Netra High Availability Suite Foundation Services 2.1 6/03 Overview

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–1761–11
September 2004

040813@9495

Contents

Tables

Figures

Preface

The *Netra High Availability Suite Foundation Services 2.1 6/03 Overview* introduces the Netra™ High Availability (HA) Suite Foundation Services 2.1 6/03. This book describes the concepts that the Foundation Services are built on and the architecture that they are built around. This book also helps you to plan to install a cluster running the Foundation Services.

Who Should Use This Book

This book is for system administrators who are maintaining a cluster running the Foundation Services, or for system developers who are developing applications for a cluster running the Foundation Services.

How This Book Is Organized

This book is organized in two parts. Part I introduces you to the Foundation Services. It describes the concepts that the Foundation Services are built on and the factors that you must consider when planning to install the Foundation Services. Part II describes each of the Foundation Services.

Chapter 11 refers to RFC standards. For further information, see the complete text of RFC papers at `http://www.ietf.org/`.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other material on or available from such sites or resources. Sun will not be responsible or liable for any damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through any such sites or resources.

Related Books

You will require some of the following books from the Foundation Services documentation set:

- *Netra High Availability Suite Foundation Services 2.1 6/03 Overview*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Glossary*
- *What's New in Netra High Availability Suite Foundation Services 2.1 6/03*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Quick Start Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Troubleshooting Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 CMM Programming Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Reference Manual*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Standalone CGTP Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes*
- *Netra High Availability Suite Foundation Services 2.1 6/03 README*

Documentation Accessibility Features

This documentation set is delivered in both PDF and HTML formats. The HTML version of the documents includes the following accessibility features:

- Text equivalents for graphics

Alternative text labels are assigned to graphics. Where graphics provide detailed descriptions, text versions of these descriptions are provided within the surrounding text.

- Tables that can be interpreted by assistive technology

  All tables include descriptive headers. A brief description of the table contents is also provided in the surrounding text.

Accessing Sun Documentation Online

The docs.sun.com[SM] Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print form. For a list of documents and how to order them, see "Buy printed documentation" at `http://docs.sun.com`.

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	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.`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface or Symbol	Meaning	Example
AaBbCc123	What you type, contrasted with onscreen computer output	`machine_name% su` `Password:`
AaBbCc123	Command-line 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*. These are called *class* options. Do *not* save the file. (Emphasis sometimes appears in bold online.)

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

PART I

Introduction

This part introduces the Foundation Services. It describes the concepts on which the Foundation Services are built, and provides information to help you plan your cluster. For an introduction to the Foundation Services, see the following chapters:

- Chapter 1 briefly describes each of the Foundation Services. This chapter also describes the tools that facilitate cluster installation and administration.
- Chapter 2 describes the concepts on which the Foundation Services are built. You must be familiar with the concepts described in this chapter before installing and using the Foundation Services.
- Chapter 3 enables you to plan your cluster configuration. It lists the hardware and software that you require, and describes the different installation methods.

CHAPTER 1

Introduction to the Foundation Services

This chapter introduces the Foundation Services. For a brief description of each of the Foundation Services, the installation tools, and the cluster administration tools, see the following sections:

What Are the Foundation Services?

The Foundation Services are a suite of reliable software services that run on the SPARC Solaris™ operating system. The Foundation Services enable you to deploy applications in a continuous availability environment. The Foundation Services can be used to create a highly available, dynamically scalable cluster of distributed nodes, or to augment existing highly available frameworks.

The following figure illustrates a basic Foundation Services cluster.

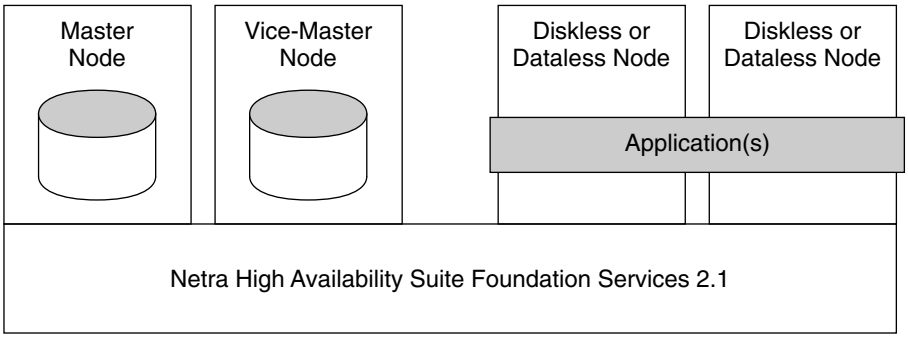


FIGURE 1–1 Basic Foundation Services Cluster

The concepts of cluster, master node, vice-master node, diskless node, and dataless node are described in "Cluster Model" on page 21.

The Foundation Services have been designed to support the following:

- Hardware replacement or upgrade, and diagnostics without incurring system outage
- Redundant services
- Redundant dual Ethernet links
- Redundant platform services such as Reliable NFS and the Reliable Boot Service

High-Level View of the Foundation Services

The following figure shows a high-level view of the Foundation Services architecture.

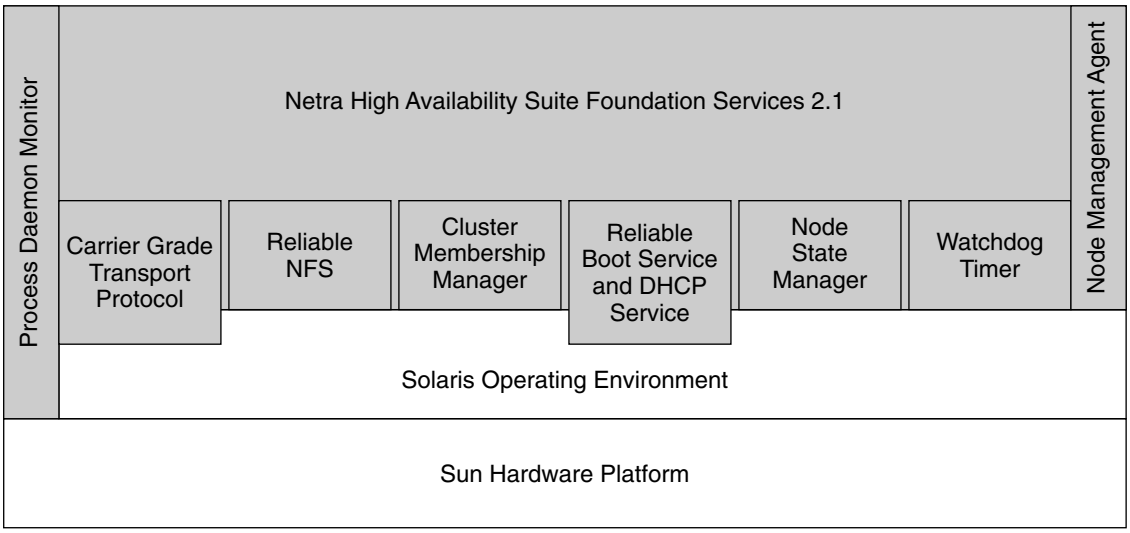


FIGURE 1–2 High-Level View of the Foundation Services Architecture

The Foundation Services software offers the following services:

- A reliable IP transport mechanism provided by the Carrier Grade Transport Protocol (CGTP). CGTP limits the consequences of single network failure by duplicating the communication links. For further information, see Chapter 6.
- Reliable NFS to ensure that data is accessible to clients, even in the event of hardware or software failure. Reliable NFS uses mounted file systems, IP mirroring of disk-based data, and IP address failover of the master role. For further information, see Chapter 7.
- A Cluster Membership Manager to provide a global view of the cluster. The Cluster Membership Manager determines which nodes are members of the cluster. It assigns the roles and attributes of nodes, detects the failure of nodes, and notifies clients of changes to the cluster. A heartbeat mechanism is used to detect node failure. For further information, see Chapter 8.
- A Reliable Boot Service and the Solaris Dynamic Host Configuration Protocol service to ensure the boot of diskless nodes regardless of software or hardware failures. For further information, see Chapter 9.
- A Node State Manager with scripts that provide access for external networks to the node holding the master role. For further information, see Chapter 5.
- A Daemon Monitor to survey Foundation Services daemons, many Solaris operating system daemons, and some companion product daemons. If any of the monitored daemons fail, the Daemon Monitor initiates a recovery response. The Daemon Monitor is itself monitored by the Node Management Agent. For further information, see Chapter 10.

- A Node Management Agent to monitor cluster statistics. The Node Management Agent can initiate a switch to the backup node, change some error recovery responses, and listen for notifications of some cluster events.

  The Node Management Agent is compliant with the Java™ Management Extensions (JMX™) and based on the Java Dynamic Management Kit. For further information, see Chapter 11.
- A Watchdog Timer for low-level system monitoring of the Foundation Services. For further information, see Chapter 12.

Each of these services is described in detail in Part II.

Foundation Services Tools

The Foundation Services include a suite of tools to facilitate installation and cluster administration. There tools and their purpose are as follows:

`nhadm`	Perform administration tasks on a cluster
`nhcmmqualif`	Qualify the current node as master
`nhcmmrole`	get the role of the current node
`nhcmmstat`	display information about peer nodes, trigger a switchover, or force the qualification of a master-eligible node
`nhcrfsadm`	command line tool for Reliable NFS administration
`nhenablesync`	trigger disk synchronization
`nhinstall`	install and configure the Foundation Services
`nhpmdadm`	process monitor daemon administration tool

For information about using `nhinstall`, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*.

In addition to the Foundation Services tools, many Solaris tools can be used for administration of a cluster. For information about using these tools, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.

CHAPTER 2

Concepts Used in the Foundation Services

This chapter describes the concepts on which the Foundation Services are built. Make sure that you are familiar with the concepts described in this chapter before installing and using the Foundation Services.

This chapter includes the following topics:

Cluster Model

This section describes the cluster environment and the types of nodes in a cluster.

A cluster is a set of interconnected nodes that collaborate to provide highly available services. The recommended cluster configuration is to have:

- A master node to coordinate the cluster membership information and to provide the highly available services
- A vice-master node that backs up the master node
- Replication of data between the master node and the vice-master node
- A redundant network between all nodes for reliable inter-node communication

In addition to the master node and the vice-master node, a cluster can contain other nodes. The following figure is an example of the nodes inside and outside a cluster.

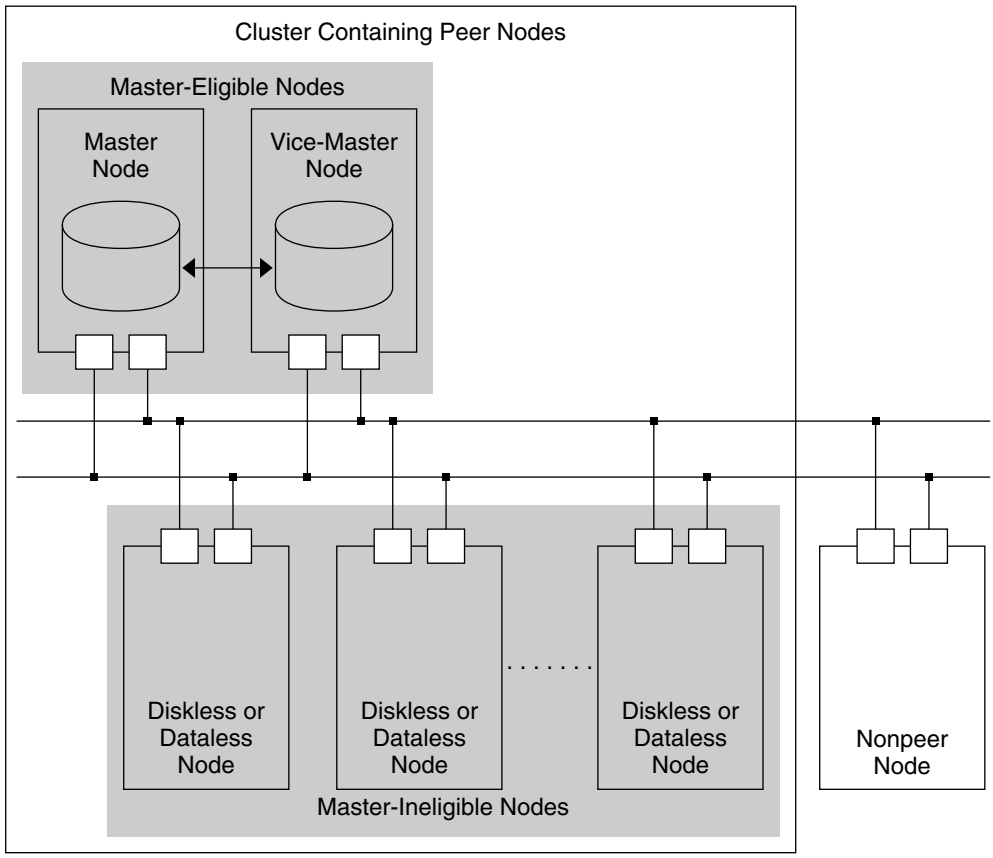


FIGURE 2–1 Example of Nodes Inside and Outside a Cluster

The types of nodes in Figure 2–1 are described in the following sections.

Peer Nodes and Nonpeer Nodes

Nodes that are configured as members of a cluster are called *peer nodes*. Peer nodes can run the Foundation Services and communicate with each other on the same network. Peer nodes can be master-eligible nodes or master-ineligible nodes.

Nodes that are not configured as members of a cluster are called *nonpeer nodes*. A nonpeer node communicates with one or more peer nodes to access resources or services provided by the cluster. In Figure 2–1, the nonpeer node is connected to both of the redundant network links. For information about the options for connecting nonpeer nodes to a cluster, see Chapter 5.

Master-Eligible Nodes

A cluster contains two *master-eligible* nodes. A master-eligible node is a peer node that can be elected as the master node or the vice-master node.

The *master node* is the node that coordinates the cluster membership information. The master node generates its view of the cluster configuration. It communicates this view to the other peer nodes. The master node provides the Reliable Boot Service and Reliable NFS to the cluster.

The *vice-master node* backs up the master node. The vice-master node has a copy of all of the cluster management information that is on the master node. It can transparently take control of the cluster if required.

You must take care that any tasks you run on the vice-master node either have a very low-level of load, or are designed in such a way that the tasks can be interrupted if the vice-master needs to become master. The vice-master must always be available to take over the master node's load if the current master is no longer able to continue in the master role.

Each master-eligible node must be a *diskfull node*. A diskfull node has at least one disk on which information can be permanently stored. A master-eligible node must be configured as master-eligible at the time of installation and configuration. The master node and vice-master node are the only nodes that are configured as diskfull in a Foundation Services cluster.

Master-Ineligible Nodes

A cluster can contain only two master-eligible nodes, the master node and the vice-master node. All other peer nodes are *master-ineligible*.

In a Foundation Services cluster, master-ineligible nodes are either *diskless nodes* or *dataless nodes*. For examples of supported cluster configurations, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

A diskless node either does not have a local disk or is configured not to use its local disk. Diskless nodes boot through the network, using the master node as a boot server.

A dataless node has a local disk from which it boots, but it cannot store data permanently on its disk. A dataless node accesses the Foundation Services through the cluster network.

Data generated on diskless nodes and dataless nodes is sent to the master node.

Reliability, Serviceability, Redundancy, and Availability

This section defines the concepts of reliability, serviceability, redundancy, and availability. These concepts use the mechanisms of failover and switchover, described in "Failover and Switchover" on page 25.

Reliability

Reliability is a measure of continuous system uptime. The Foundation Services include Reliable NFS, CGTP, and the Reliable Boot Service to increase the reliability of your system.

Serviceability

Serviceability is the probability that a service can be restored within a specified period of time following a service failure. The Foundation Services have a high degree of serviceability through node switchover. Switchover ensures a transfer of services and data from a node requiring maintenance to a backup node.

Redundancy

Redundancy increases the availability of a service by providing a backup to take over in the event of failure.

The Foundation Services use the 2N redundancy model. The master node is backed up by the vice-master node. If the master node fails, there is a transparent transfer of services to the vice-master node. In the Foundation Services, the instance of the service running on the master node is the primary instance. The instance of the service running on the vice-master node is the secondary instance.

The Foundation Services provide file system redundancy. The file system on the master node is replicated on the vice-master node.

The Foundation Services also provide transport redundancy. All cluster transport is duplicated over dual, redundant Ethernet links. If a link fails, a copy of a data packet can still reach its destination through the other link.

Availability

Availability is the probability that a service is available for use at any given time. Availability is a function of system reliability and serviceability, supported by redundancy.

Failover and Switchover

Failover and switchover are the mechanisms that ensure the high availability of a cluster.

Failover occurs if the master node fails, or if a vital service running on the master node fails. The services on the master node fail over to the vice-master node. The vice-master node has all of the necessary state information to take over from the master node. The vice-master node expects no cooperation or coordination from the failed master node.

Switchover is the planned transfer of services from the master node to the vice-master node. Switchover is orchestrated by the system or by an operator so that a node can be maintained without affecting system performance. Switchover is not linked to node failure. As in the case of a failover, the backup must have all of the necessary state information to take over at the moment of the switchover. Unlike failover, in switchover the master node can help the vice-master node by, for example, flushing caches for shared files.

Only the master node and vice-master node take part in failover and switchover. If a diskless node or dataless node fails, there is no failover. If a diskless node or dataless node is the only node running an application, the application fails. If there are other diskless nodes or dataless nodes running the application, the application will continue to run on these other nodes.

Service Models

The Foundation Services can be divided into two categories of service: highly available services and distributed services.

Highly available services run on the master node and vice-master node only. The Reliable Boot Service and Reliable NFS are highly available services. If the master node or one of these services on the master node fails, a failover occurs.

Distributed services are services that run on all peer nodes. The distributed services include the Cluster Membership Manager, the Node State Manager, the Node Management Agent, the Daemon Monitor, and the Watchdog Timer. If a distributed service fails and cannot be restarted, the node running the service is removed from the cluster. If the node is the master node, a failover occurs.

Fault Management Models

This section describes some of the faults that can occur in a cluster, and how those faults are managed.

Fault Types

When one critical fault occurs, it is called a *single fault*. A single fault can be the failure of one master-eligible node, the failure of a service, or the failure of one of the redundant networks. After a single fault, the cluster continues to operate correctly but it is not highly available until the fault is repaired.

When two critical faults occur that affect both parts of the redundant system, it is called a *double fault*. A double fault can be the simultaneous failure of both master-eligible nodes, or the simultaneous failure of both redundant network links. Although many double faults can be detected, it might not be possible to recover from all double faults. Although rare, double faults can result in cluster failure.

Some faults can result in the election of two master nodes. This error scenario is called *split brain*. Split brain is usually caused by communication failure between master-eligible nodes. When the communication between the master-eligible nodes is restored, the last elected master node remains the master node. The other master-eligible node is elected as the vice-master node. The synchronized state must then be restored using Reliable NFS.

When a peer node does not receive information from the master node for more than 10 seconds, a *stale cluster* error occurs. A cluster becomes stale if the master node does not send information to the peer node, or if information does not reach the peer node.

When a cluster restarts with stale cluster configuration data, the fault is called *amnesia*. Amnesia is caused by restarting the cluster from a node that was not previously part of the most recent cluster membership list.

Fault Detection

Fault detection is critical for a cluster running highly available applications. The Foundation Services have the following fault detection mechanisms:

- The Cluster Membership Manager detects the failure of peer nodes. It notifies the other peer nodes of the failure. For information about the Cluster Membership Manager, see Chapter 8.
- The Daemon Monitor supervises Foundation Services daemons, many Solaris operating system daemons, and some companion products daemons. When a critical service or a descendent of a critical service fails, the Daemon Monitor

detects the failure and triggers a recovery response. For information about the Daemon Monitor, see Chapter 10.

- The Watchdog Timer monitors hardware watchdogs at the lights-off management level. For information about the Watchdog Timer, see Chapter 12.

Fault Reporting

Errors that indicate potential failure are reported so that you can understand the sequence of events that have led to the problem. The Foundation Services have the following fault reporting mechanisms:

- All error messages are sent to system log files. For information about how to configure log files, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.
- The Cluster Membership Manager on the master node notifies clients when a node fails, a failover occurs, or the cluster membership changes. Clients can be subscribed system services or applications.
- The Node Management Agent can be used to develop applications that retrieve statistics on the Cluster Membership Manager, CGTP, Reliable NFS, and the Daemon Monitor. These applications can be used to detect faults or diminished levels of service on your system. For further information on how to collect and manage node and cluster statistics, see the *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*.

Fault Isolation

Fault isolation has two aspects: isolation and redundancy. When a fault occurs in the cluster, the node on which the fault occurred is isolated. The Cluster Membership Manager ensures that the failed node cannot communicate with the other peer nodes.

If the master node fails, a failover occurs. If a diskless node or dataless node fails, there is no failover.

Fault Recovery

The first recovery response to a critical failure is the failover to a backup node or service. Failover ensures the continuation of a service until the failure is repaired.

Failed nodes are often repaired by reboot. Overload errors are often repaired by waiting for an acceptable delay and then rebooting or restarting the failed service. The Foundation Services are designed so that individual nodes can be shut down and restarted independently, reducing the impact of errors. After failover, the master node and vice-master node are synchronized so that the repaired vice-master node can rejoin the cluster in its current state.

CHAPTER 3

Planning Your Cluster

Planning your cluster configuration is essential. Before you start to install your cluster, you must consider what hardware and software you require.

This chapter helps you to choose your cluster configuration and a method to install the Foundation Services on your cluster. This chapter describes the hardware and software required for a two-node cluster and for larger clusters. It also points to documents that provide detailed information about hardware configurations and installation.

For information about cluster requirements and installation methods, see the following sections:

Defining Your Cluster

You can use a cluster to perform the following tasks:

- Test newly developed applications.
- Run existing user applications.
- Run the Foundation Services without a user application, to take advantage of the high availability of Reliable NFS, the Reliable Boot Service, and the DHCP services.

Your choice of cluster hardware and software is influenced by the type of applications you have or the type of applications that you are developing. Before you choose your cluster hardware, consider the following questions:

- Do you want diskless nodes or dataless nodes in your cluster?

- How many nodes do you require in your cluster?
- What types of boards, network cards, and disks do you need?
- What switches, I/O cards, and terminal concentrator do you need?
- How can you access the cluster from an external network?

For example hardware configurations that represent a range of typical clusters, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

If you plan to configure a small cluster now and add nodes to it in the future, configure your cluster for its maximum size. A cluster can contain fewer diskless nodes or dataless nodes than it is configured for, but it cannot contain more. For information about adding nodes to a cluster, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.

Hardware and Software Requirements

This section summarizes the hardware and software that you require for a cluster. To install a cluster you need the following hardware:

- An installation server with which to install the software on the cluster
- A cluster on which to run the software

The following figure illustrates the hardware required to install a cluster.

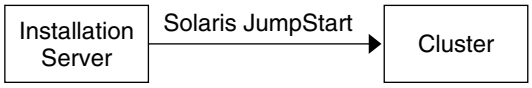


FIGURE 3–1 Hardware Required for Installation

For detailed information about setting up your installation server and about your example cluster configurations, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

For information about how to install a cluster, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*.

Hardware Requirements for a Cluster

For a cluster, you require the following hardware types:

- Master-eligible node hardware
- Optional diskless node hardware or dataless node hardware

- Ethernet switches
- A terminal server
- Two network interface cards on each node
- Optional supplementary Ethernet interface cards for external access

For information about the boards, cards, and disks that you can use, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

Software Requirements for a Cluster

To run the Foundation Services on a cluster of two master-eligible nodes, you require the Solaris operating system. For information about other supported software, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes*.

Installation Methods

The Foundation Services provide `nhinstall`, a tool for installing a cluster. The `nhinstall` tool provides an easy installation with a high level of flexibility. You can also install a cluster manually.

Note – You cannot install dataless nodes using the `nhinstall` tool.

For information about the requirements of the `nhinstall` tool and manual installation, see "Hardware and Software Requirements" on page 30.

PART II

Description of the Foundation Services

This part describes in detail each of the Foundation Services:

- Chapter 4 describes the cluster addressing and networking of the Foundation Services.
- Chapter 5 describes the external addressing scheme. It explains the options for connecting external networks to the cluster network.
- Chapter 6 discusses the reliable IP transport mechanism provided by the Carrier Grade Transport Protocol.
- Chapter 7 explains how the master node disk and vice-master node disk are partitioned and mirrored using Reliable NFS.
- Chapter 8 describes how the cluster membership is managed and configured, and how peer nodes are monitored.
- Chapter 9 covers the Reliable Boot Service. It describes how diskless nodes are booted.
- Chapter 10 describes how the Daemon Monitor is used to survey other process daemons. It also includes how the Daemon Monitor itself can be monitored and manipulated.
- Chapter 11 explains how the Node Management Agent can be used to monitor cluster statistics. It describes how the Node Management Agent initiates a switchover, changes the recovery response for daemon failure, and listens for notifications of cluster events.
- Chapter 12 describes how the Watchdog Timer guards against operating system hang and boot failure.

CHAPTER 4

Cluster Addressing

For a description of how a Foundation Services cluster can be addressed, see the following sections:

Introduction to Cluster Addressing

Peer nodes communicate with each other over a private network, called the *cluster network*.

The addressing scheme used by the cluster network is classless. The IP addresses of peer nodes can be in a private network or a public network. It is advantageous to configure a cluster in a private network for the following reasons:

- There is no need to allocate addresses from a public address space for each cluster.
- External data can be added into or taken out of the cluster network in a controlled way, making the cluster network more secure.
- The cluster network cannot be overloaded by external traffic.
- The redundant network symmetry is protected.

Cluster Addressing Scheme

All peer nodes are assigned IPv4 addresses. The address is split into a *network part* and a *host part*. The dividing point between the network part and the host part is defined by the *netmask*.

The following figure illustrates the structure of an IP address on a peer node.

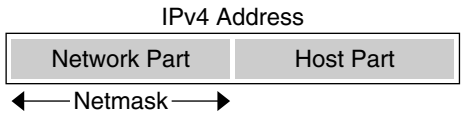


FIGURE 4–1 Structure of an IP Address on a Peer Node

The network part of an IP address represents the identity of the network to which a peer node is connected. The host part of an IP address represents the `nodeid` of the peer node. The `nodeid` is the decimal equivalent of the host part of the IP address.

In a class B addressing scheme, the value of the netmask is `ffff0000`. The network part of an IP address is 16 bits long, and the host part of an IP address is 16 bits long.

In the class C addressing scheme, the value of the netmask is `ffffff00`. The network part of an IP address is 24 bits long, and the host part of an IP address is 8 bits long. By default, a class C address on a Foundation Services cluster has the following format:

`10`.*domainid.interfaceid.nodeid*

The Foundation Services can use a classless addressing scheme. The value of the netmask is not restricted to any address class.

The following values of the host part of the IP address are reserved:

- Value “0” is reserved for the identification of the network part of the IP address.
- Value “1” is reserved for the floating address triplet. For information about the floating address triplet, see “Floating Address Triplet” on page 38.
- Value 2^n-1 is reserved for the broadcast address. The parameter *n* is the number of bits in the host part of the IP address.

Node Address Triplets

A *node address triplet* is assigned to each peer node in a cluster. The node address triplet consists of three IP addresses:

- An IP address for each of the two physical interfaces, `NIC`*0* and `NIC`*1*
- An IP address for the virtual interface, called the *CGTP address*

The CGTP address is used for IP routing. It hides the multirouting capability of CGTP. The CGTP address presents the redundant network as a single network interface. Applications can use the CGTP address to communicate with the master node. The CGTP address supports unicast, broadcast, and multicast transmissions. For more information about CGTP, see Chapter 6.

The following table shows an example of node address triplets for a cluster containing two master-eligible nodes and two diskless nodes. The cluster uses a default class C addressing scheme.

TABLE 4–1 Example of Node Address Triplets for a Four-Node Cluster

Node Name	Address for Physical Interface `hme`*0*	Address for Physical Interface `hme`*1*	Address for Virtual Interface `cgtp`*0*
Master Node	`10.200.1.10`	`10.200.2.10`	`10.200.3.10`
Vice-Master Node	`10.200.1.11`	`10.200.2.11`	`10.200.3.11`
Diskless Node 1	`10.200.1.12`	`10.200.2.12`	`10.200.3.12`
Diskless Node 2	`10.200.1.13`	`10.200.2.13`	`10.200.3.13`

The following figure shows the address triplets of the preceding cluster.

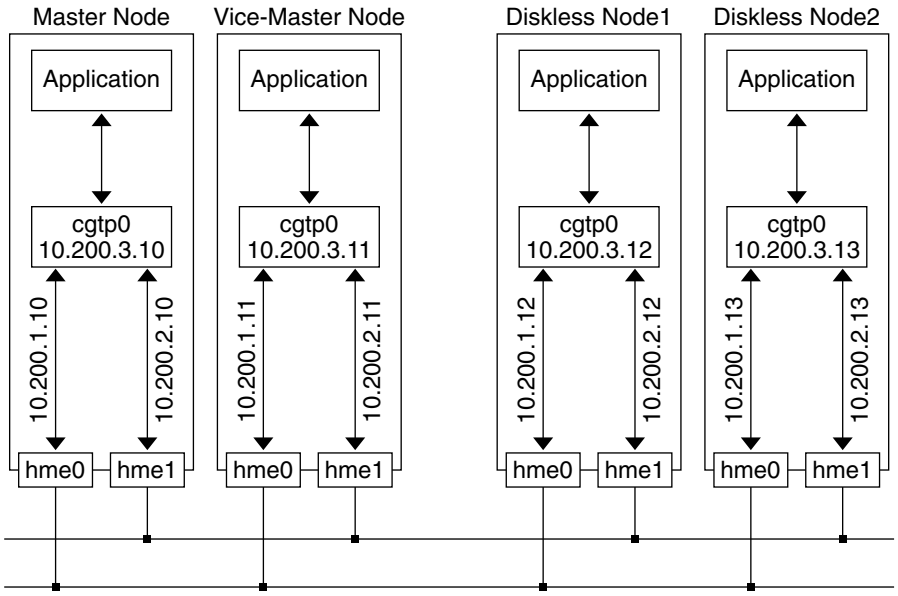


FIGURE 4–2 Node Address Triplets

Floating Address Triplet

In addition to a node address triplet, the master node and vice-master node have an address triplet called the *floating address triplet*. The floating address triplet is activated on the master node only. If the master node fails over or is switched over, the floating address triplet is activated on the new master node. It is deactivated on the old master node.

Diskless nodes and dataless nodes access services and data on the master node, through the floating address triplet. Because the floating address triplet is always up on the master node, the diskless nodes and dataless nodes can access the master node even after a failover or switchover.

The floating address triplet has a *logical address*. A logical address is an address that is assigned to a physical interface or virtual interface. A logical address for an `hme0` or `cgtp0` interface has the format `hme0:x` or `cgtp0:x`.

The following table shows an example of the node address triplet and floating address triplet of a master node. The cluster is using a default class C addressing scheme.

TABLE 4–2 Example of Master Node Address Triplets

Triplet Type	Address Type	Interface	Address Example
Node Address Triplet	Physical	`hme`*0*	`10.200.1.10`
	Physical	`hme`*1*	`10.200.2.10`
	Virtual Physical	`cgtp`*0*	`10.200.3.10`
Floating Address Triplet	Logical	`hme`*0:1*	`10.200.1.1`
	Logical	`hme`*1:1*	`10.200.2.1`
	Virtual Logical	`cgtp`*0:1*	`10.200.3.1`

Figure 4–3 shows the node address triplet and floating address triplet of the master node and vice-master node. The diskless nodes are mounted onto the master node. The floating address triplet of the vice-master node is barred out because it is down.

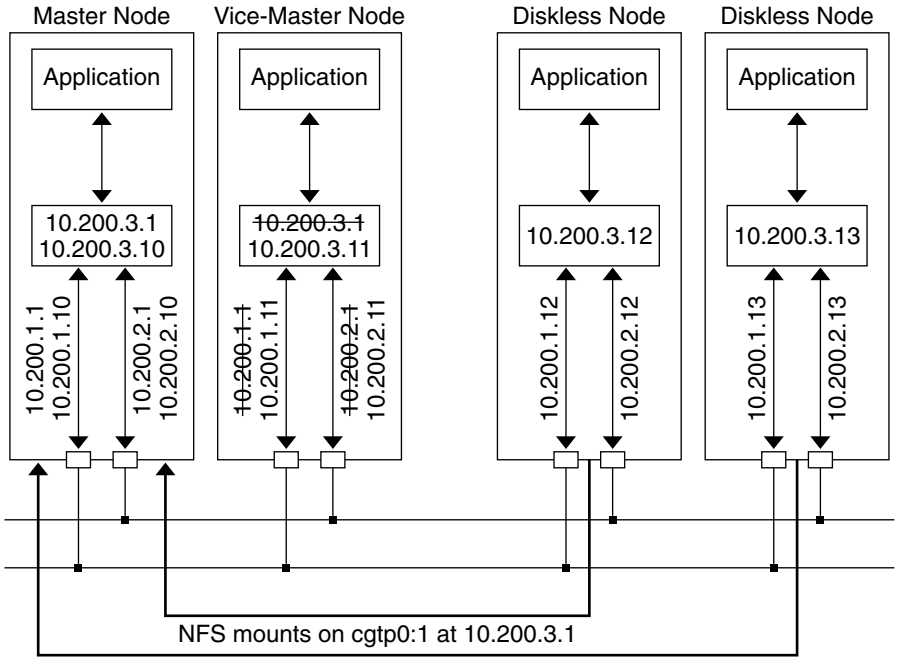


FIGURE 4–3 Example of the Floating Address Triplet of a Master Node and Vice-Master Node

Figure 4–4 shows the cluster in Figure 4–3 after a failover. The diskless nodes are mounted onto the new master node.

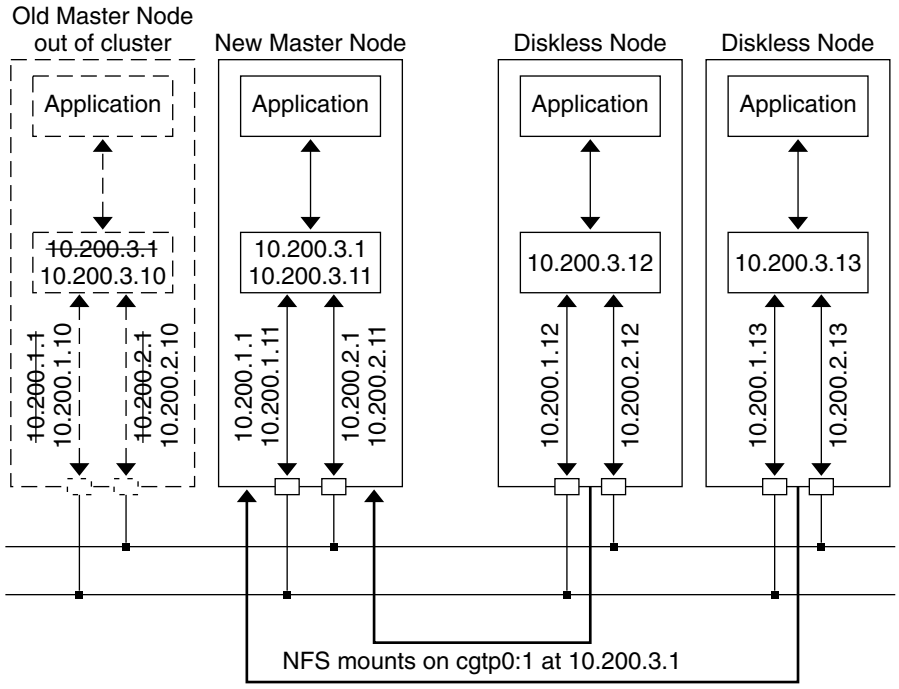


FIGURE 4–4 Example of the Floating Address Triplet After Failover

CHAPTER 5

External Addressing

For a description of the options for connecting nonpeer nodes to the cluster network, see the following sections:

Introduction to External Addressing

An *external network* communicates with a cluster running the Foundation Services to perform one or more of the following tasks:

- Access one or more of the Foundation Services, such as the Reliable NFS service
- Access services provided by user applications
- Retrieve cluster information and statistics from the Node Management Agent
- Debug and maintain the cluster
- Install software from a build server or an installation server
- Install software from a development host

The external network can be an Ethernet, ATM, or any other network type supported by the Solaris operating system.

The addressing scheme used by the cluster network is classless. The IP addresses of peer nodes can be in a private network or a public network. An external network can connect to a cluster network in the following ways:

- Directly to the cluster network

If the cluster IP addresses are in a private network, logical interfaces must be created to connect the external network to the cluster network.

If the cluster IP addresses are in the same subnetwork as the external network, it is not necessary to create logical interfaces.

- Through additional physical interfaces on the peer nodes
- Through a router

External Addressing Scheme

External addresses have no inherent relationship to internal cluster addresses. External addresses are flexible. They can be single addresses, or multiple addresses combined using IP multipathing. They can be IPv4 or IPv6.

Floating External Addresses

A logical address assigned to an interface that connects the master node to an external network is called a *floating external address*.

The Node State Manager (NSM) uses Cluster Membership Manager notifications to determine when a node is promoted to or demoted from the master role. When a node is promoted to the master role, the NSM configures a floating external address for one of the node's external interfaces. When a node is demoted from the master role, the NSM unconfigures the floating external address.

The floating external address enables clients on an external network to access the master node. Because the floating external address is always configured on the master node, clients on an external network can always access the master node, even after failover and switchover.

The NSM can be used for tasks other than address management. For information about how to configure the NSM, see the `nhnsmd`(1M) and `nhfs.conf`(4) man pages.

Connecting Nonpeer Nodes Directly to a Cluster Network

This section describes how a nonpeer node can be connected directly to a cluster network. Connecting a nonpeer node directly to the cluster network is disadvantageous for the following reasons:

- Internal traffic can leave the cluster network, compromising security.
- External traffic can enter the cluster network, reducing network performance.
- If the external network is connected to one of the cluster networks only, the traffic on the two cluster network paths can become asymmetric. This could affect the performance of the redundant transport mechanism provided by CGTP.

Figure 5–1 and Figure 5–2 show examples of how a nonpeer node can be connected directly to a cluster network.

In Figure 5–1, the nonpeer node is connected to the `hme`*0* interface of each peer node. Each `hme`*0* interface has a logical interface called `hme`*0:100*, configured with an address in the public IP address space. The nonpeer node can access the cluster network through these logical interfaces.

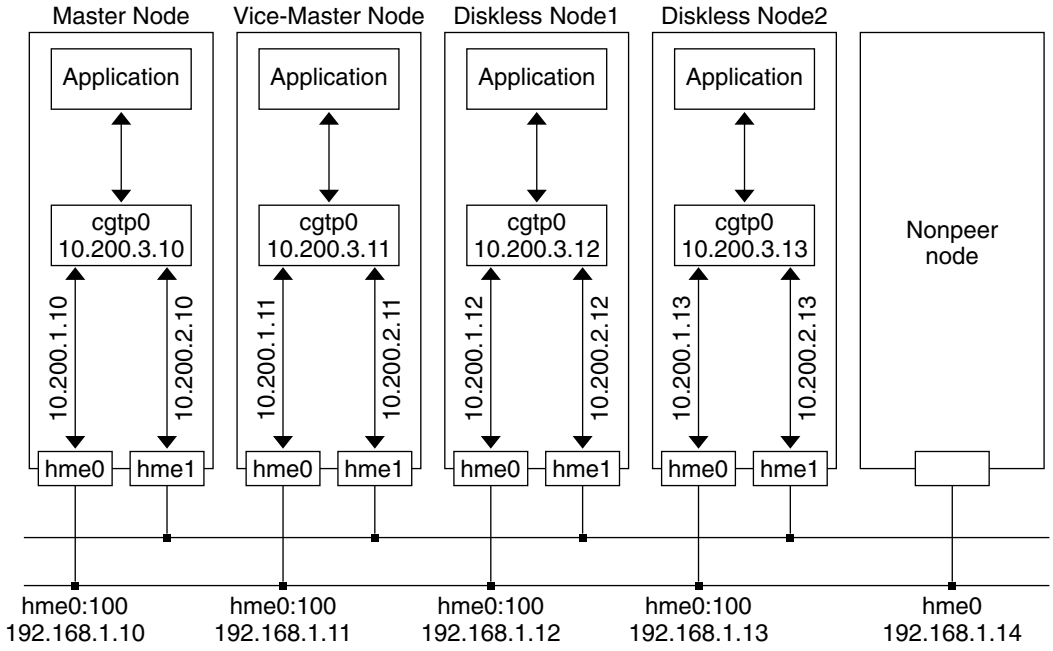


FIGURE 5–1 Example of a Nonpeer Node Connected Directly to a Cluster Network Using a Private IP Address Space

Figure 5–2 shows the same cluster as Figure 5–1. In Figure 5–2 the cluster network uses a public IP address space. The nonpeer node is connected directly to the cluster network without the use of logical interfaces.

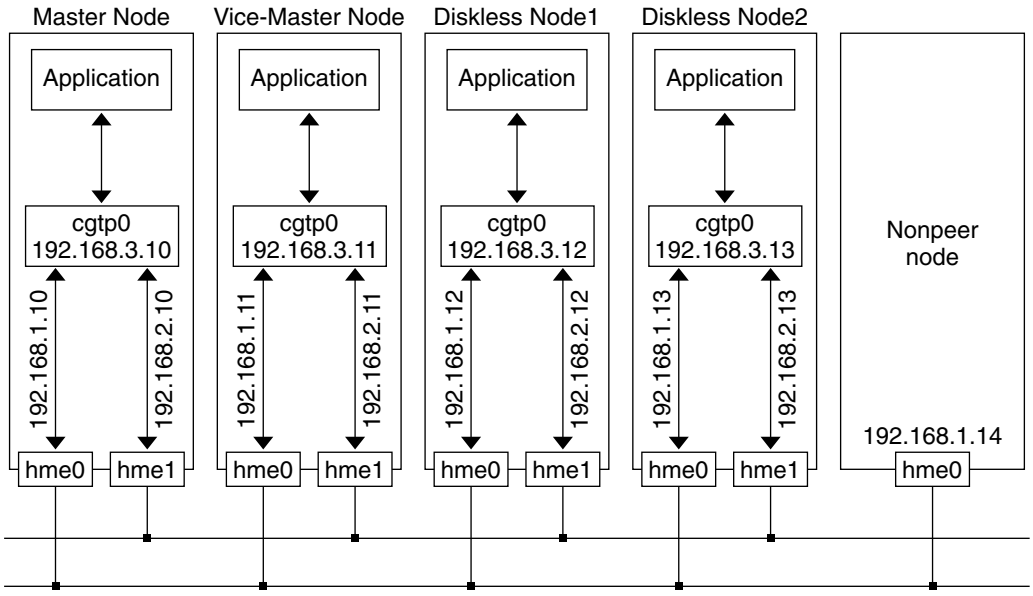


FIGURE 5–2 Example of a Nonpeer Node Connected Directly to a Cluster Network Using a Public IP Address Space

Addressing a Shared Cluster Network and External Network

Table 5–1 shows the IP addresses of the master node in Figure 5–1. In addition to the addresses shown in Figure 5–1, the master node has a floating address for each interface. The Node State Manager configures the floating external address, `hme`*0:101*.

TABLE 5–1 Example IP Addresses for a Master Node With a Logical Interface Configured for External Access

Address Type	Interface	IP Address
Master Node Addresses	`hme`*0*	`10.200.1.10`
	`hme`*1*	`10.200.2.10`
	`cgtp`*0*	`10.200.3.10`
External Address	`hme`*0:100*	`192.168.1.10`

TABLE 5–1 Example IP Addresses for a Master Node With a Logical Interface Configured for External Access *(Continued)*

Address Type	Interface	IP Address
Floating Addresses	`hme`*0:1*	`10.200.1.1`
	`hme`*1:1*	`10.200.2.1`
	`cgtp`*0:1*	`10.200.3.1`
Floating External Address	`hme`*0:101*	`192.168.1.1`

Connecting Nonpeer Nodes to the Cluster Through Additional Physical Interfaces

This section describes how a nonpeer node can be connected to the cluster network through additional physical interfaces on the peer nodes. This configuration is preferable to that in "Connecting Nonpeer Nodes Directly to a Cluster Network" on page 42 for the following reasons:

- The cluster network is separate from the external network, giving better security and performance.
- The cluster network management is simplified.
- There are no restrictions on the external network addressing model.

All of the supported node hardware for the Foundation Services can be configured with more than two physical interfaces.

Figure 5–3 shows an example of how a nonpeer node can be connected to a cluster through the physical interface `hme`*2*.

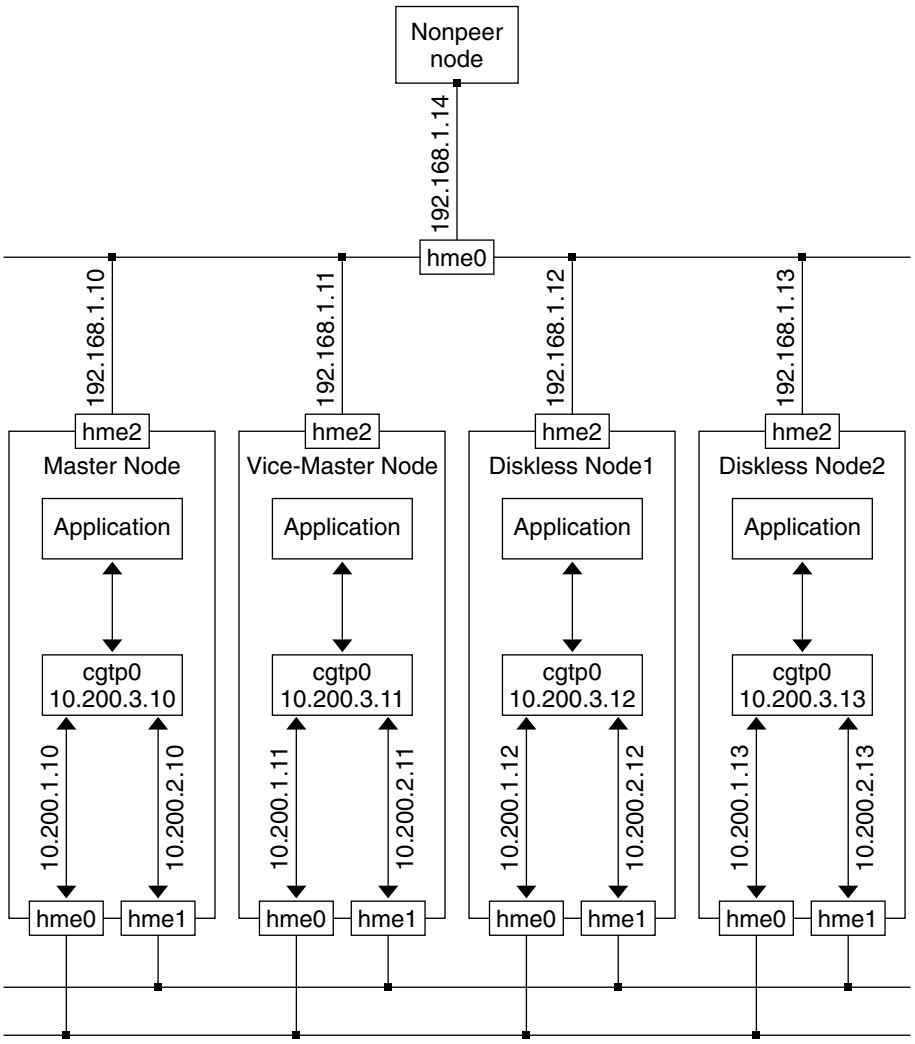


FIGURE 5–3 Example of a Nonpeer Node Connected to the Cluster Network Through Additional Physical Interfaces on Peer Nodes

For simplicity, in Figure 5–3 the nonpeer node is connected to each peer node through a single interface. This configuration would introduce a single point of failure. In highly available platforms, single points of failure must be avoided.

Addressing Physical Interfaces That Are Connected to an External Network

Table 5–2 shows the IP addresses of the master node in Figure 5–3. In addition to the addresses in Figure 5–3, the master node has a floating address for each interface. The Node State Manager configures the floating external address, `hme2:1`.

TABLE 5–2 Example IP Addresses for a Master Node With Three Physical Interfaces

Address Group	Interface	IP Address
Master Node Addresses	`hme0`	`10.200.1.10`
	`hme1`	`10.200.2.10`
	`cgtp0`	`10.200.3.10`
	`hme2`	`192.168.1.10`
Floating Addresses	`hme0:1`	`10.200.1.1`
	`hme1:1`	`10.200.2.1`
	`cgtp0:1`	`10.200.3.1`
Floating External Address	`hme2:1`	`192.168.1.0`

Connecting Nonpeer Nodes to the Cluster Network Through a Router

This section describes how a nonpeer node can be connected to the cluster network through a router. The router node can contain the Network Address Translation (NAT) service to protect the cluster from unwanted external traffic.

The use of a router is advantageous compared to the scenario in "Connecting Nonpeer Nodes Directly to a Cluster Network" on page 42 for the following reasons:

- Internal traffic can be prevented from leaving the cluster network.
- External traffic can be prevented from entering the cluster network.

However, the use of a router is disadvantageous compared to the scenario in "Connecting Nonpeer Nodes to the Cluster Through Additional Physical Interfaces" on page 45 for the following reasons:

- It complicates the network configuration because routers must be configured.
- It could allow external traffic to enter the cluster network, reducing network performance.

- If the external network is connected to one of the cluster networks only, the traffic on the two cluster network paths can become asymmetric.

The following figure shows an example of how several nonpeer nodes can be connected to a cluster network through a router.

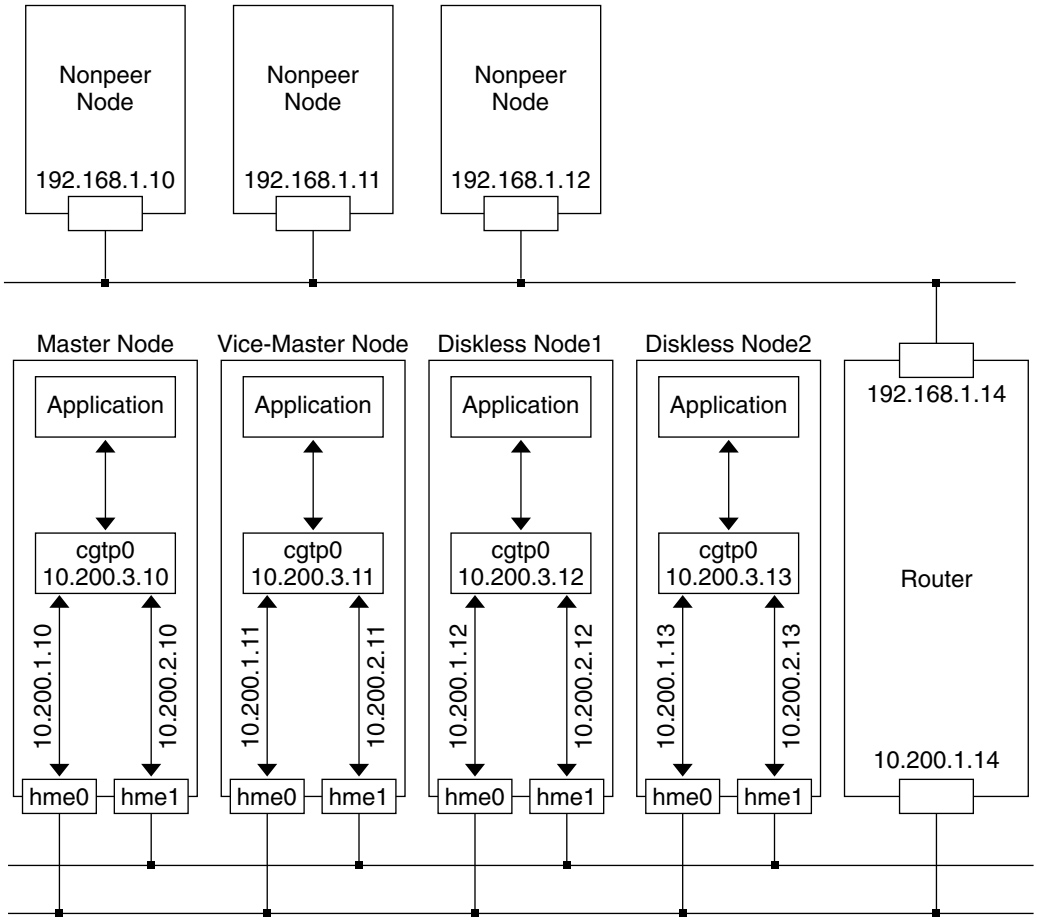


FIGURE 5–4 Example of Nonpeer Nodes Connected to the Cluster Network Through a Router

CHAPTER 6

Carrier Grade Transport Protocol

For information about the reliable IP transport mechanism provided by the Carrier Grade Transport Protocol (CGTP), see the following sections:

Introduction to CGTP

The Foundation Services use a reliable IP transport mechanism provided by CGTP. CGTP is based on transparent multirouting using redundant routes.

In the Foundation Services, each peer node is connected by two high-speed Ethernet networks. When data is sent from a source node to a destination node, it is sent along both Ethernet networks. If data on one network fails to reach the destination node, the data on the other network is still able to reach the destination node. To ensure symmetry in the redundant routes, the Ethernet networks must have equal bandwidth and latency. To prevent single points of failure, the Ethernet networks must not share switching equipment or communication links.

CGTP on peer nodes is configured by the Cluster Membership Manager. CGTP installed on nonpeer nodes is called *standalone CGTP*. Standalone CGTP must be configured manually. For information about installing and configuring CGTP on nonpeer nodes, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Standalone CGTP Guide*.

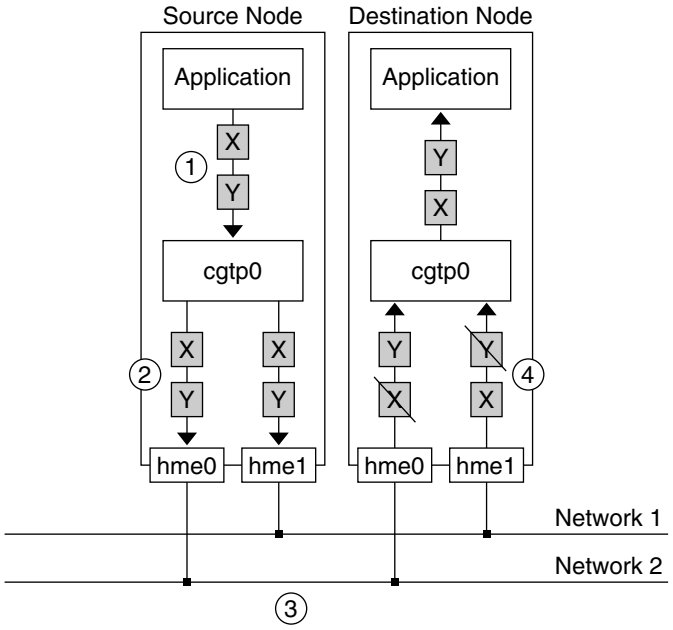
You can configure your cluster without CGTP. When CGTP is not used, each peer node is connected by one Ethernet network only. This configuration introduces a single point of failure. If the Ethernet network fails, there is no backup network. To configure your cluster without CGTP, you must install your cluster either manually or by using the `nhinstall` tool.

Data Transfer Using CGTP

This section describes how CGTP transfers data from a source node to a destination node.

CGTP adds a CGTP source address and CGTP destination address to the header of a data packet on a source node, creating an *IP data packet*. The header contains all the information necessary to uniquely identify an IP data packet.

The Cluster Membership Manager defines routing tables on the source node and destination node. CGTP duplicates the IP data packet, and, using the routing tables, sends one copy of each IP data packet on each of the Ethernet networks. Figure 6–1 illustrates the transfer of data packets from a source node to a destination node, using CGTP.

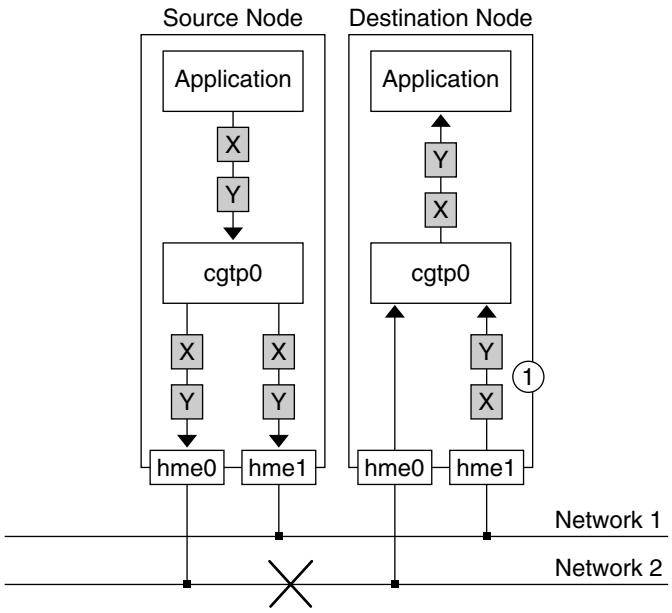


1 Data packets are generated.
2 Data packets are duplicated.
3 Data packets are sent on redundant paths.
4 First packet is transmitted, the second packet is discarded.

FIGURE 6–1 CGTP Transfer of Data Packets From a Source Node to a Destination Node

When the first IP data packet reaches its destination address, CGTP consults the filtering table on the destination node. CGTP verifies that the destination node has not already received the IP data packet. If it has not, CGTP sends the IP data packet to the higher protocols for processing. When the second incoming packet is detected, CGTP identifies it as a duplicate and filters it out. CGTP supports packet filtering on IPv4 and IPv6.

Figure 6–2 illustrates how CGTP is able to deliver the data packets when one of the redundant networks fails.



1 Both data packets are delivered even though Network 2 is broken.

FIGURE 6–2 CGTP Link Failure

If one link fails, the data packet sent on the other network path is still able to reach the destination address. However, until Network 2 is repaired, the system is not highly available.

CHAPTER 7

File Sharing and Data Replication

File sharing and data replication on a cluster are provided by the highly available NFS service, Reliable NFS. This chapter describes how the disks on the master node and vice-master node are partitioned and are mirrored. This chapter refers to the master node disk and the vice-master node disk that contain the shared cluster configuration data. This chapter contains the following sections:

Introduction to Reliable NFS

Reliable NFS provides the following services:

- A reliable file system that gives the vice-master node, the diskless nodes, and the dataless nodes access to data on the master node
- IP mirroring of disk-based data from the master node to the vice-master node
- IP addresses failover of the master role

The Reliable NFS service is controlled by the `nhcrfsd` daemon. The `nhcrfsd` daemon runs on the master node and vice-master node. It controls the failover or switchover from the master node to the vice-master node. If the master node fails, the vice-master node becomes master and the NFS server on the new master node becomes active.

The `nhcrfsd` daemon responds to changes in the cluster state as it receives notifications from the Cluster Membership Manager. For further information about the Cluster Membership Manager, see Chapter 8. The Reliable NFS daemon is monitored by the Daemon Monitor, `nhpmd`. For further information about the Daemon Monitor, see Chapter 10.

If the impact on performance is acceptable, do not use data and attribute caches when writing to shared file systems. If it is necessary to use data caches to improve performance, ensure that your applications minimize the risk of using inconsistent data. For guidelines on how to use data and attribute caches when writing to shared file systems, see "Using Data Caches in Shared File Systems" in the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.

For reference information about network tunable parameters and the Solaris kernel, see the *Solaris Tunable Parameters Reference Manual* for your version of the Solaris operating system.

Volume Management

This section describes how the master node disk and vice-master node disk are partitioned.

The master node, vice-master node, and dataless nodes access their local disks. The vice-master node and dataless nodes also access some disk partitions of the master node. Diskless nodes do not have, or are not configured to use, local disks. Diskless nodes rely entirely on the master node to boot and access services and data.

You can partition your disks as described in "Standard Disk Partitioning" on page 54, or as described in "Virtual Disk Partitioning" on page 55.

Standard Disk Partitioning

To use standard disk partitioning, you must specify your disk partitions in the cluster configuration files. During installation, the `nhinstall` tool partitions the disks according to the specifications in the cluster configuration files. If you manually install the Foundation Services, you must partition the system disk and create the required file systems manually.

The master node disk and vice-master node disk can be split identically into a maximum of eight partitions. For a cluster containing diskless nodes, the partitions can be arranged as follows:

- Three partitions for the system configuration

- Two partitions for data
- Two partitions for scoreboard bitmaps
- One free partition

Partitions that contain data are called *data partitions*. One data partition might typically contain the exported file system for diskless nodes. The other data partition might contain configuration and status files for the Foundation Services. Data partitions are mirrored from the master node to the vice-master node.

To be mirrored, a data partition must have a corresponding *scoreboard bitmap partition*. If a data partition does not have a corresponding scoreboard bitmap partition, it cannot be mirrored. For information about the scoreboard bitmap, see "IP Mirroring" on page 57.

Table 7–1 shows an example disk partition for a cluster containing master-eligible nodes and diskless nodes. This example indicates which partitions are mirrored.

TABLE 7–1 Example Disk Partition for a Cluster of Master-Eligible Nodes and Diskless Nodes

Partition	Use	Mirrored
`s0`	Solaris boot	Not mirrored
`s1`	Swap	Not mirrored
`s2`	Whole disk	Not applicable
`s3`	Data partition for diskless Solaris images	Mirrored read/write for the diskless nodes
`s4`	Data partition for middleware data and binaries	Mirrored read/write for applications
`s5`	Scoreboard bitmap partition	Used to mirror partition `s3`
`s6`	Scoreboard bitmap partition	Used to mirror partition `s4`
`s7`	Free	

Master-eligible nodes in a cluster that does not contain diskless nodes do not require partitions `s3` and `s5`.

Virtual Disk Partitioning

Virtual disk partitioning is provided by Solstice DiskSuite 4.2.1 for Solaris 8, and is integrated into Solaris 9 in the Solaris Volume Manager software.

One of the partitions of a physical disk can be configured as a *virtual disk* by using Solstice DiskSuite or Solaris Volume Manager. A virtual disk can be partitioned into a maximum of 128 *soft partitions*. To an application, a virtual disk is functionally identical to a physical disk. The following figure shows one partition of a physical disk configured as a virtual disk with soft partitions.

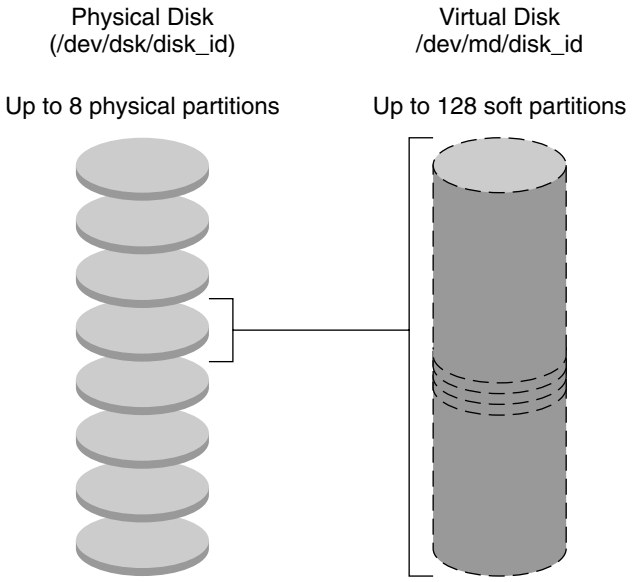


FIGURE 7–1 One Partition of a Physical Disk Configured as a Virtual Disk

In Solaris Volume Manager, a virtual disk is called a *volume*. In Solstice DiskSuite, a virtual disk is called a *metadevice*.

To use virtual disk partitioning, you must install and configure the Solaris operating system and virtual disk partitioning manually on your cluster. You can then configure the `nhinstall` tool to install the Foundation Services only, or you can install the Foundation Services manually.

For more information about virtual disk partitioning, see the Solaris documentation.

Logical Mirroring

Logical mirroring is provided by Solstice DiskSuite 4.2.1 for Solaris 8, and by Solaris Volume Manager for Solaris 9.

Logical mirroring can be used on master-eligible nodes with two or more disks. The disks are mirrored locally on the master-eligible nodes. They always contain identical information. If a disk on the master node is replaced or crashes, the second local disk takes over without a failover.

For more information about logical mirroring, see the Solaris documentation.

IP Mirroring

This section describes how data is replicated from the master node to the vice-master node, and how these nodes are resynchronized after failover or switchover.

Data Partitions and Scoreboard Bitmaps

When data is written to a replicated partition on the master node disk, the corresponding scoreboard bitmap is updated.

The scoreboard bitmap maps one bit to a block of data on a replicated partition. When a block of data is changed, the corresponding bit in the scoreboard bitmap is set to 1. When the data has been replicated to the vice-master node, the corresponding bit in the scoreboard bitmap is set to zero.

The scoreboard bitmap can reside on a partition on the master node disk or in memory. There are advantages and disadvantages to storing the scoreboard bitmap on the master node disk or in memory:

- Storing the scoreboard bitmap in memory means better performance because writing to memory is quicker than writing to disk. Scoreboard bitmaps in memory are copied to disk when the master node is shut down gracefully. Each replicated partition on a disk must have a corresponding partition for a scoreboard bitmap, even if the scoreboard bitmap is stored in memory.
- Storing the scoreboard bitmap in memory is a problem if the master node and vice-master node fail simultaneously. In this case, the scoreboard bitmap is lost and a full resynchronization is required when the nodes are rebooted.
- Storing the scoreboard bitmap on a disk partition is slower during normal operation because writing to disk is slower than writing to memory. However, if the master node and vice-master node fail simultaneously, the scoreboard bitmap can be used to resynchronize the nodes, without the need for a full resynchronization.

For information about how to configure the scoreboard bitmap in memory or on disk, see “Changing the Location of the Scoreboard Bitmap” in the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.

Replication During Normal Operation

Replication is the act of copying data from the master node to the vice-master node. Through replication, the vice-master node has an up-to-date copy of the data on the master node. Replication enables the vice-master node to take over the master role at any time, transparently. After replication, the master node disk and vice-master node disk are *synchronized*, that is, the mirrored partitions contain exactly the same data.

Replication occurs at the following times:

- When the master node and vice-master node are running, and data on the master node disk is changed
- After startup, to replicate the software installed on the replicated partitions, such as the diskless Solaris image
- After a failover or switchover

The following figure illustrates a diskless node writing data to the master node, and that data being replicated to the vice-master node.

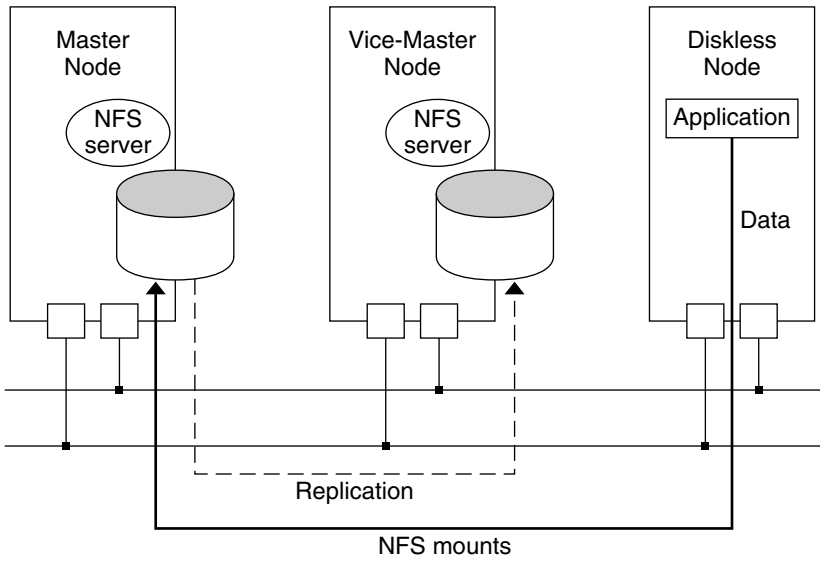


FIGURE 7–2 Data Replication

Replication During Failover and Switchover

During failover or switchover, the master node goes out of service for a time before being re-established as the vice-master node. During this time, changes that are made to the new master node disk cannot be replicated to the vice-master node. Consequently, the cluster becomes unsynchronized.

While the vice-master node is out of service, data continues to be updated on the master node disk, and the modified data blocks are identified in the scoreboard bitmap. Figure 7–3 illustrates Reliable NFS during failover or switchover.

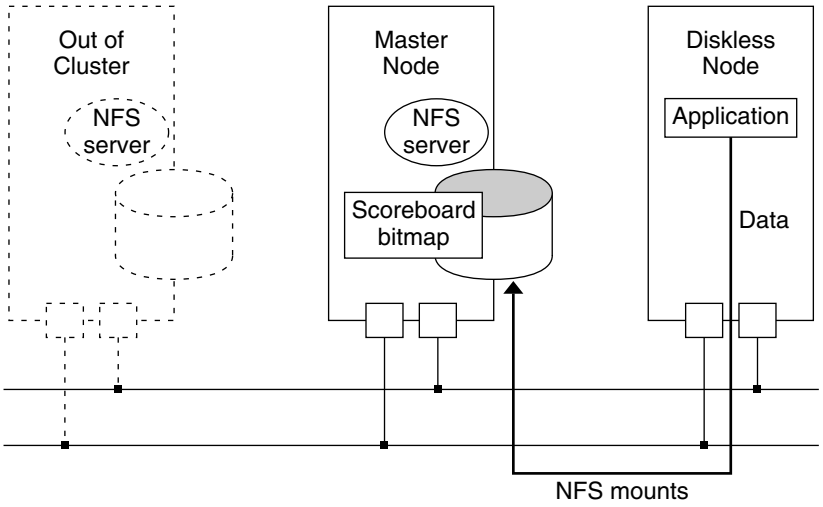


FIGURE 7–3 Reliable NFS During Failover or Switchover

When the vice-master node is re-established, replication resumes. Any data written to the master node is replicated to the vice-master node. In addition, the scoreboard bitmap is examined to determine which data blocks have been changed while the vice-master node was out of service. Any changed data blocks are also replicated to the vice-master node. In this way, the cluster becomes synchronized again. The following figure illustrates the restoration of the synchronized state.

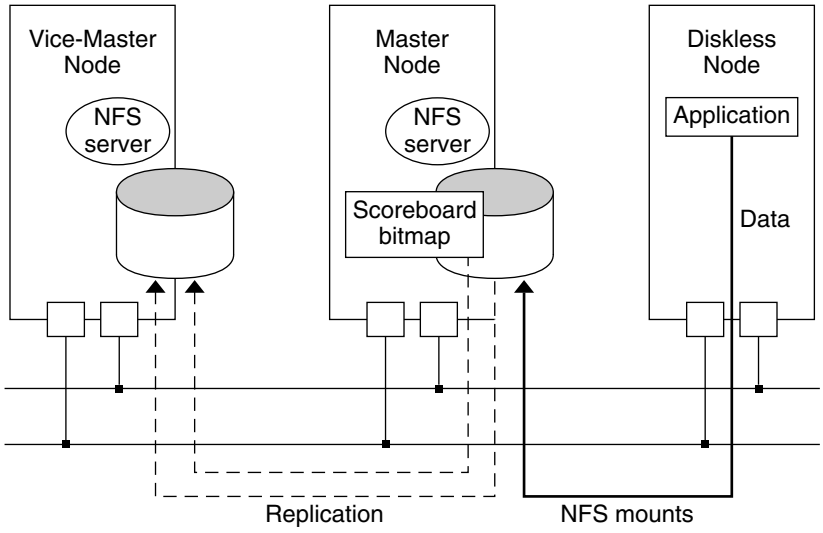


FIGURE 7–4 Restoration of the Synchronized State

While a cluster is unsynchronized, the data on the master node disk is not fully backed up. Do not schedule major tasks when a cluster is unsynchronized.

You can verify whether a cluster is synchronized, as described in "To Verify That the Master Node and Vice-Master Node Are Synchronized" in the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.

You can collect replication statistics by using the Node Management Agent as described in the *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*.

Master Node IP Address Failover

For a failover to be transparent to a diskless node or dataless node, the following must be true:

- The diskless node or dataless node must be mounted onto the floating address of the master node.
- The floating address must always be configured and active on the node holding the master role, even after failover or switchover.

For further information about the floating address of the master node, see "Floating Address Triplet" on page 38.

CHAPTER 8

Cluster Membership Manager

For information about how the cluster membership is managed and configured, and how the presence of peer nodes is monitored, see the following sections:

Introduction to the Cluster Membership Manager

The Cluster Membership Manager (CMM) is implemented by the `nhcmmd` daemon. There is a `nhcmmd` daemon on each peer node.

The `nhcmmd` daemon on the master node has the current view of the cluster configuration. It communicates its view to the `nhcmmd` daemons on the other peer nodes. The `nhcmmd` daemon on the master node determines which nodes are members of the cluster, and assigns roles and attributes to the nodes. It detects the failure of nodes and configures routes for reliable transport.

The `nhcmmd` daemon on the vice-master node monitors the status of the master node. If the master node fails, the vice-master node is able to take over as the master node.

The `nhcmmd` daemons on the master-ineligible nodes do not communicate with one another. Each `nhcmmd` daemon exports an API to do the following:

- Notify clients of changes to the cluster
- Notify services and applications when the cluster membership or master changes

Notification messages describe the change and the *nodeid* of the affected node. Clients can use notifications to maintain an accurate view of the peer nodes in the cluster.

For further information about the nhcmmd daemon, see the nhcmmd(1M) man page.

You can use the CMM API to write applications that manage peer nodes or that register clients to receive notifications. For further information about writing applications that use the CMM API, see the *Netra High Availability Suite Foundation Services 2.1 6/03 CMM Programming Guide*.

Configuring the Cluster Membership

Cluster membership information is stored in the configuration files, cluster_nodes_table and nhfs.conf.

At cluster startup, the cluster membership is configured as follows:

1. Both of the master-eligible nodes retrieve the list of peer nodes and their attributes from the cluster_nodes_table, and configuration information from nhfs.conf. All other peer node retrieve configuration information from nhfs.conf.
2. The nhcmmd daemon on the master node uses the list of nodes and their attributes to generate its view of the cluster configuration. It communicates this view to the nhcmmd daemons on the other peer nodes, including the vice-master node.
3. Using the master node view of the cluster, the nhcmmd daemon on the vice-master node updates its local cluster_nodes_table.

The nhcmmd daemon on the master node updates its cluster_nodes_table and its view of the cluster configuration when a peer node is added, removed, or disqualified. The nhcmmd daemon on the master node communicates the updated view to the nhcmmd daemons on the other peer nodes. The vice-master node uses this view to update its local cluster_nodes_table. In this way, the master node and vice-master node always have an up-to-date view of the cluster.

Monitoring the Presence of Peer Nodes

Each peer node runs a daemon called nhprobed that periodically sends a heartbeat in the form of an IP packet. Heartbeats are sent through each of the two physical interfaces of each peer node. When a heartbeat is detected through a physical interface, it indicates that the node is reachable and that the physical interface is alive. If a heartbeat is not detected for a period of time exceeding the detection delay, the physical interface is considered to have failed. If both of the node's physical interfaces fail, the node itself is considered to have failed. The detection delay is 900 milliseconds. At least one heartbeat must be detected each 900 milliseconds.

For more information about the nhprobed daemon, see the nhprobed(1M) man page.

Interaction Between the nhprobed Daemon and the nhcmmd Daemon

On the master-eligible nodes, the nhprobed daemon receives a list of nodes from the nhcmmd daemon. The nhprobed daemon monitors the heartbeats of the nodes on the list. On the master node, the list contains all of the master-ineligible nodes and the vice-master node. On the vice-master node, the list contains the master node only.

On the master-eligible nodes, the nhprobed daemon notifies the nhcmmd daemon when, for any node on its list, any of the following events occur:

- One link becomes available, indicating that the node is accessible through the link.
- One link becomes unavailable, indicating that the node is not accessible through the link.
- The node becomes available, indicating that the first link to the node becomes available.
- The node becomes unavailable, indicating that the last available link to the node becomes unavailable.

When a node other than the master node becomes unavailable, the master node eliminates the node from the cluster. The master node uses the TCP abort facility to close communication to the node. When the master node becomes unavailable, a failover is provoked.

Using the Direct Link to Prevent Split Brain Errors

Split brain is an error scenario in which the cluster has two master nodes. A direct communication link between the master-eligible nodes prevents the occurrence of split brain when the communication between the master node and vice-master node fails.

As described in "Monitoring the Presence of Peer Nodes" on page 62, the nhprobed daemon on the vice-master node monitors the presence of the master node. If the nhprobed daemon on the vice-master node fails to detect the master node, the master node itself or the communication to the master node has failed. If this happens, the vice-master node uses the direct link to try to contact the master node.

- If the vice-master node does not receive a reply from the master node by using the direct link, it is assumed that the master node has failed. The vice-master node becomes the master node.
- If the vice-master node receives a reply from the master node by using the direct link, it is assumed that the communication to the master node has failed but the master node is alive. The vice-master node is rebooted.

The Node Management Agent can monitor the following statistics on the direct link:

- The number of times that the vice-master node has requested to become the master node.
- The state of the direct communication link. The state can be *up* or *down*.

For information about how to connect the direct link between the master-eligible nodes, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

Multicast Transmission of Heartbeats

Probe heartbeats are multicast. Each cluster on a local area network (LAN) is assigned to a different multicast group, and each network interface card (NIC) on a node is assigned to a different multicast group. For example, NICs connected to an `hme0` Ethernet network are assigned to one multicast group, and NICs connected to an `hme1` Ethernet network are assigned to another multicast group.

A heartbeat sent from one multicast group cannot be detected by another multicast group. Therefore, heartbeats sent from one cluster cannot be detected by another cluster on the same LAN. Similarly, for a cross-switched topology, heartbeats sent from one Ethernet network cannot be detected on another Ethernet network.

Multicast addresses are 32–bit. The lower 28 bits of the multicast address represent the multicast group. The multicast address is broken into the following parts:

- Bits 28 to 31 are fixed.
- Bits 23 to 27 identify the Foundation Services. For the Foundation Services bits 23 to 27 are always set to `10100`.
- Bits 8 to 22 identify the cluster. The value for a given cluster is specified in the `nhfs.conf` file by the `Node.DomainId` parameter.
- Bits 0 to 7 identify the NIC. The value for a given NIC is specified in the `nhfs.conf` file by the `Node.NIC0` and `Node.NIC1` parameters.

When you are defining multicast groups for applications, follow these recommendations:

- Bits 23 to 27 of the multicast address must not have the value `10100`. This value is reserved for the Foundation Services.
- Bits 0 to 22 of the multicast address should not have the same value as any of your Foundation Services clusters.

  When a message is sent, the IP stack uses the lower 23 bits of the multicast address to define the destination MAC address. If several multicast addresses have the same value for the lower 23 bits, even if they have different values for the upper five bits, the addresses must be filtered at the IP level. The IP filtering would create a corresponding reduction in performance.
- Because of the way that `hme` and `le` interfaces filter multicast packets, one in four clusters share the same multicast filter. To reduce the need to filter at the IP level, clusters in the same LAN should have sequential cluster identities.

CHAPTER 9

Reliable Boot Service

The Reliable Boot Service uses the Solaris Dynamic Host Configuration Protocol (DHCP) service and the other Foundation Services to ensure the boot of diskless nodes regardless of software or hardware failure. For information about how diskless nodes are booted and how they are allocated IP addresses, see the following sections:

Introduction to the Reliable Boot Service

In a Foundation Services cluster, diskless nodes rely on network services to boot their operating system and run their software. The Reliable Boot Service provides a standard Solaris DHCP server on each of the master-eligible nodes. The Reliable Boot Service keeps the DHCP service operational even after failover or switchover.

The Solaris DHCP servers use a public DHCP module that is configured by the DHCP administration utilities `dhcpconfig`, `pntadm`, and `dhtadm`. For information about these utilities, see their man pages.

By default, the DHCP configuration files are stored on the master node. They are mounted on a mirrored file system and replicated on the vice-master node by Reliable NFS. For information about Reliable NFS, see Chapter 7. You can also put the DHCP configuration files locally on the master node and copy them to the vice-master node. For information on putting DHCP configuration files locally, see the `nhfs.conf`(4) man page.

The DHCP service provides several methods of allocating IP addresses to diskless nodes at boot.

The DHCP daemon is started under the control of the Daemon Monitor. If the Reliable Boot Service fails, the Daemon Monitor takes the recovery action described in the `nhpmd`(1M) man page.

For further information about DHCP in the Solaris operating system, see the *Solaris DHCP Service Developer's Guide*.

Booting Diskless Nodes

Figure 9–1 shows the Reliable Boot Service on the master node and vice-master node, and the request for boot from a diskless node. The diskless node broadcasts a *DHCP discover request* to the DHCP servers on the master-eligible nodes. Only the master node responds to the DHCP discover request. After failover or switchover, the boot server on the new master node responds to the DHCP requests from diskless nodes.

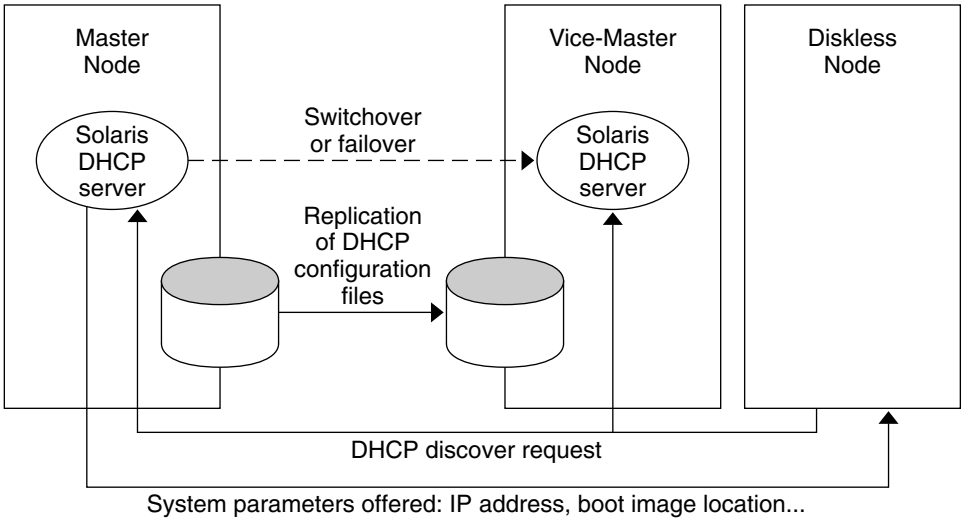


FIGURE 9–1 Request for Boot Broadcast From a Diskless Node

The Reliable Boot Service is notified by the Cluster Membership Manager when a new master node is elected, or when a node joins or leaves the cluster.

When a diskless node boots, the Reliable Boot Service assigns an IP address to it. If the node does not boot successfully within a specific time period, the Reliable Boot Service frees the allocated resources for the node. When a node leaves the cluster, the Reliable Boot Service retrieves the IP address that was being used by the node and clears the associated entry in the DHCP configuration files.

Boot Policies for Diskless Nodes

The method of booting diskless nodes at cluster startup is called the *boot policy*. There are advantages and disadvantages to each of the boot policies. Your choice depends on the hardware configuration of your cluster. For information about how to configure a boot policy the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*. This section describes the three boot policies used by the Foundation Services.

DHCP Dynamic

This boot policy creates a dynamic map between the Ethernet address of a diskless node and an IP address taken from a pool of available addresses. The map is stored in the data files for the DHCP module.

The map remains valid as long as the node remains in the cluster. When the node fails or leaves the cluster, the DHCP module deletes the Ethernet address to IP address mapping. This enables the IP address to be re-enter the pool and be used by another diskless node.

When a diskless node is rebooted, the DHCP module assigns a new IP address to the node. Previously allocated IP addresses are stored in a cache. The DHCP module attempts to reallocate a rebooted node with the same IP address as it had before.

DHCP Static

This boot policy maps the Ethernet address of a diskless node to a fixed IP address.

This system has the advantage of statically dedicating IP addresses to specific machines, making it possible to attribute groups of software to specific machines. However, this system does not support hardware hot-swap. Furthermore, when a node fails, the Ethernet address to IP address mapping remains assigned and cannot be reused.

DHCP Client ID

This boot policy associates a `CLIENT_ID` string with a diskless node. When the diskless node is replaced, the `CLIENT_ID` string must be associated with the new node.

CHAPTER 10

Daemon Monitor

This chapter describes how the Daemon Monitor is used to survey other process daemons. It describes how the Daemon Monitor can be monitored and its recovery response changed and reset.

This chapter includes the following sections:

The nhpmd Daemon

The nhpmd daemon provides the Daemon Monitor service. The nhpmd daemon runs at the multiuser level on all nodes in the cluster. The nhpmd daemon surveys other Foundation Services daemons, many Solaris operating system daemons, and some companion product daemons. If a daemon that provides a critical service fails, the nhpmd daemon detects the failure and triggers a recovery response. The recovery response is specific to the daemon that has failed. For a list of monitored daemons and their recovery responses, see the nhpmd(1M) man page.

The nhpmd daemon operates at a higher priority than the other Foundation Services daemons.

The Daemon Monitor is surveyed by a kernel module. When the kernel module detects an abnormal exit of the Daemon Monitor, it implements a panic that results in the crash and reboot of the node.

Foundation Services daemons and Solaris operating system daemons are launched by *startup scripts*. A *nametag* is assigned to the daemon or group of daemons that is launched by each startup script. In some cases, such as for syslogd, a nametag is assigned to only one daemon. In other cases, such as for nfs_client, a nametag is

assigned to a group of daemons. If one of the daemons covered by a nametag fails, the recovery response is performed on all of the daemons covered by that nametag. If the recovery response is to restart the failed daemon, all of the daemons grouped under that nametag are killed and then restarted.

Information about monitored daemons can be collected using the `nhpmdadm` command, as described in the `nhpmdadm`(1M) man page.

Information about the actions taken by the `nhpmd` daemon can be gathered from the system log files. For information on how to configure the system log files, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*.

Using the Node Management Agent With the Daemon Monitor

The Node Management Agent (NMA) can be used to collect the following information from a Daemon Monitor:

- Which daemons are monitored
- Which monitored processes have failed
- The number of times a failed daemon has been restarted
- The maximum number of times a failed daemon is allowed to be restarted

The NMA can be used to change the following parameters of the Daemon Monitor:

- The maximum number of times that the Daemon Monitor attempts to restart a daemon or group of daemons
- The reset of the current retry count for a monitored daemon

For information about the NMA, see Chapter 11 and also the *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*.

CHAPTER 11

Node Management Agent

This chapter describes how the Node Management Agent (NMA) can be used to monitor and manipulate a cluster. This chapter contains the following sections:

Introduction to the Node Management Agent

The NMA is compliant with the Java Management Extensions (JMX) and based on the Java Dynamic Management Kit. The NMA provides access to cluster statistics through the Simple Network Management Protocol (SNMP) or through JMX clients using HTTP. The NMA supports the Internet Engineering Task Force standard RFC 2573.

The NMA retrieves statistics about the cluster membership, the reliable transport mechanism, the network file system, and the monitoring of process daemons. The NMA can be used to initiate a switchover, change the maximum number of times that it attempts to restart a daemon, reset the current retry count for a daemon, and listen for certain cluster notifications.

The following figure shows a remote client accessing nodes in a cluster.

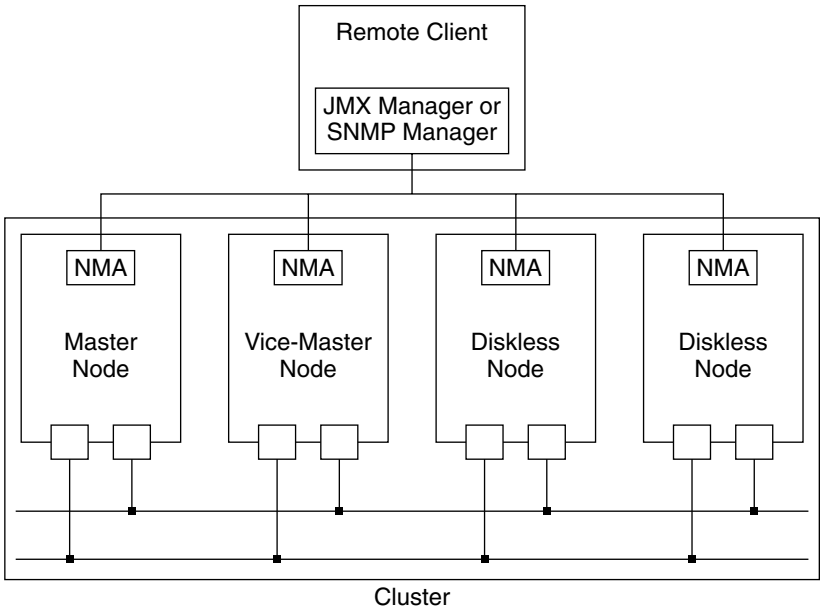


FIGURE 11–1 Remote Access to the Cluster

Monitoring Statistics With the NMA

The NMA collects two types of statistics: *node-level statistics* and *cluster-level statistics*.

There is an NMA on each peer node that collects node-level statistics, that is, statistics for that node. Each NMA collects statistics about CGTP, CMM, and the Daemon Monitor. The NMA on each master-eligible node collects node-level statistics about Reliable NFS.

The NMA on the master node collects cluster-level statistics, that is, statistics about the cluster.

The following table describes the statistics that are collected by the NMA on each peer node and on the master node.

TABLE 11–1 Statistics Collected by the NMA

Type of Statistics	Collected From Which Nodes	Statistics
Node-level statistics	All peer nodes, including the master node	CGTP statistics, CMM statistics, and Daemon Monitor statistics
		Reliable NFS statistics on the master-eligible nodes
Cluster-level statistics	Master node only	CMM statistics, list of peer nodes, cluster Reliable NFS statistics, and high-level statistics for each peer node

The following statistics can be collected by the NMA:

- CMM statistics

  Some cluster membership statistics are collected on each node. Other cluster membership statistics can be collected by the NMA on the master node only. Statistics collected from the master node include the following:

  – Information about mastership elections
  – Information about switchovers
  – Information about the direct link
  – Details of the individual cluster nodes
  – Activity of the `nhcmmd` and `nhprobed` daemons

- CGTP statistics

  These statistics include a set of general CGTP statistics and a set of dedicated packet filtering statistics. Packet filtering statistics count the number of packets successfully received through each of the CGTP redundant links. In this way, packet filtering statistics measure the quality of the communication.

- Reliable NFS statistics

  These statistics give the file status and disk replication status. These statistics can be collected for master-eligible nodes only.

- Daemon Monitor statistics

  These statistics provide the following information:

  – Which daemons are monitored
  – Which monitored processes have failed
  – The number of times a failed daemon has been restarted
  – The maximum number of times a failed daemon is allowed to be restarted

The NMA running on the master node *cascades* the statistics from the NMA on each of the peer nodes into its namespace. By cascading, the NMA on the master node can see the statistics on all of the peer nodes. In this way, the NMA on the master node has a view of the entire cluster. The following figure illustrates the cascade of statistics from the peer nodes to the master node.

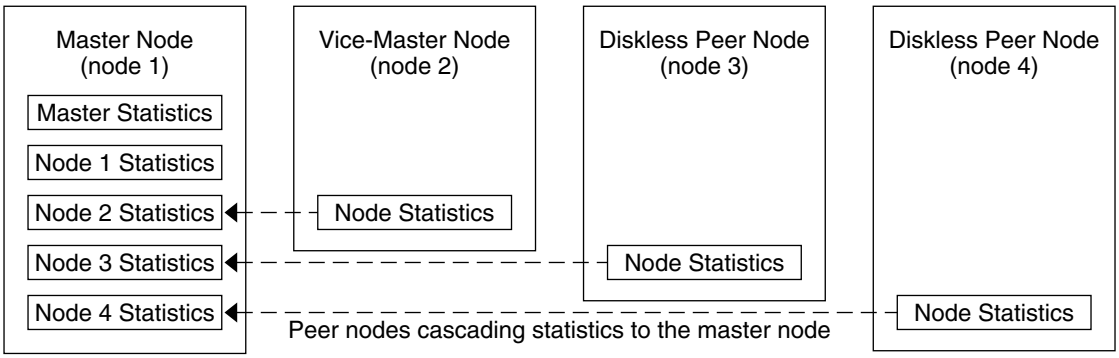


FIGURE 11–2 Cascading Data From Peer Nodes to the Master Node

A set of NMA APIs can be used to develop applications that monitor and react to the statistics produced by a cluster. For information about developing applications with the NMA APIs, see the *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*.

The statistics that a cluster generates depend on the type, size, and arrangement of the cluster. Each cluster has an individual set of statistics. Knowing the statistics that your cluster generates when it runs correctly can help you to interpret the statistics when the cluster is failing. Use the NMA and its APIs to establish a set of statistics or a benchmark for your cluster when it is working correctly.

Manipulating the Cluster With the NMA

The NMA can be configured to initiate a switchover or to change the following Daemon Monitor parameters:

- The maximum number of times that the Daemon Monitor attempts to restart a daemon or group of daemons
- The current retry count for a monitored daemon

For more information, see the *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*.

Receiving Notifications With the NMA

The NMA can be used to "listen" for notifications of the following cluster events:

- A node joins or leaves the cluster
- A master node or vice-master node is elected
- The maximum number of times that the Daemon Monitor attempts to restart a daemon or group of daemons is reached
- The maximum number of times that the Daemon Monitor attempts to restart a daemon or group of daemons is changed by the NMA
- The current number of times that the Daemon Monitor has attempted to restart a daemon or group of daemons is reset to zero by the NMA
- A nametag is created by the Daemon Monitor
- A nametag is removed by the Daemon Monitor

For more information, see the *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*.

CHAPTER 12

Watchdog Timer

For information about how the Watchdog Timer guards against operating system hang, see the following sections:

The nhwdt Daemon

Low-level system monitoring in the Foundation Services is provided by the Watchdog Timer service. This service is implemented by the `nhwdtd` daemon.

The Watchdog Timer service monitors the hardware watchdog on the nodes of the cluster, at the lights-off management (LOM) level. The `nhwdtd` daemon monitors a node for operating system hang, but does not monitor the boot process.

Hardware watchdogs that operate at the OpenBoot™ PROM (OBP) level are monitored by the platform's software, not by the `nhwdtd` daemon.

For information about which hardware is monitored at the OBP level and the LOM level, see "Types of Hardware Watchdogs" in the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

The Watchdog Timer can be configured differently on each peer node depending on your requirements. For information about how to configure the Watchdog Timer, see the `nhfs.conf`(4) man page.

Monitoring by the Daemon Monitor

The Watchdog Timer is monitored by the Daemon Monitor. If the `nhwdtd` daemon fails, it is relaunched three times. If the `nhwdtd` daemon fails a third time, the node is rebooted.

The `nhwdtd` daemon operates at a lower priority than the `nhpmd` daemon, but at a higher priority than the other Foundation Services daemons.

Index

D

E

F

H

I

J

L

M

N

O

P

R

S

T

V

W