications that use raw devices) uses a consistent path across the cluster. This consistency is especially important for multihost disks because the local major and minor numbers for each device can vary from node to node, thus changing the Oracle Solaris device naming conventions as well. For example, `Host1` might identify a multihost disk as `c1t2d0`, and `Host2` might identify the same disk completely differently as `c3t2d0`. The DID framework assigns a common (normalized) logical name, such as `d10`, that the nodes use instead, giving each node a consistent mapping to the multihost disk.

You update and administer device IDs with the `cldevice` command. See the `cldevice(1CL)` man page.

Zone Membership

Oracle Solaris Cluster software also tracks zone membership by detecting when a zone boots up or halts. These changes also trigger a reconfiguration. A reconfiguration can redistribute cluster resources among the nodes in the cluster.

Cluster Membership Monitor

To ensure that data is kept safe from corruption, all nodes must reach a consistent agreement on the cluster membership. When necessary, the CMM coordinates a cluster reconfiguration of cluster services (applications) in response to a failure.

The CMM receives information about connectivity to other nodes from the cluster transport layer. The CMM uses the cluster interconnect to exchange state information during a reconfiguration. A problem called *split brain* can occur when the cluster interconnect between cluster nodes is lost. The cluster becomes partitioned into subclusters, and each subcluster is no longer aware of other subclusters. A subcluster that is not aware of the other subclusters could cause a conflict in shared resources, such as duplicate network addresses and data corruption. The quorum subsystem manages the situation to ensure that split brain does not occur, and that one partition survives. For more information, see "Quorum and Quorum Devices" on page 52.

After detecting a change in cluster membership, the CMM performs a synchronized configuration of the cluster. In a synchronized configuration, cluster resources might be redistributed, based on the new membership of the cluster.

Failfast Mechanism

The *failfast* mechanism detects a critical problem on either a global-cluster voting node or global-cluster non-voting node. The action that Oracle Solaris Cluster takes when failfast detects a problem depends on whether the problem occurs in a voting node or a non-voting node.

If the critical problem is located in a voting node, Oracle Solaris Cluster forcibly shuts down the node. Oracle Solaris Cluster then removes the node from cluster membership.

If the critical problem is located in a non-voting node, Oracle Solaris Cluster reboots that non-voting node.

If a node loses connectivity with other nodes, the node attempts to form a cluster with the nodes with which communication is possible. If that set of nodes does not form a quorum, Oracle Solaris Cluster software halts the node and "fences" the node from the shared disks, that is, prevents the node from accessing the shared disks. Fencing is a mechanism that is used by the cluster to protect the data integrity of a shared disk during split-brain situations. By default, the `scinstall` utility in Typical Mode leaves global fencing enabled.

You can turn off fencing for selected disks or for all disks.

Caution – If you turn off fencing under the wrong circumstances, your data can be vulnerable to corruption during application failover. Examine this data corruption possibility carefully when you are considering turning off fencing. If your shared storage device does not support the SCSI protocol, such as a Serial Advanced Technology Attachment (SATA) disk, or if you want to allow access to the cluster's storage from nodes outside the cluster, turn off fencing.

If one or more cluster-specific daemons die, Oracle Solaris Cluster software declares that a critical problem has occurred. Oracle Solaris Cluster software runs cluster-specific daemons on both voting nodes and non-voting nodes. If a critical problem occurs, Oracle Solaris Cluster either shuts down and removes the node or reboots the non-voting node where the problem occurred.

When a cluster-specific daemon that runs on a non-voting node fails, a message similar to the following is displayed on the console:

```
cl_runtime: NOTICE: Failfast: Aborting because "pmfd" died in zone "zone4" (zone id 3)
35 seconds ago.
```

When a cluster-specific daemon that runs on a voting node fails and the node panics, a message similar to the following is displayed on the console:

```
panic[cpu1]/thread=2a10007fcc0: Failfast: Aborting because "pmfd" died in zone "global" (zone id 0)
35 seconds ago.
409b8 cl_runtime:__0FZsc_syslog_msg_log_no_argsPviTCPCcTB+48 (70f900, 30, 70df54, 407acc, 0)
%l0-7: 1006c80 000000a 000000a 10093bc 406d3c80 7110340 0000000 4001 fbf0
```

After the panic, the node might reboot and the node might attempt to rejoin the cluster. Alternatively, if the cluster is composed of SPARC based systems, the node might remain at the OpenBoot PROM (OBP) prompt. The next action of the node is determined by the setting of the `auto-boot?` parameter. You can set `auto-boot?` with the `eeprom` command at the OpenBoot PROM `ok` prompt. See the `eeprom(1M)` man page.

Cluster Configuration Repository (CCR)

To maintain an accurate representation of an Oracle Solaris Cluster configuration, clustered systems require configuration control. The Oracle Solaris Cluster software uses a Cluster Configuration Repository (CCR) to store the current cluster configuration information. The CCR uses a two-phase commit algorithm for updates: An update must be successfully completed on all cluster members or the update is rolled back. The CCR uses the cluster interconnect to apply the distributed updates.

Caution – Although the CCR consists of text files, never edit the CCR files yourself. Each file contains a checksum record to ensure consistency between nodes. Updating CCR files yourself can cause a node or the entire cluster to stop working.

The CCR relies on the CMM to guarantee that a cluster is running only when quorum is established. The CCR is responsible for verifying data consistency across the cluster, performing recovery as necessary, and facilitating updates to the data.

Device Groups

In the Oracle Solaris Cluster software, all multihost devices must be under control of the Oracle Solaris Cluster software. The Oracle Solaris Cluster software automatically creates a raw device group for each disk and tape device in the cluster. However, these cluster device groups remain in an offline state until you access them as global devices.

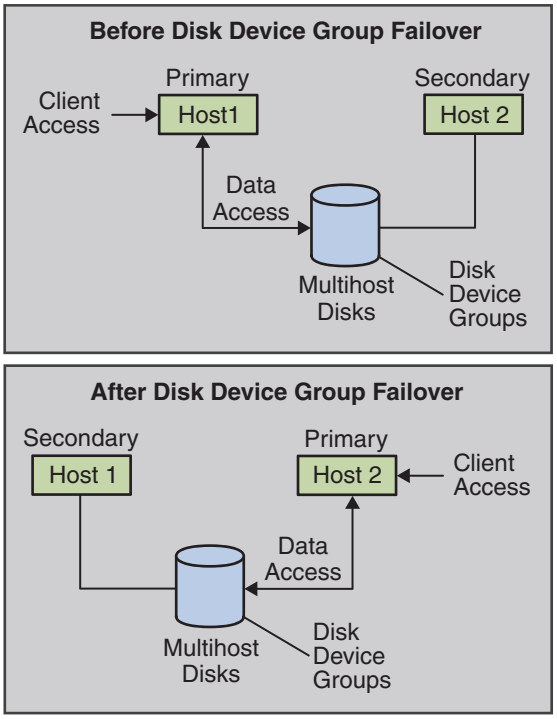
Note – Device groups are independent of resource groups. One node can master a resource group (representing a group of data service processes). Another node can master the device groups that are being accessed by the data services. However, the best practice is to keep on the same node the device group that stores a particular application's data and the resource group that contains the application's resources (the application daemon). Refer to "Relationship Between Resource Groups and Device Groups" in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for more information about the association between device groups and resource groups.

Each cluster node that is physically attached to the multihost disks provides a path to the device group.

Device Group Failover

Because a disk enclosure is connected to more than one cluster node, all device groups in that enclosure are accessible through an alternate path if the node currently mastering the device group fails. The failure of the node that is mastering the device group does not affect access to the device group except for the time it takes to perform the recovery and consistency checks. During this time, all requests are blocked (transparently to the application) until the system makes the device group available.

FIGURE 3–1 Device Group Before and After Failover



Device Group Ownership

This section describes device group properties that enable you to balance performance and availability in a multiported disk configuration. Oracle Solaris Cluster software provides two properties that configure a multiported disk configuration: `preferenced` and `numsecondaries`. You can control the order in which nodes attempt to assume control if a failover occurs by using the `preferenced` property. Use the `numsecondaries` property to set the number of secondary nodes for a device group that you want.

A highly available service is considered down when the primary node fails and when no eligible secondary nodes can be promoted to primary nodes. If service failover occurs and the `preferenced` property is `true`, then the nodes follow the order in the node list to select a secondary node. The node list defines the order in which nodes attempt to assume primary control or transition from spare to secondary. You can dynamically change the preference of a device service by using the `clsetup` command. The preference that is associated with dependent service providers, for example, a global file system, is identical to the preference of the device service.

Secondary nodes are check-pointed by the primary node during normal operation. In a multiported disk configuration, checkpointing each secondary node causes cluster performance degradation and memory overhead. Spare node support was implemented to minimize the performance degradation and memory overhead that checkpointing caused. By default, your device group has one primary and one secondary. The remaining available provider nodes become spares. If failover occurs, the secondary becomes primary and the node highest in priority on the node list becomes secondary.

You can set the number of secondary nodes that you want to any integer between one and the number of operational nonprimary provider nodes in the device group.

Note – If you are using Solaris Volume Manager, you must create the device group first. Use the `metaset` command before you use the `cldevicegroup` command to set the `numsecondaries` property.

The default number of secondaries for device services is 1. The actual number of secondary providers that is maintained by the replica framework is the number that you want, unless the number of operational nonprimary providers is less than the number that you want. You must alter the `numsecondaries` property and double-check the node list if you are adding or removing nodes from your configuration. Maintaining the node list and number of secondaries prevents conflict between the configured number of secondaries and the actual number that is allowed by the framework.

- Use the `cldevicegroup` command for Solaris Volume Manager device groups, in conjunction with the `preferenced` and `numsecondaries` property settings, to manage the addition of nodes to and the removal of nodes from your configuration.
- Refer to "Overview of Administering Cluster File Systems" in *Oracle Solaris Cluster System Administration Guide* for procedural information about changing device group properties.

Global Namespace

The Oracle Solaris Cluster software mechanism that enables global devices is the *global namespace*. The global namespace includes the `/dev/global/` hierarchy as well as the volume manager namespaces. The global namespace reflects both multihost disks and local disks (and any other cluster device, such as CD-ROMs and tapes). Each cluster node that is physically connected to multihost disks provides a path to the storage for any node in the cluster.

For Solaris Volume Manager, the volume manager namespaces are located in the `/dev/md/`*diskset*`/dsk` (and `rdsk`) directories. These namespaces consist of directories for each Solaris Volume Manager disk set imported throughout the cluster.

In the Oracle Solaris Cluster software, each device host in the local volume manager namespace is replaced by a symbolic link to a device host in the `/global/.devices/node@`*nodeID* file system. *nodeID* is an integer that represents the nodes in the cluster. Oracle Solaris Cluster software continues to present the volume manager devices as symbolic links in their standard locations as well. Both the global namespace and standard volume manager namespace are available from any cluster node.

The advantages of the global namespace include the following:

- Each host remains fairly independent, with little change in the device administration model.
- Third-party generated device trees are still valid and continue to work.
- Given a local device name, an easy mapping is provided to obtain its global name.

Local and Global Namespaces Example

The following table shows the mappings between the local and global namespaces for a multihost disk, `c0t0d0s0`.

TABLE 3–2 Local and Global Namespace Mappings

Component or Path	Local Host Namespace	Global Namespace
Oracle Solaris logical name	`/dev/dsk/c0t0d0s0`	`/global/.devices/node@`*nodeID*`/dev/dsk/c0t0d0s0`
DID name	`/dev/did/dsk/d0s0`	`/global/.devices/node@`*nodeID*`/dev/did/dsk/d0s0`
Solaris Volume Manager	`/dev/md/`*diskset*`/dsk/d0`	`/global/.devices/node@`*nodeID*`/dev/md/`*diskset*`/dsk/d0`

The global namespace is automatically generated on installation and updated with every reconfiguration reboot. You can also generate the global namespace by using the `cldevice` command. See the `cldevice(1CL)` man page.

Cluster File Systems

Oracle Solaris Cluster software provides a cluster file system based on the Oracle Solaris Cluster Proxy File System (PxFS). The cluster file system has the following features:

- File access locations are transparent. A process can open a file that is located anywhere in the system. Processes on all cluster nodes can use the same path name to locate a file.

  Note – When the cluster file system reads files, it does not update the access time on those files.

- Coherency protocols are used to preserve the UNIX file access semantics even if the file is accessed concurrently from multiple nodes.
- Extensive caching is used along with zero-copy bulk I/O movement to move file data efficiently.
- The cluster file system provides highly available, advisory file-locking functionality by using the `fcntl` command interfaces. Applications that run on multiple cluster nodes can synchronize access to data by using advisory file locking on a cluster file system. File locks are recovered immediately from nodes that leave the cluster and from applications that fail while holding locks.
- Continuous access to data is ensured, even when failures occur. Applications are not affected by failures if a path to disks is still operational. This guarantee is maintained for raw disk access and all file system operations.
- Cluster file systems are independent from the underlying file system and volume management software.

You can mount a file system on a global device globally with `mount -g` or locally with `mount`.

Programs can access a file in a cluster file system from any node in the cluster through the same file name (for example, `/global/foo`).

A cluster file system is mounted on all cluster members. You cannot mount a cluster file system on a subset of cluster members.

A cluster file system is not a distinct file system type. Clients verify the underlying file system (for example, UFS).

Using Cluster File Systems

In the Oracle Solaris Cluster software, all multihost disks are placed into device groups, which can be Solaris Volume Manager disk sets, raw-disk groups, or individual disks that are not under control of a software-based volume manager.

For a cluster file system to be highly available, the underlying disk storage must be connected to more than one cluster node. Therefore, a local file system (a file system that is stored on a node's local disk) that is made into a cluster file system is not highly available.

You can mount cluster file systems as you would mount file systems:

- **Manually.** Use the `mount` command and the `-g` or `-o global` mount options to mount the cluster file system from the command line, for example, on SPARC:

  ```
  # mount -g /dev/global/dsk/d0s0 /global/oracle/data
  ```

- **Automatically.** Create an entry in the `/etc/vfstab` file with a global mount option to mount the cluster file system at boot. You then create a mount point under the `/global` directory on all nodes. The directory `/global` is a recommended location, not a requirement. Here's a sample line for a cluster file system on SPARC from an `/etc/vfstab` file:

```
/dev/md/oracle/dsk/d1 /dev/md/oracle/rdsk/d1 /global/oracle/data ufs 2 yes global,logging
```

Note – While Oracle Solaris Cluster software does not impose a naming policy for cluster file systems, you can ease administration by creating a mount point for all cluster file systems under the same directory, such as `/global/`*disk-group*. See the *Oracle Solaris Cluster Data Services Planning and Administration Guide* for more information.

HAStoragePlus Resource Type

Use the `HAStoragePlus` resource type to make local and global file system configurations highly available. You can use the `HAStoragePlus` resource type to integrate your shared local or global file system into the Oracle Solaris Cluster environment and make the file system highly available.

You can use the `HAStoragePlus` resource type to make a file system available to a global-cluster non-voting node. You must create a mount point on the global-cluster voting node and on the global-cluster non-voting node. The `HAStoragePlus` resource type makes the file system available to the global-cluster non-voting node by mounting the file system in the global-cluster voting node. The resource type then performs a loopback mount in the global-cluster node.

Oracle Solaris Cluster systems support the following cluster file systems:

- UNIX® File System (UFS) – Uses Oracle Solaris Cluster Proxy File System (PxFS)

Oracle Solaris Cluster software supports the following as highly available failover local file systems:

- UFS
- Solaris ZFS

  Systems with Solaris ZFS are only mounted directly into the non-global zone

- Sun QFS shared file system
- Oracle Automatic Storage Management Cluster File System (Oracle ACFS) - Integrates with Oracle ASM and Oracle Clusterware

The `HAStoragePlus` resource type provides additional file system capabilities such as checks, mounts, and forced unmounts. These capabilities enable Oracle Solaris Cluster to fail over local file systems. In order to fail over, the local file system must reside on global disk groups with affinity switchovers enabled.

See "Enabling Highly Available Local File Systems" in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for information about how to use the `HAStoragePlus` resource type.

You can also use the `HAStoragePlus` resource type to synchronize the startup of resources and device groups on which the resources depend. For more information, see "Resources, Resource Groups, and Resource Types" on page 69.

syncdir Mount Option

You can use the `syncdir` mount option for cluster file systems that use UFS as the underlying file system. However, performance significantly improves if you do not specify `syncdir`. If you specify `syncdir`, the writes are guaranteed to be POSIX compliant. If you do not specify `syncdir`, you experience the same behavior as in NFS file systems. For example, without `syncdir`, you might not discover an out-of-space condition until you close a file. With `syncdir` (and POSIX behavior), the out-of-space condition would have been discovered during the write operation. The cases in which you might have problems if you do not specify `syncdir` are rare.

Disk Path Monitoring

The current release of Oracle Solaris Cluster software supports disk path monitoring (DPM). This section provides conceptual information about DPM, the DPM daemon, and administration tools that you use to monitor disk paths. Refer to *Oracle Solaris Cluster System Administration Guide* for procedural information about how to monitor, unmonitor, and check the status of disk paths.

DPM Overview

DPM improves the overall reliability of failover and switchover by monitoring secondary disk path availability. Use the `cldevice` command to verify the availability of the disk path that is used by a resource before the resource is switched. Options that are provided with the `cldevice` command enable you to monitor disk paths to a single node or to all nodes in the cluster. See the `cldevice(1CL)` man page for more information about command-line options.

The following table describes the default location for installation of DPM components.

Location	Component
Daemon	`/usr/cluster/lib/sc/scdpmd`
Command-line interface	`/usr/cluster/bin/cldevice`
Daemon status file (created at runtime)	`/var/run/cluster/scdpm.status`

A multithreaded DPM daemon runs on each node. The DPM daemon (`scdpmd`) is started by an `rc.d` script when a node boots. If a problem occurs, the daemon is managed by `pmfd` and restarts automatically.

Note – At startup, the status for each disk path is initialized to `UNKNOWN`.

The following list describes how the `scdpmd` works on initial startup:

1. The DPM daemon gathers disk path and node name information from the previous status file or from the CCR database. See "Cluster Configuration Repository (CCR)" on page 43 for more information about the CCR. After a DPM daemon is started, you can force the daemon to read the list of monitored disks from a specified file name.
2. The DPM daemon initializes the communication interface to respond to requests from components that are external to the daemon, such as the command-line interface.
3. The DPM daemon pings each disk path in the monitored list every 10 minutes by using `scsi_inquiry` commands. Each entry is locked to prevent the communication interface access to the content of an entry that is being modified.
4. The DPM daemon notifies the Oracle Solaris Cluster Event Framework and logs the new status of the path through the UNIX `syslogd` command. See the `syslogd(1M)` man page.

Note – All errors that are related to the daemon are reported by `pmfd`. All the functions from the API return `0` on success and `-1` for any failure.

The DPM daemon monitors the availability of the logical path that is visible through multipath drivers such as Oracle Solaris I/O multipathing(formerly named Sun StorEdge Traffic Manager) and EMC PowerPath. The individual physical paths that are managed by these drivers are not monitored because the multipath driver masks individual failures from the DPM daemon.

Monitoring Disk Paths

This section describes two methods for monitoring disk paths in your cluster. The first method uses the `cldevice` command to monitor, unmonitor, or display the status of disk paths in your cluster. You can also use this command to print a list of faulted disks and to monitor disk paths from a file. See the `cldevice(1CL)` man page.

The second method for monitoring disk paths in your cluster is provided by the Oracle Solaris Cluster Manager graphical user interface (GUI). Oracle Solaris Cluster Manager provides a topological view of the monitored disk paths in your cluster. The view is updated every 10 minutes to provide information about the number of failed pings. Use the information that is provided by the Oracle Solaris Cluster Manager GUI in conjunction with the `cldevice` command to administer disk paths. See Chapter 13, "Administering Oracle Solaris Cluster With the Graphical User Interfaces," in *Oracle Solaris Cluster System Administration Guide* for information about Oracle Solaris Cluster Manager.

Using the `cldevice` Command to Monitor and Administer Disk Paths

The `cldevice` command enables you to perform the following tasks:

- Monitor a new disk path
- Unmonitor a disk path
- Reread the configuration data from the CCR database
- Read the disks to monitor or unmonitor from a specified file
- Report the status of a disk path or all disk paths in the cluster
- Print all the disk paths that are accessible from a node

Note – Always specify a global disk path name rather than a UNIX disk path name because a global disk path name is consistent throughout a cluster. A UNIX disk path name is not. For example, the disk path name can be `c1t0d0` on one node and `c2t0d0` on another node. To determine a global disk path name for a device that is connected to a node, use the `cldevice list` command before issuing DPM commands. See the `cldevice(1CL)` man page.

TABLE 3–3 Sample Disk Path Names

Name Type	Sample Disk Path Name	Description
Global disk path	`schost-1:/dev/did/dsk/d1`	Disk path `d1` on the `schost-1` node
	`all:d1`	Disk path `d1` on all nodes in the cluster
UNIX disk path	`schost-1:/dev/rdsk/c0t0d0s0`	Disk path `c0t0d0s0` on the `schost-1` node

TABLE 3–3 Sample Disk Path Names *(Continued)*

Name Type	Sample Disk Path Name	Description
	`schost-1:all`	All disk paths on the `schost-1` node
All disk paths	`all:all`	All disk paths on all nodes of the cluster

Using Oracle Solaris Cluster Manager to Monitor Disk Paths

Oracle Solaris Cluster Manager enables you to perform the following basic DPM administration tasks:

- Monitor a disk path
- Unmonitor a disk path
- View the status of all monitored disk paths in the cluster
- Enable or disable the automatic rebooting of a cluster node when all monitored shared-disk paths fail

The Oracle Solaris Cluster Manager online help provides procedural information about how to administer disk paths

Using the `clnode set` Command to Manage Disk Path Failure

You use the `clnode set` command to enable and disable the automatic rebooting of a node when all monitored shared-disk paths fail. When you enable the `reboot_on_path_failure` property, the states of local-disk paths are not considered when determining whether a node reboot is necessary. Only monitored shared disks are affected. You can also use Oracle Solaris Cluster Manager to perform these tasks.

Quorum and Quorum Devices

This section contains the following topics:

Note – For information about the specific devices that Oracle Solaris Cluster software supports as quorum devices, contact your Oracle service provider.

Because cluster nodes share data and resources, a cluster must never split into separate partitions that are active at the same time because multiple active partitions might cause data corruption. The Cluster Membership Monitor (CMM) and quorum algorithm guarantee that, at most, one instance of the same cluster is operational at any time, even if the cluster interconnect is partitioned.

For more information on the CMM, see "Cluster Membership Monitor" on page 41.

Two types of problems arise from cluster partitions:

- Split brain
- Amnesia

Split brain occurs when the cluster interconnect between nodes is lost and the cluster becomes partitioned into subclusters. Each partition is not aware of the other partitions because the nodes in one partition cannot communicate with the nodes in the other partition.

Amnesia occurs when the cluster restarts after a shutdown with cluster configuration data that is older than the data was at the time of the shutdown. This problem can occur when you start the cluster on a node that was not in the last functioning cluster partition.

Oracle Solaris Cluster software avoids split brain and amnesia by:

- Assigning each node one vote
- Mandating a majority of votes for an operational cluster

A partition with the majority of votes gains *quorum* and is allowed to operate. This majority vote mechanism prevents split brain and amnesia when more than two nodes are configured in a cluster. However, counting node votes alone is not sufficient when more than two nodes are configured in a cluster. In a two-node cluster, a majority is two. If such a two-node cluster becomes partitioned, an external vote is needed for either partition to gain quorum. This external vote is provided by a *quorum device*.

A quorum device is a shared storage device or quorum server that is shared by two or more nodes and that contributes votes that are used to establish a quorum. The cluster can operate only when a quorum of votes is available. The quorum device is used when a cluster becomes partitioned into separate sets of nodes to establish which set of nodes constitutes the new cluster.

Oracle Solaris Cluster software supports the monitoring of quorum devices. Periodically, each node in the cluster tests the ability of the local node to work correctly with each configured quorum device that has a configured path to the local node and is not in maintenance mode. This test consists of an attempt to read the quorum keys on the quorum device.

When the Oracle Solaris Cluster system discovers that a formerly healthy quorum device has failed, the system automatically marks the quorum device as unhealthy. When the Oracle Solaris Cluster system discovers that a formerly unhealthy quorum device is now healthy, the system marks the quorum device as healthy and places the appropriate quorum information on the quorum device.

The Oracle Solaris Cluster system generates reports when the health status of a quorum device changes. When nodes reconfigure, an unhealthy quorum device cannot contribute votes to membership. Consequently, the cluster might not continue to operate.

About Quorum Vote Counts

Use the `clquorum show` command to determine the following information:

- Total configured votes
- Current present votes
- Votes required for quorum

See the `cluster(1CL)` man page.

Both nodes and quorum devices contribute votes to the cluster to form quorum.

A node contributes votes depending on the node's state:

- A node has a vote count of *one* when it boots and becomes a cluster member.
- A node has a vote count of *zero* when the node is being installed.
- A node has a vote count of *zero* when a system administrator places the node into maintenance state.

Quorum devices contribute votes that are based on the number of votes that are connected to the device. When you configure a quorum device, Oracle Solaris Cluster software assigns the quorum device a vote count of *N*-1 where *N* is the number of connected nodes to the quorum device. For example, a quorum device that is connected to two nodes with nonzero vote counts has a quorum count of one (two minus one).

A quorum device contributes votes if *one* of the following two conditions are true:

- At least one of the nodes to which the quorum device is currently attached is a cluster member.
- At least one of the nodes to which the quorum device is currently attached is booting, and that node was a member of the last cluster partition to own the quorum device.

You configure quorum devices during the cluster installation, or afterwards, by using the procedures that are described in Chapter 6, "Administering Quorum," in *Oracle Solaris Cluster System Administration Guide*.

About Quorum Configurations

The following list contains facts about quorum configurations:

- Quorum devices can contain user data.
- In an N+1 configuration where *N* quorum devices are each connected to one of the `1` through *N* cluster nodes and the `N+1` cluster node, the cluster survives the death of either all `1` through *N* cluster nodes or any of the *N*/**2** Oracle cluster nodes. This availability assumes that the quorum device is functioning correctly and is accessible.
- In an *N*-node configuration where a single quorum device connects to all nodes, the cluster can survive the death of any of the *N*-**1** nodes. This availability assumes that the quorum device is functioning correctly.
- In an *N*-node configuration where a single quorum device connects to all nodes, the cluster can survive the failure of the quorum device if all cluster nodes are available.

For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 56.

Adhering to Quorum Device Requirements

Ensure that Oracle Solaris Cluster software supports your specific device as a quorum device. If you ignore this requirement, you might compromise your cluster's availability.

Note – For information about the specific devices that Oracle Solaris Cluster software supports as quorum devices, contact your Oracle service provider.

Oracle Solaris Cluster software supports the following types of quorum devices:

- Multihosted shared disks that support SCSI-3 PGR reservations.
- Dual-hosted shared disks that support SCSI-2 or SCSI-3 PGR reservations.
- A Network-Attached Storage (NAS) device from Oracle's Sun product line.
- A quorum server process that runs on the quorum server machine.
- Any shared disk, provided that you have turned off fencing for this disk, and are therefore using software quorum. Software quorum is a protocol developed by Oracle that emulates a form of SCSI Persistent Group Reservations (PGR).

Caution – If you are using disks that do not support SCSI, such as Serial Advanced Technology Attachment (SATA) disks, turn off fencing.

- Sun ZFS Storage Appliance iSCSI quorum device from Oracle.

Note – You cannot use a storage-based replicated device as a quorum device.

In a two–node configuration, you must configure at least one quorum device to ensure that a single node can continue if the other node fails. See Figure 3–2.

For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 56.

Adhering to Quorum Device Best Practices

Use the following information to evaluate the best quorum configuration for your topology:

- Do you have a device that is capable of being connected to all cluster nodes of the cluster?
  - If yes, configure that device as your one quorum device. You do *not* need to configure another quorum device because your configuration is the most optimal configuration.

Caution – If you ignore this requirement and add another quorum device, the additional quorum device reduces your cluster's availability.

  - If no, configure your dual-ported device or devices.

Note – In particular environments, you might want fewer restrictions on your cluster. In these situations, you can ignore this best practice.

- If the addition of a quorum device makes the total cluster vote even, the total cluster availability decreases. If you do not have more than 50% of the total cluster vote to continue the operation, both subclusters will panic with a loss of operational quorum.
- Quorum devices slightly slow reconfigurations after a node joins or a node dies. Therefore, do not add more quorum devices than are necessary.

For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 56.

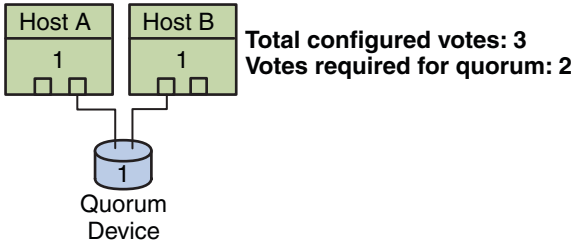
Recommended Quorum Configurations

This section provides examples of quorum configurations that are recommended.

Quorum in Two–Node Configurations

Two quorum votes are required for a two-node cluster to form. These two votes can derive from the two cluster nodes, or from just one node and a quorum device.

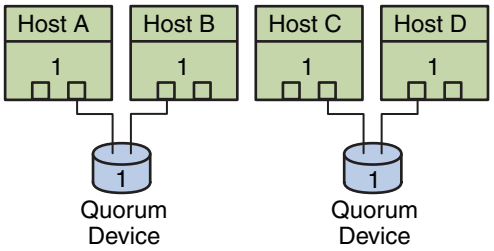
FIGURE 3–2 Two–Node Configuration



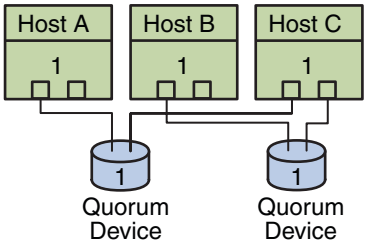
Quorum in Greater Than Two–Node Configurations

Quorum devices are not required when a cluster includes more than two nodes, because the cluster survives failures of a single node without a quorum device. However, under these conditions, you cannot start the cluster without a majority of nodes in the cluster.

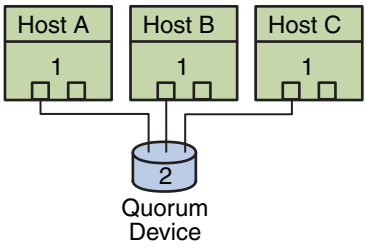
You can add a quorum device to a cluster that includes more than two nodes. A partition can survive as a cluster when that partition has a majority of quorum votes, including the votes of the nodes and the quorum devices. Consequently, when adding a quorum device, consider the possible node and quorum device failures when choosing whether and where to configure quorum devices.



In this configuration, each pair must be available for either pair to survive.



In this configuration, usually applications are configured to run on Host A and Host B and use Host C as a hot spare.



In this configuration, the combination of any one or more hosts and the quorum device can be from a cluster.

Load Limits

You can enable the automatic distribution of resource group load across nodes or zones by setting load limits. You can configure a set of load limits for each cluster node. You assign load factors to resource groups, and the load factors correspond to the defined load limits of the nodes. Each resource group is started on a node from its node list. The RGM chooses a node that best satisfies the configured load distribution policy. As resource groups are assigned to nodes by the RGM, the resource groups' load factors on each node are totaled to provide a total load. The total load is then compared against that node's load limits.

The default behavior is to distribute resource group load evenly across all the available nodes in the resource group's node list. You can configure load limits in a global cluster or a zone cluster.

For instructions on how to configure load limits using the command line, see "How to Configure Load Limits on a Node" in *Oracle Solaris Cluster System Administration Guide*. For instructions on how to configure load limits using the `clsetup` utility, see "Configuring the Distribution of Resource Group Load Across Nodes" in *Oracle Solaris Cluster Software Installation Guide*.

Data Services

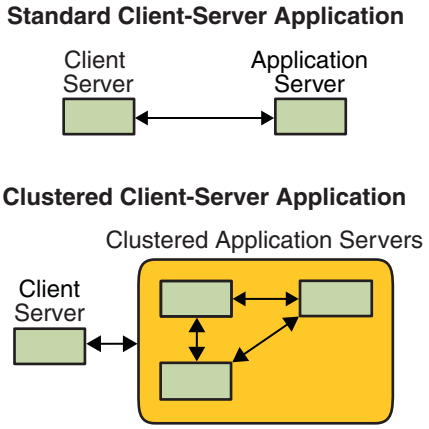
The term *data service* describes an application, such as Oracle iPlanet Web Server, that has been configured to run on a cluster rather than on a single server. Data services enable applications to become highly available and scalable services help prevent significant application interruption after any single failure within the cluster. A data service consists of an application, specialized Oracle Solaris Cluster configuration files, and Oracle Solaris Cluster management methods that control the following actions of the application:

- Start
- Stop
- Monitor and take corrective measures

The two types of data services are single-mastered (or *failover*) and multi-mastered (or *parallel*). Some multi-mastered data services, referred to as *scalable services*, are designed to take advantage of Oracle Solaris Cluster network load balancing when configured with a `SharedAddress` resource.

The following figure compares an application that runs on a single application server (the single-server model) to the same application running on a cluster (the clustered-server model). The only difference between the two configurations is that the clustered application might run faster and is more highly available.

FIGURE 3–3 Standard Compared to Clustered Client-Server Configuration



In the single-server model, you configure the application to access the server through a particular public network interface (a host name). The host name is associated with that physical server.

In the clustered-server model, the public network interface is a *logical host name* or a *shared address*. The term *network resources* is used to refer to both logical host names and shared addresses.

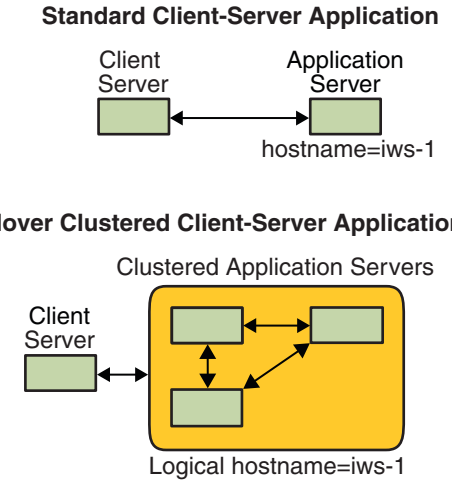
Some data services require you to specify either logical host names or shared addresses as the network interfaces. Logical host names and shared addresses are not always interchangeable. Other data services allow you to specify either logical host names or shared addresses. Refer to the installation and configuration for each data service for details about the type of interface you must specify.

A network resource is not associated with a specific physical server. A network resource can migrate between physical servers.

A network resource is initially associated with one node, the *primary*. If the primary fails, the network resource and the application resource fail over to a different cluster node (a secondary). When the network resource fails over, after a short delay, the application resource continues to run on the secondary.

The following figure compares the single-server model with the clustered-server model. Note that in the clustered-server model, a network resource (logical host name, in this example) can move between two or more of the cluster nodes. The application is configured to use this logical host name in place of a host name that is associated with a particular server.

FIGURE 3–4 Fixed Host Name Compared to Logical Host Name



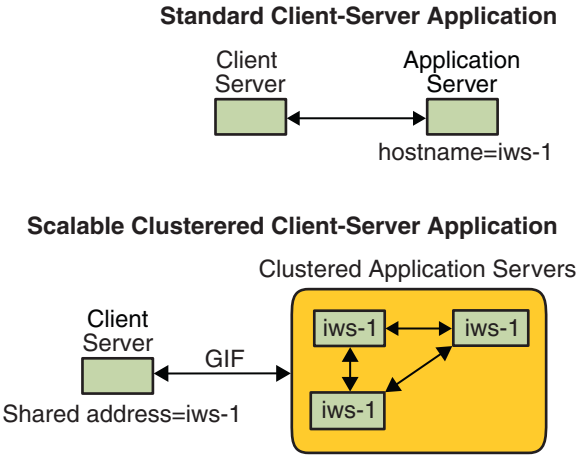
A shared address is also initially associated with one node. This node is called the global interface node. A shared address (known as the *global interface*) is used as the single network interface to the cluster.

The difference between the logical host name model and the scalable service model is that in the latter, each node also has the shared address actively configured on its loopback interface. This configuration enables multiple instances of a data service to be active on several nodes simultaneously. The term "scalable service" means that you can add more CPU power to the application by adding additional cluster nodes and the performance scales.

If the global interface node fails, the shared address can be started on another node that is also running an instance of the application (thereby making this other node the new global interface node). Or, the shared address can fail over to another cluster node that was not previously running the application.

The following figure compares the single-server configuration with the clustered scalable service configuration. Note that in the scalable service configuration, the shared address is present on all nodes. The application is configured to use this shared address in place of a host name that is associated with a particular server. This scheme is similar to how a logical host name is used for a failover data service.

FIGURE 3–5 Fixed Host Name Compared to Shared Address



Data Service Methods

The Oracle Solaris Cluster software supplies a set of service management methods. These methods run under the control of the Resource Group Manager (RGM), which uses them to start, stop, and monitor the application on the cluster nodes. These methods, along with the cluster framework software and multihost devices, enable applications to become failover or scalable data services.

The RGM also manages resources in the cluster, including instances of an application and network resources (logical host names and shared addresses).

In addition to Oracle Solaris Cluster software-supplied methods, the Oracle Solaris Cluster software also supplies an API and several data service development tools. These tools enable application developers to develop the data service methods that are required to make other applications run as highly available data services with the Oracle Solaris Cluster software.

Failover Data Services

If the node on which the data service is running (the primary node) fails, the service is migrated to another working node without user intervention. Failover services use a *failover resource group*, which is a container for application instance resources and network resources (*logical host names*). Logical host names are IP addresses that can be configured on one node and, at a later time, automatically configured down on the original node and configured on another node.

For failover data services, application instances run only on a single node. If the fault monitor detects an error, it either attempts to restart the instance on the same node or to start the instance on another node (failover). The outcome depends on how you have configured the data service.

Scalable Data Services

The scalable data service has the potential for active instances on multiple nodes.

Scalable services use the following two resource groups:

- A *scalable* resource group contains the application resources.
- A *failover* resource group, which contains the network resources (shared address) on which the scalable service depends. A shared address is a network address. This network address can be bound by all scalable services that are running on nodes within the cluster. This shared address enables these scalable services to scale on those nodes. A cluster can have multiple shared addresses, and a service can be bound to multiple shared addresses.

A scalable resource group can be online on multiple nodes simultaneously. As a result, multiple instances of the service can be running at once. All scalable resource groups use load balancing. All nodes that host a scalable service use the same shared address to host the service. The failover resource group that hosts the shared address is online on only one node at a time.

Service requests enter the cluster through a single network interface (the global interface). These requests are distributed to the nodes, based on one of several predefined algorithms that are set by the *load-balancing policy*. The cluster can use the load-balancing policy to balance the service load between several nodes. Multiple global interfaces can exist on different nodes that host other shared addresses.

For scalable services, application instances run on several nodes simultaneously. If the node that hosts the global interface fails, the global interface fails over to another node. If an application instance that is running fails, the instance attempts to restart on the same node.

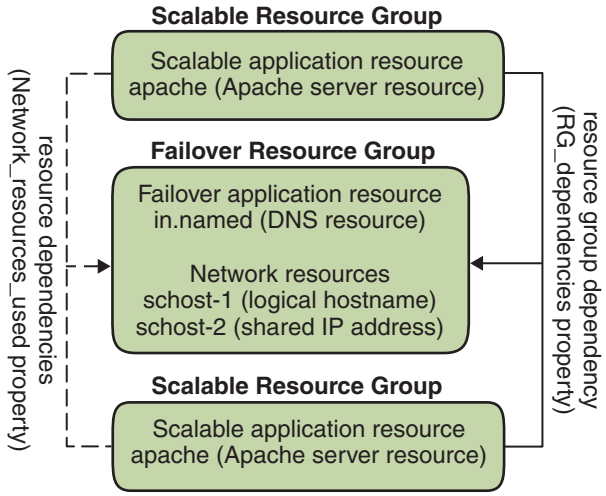
If an application instance cannot be restarted on the same node and another unused node is configured to run the service, the service fails over to the unused node. Otherwise, the service continues to run on the remaining nodes, possibly causing a degradation of service throughput.

Note – TCP state for each application instance is kept on the node with the instance, not on the global interface node. Therefore, failure of the global interface node does not affect the connection.

The following figure shows an example of failover and scalable resource groups and the dependencies that exist between them for scalable services. This example shows three resource groups. The failover resource group contains application resources for highly available DNS,

and network resources used by both highly available DNS and highly available Apache Web Server. The scalable resource groups contain only application instance of the Apache Web Server. Note that all the Apache application resources are configured with resource dependencies on the network resource `schost-2`, which is a shared address (dashed lines).

FIGURE 3–6 SPARC: Instances of Failover and Scalable Resource Groups



Load-Balancing Policies

Load balancing improves performance of the scalable service, both in response time and in throughput. The two classes of scalable data services are pure and sticky.

A *pure* service is capable of having any of its instances respond to client requests. A *sticky* service is capable of having a client send requests to the same instance. Those requests are not redirected to other instances.

A pure service uses a weighted load-balancing policy. Under this load-balancing policy, client requests are by default uniformly distributed over the server instances in the cluster. The load is distributed among various nodes according to specified weight values. For example, in a three-node cluster, suppose that each node has the weight of 1. Each node services one-third of the requests from any client on behalf of that service. The cluster administrator can change weights at any time with an administrative command or with Oracle Solaris Cluster Manager.

The weighted load-balancing policy is set by using the `LB_WEIGHTED` value for the `Load_balancing_weights` property. If a weight for a node is not explicitly set, the weight for that node is set to 1 by default.

The weighted policy redirects a certain percentage of the traffic from clients to a particular node. Given X=weight and A=the total weights of all active nodes, an active node can expect approximately X/A of the total new connections to be directed to the active node. However, the total number of connections must be large enough. This policy does not address individual requests.

Note that the weighted policy is not round robin. A round-robin policy would always cause each request from a client to go to a different node. For example, the first request would go to node 1, the second request would go to node 2, and so on.

A sticky service has two flavors: *ordinary sticky* and *wildcard sticky*.

Sticky services enable concurrent application-level sessions over multiple TCP connections to share in-state memory (application session state).

Ordinary sticky services enable a client to share the state between multiple concurrent TCP connections. The client is said to be "sticky" toward the server instance that is listening on a single port.

The client is guaranteed that all requests go to the same server instance, provided that the following conditions are met:

- The instance remains up and accessible
- The load-balancing policy is not changed while the service is online

For example, a web browser on the client connects to a shared IP address on port 80 using three different TCP connections. However, the connections exchange cached session information between them at the service.

A generalization of a sticky policy extends to multiple scalable services that exchange session information in the background and at the same instance. When these services exchange session information in the background and at the same instance, the client is said to be "sticky" toward multiple server instances on the same node that is listening on different ports.

For example, a customer on an e-commerce web site fills a shopping cart with items by using HTTP on port 80. The customer then switches to SSL on port 443 to send secure data to pay by credit card for the items in the cart.

In the ordinary sticky policy, the set of ports is known at the time the application resources are configured. This policy is set by using the `LB_STICKY` value for the `Load_balancing_policy` resource property.

Wildcard sticky services use dynamically assigned port numbers, but still expect client requests to go to the same node. The client is "sticky wildcard" over ports that have the same IP address.

A good example of this policy is passive mode FTP. For example, a client connects to an FTP server on port 21. The server then instructs the client to connect back to a listener port server in the dynamic port range. All requests for this IP address are forwarded to the same node that the server informed the client through the control information.

The sticky-wildcard policy is a superset of the ordinary sticky policy. For a scalable service that is identified by the IP address, ports are assigned by the server and are not known in advance. The ports might change. This policy is set by using the `LB_STICKY_WILD` value for the `Load_balancing_policy` resource property.

For each one of these sticky policies, the weighted load-balancing policy is in effect by default. Therefore, a client's initial request is directed to the instance that the load balancer dictates. After the client establishes an affinity for the node where the instance is running, future requests are conditionally directed to that instance. The node must be accessible and the load-balancing policy must not have changed.

Failback Settings

Resource groups fail over from one node to another. When this failover occurs, the original secondary becomes the new primary. The failback settings specify the actions that occur when the original primary comes back online. The options are to have the original primary become the primary again (failback) or to allow the current primary to remain. You specify the option you want by using the `Failback` resource group property setting.

If the original node that hosts the resource group fails and reboots repeatedly, setting failback might result in reduced availability for the resource group.

Data Services Fault Monitors

Each Oracle Solaris Cluster data service supplies a fault monitor that periodically probes the data service to determine its health. A fault monitor verifies that the application daemon or daemons are running and that clients are being served. Based on the information that probes return, predefined actions such as restarting daemons or causing a failover can be initiated.

Developing New Data Services

Oracle supplies configuration files and management methods templates that enable you to make various applications operate as failover or scalable services within a cluster. If Oracle does not offer the application that you want to run as a failover or scalable service, you can use an Oracle Solaris Cluster API or the DSET API to configure the application to run as a failover or scalable service. However, not all applications can become a scalable service.

Characteristics of Scalable Services

A set of criteria determines whether an application can become a scalable service. To determine whether your application can become a scalable service, see "Analyzing the Application for Suitability" in *Oracle Solaris Cluster Data Services Developer's Guide*.

This set of criteria is summarized as follows:

- This type of service is composed of one or more server *instances*. Each instance runs on a different node. Two or more instances of the same service cannot run on the same node.
- If the service provides an external logical data store, you must exercise caution. Concurrent access to this store from multiple server instances must be synchronized to avoid losing updates or reading data as it is being changed. Note the use of "external" to distinguish the store from in-memory state. The term "logical" indicates that the store appears as a single entity, although it might itself be replicated. Furthermore, in this data store, when any server instance updates the data store, this update is immediately "seen" by other instances.

  The Oracle Solaris Cluster software provides such an external storage through its cluster file system and its global raw partitions. As an example, suppose a service writes new data to an external log file or modifies existing data in place. When multiple instances of this service run, each instance has access to this external log, and each might simultaneously access this log. Each instance must synchronize its access to this log, or else the instances interfere with each other. The service could use ordinary Oracle Solaris file locking through `fcntl` and `lockf` to achieve the synchronization that you want.

  Another example of this type of store is a back-end database. This type of back-end database server provides built-in synchronization by using database query or update transactions. Therefore, multiple server instances do not need to implement their own synchronization.

  Oracle's Sun IMAP server is an example of a service that is not a scalable service. The service updates a store but that store is private. When multiple IMAP instances write to this store, they overwrite each other because the updates are not synchronized. The IMAP server must be rewritten to synchronize concurrent access.
- Note that instances can have private data that is separate from the data of other instances. In such a case, the service does not need synchronized concurrent access because the data is private, and only that instance can manipulate it. In this case, you must be careful not to store this private data under the cluster file system because this data can become globally accessible.

Data Service API and Data Service Development Library API

The Oracle Solaris Cluster software provides the following to make applications highly available:

- Data services that are supplied as part of the Oracle Solaris Cluster software
- A data service API
- A development library API for data services
- A "generic" data service

The *Oracle Solaris Cluster Data Services Planning and Administration Guide* describes how to install and configure the data services that are supplied with the Oracle Solaris Cluster software. Other applications can also be configured to be highly available under the Oracle Solaris Cluster framework.

The Oracle Solaris Cluster APIs enable application developers to develop fault monitors and scripts that start and stop data service instances. With these tools, an application can be implemented as a failover or a scalable data service. The Oracle Solaris Cluster software provides a generic data service. Use this generic data service to quickly generate an application's required start and stop methods and to implement the data service as a failover or scalable service.

Using the Cluster Interconnect for Data Service Traffic

A cluster must usually have multiple network connections between cluster nodes, forming the cluster interconnect.

Oracle Solaris Cluster software uses multiple interconnects to achieve the following goals:

- Ensure high availability
- Improve performance

For both internal and external traffic, such as file system data or scalable services data, messages are striped across all available interconnects. The cluster interconnect is also available to applications for highly available communication between nodes. For example, a distributed application might have components that are running on different nodes that need to communicate. By using the cluster interconnect rather than the public transport, these connections can withstand the failure of an individual link.

To use the cluster interconnect for communication between nodes, an application must use the private host names that you configured during the Oracle Solaris Cluster installation. For example, if the private host name for `host1` is `clusternode1-priv`, use this name to communicate with `host1` over the cluster interconnect. TCP sockets that are opened by using this name are routed over the cluster interconnect and can be transparently rerouted if a private network adapter fails. Application communication between any two nodes is striped over all interconnects. The traffic for a given TCP connection flows on one interconnect at any point. Different TCP connections are striped across all interconnects. Additionally, UDP traffic is always striped across all interconnects.

An application can optionally use a zone's private host name to communicate over the cluster interconnect between zones. However, you must first set each zone's private host name before the application can begin communicating. Each zone must have its own private host name to communicate. An application that is running in one zone must use the private host name in the same zone to communicate with private host names in other zones.

You can change the private host names after you install the Oracle Solaris Cluster software. To determine the actual name, use the `scha_cluster_get` command with the `scha_privatelink_hostname_node` argument. See the `scha_cluster_get(1HA)` man page.

Each host is also assigned a fixed per-host address. This per-host address is plumbed on the `clprivnet` driver. The IP address maps to the private host name for the host: `clusternode1-priv`. See the `clprivnet(7)` man page.

Resources, Resource Groups, and Resource Types

Data services use several types of *resources*: applications such as Oracle iPlanet Web Server or Apache Web Server use network addresses (logical host names and shared addresses) on which the applications depend. Application and network resources form a basic unit that is managed by the Resource Group Manager (RGM).

For example, Oracle Solaris Cluster HA for Oracle is the resource type `SUNW.oracle-server` and Oracle Solaris Cluster HA for Apache is the resource type `SUNW.apache`.

A resource is an instance of a *resource type* that is defined cluster wide. Several resource types are defined.

Network resources are either `SUNW.LogicalHostname` or `SUNW.SharedAddress` resource types. These two resource types are preregistered by the Oracle Solaris Cluster software.

The `HAStoragePlus` resource type is used to synchronize the startup of resources and device groups on which the resources depend. This resource type ensures that before a data service starts, the paths to a cluster file system's mount points, global devices, and device group names are available. For more information, see "Synchronizing the Startups Between Resource Groups and Device Groups" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*. The `HAStoragePlus` resource type also enables local file systems to be highly available. For more information about this feature, see "`HAStoragePlus` Resource Type" on page 48.

RGM-managed resources are placed into groups, called *resource groups*, so that they can be managed as a unit. A resource group is migrated as a unit if a failover or switchover is initiated on the resource group.

Note – When you bring a resource group that contains application resources online, the application is started. The data service start method waits until the application is running before exiting successfully. The determination of when the application is up and running is accomplished the same way the data service fault monitor determines that a data service is serving clients. Refer to the *Oracle Solaris Cluster Data Services Planning and Administration Guide* for more information about this process.

Resource Group Manager (RGM)

The RGM controls data services (applications) as resources, which are managed by *resource type* implementations. These implementations are either supplied by Oracle or created by a developer with a generic data service template, the Data Service Development Library API (DSDL API), or the Resource Management API (RMAPI). The cluster administrator creates and manages resources in containers called resource groups. The RGM stops and starts resource groups on selected nodes in response to cluster membership changes.

The RGM acts on resources and resource groups RGM actions cause resources and resource groups to move between online and offline states. For a complete description of the states and settings that can be applied to resources and resource groups, see "Resource and Resource Group States and Settings" on page 70.

Refer to "Data Service Project Configuration" on page 78 for information about how to launch Oracle Solaris projects under RGM control.

Resource and Resource Group States and Settings

A system administrator applies static settings to resources and resource groups. You can change these settings only by administrative action. The RGM moves resource groups between dynamic states.

These settings and states are as follows:

- **Managed or unmanaged settings.** These cluster-wide settings apply only to resource groups. The RGM manages resource groups. You can use the `clresourcegroup` command to request that the RGM manage or unmanage a resource group. These resource group settings do not change when you reconfigure a cluster.

  When a resource group is first created, it is unmanaged. A resource group must be managed before any resources placed in the group can become active.

  In some data services, for example, a scalable web server, work must be done prior to starting network resources and after they are stopped. This work is done by initialization (`INIT`) and finish (`FINI`) data service methods. The `INIT` methods only run if the resource group in which the resources are located is in the managed state.

  When a resource group is moved from unmanaged to managed, any registered `INIT` methods for the group are run on the resources in the group.

  When a resource group is moved from managed to unmanaged, any registered `FINI` methods are called to perform cleanup.

  The most common use of the `INIT` and `FINI` methods are for network resources for scalable services. However, a data service developer can use these methods for any initialization or cleanup work that is not performed by the application.

- **Enabled or disabled settings.** These settings apply to resources on one or more nodes. A system administrator can use the `clresource` command to enable or disable a resource on one or more nodes. These settings do not change when the cluster administrator reconfigures a cluster.

  The normal setting for a resource is that it is enabled and actively running in the system.

  If you want to make the resource unavailable on all cluster nodes, disable the resource on all cluster nodes. A disabled resource is not available for general use on the cluster nodes that you specify.
- **Online or offline states.** These dynamic states apply to both resources and resource groups.

  Online and offline states change as the cluster transitions through cluster reconfiguration steps during switchover or failover. You can also change the online or offline state of a resource or a resource group by using the `clresource` and `clresourcegroup` commands.

  A failover resource or resource group can only be online on one node at any time. A scalable resource or resource group can be online on some nodes and offline on others. During a switchover or failover, resource groups and the resources within them are taken offline on one node and then brought online on another node.

  If a resource group is offline, all of its resources are offline. If a resource group is online, all of its enabled resources are online.

  You can temporarily suspend the automatic recovery actions of a resource group. You might need to suspend the automatic recovery of a resource group to investigate and fix a problem in the cluster. Or, you might need to perform maintenance on resource group services.

  A suspended resource group is *not* automatically restarted or failed over until you explicitly issue the command that resumes automatic recovery. Whether online or offline, suspended data services remain in their current state. You can still manually switch the resource group to a different state on specified nodes. You can also still enable or disable individual resources in the resource group.

  Resource groups can contain several resources, with dependencies between resources. These dependencies require that the resources be brought online and offline in a particular order. The methods that are used to bring resources online and offline might take different amounts of time for each resource. Because of resource dependencies and start and stop time differences, resources within a single resource group can have different online and offline states during a cluster reconfiguration.

Resource and Resource Group Properties

You can configure property values for resources and resource groups for your Oracle Solaris Cluster data services. Standard properties are common to all data services. Extension properties are specific to each data service. Some standard and extension properties are configured with default settings so that you do not have to modify them. Others need to be set as part of the

process of creating and configuring resources. The documentation for each data service specifies which resource properties can be set and how to set them.

The standard properties are used to configure resource and resource group properties that are usually independent of any particular data service. For the set of standard properties, see the following man pages: `cluster(1CL)`, `rt_properties(5)`, `rg_properties(5)`, `r_properties(5)`, and `property_attributes(5)`.

The RGM extension properties provide information such as the location of application binaries and configuration files. You modify extension properties as you configure your data services. The set of extension properties is described in the individual guide for the data service.

Support for Oracle Solaris Zones

Oracle Solaris zones provide a means of creating virtualized operating system environments within an instance of the Oracle Solaris OS. Oracle Solaris zones enable one or more applications to run in isolation from other activity on your system. The Oracle Solaris zones facility is described in Part II, "Zones," in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*.

You can create any number of global-cluster non-voting nodes.

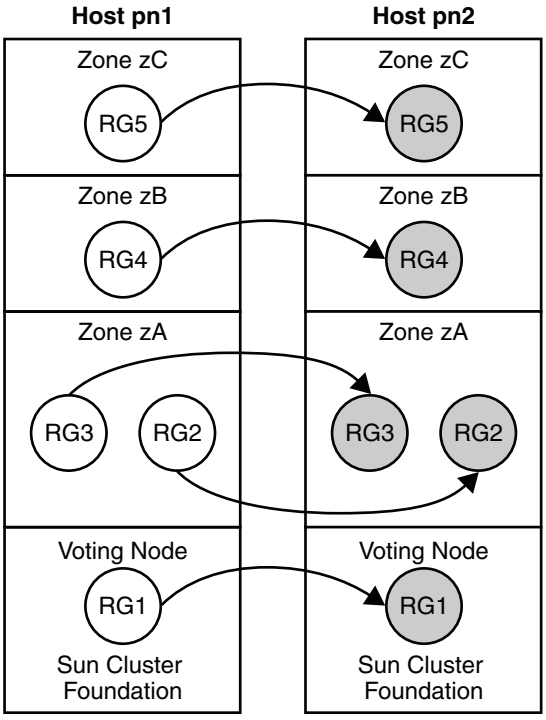
You can use Oracle Solaris Cluster software to manage the availability and scalability of applications that are running on global-cluster non-voting nodes.

Support for Global-Cluster Non-Voting Nodes (Oracle Solaris Zones) Directly Through the RGM

You can configure a resource group to run on a global-cluster voting node or a global-cluster non-voting node. The RGM manages each global-cluster non-voting node as a switchover target. If a global-cluster non-voting node is specified in the node list of a resource group, the RGM brings the resource group online in the specified node.

The following figure illustrates the failover of resource groups between nodes in a two-node cluster. In this example, identical nodes are configured to simplify the administration of the cluster.

FIGURE 3–7 Failover of Resource Groups Between Nodes



You can configure a scalable resource group (which uses network load balancing) to run in a cluster non-voting node as well.

Using Oracle Solaris Cluster commands, you specify a zone by appending the name of the zone to the name of the node and separating the names with a colon. For example:

```
phys-schost-1:zoneA
```

Criteria for Using Support for Oracle Solaris Zones Directly Through the RGM

Use support for Oracle Solaris zones directly through the RGM if any of following criteria is met:

- Your application cannot tolerate the additional failover time that is required to boot a zone.
- You require minimum downtime during maintenance.
- You require dual-partition software upgrade.
- You are configuring a data service that uses a shared address resource for network load balancing.

Requirements for Using Support for Oracle Solaris Zones Directly Through the RGM

If you plan to use support for Oracle Solaris zones directly through the RGM for an application, ensure that the following requirements are met:

- The application is supported to run in non-global zones.
- The data service for the application is supported to run on a global-cluster non-voting node.

If you use support for Oracle Solaris zones directly through the RGM, ensure that resource groups that are related by an affinity are configured to run on the same cluster node.

Additional Information About Support for Solaris Zones Directly Through the RGM

For information about how to configure support for Oracle Solaris zones directly through the RGM, see the following documentation:

- "Guidelines for Non-Global Zones in a Global Cluster" in *Oracle Solaris Cluster Software Installation Guide*
- "Zone Names" in *Oracle Solaris Cluster Software Installation Guide*
- "Configuring a Non-Global Zone on a Global-Cluster Node" in *Oracle Solaris Cluster Software Installation Guide*
- *Oracle Solaris Cluster Data Services Planning and Administration Guide*
- Individual data service guides

Support for Oracle Solaris Zones on Cluster Nodes Through Oracle Solaris Cluster HA for Solaris Zones

The Oracle Solaris Cluster HA for Solaris Zones data service manages each zone as a resource that is controlled by the RGM.

Criteria for Using Oracle Solaris Cluster HA for Solaris Zones

Use the Oracle Solaris Cluster HA for Solaris Zones data service if any of following criteria is met:

- You require delegated root access.
- The application is not supported in a cluster.
- You require affinities between resource groups that are to run in different zones on the same node.

Requirements for Using Oracle Solaris Cluster HA for Solaris Zones

If you plan to use the Oracle Solaris Cluster HA for Solaris Zones data service for an application, ensure that the following requirements are met:

- The application is supported to run on global-cluster non-voting nodes.
- The application is integrated with the Oracle Solaris OS through a script, a run-level script, or an Oracle Solaris Service Management Facility (SMF) manifest.
- The additional failover time that is required to boot a zone is acceptable.
- Some downtime during maintenance is acceptable.

Additional Information About Oracle Solaris Cluster HA for Solaris Zones

For information about how to use the Oracle Solaris Cluster HA for Solaris Zones data service, see *Oracle Solaris Cluster Data Service for Solaris Containers Guide*.

Service Management Facility

The Oracle Solaris Service Management Facility (SMF) enables you to run and administer applications as highly available and scalable resources. Like the Resource Group Manager (RGM), the SMF provides high availability and scalability, but for the Oracle Solaris OS.

Oracle Solaris Cluster provides three proxy resource types that you can use to enable SMF services in a cluster. These resource types, `SUNW.Proxy_SMF_failover`, `SUNW.Proxy_SMF_loadbalanced`, and `SUNW.Proxy_SMF_multimaster`, enable you to run SMF services in a failover, scalable, and multi-master configuration, respectively. The SMF manages the availability of SMF services on a single cluster node. The SMF uses the callback method execution model to run services.

The SMF also provides a set of administrative interfaces for monitoring and controlling services. These interfaces enable you to integrate your own SMF-controlled services into Oracle Solaris Cluster. This capability eliminates the need to create new callback methods, rewrite existing callback methods, or update the SMF service manifest. You can include multiple SMF resources in a resource group and you can configure dependencies and affinities between them.

The SMF is responsible for starting, stopping, and restarting these services and managing their dependencies. Oracle Solaris Cluster is responsible for managing the service in the cluster and for determining the nodes on which these services are to be started.

The SMF runs as a daemon, `svc.startd`, on each cluster node. The SMF daemon automatically starts and stops resources on selected nodes according to preconfigured policies.

The services that are specified for an SMF proxy resource can be located on global cluster voting node or global cluster non-voting node. However, all the services that are specified for the same SMF proxy resource must be located on the same node. SMF proxy resources work on any node.

System Resource Usage

System resources include aspects of CPU usage, memory usage, swap usage, and disk and network throughput. Oracle Solaris Cluster enables you to monitor how much of a specific system resource is being used by an *object type*. An object type includes a host, node, zone, disk, network interface, or resource group. Oracle Solaris Cluster also enables you to control the CPU that is available to a resource group.

Monitoring and controlling system resource usage can be part of your resource management policy. The cost and complexity of managing numerous machines encourages the consolidation of several applications on larger hosts. Instead of running each workload on separate systems with full access to each system's resources, you use resource management to segregate workloads within the system.

Resource management ensures that your applications have the required response times. Resource management can also increase resource use. By categorizing and prioritizing usage, you can effectively use reserve capacity during off-peak periods, often eliminating the need for additional processing power. You can also ensure that resources are not wasted because of load variability.

To use the data that Oracle Solaris Cluster collects about system resource usage, you must do the following:

- Analyze the data to determine what it means for your system.
- Make a decision about the action that is required to optimize your usage of hardware and software resources.
- Take action to implement your decision.

By default, system resource monitoring and control are not configured when you install Oracle Solaris Cluster. For information about configuring these services, see Chapter 10, "Configuring Control of CPU Usage," in *Oracle Solaris Cluster System Administration Guide*.

System Resource Monitoring

By monitoring system resource usage, you can do the following:

- Collect data that reflects how a service that is using specific system resources is performing.
- Discover resource bottlenecks or overload and so preempt problems.

- More efficiently manage workloads.

Data about system resource usage can help you determine the hardware resources that are underused and the applications that use many resources. Based on this data, you can assign applications to nodes that have the necessary resources and choose the node to which to failover. This consolidation can help you optimize the way that you use your hardware and software resources.

Monitoring all system resources at the same time might be costly in terms of CPU. Choose the system resources that you want to monitor by prioritizing the resources that are most critical for your system.

When you enable monitoring, you choose the *telemetry attribute* that you want to monitor. A telemetry attribute is an aspect of system resources. Examples of telemetry attributes include the amount of free CPU or the percentage of blocks that are used on a device. If you monitor a telemetry attribute on an object type, Oracle Solaris Cluster monitors this telemetry attribute on all objects of that type in the cluster. Oracle Solaris Cluster stores a history of the system resource data that is collected for seven days.

If you consider a particular data value to be critical for a system resource, you can set a *threshold* for this value. When setting a threshold, you also choose how critical this threshold is by assigning it a severity level. If the threshold is crossed, Oracle Solaris Cluster changes the severity level of the threshold to the severity level that you choose.

Control of CPU

Each application and service that is running on a cluster has specific CPU needs. The following table lists the CPU control activities that are available.

TABLE 3–4 CPU Control

Zone	Control
Global-cluster voting node	Assign CPU shares
Global-cluster non-voting node	Assign CPU shares
	Assign number of CPU
	Create dedicated processor sets

A CPU share is the portion of the system's CPU resources that is allocated to a project. Shares define the relative importance of workloads in relation to other workloads. If you want to apply CPU shares, you must specify the Fair Share Scheduler (FFS) as the default scheduler in the cluster. When you assign CPU shares to a project, your primary concern is not the number of shares the project has. Rather, you should know how many of those other projects will be

competing with it for CPU resources. Controlling the CPU that is assigned to a resource group in a dedicated processor set in a global-cluster non-voting node offers the strictest level of control. If you reserve CPU for a resource group, this CPU is not available to other resource groups.

Viewing System Resource Usage

You can view system resource data and CPU assignments by using the command line or through Oracle Solaris Cluster Manager. The system resources that you choose to monitor determine the tables and graphs that you can view.

By viewing the output of system resource usage and CPU control, you can do the following:

- Anticipate failures due to the exhaustion of system resources
- Detect unbalanced usage of system resources
- Validate server consolidation
- Obtain information that enables you to improve the performance of applications

For more information, see the `clresource(1CL)`, `cltelemetryattribute(1CL)`, and `rg_properties(5)` man pages.

Data Service Project Configuration

This section provides a conceptual description of configuring data services to launch processes on a specified Oracle Solaris OS UNIX project. This section also describes several failover scenarios and suggestions for using the management functionality provided by the Oracle Solaris OS. See the `project(4)` man page for more information.

Data services can be configured to launch under an Oracle Solaris project name when brought online by using the RGM. The configuration associates a resource or resource group managed by the RGM with an Oracle Solaris project ID. The mapping from your resource or resource group to a project ID gives you the ability to use sophisticated controls that are available in the Oracle Solaris OS to manage workloads and consumption within your cluster.

Using the Oracle Solaris management functionality in an Oracle Solaris Cluster environment ensures that your most important applications are given priority when sharing a node with other applications. Applications might share a node if you have consolidated services or because applications have failed over. Use of the management functionality described here might improve availability of a critical application by preventing lower-priority applications from over-consuming system supplies such as CPU time.

Note – The Oracle Solaris documentation for this feature describes CPU time, processes, tasks and similar components as "resources." However, the Oracle Solaris Cluster documentation uses the term "resources" to describe entities that are under the control of the RGM. The section uses the term "supplies" to refer to CPU time, processes, and tasks.

For detailed conceptual and procedural documentation about the management feature, refer to Chapter 1, "Network Service (Overview)," in *System Administration Guide: Network Services*.

When configuring resources and resource groups to use Oracle Solaris management functionality in a cluster, use the following high-level process:

1. Configure applications as part of the resource.
2. Configure resources as part of a resource group.
3. Enable resources in the resource group.
4. Make the resource group managed.
5. Create an Oracle Solaris project for your resource group.
6. Configure standard properties to associate the resource group name with the project you created in step 5.
7. Bring the resource group online.

To configure the standard `Resource_project_name` or `RG_project_name` properties to associate the Oracle Solaris project ID with the resource or resource group, use the `-p` option with the `clresource set` and the `clresourcegroup set` command. Set the property values to the resource or to the resource group. See the `r_properties(5)` and `rg_properties(5)` man pages for descriptions of properties.

The specified project name must exist in the projects database (`/etc/project`) and the root user must be configured as a member of the named project. Refer to Chapter 2, "Projects and Tasks (Overview)," in *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones* for conceptual information about the project name database. Refer to `project(4)` for a description of project file syntax.

When the RGM brings resources or resource groups online, it launches the related processes under the project name.

Note – Users can associate the resource or resource group with a project at any time. However, the new project name is not effective until the resource or resource group is taken offline and brought back online by using the RGM.

Launching resources and resource groups under the project name enables you to configure the following features to manage system supplies across your cluster:

- Extended Accounting. Provides a flexible way to record consumption on a task or process basis. Extended accounting enables you to examine historical usage and make assessments of capacity requirements for future workloads.
- Controls. Provide a mechanism for constraint on system supplies. Processes, tasks, and projects can be prevented from consuming large amounts of specified system supplies.
- Fair Share Scheduling (FSS). Provides the ability to control the allocation of available CPU time among workloads based on their importance. Workload importance is expressed by the number of shares of CPU time that you assign to each workload. Refer to the following man pages for more information:
  - `dispadmin(1M)`
  - `priocntl(1)`
  - `ps(1)`
  - `FSS(7)`
- Pools. Provide the ability to use partitions for interactive applications according to the application's requirements. Pools can be used to partition a host that supports a number of different software applications. The use of pools results in a more predictable response for each application.

Determining Requirements for Project Configuration

Before you configure data services to use the controls provided by Oracle Solaris in an Oracle Solaris Cluster environment, you must decide how to control and track resources across switchovers or failovers. Identify dependencies within your cluster before configuring a new project. For example, resources and resource groups depend on device groups.

Use the `nodelist`, `failback`, `maximum_primaries`, and `desired_primaries` resource group properties that you configure with the `clresourcegroup set` command to identify node list priorities for your resource group.

- For a brief discussion of the node list dependencies between resource groups and device groups, refer to "Relationship Between Resource Groups and Device Groups" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.
- For detailed property descriptions, refer to `rg_properties(5)`.

Use the `preferenced` property and `failback` property that you configure with the `cldevicegroup` and `clsetup` commands to determine device group node list priorities. See the `clresourcegroup(1CL)`, `cldevicegroup(1CL)`, and `clsetup(1CL)` man pages.

- For conceptual information about the `preferenced` property, see "Device Group Ownership" on page 44.
- For procedural information, see "How To Change Disk Device Properties" in "Administering Device Groups" in *Oracle Solaris Cluster System Administration Guide*.

- For conceptual information about node configuration and the behavior of failover and scalable data services, see "Oracle Solaris Cluster System Hardware and Software Components" on page 17.

If you configure all cluster nodes identically, usage limits are enforced identically on primary and secondary nodes. The configuration parameters of projects do not need to be identical for all applications in the configuration files on all nodes. All projects that are associated with the application must at least be accessible by the project database on all potential masters of that application. Suppose that Application 1 is mastered by `phys-schost-1` but could potentially be switched over or failed over to `phys-schost-2` or `phys-schost-3`. The project that is associated with Application 1 must be accessible on all three nodes (`phys-schost-1`, `phys-schost-2`, and `phys-schost-3`).

Note – Project database information can be a local `/etc/project` database file or can be stored in the NIS map or the LDAP directory service.

The Oracle Solaris OS enables flexible configuration of usage parameters, and few restrictions are imposed by Oracle Solaris Cluster. Configuration choices depend on the needs of the site. Consider the general guidelines in the following sections before configuring your systems.

Setting Per-Process Virtual Memory Limits

Set the `process.max-address-space` control to limit virtual memory on a per-process basis. See the `rctladm(1M)` man page for information about setting the `process.max-address-space` value.

When you use management controls with Oracle Solaris Cluster software, configure memory limits appropriately to prevent unnecessary failover of applications and a "ping-pong" effect of applications. Observe the following guidelines:

- Do not set memory limits too low.

  When an application reaches its memory limit, it might fail over. This guideline is especially important for database applications, when reaching a virtual memory limit can have unexpected consequences.

- Do not set memory limits identically on primary and secondary nodes.

  Identical limits can cause a ping-pong effect when an application reaches its memory limit and fails over to a secondary node with an identical memory limit. Set the memory limit slightly higher on the secondary node. The difference in memory limits helps prevent the ping-pong scenario and gives the system administrator a period of time in which to adjust the parameters as necessary.

- Do use the resource management memory limits for load balancing.

For example, you can use memory limits to prevent an errant application from consuming excessive swap space.

Failover Scenarios

You can configure management parameters so that the allocation in the project configuration (`/etc/project`) works in normal cluster operation and in switchover or failover situations.

The following sections are example scenarios.

- The first two sections, "Two-Node Cluster With Two Applications" on page 83 and "Two-Node Cluster With Three Applications" on page 84, show failover scenarios for entire nodes.
- The section "Failover of Resource Group Only" on page 86 illustrates failover operation for an application only.

In an Oracle Solaris Cluster environment, you configure an application as part of a resource. You then configure a resource as part of a resource group (RG). When a failure occurs, the resource group, along with its associated applications, fails over to another node. In the following examples the resources are not shown explicitly. Assume that each resource has only one application.

Note – Failover occurs in the order in which nodes are specified in the node list and set in the RGM.

The following examples have these constraints:

- Application 1 (App-1) is configured in resource group RG-1.
- Application 2 (App-2) is configured in resource group RG-2.
- Application 3 (App-3) is configured in resource group RG-3.

Although the numbers of assigned shares remain the same, the percentage of CPU time that is allocated to each application changes after failover. This percentage depends on the number of applications that are running on the node and the number of shares that are assigned to each active application.

In these scenarios, assume the following configurations:

- All applications are configured under a common project.
- Each resource has only one application.
- The applications are the only active processes on the nodes.
- The projects databases are configured the same on each node of the cluster.

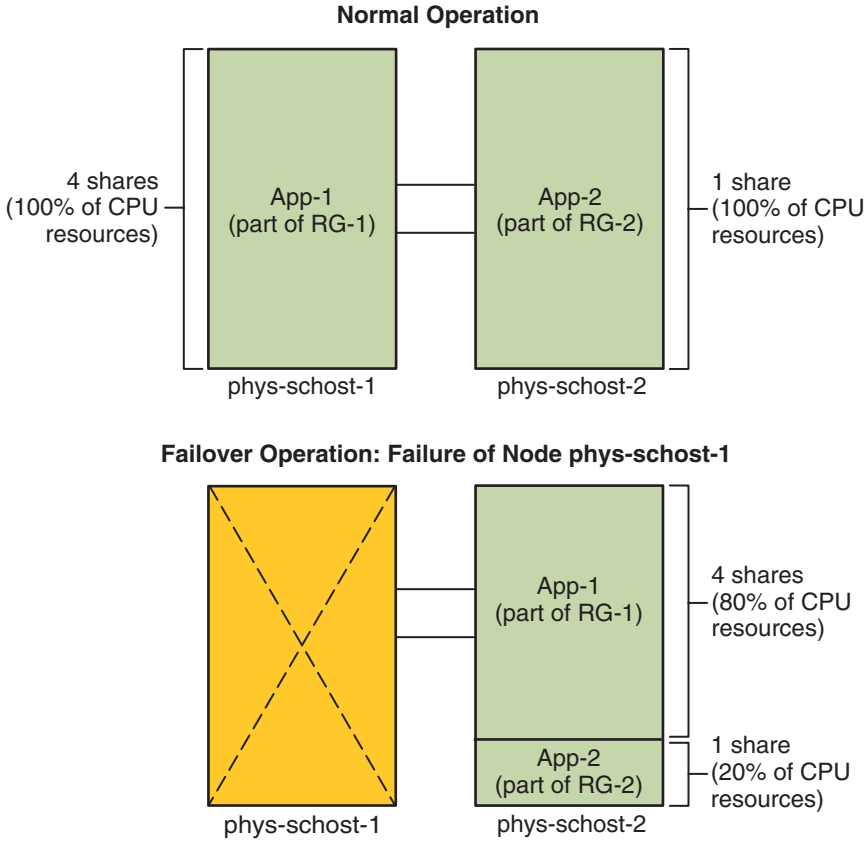
Two-Node Cluster With Two Applications

You can configure two applications on a two-node cluster to ensure that each physical node(`phys-schost-1`, `phys-schost-2`) acts as the default master for one application. Each physical node acts as the secondary node for the other physical node. All projects that are associated with Application 1 and Application 2 must be represented in the projects database files on both nodes. When the cluster is running normally, each application is running on its default master, where it is allocated all CPU time by the management facility.

After a failover or switchover occurs, both applications run on a single node where they are allocated shares as specified in the configuration file. For example, this entry in the`/etc/project` file specifies that Application 1 is allocated 4 shares and Application 2 is allocated 1 share.

```
Prj_1:100:project for App-1:root::project.cpu-shares=(privileged,4,none)
Prj_2:101:project for App-2:root::project.cpu-shares=(privileged,1,none)
```

The following diagram illustrates the normal and failover operations of this configuration. The number of shares that are assigned does not change. However, the percentage of CPU time available to each application can change. The percentage depends on the number of shares that are assigned to each process that demands CPU time.



Two-Node Cluster With Three Applications

On a two-node cluster with three applications, you can configure one node (`phys-schost-1`) as the default master of one application. You can configure the second physical node (`phys-schost-2`) as the default master for the remaining two applications. Assume the following example projects database file is located on every node. The projects database file does not change when a failover or switchover occurs.

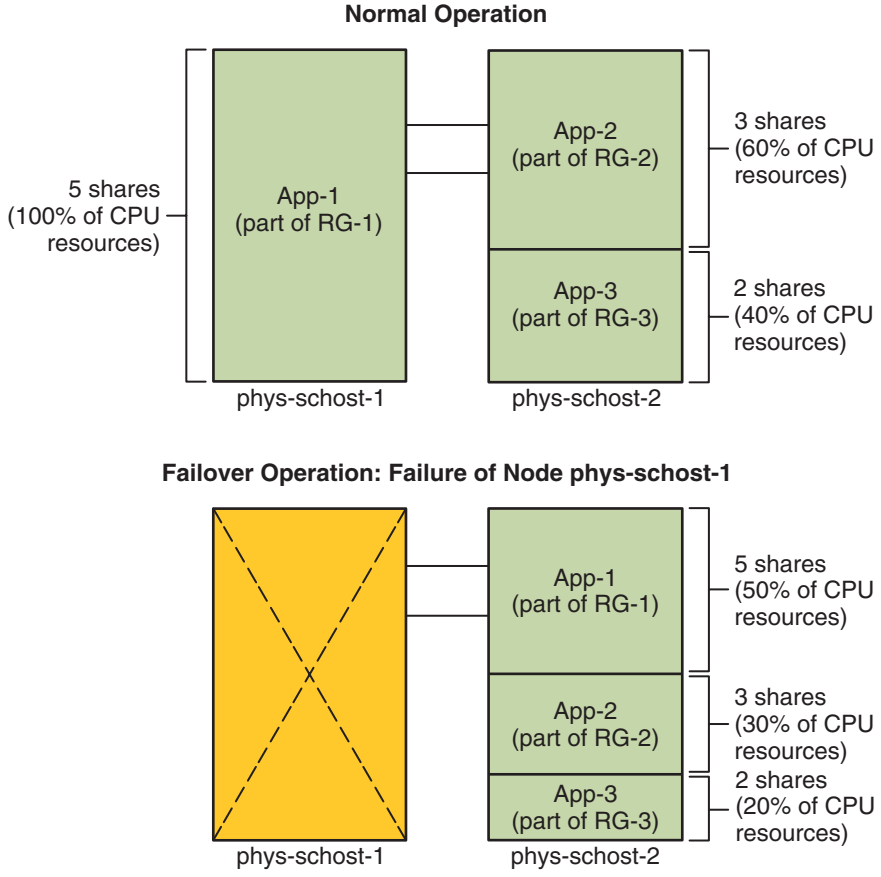
```
Prj_1:103:project for App-1:root::project.cpu-shares=(privileged,5,none)
Prj_2:104:project for App_2:root::project.cpu-shares=(privileged,3,none)
Prj_3:105:project for App_3:root::project.cpu-shares=(privileged,2,none)
```

When the cluster is running normally, Application 1 is allocated 5 shares on its default master, `phys-schost-1`. This number is equivalent to 100 percent of CPU time because it is the only application that demands CPU time on that node. Applications 2 and 3 are allocated 3 and 2 shares, respectively, on their default master, `phys-schost-2`. Application 2 would receive 60 percent of CPU time and Application 3 would receive 40 percent of CPU time during normal operation.

If a failover or switchover occurs and Application 1 is switched over to `phys-schost-2`, the shares for all three applications remain the same. However, the percentages of CPU resources are reallocated according to the projects database file.

- Application 1, with 5 shares, receives 50 percent of CPU.
- Application 2, with 3 shares, receives 30 percent of CPU.
- Application 3, with 2 shares, receives 20 percent of CPU.

The following diagram illustrates the normal operations and failover operations of this configuration.



Failover of Resource Group Only

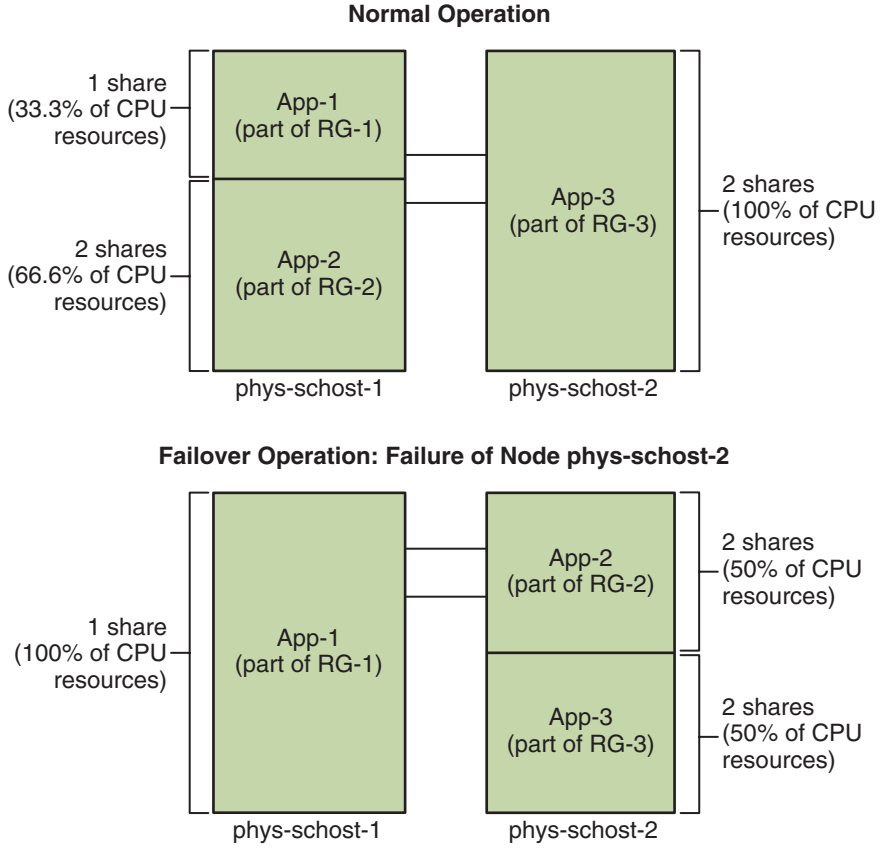
In a configuration in which multiple resource groups have the same default master, a resource group (and its associated applications) can fail over or be switched over to a secondary node. Meanwhile, the default master is running in the cluster.

Note – During failover, the application that fails over is allocated resources as specified in the configuration file on the secondary node. In this example, the project database files on the primary and secondary nodes have the same configurations.

For example, this sample configuration file specifies that Application 1 is allocated 1 share, Application 2 is allocated 2 shares, and Application 3 is allocated 2 shares.

```
Prj_1:106:project for App_1:root::project.cpu-shares=(privileged,1,none)
Prj_2:107:project for App_2:root::project.cpu-shares=(privileged,2,none)
Prj_3:108:project for App_3:root::project.cpu-shares=(privileged,2,none)
```

The following diagram illustrates the normal and failover operations of this configuration, where RG-2, containing Application 2, fails over to `phys-schost-2`. Note that the number of shares assigned does not change. However, the percentage of CPU time available to each application can change, depending on the number of shares that are assigned to each application that demands CPU time.



Public Network Adapters and IP Network Multipathing

Clients make data requests to the cluster through the public network. Each cluster node is connected to at least one public network through a pair of public network adapters.

Oracle Solaris Internet Protocol (IP) Network Multipathing software on Oracle Solaris Cluster provides the basic mechanism for monitoring public network adapters and failing over IP addresses from one adapter to another when a fault is detected. Each node has its own IP network multipathing configuration, which can be different from the configuration on other nodes.

Public network adapters are organized into *IP multipathing groups*. Each multipathing group has one or more public network adapters. Each adapter in a multipathing group can be active. Alternatively, you can configure standby interfaces that are inactive unless a failover occurs.

The `in.mpathd` multipathing daemon uses a test IP address to detect failures and repairs. If a fault is detected on one of the adapters by the multipathing daemon, a failover occurs. All network access fails over from the faulted adapter to another functional adapter in the

multipathing group. Therefore, the daemon maintains public network connectivity for the node. If you configured a standby interface, the daemon chooses the standby interface. Otherwise, the daemon chooses the interface with the least number of IP addresses. Because the failover occurs at the adapter interface level, higher-level connections such as TCP are not affected except for a brief transient delay during the failover. When the failover of IP addresses completes successfully, ARP broadcasts are sent. Therefore, the daemon maintains connectivity to remote clients.

Note – Because of the congestion recovery characteristics of TCP, TCP endpoints can experience further delay after a successful failover. Some segments might have been lost during the failover, activating the congestion control mechanism in TCP.

Multipathing groups provide the building blocks for logical host name and shared address resources. The same multipathing group on a node can host any number of logical host name or shared address resources. For more information about logical host name and shared address resources, see the *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

Note – TheIP network multipathing mechanism is meant to detect and mask adapter failures. It is not intended to recover from an administrator's use of `ifconfig` to remove one of the logical (or shared) IP addresses. The Oracle Solaris Cluster software views the logical and shared IP addresses as resources that are managed by the RGM. The correct way for an administrator to add or to remove an IP address is to use `clresource` and `clresourcegroup` to modify the resource group that contains the resource.

For more information about the Oracle Solaris implementation of IP Network Multipathing, see Part V, "IPMP," in *Oracle Solaris Administration: IP Services*.

SPARC: Dynamic Reconfiguration Support

Oracle Solaris Cluster supports the dynamic reconfiguration (DR) software feature. This section describes concepts and considerations for Oracle Solaris Cluster support of the DR feature.

All the requirements, procedures, and restrictions that are documented for the Oracle Solaris DR feature also apply to Oracle Solaris Cluster DR support except for the operating environment quiescence operation. Therefore, review the documentation for the Oracle Solaris DR feature *before* by using the DR feature with Oracle Solaris Cluster software. You should review in particular the issues that affect non-network IO devices during a DR detach operation.

DR implementation can be system dependent, and might be implemented differently as technology changes. For more information, see *Oracle Solaris Cluster Data Service for Oracle VM Server for SPARC Guide*.

SPARC: Dynamic Reconfiguration General Description

The DR feature enables operations such as the removal of system hardware in running systems. The DR processes are designed to ensure continuous system operation with no need to halt the system or interrupt cluster availability.

DR operates at the board level. Therefore, a DR operation affects all the components on a board. Each board can contain multiple components, including CPUs, memory, and peripheral interfaces for disk drives, tape drives, and network connections.

Removing a board that contains active components would result in system errors. Before removing a board, the DR subsystem queries other subsystems, such as Oracle Solaris Cluster, to determine whether the components on the board are being used. If the DR subsystem finds that a board is in use, the DR remove-board operation is not done. Therefore, issuing a DR remove-board operation is always safe because the DR subsystem rejects operations on boards that contain active components.

The DR add-board operation is also always safe. CPUs and memory on a newly added board are automatically brought into service by the system. However, the system administrator must manually configure the cluster to actively use components that are on the newly added board.

Note – The DR subsystem has several levels. If a lower level reports an error, the upper level also reports an error. However, when the lower level reports the specific error, the upper level reports `Unknown error`. You can safely ignore this error.

The following sections describe DR considerations for the different device types.

SPARC: DR Clustering Considerations for CPU Devices

Oracle Solaris Cluster software does not reject a DR remove-board operation because of the presence of CPU devices.

When a DR add-board operation succeeds, CPU devices on the added board are automatically incorporated in system operation.

SPARC: DR Clustering Considerations for Memory

For the purposes of DR, there are two types of memory:

- Kernel memory cage
- Non-kernel memory cage

These two types differ only in usage. The actual hardware is the same for both types. Kernel memory cage is the memory that is used by the Oracle Solaris OS. Careful consideration must be taken before performing a DR remove-board operation which will impact kernel memory cage. Oracle Solaris Cluster software does not reject the operation, but in most cases, such a DR operation will have a significant impact on the entire cluster. The tight coupling between cluster nodes, between multiple instances of scalable applications, and between the primary and secondary nodes of HA applications and services means that the quiescing of one node for repair can cause operations on non-quiesced nodes to be delayed until the repair operation is complete and the node is unquiesced.

In most cases, the preferred method of removing or replacing a system board with kernel cage memory is to bring the node requiring repair down. The remainder of the cluster can then cleanly take over the duties of the node being repaired. Only when circumstances prevent the node requiring repair from being brought out of the cluster should DR be used to remove or replace a system board with kernel cage memory while the node is still part of the operating cluster. For suggestions on preparing the cluster for a DR kernel cage remove-board operation, see "Preparing the Cluster for Kernel Cage DR" in *Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual*.

When a DR add-board operation that pertains to memory succeeds, memory on the added board is automatically incorporated in system operation.

If the node being repaired panics during the DR operation, or if the DR operation is otherwise interrupted, you might have to manually re-enable heartbeat monitoring and reset the repaired node's quorum vote count. These two actions are normally done automatically at the completion of the DR operation to return the cluster to a stable state. For instructions on recovering in this case, see "How to Recover From an Interrupted Kernel Cage DR Operation" in *Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual*.

SPARC: DR Clustering Considerations for Disk and Tape Drives

Oracle Solaris Cluster rejects dynamic reconfiguration (DR) remove-board operations on active drives on the primary node. You can perform DR remove-board operations on inactive drives on the primary node and on any drives in the secondary node. After the DR operation, cluster data access continues as before.

Note – Oracle Solaris Cluster rejects DR operations that affect the availability of quorum devices. For considerations about quorum devices and the procedure for performing DR operations on them, see "SPARC: DR Clustering Considerations for Quorum Devices" on page 91.

See "Dynamic Reconfiguration With Quorum Devices" in *Oracle Solaris Cluster System Administration Guide* for detailed instructions about how to perform these actions.

SPARC: DR Clustering Considerations for Quorum Devices

If the DR remove-board operation pertains to a board that contains an interface to a device configured for quorum, Oracle Solaris Cluster software rejects the operation. Oracle Solaris Cluster software also identifies the quorum device that would be affected by the operation. You must disable the device as a quorum device before you can perform a DR remove-board operation.

See Chapter 6, "Administering Quorum," in *Oracle Solaris Cluster System Administration Guide* for detailed instructions about how administer quorum.

SPARC: DR Clustering Considerations for Cluster Interconnect Interfaces

If the DR remove-board operation pertains to a board containing an active cluster interconnect interface, Oracle Solaris Cluster software rejects the operation. Oracle Solaris Cluster software also identifies the interface that would be affected by the operation. You must use an Oracle Solaris Cluster administrative tool to disable and remove the active interface before the DR operation can succeed.

Caution – Oracle Solaris Cluster software requires each cluster node to have at least one functioning path to every other cluster node. Do not disable a private interconnect interface that supports the last path to any node in the cluster.

See "Administering the Cluster Interconnects" in *Oracle Solaris Cluster System Administration Guide* for detailed instructions about how to perform these actions.

SPARC: DR Clustering Considerations for Public Network Interfaces

If the DR remove-board operation pertains to a board that contains an active public network interface, Oracle Solaris Cluster software rejects the operation. Oracle Solaris Cluster software also identifies the interface that would be affected by the operation. Before you remove a board with an active network interface present, switch over all traffic on that interface to another functional interface in the multipathing group by using the `if_mpadm` command.

Caution – If the remaining network adapter fails while you are performing the DR remove operation on the disabled network adapter, availability is affected. The remaining adapter has no place to fail over for the duration of the DR operation.

See "Administering the Public Network" in *Oracle Solaris Cluster System Administration Guide* for detailed instructions about how to perform a DR remove operation on a public network interface.

Index

D

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

Z