Configuring OPS Clusters
with MC/LockManager

HEWLETT PACKARD

Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document edition.

Edition	Date
First	March 1995
Second	February 1996

This version of MC/LockManager operates under HP-UX 10.10. The product number for MC/LockManager media and manuals is B5158AA. Use the HP-UX `what` command together with the complete pathname of a particular product component to obtain a version string for that component.

List of Related HP and Oracle Documents. The following documents contain additional related information:

- *Oracle7 Parallel Server Administrator's Guide* (Oracle)
- *Oracle Parallel Server for HP 9000 Series 800 Installation and Configuration Guide* (Oracle)
- *Addendum to the OPS ICG* (Oracle)
- *Oracle7 Server for UNIX Administrator's Reference Guide* (Oracle)
- *Oracle Tools for UNIX Administrators Reference Guide* (Oracle)
- *System Administration Tasks* manual for HP-UX 10.0 Series 800 (HP)
- *HP 9000 Series 800 Business Servers Configuration Guide* (available through your HP representative)

Preface

MC/LockManager software enables the Oracle Parallel Server RDBMS to run
on HP 9000 high availability clusters using the HP-UX operating system.
High availability clusters configured with Oracle Parallel Server are known as
OPS clusters. This guide describes how to use MC/LockManager to install,
configure and maintain OPS clusters. The following chapters are included:

- Chapter 1, "Introducing MC/LockManager," describes the hardware and
  software elements used by MC/LockManager and provides a general view of
  how they work with Oracle Parallel Server software.
- Chapter 2, "Planning and Documenting an OPS Cluster," gives a set of
  planning worksheets to assist in identifying and documenting the exact
  components that must be configured.
- Chapter 3, "Building an OPS Cluster Configuration," presents detailed steps
  for configuration using SAM options and/or HP-UX system commands and
  scripts.
- Chapter 4, "Configuring Packages and their Services," shows how to build
  package configurations for use in a OPS cluster.
- Chapter 5, "Maintaining an OPS Cluster," identifies a set of common
  maintenance tasks and shows how to carry them out using SAM and/or
  HP-UX system commands.
- Chapter 6, "Troubleshooting Your Cluster," describes some ways of assessing
  the state of cluster elements when problems arise.
- Appendix A, "Moving from HP-UX 9.04 to HP-UX 10.10," gives instructions
  for migrating an OPS cluster to the HP-UX 10.10 operating system.
- Appendix B, "Sample Planning Worksheets," contains a set of blank
  worksheets for preparing an OPS configuration on HP-UX.
- Appendix C, "Man Pages for MC/LockManager Configuration," lists the
  man pages relating to OPS on HP-UX that are available on your system.
- Appendix D, "Designing Highly Available Cluster Applications," describes
  issues specific to applications running on highly available clusters.
- Appendix E, "Distributed Lock Manager (DLM) Error Messages," lists DLM
  error messages with cause and action text.

Since MC/LockManager is a very complex product to configure and maintain,
it is strongly recommended that you use Hewlett-Packard high availability
consulting services to ensure a smooth installation and rollout. Please contact
your HP representative to inquire about high availability consulting.

Contents

Contents-4

Index

Figures

Tables

1

Introducing MC/LockManager

MC/LockManager enables the Oracle Parallel Server RDBMS to run on HP 9000 high availability clusters under the HP-UX operating system.

This chapter introduces OPS on HP-UX and gives a broad overview of how the MC/LockManager components work. The following topics are presented:

- Using this Guide
- What is an OPS Cluster?
- Basic Concepts and Components
- How the Cluster Manager Works
- How the Distributed Lock Manager Works
- How the Package Manager Works
- How the Network Manager Works
- Responses to Failures

If you are ready to start setting up OPS clusters, skip ahead to the "Planning" chapter. Specific steps for setup are given in the chapter "Building an OPS Cluster Configuration."

Specific information about Oracle software installation is provided in the *Oracle Parallel Server for HP 9000 Series 800 Installation and Configuration Guide*, which you should read before using the Oracle *installer* program.

Using this Guide

This manual presents the tasks you need to perform in order to create a functioning OPS cluster. These tasks are shown in Figure 1-1.
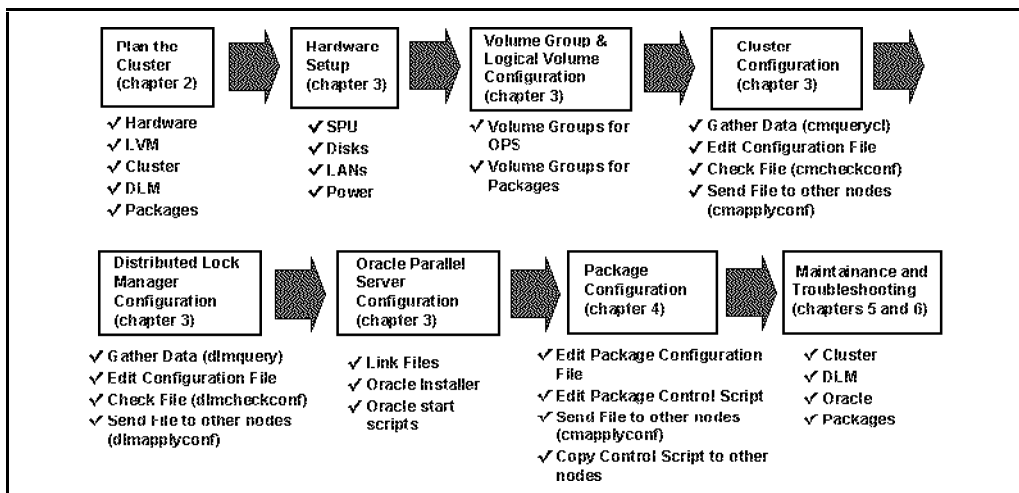
Figure 1-1. Tasks in Configuring an OPS Cluster

The tasks in Figure 1-1 are covered in step-by-step detail in chapters 2 through 6. It is strongly recommended that you gather all the data that is needed for configuration *before you start.* Refer to Chapter 2, "Planning," for tips on gathering data.

What is an OPS Cluster?

A **high availablity cluster** is a grouping of HP 9000 series 800 servers having sufficient redundancy of software and hardware components that a single point of failure will not disrupt the availability of computer services. High availability clusters configured with Oracle Parallel Server are known as **OPS clusters**. Figure 1-2 shows a very simple picture of the basic configuration of an OPS cluster on HP-UX.
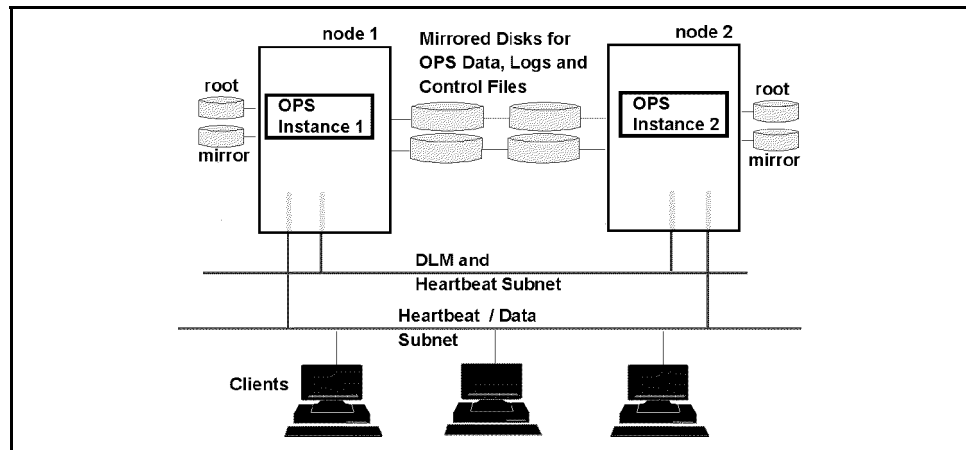
Figure 1-2. Overview of Oracle Parallel Server Configuration on HP-UX

In the figure, two loosely coupled HP 9000 series 800 systems (each one known as a **node**) are running separate instances of Oracle software that read data from and write data to a shared set of disks. Clients connect to one node or the other via LAN.

OPS on HP-UX lets you maintain a single database image that is accessed by the HP 9000 servers in parallel, thereby gaining added processing power without the need to administer separate databases. MC/LockManager handles issues of concurrent access to the same resources by different servers and ensures data integrity. Further, when redundant LAN hardware and disk mirroring are used, MC/LockManager provides a highly available database that continues to operate even if one hardware component should fail.

Using Packages in an OPS Cluster

In order to make other important applications highly available (in addition to the Oracle Parallel Server RDBMS), you can configure your OPS cluster to use **packages**. Packages group applications and services together; in the event of a service, node, or network failure, MC/LockManager can automatically transfer control of all system resources in a designated package to another node within the cluster, allowing your applications to remain available with minimal interruption.

Note	Note that it is *not* the OPS instances themselves which are grouped in packages. Rather, packages contain other types of applications and services, including those which may access the OPS database.

Basic Concepts and Components

An OPS cluster is a set of nodes running as independent servers but accessing the same database files. In addition, an OPS cluster is intended to be highly available. The health of the OPS cluster is continually monitored, and in the event of one node's failure, the cluster reconfigures itself as a single node, and clients on the failed node can reconnect to the other node. In such a case, the failed node may rejoin the cluster after the failure has been corrected.

Redundant Cluster Components

A clear understanding of the hardware requirements for OPS clusters makes
the software configuration more intuitive. OPS on HP-UX requires HP9000
Series 800 servers, redundant sets of LAN hardware, and several disks or disk
arrays configured with Fast/Wide SCSI I/O bus architecture. A sketch of the
hardware components for one possible configuration is given in Figure 1-3.
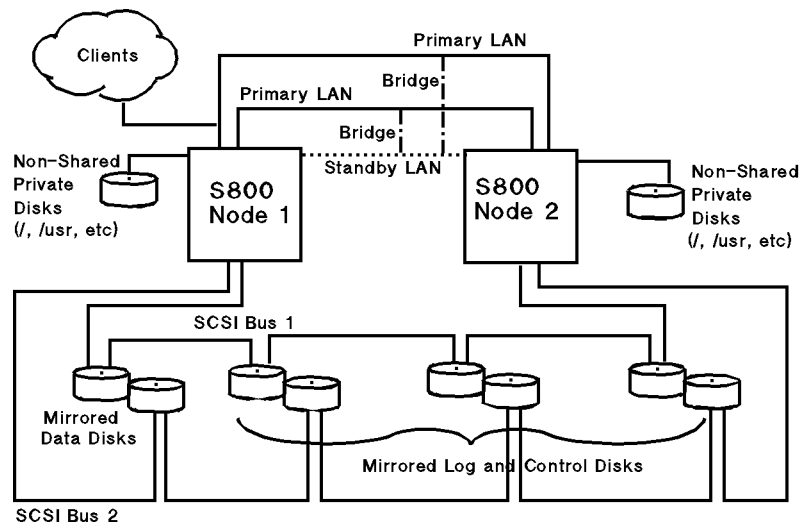
Figure 1-3. Oracle Parallel Server Hardware Configuration

HP 9000 Systems

The nodes in an OPS cluster are HP 9000 systems with similar memory
configuration and processor architecture. A node can be any S800 model; S700s
are not supported as OPS cluster nodes. It is recommended that both nodes be
of similar processing power and memory capacity. If the nodes to be clustered
have different amounts of processing power and memory size, you may observe
the following behavior:

■ The node with less memory may become a bottleneck. The reason is that the
distributed lock manager (DLM), which provides parallel cache management
for OPS, has shared memory segments which must be the same size on both
nodes.

■ The node with less processing power may become a bottleneck, since roughly half the DLM locks requested by one node will be serviced by the other node.

Network Components

MC/LockManager uses TCP/IP network services for reliable communication among nodes in the cluster.

MC/LockManager supports the following local area networks:

■ Ethernet

■ FDDI

Redundancy in Network Interfaces

Networked communication is required for the exchange of data and for the passage of cluster heartbeat information. Therefore, at least one LAN is required for an OPS configuration. If a single LAN only is provided, then the cluster will still survive as a single-node cluster in the event of LAN failure; however, all LAN-connected clients could lose connectivity if the LAN fails. Hence, for the case of LAN-connected clients, at least two LAN interfaces per node are required. (If the clients connect via some other mechanism, such as X.25, the requirement of multiple interfaces per node does not exist.)

You obtain the greatest availability by using a **bridged net** consisting of one or more primary LAN interfaces and at least one standby LAN interface on each cluster node. A bridged net is a set of interfaces that are interconnected through bridges or the use of common cabling. If the LAN interfaces on a node are connected to the same bridged net, and if a standby is available, the system can switch the IP address assigned to a primary interface card to a standby card in the event of a failure. **Primary** interfaces are those which are mapped by the operating system to a particular IP subnet at boot time; **standby** interfaces are those which are available for switching by MC/LockManager software if a failure occurs on the primary. Use of a bridged net preserves access to Oracle services on each node and maintains the full strength of the cluster in the event of a network failure

Using a Serial (RS232) Heartbeat Line

MC/LockManager supports using serial (RS232) communication for cluster heartbeats only. (The heartbeat messages are used to monitor the health of the cluster.) You can select a serial (RS232) line as an alternate heartbeat interface to provide redundant heartbeat. If you configure a serial line as a heartbeat line, MC/LockManager will send the heartbeat continuously on both the LAN configured for heartbeat and the serial line. Note that even if you have a serial line configured for redundant heartbeat, one LAN is still required to carry a heartbeat signal. If user traffic overloads the heartbeat LAN such that heartbeats are delayed for a short period of time, the cluster continues to run using the redundant serial heartbeat.

Figure 1-4 shows an OPS cluster configured with primary and standby LANs, and a serial (RS232) line providing redundant heartbeat.
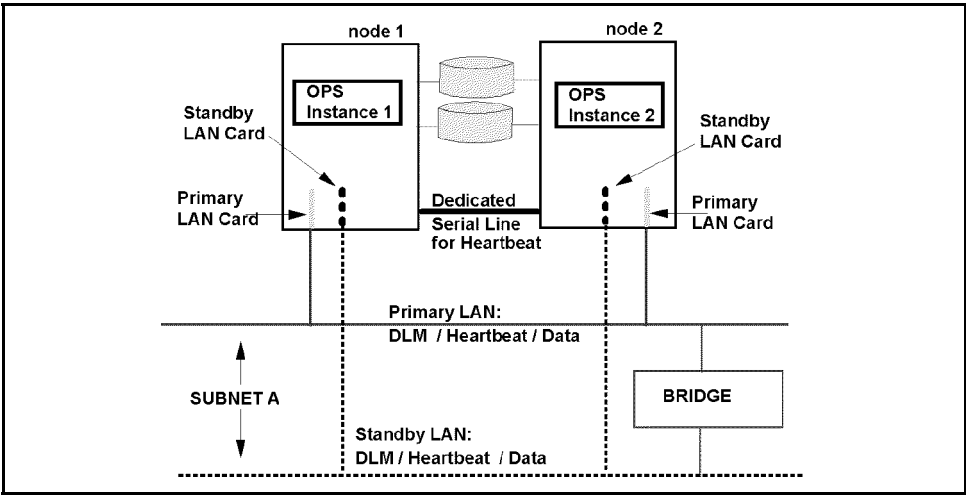
Figure 1-4. Two-LAN Configuration with Redundant Serial (RS232) Line

Networking Guidelines

The recommended distribution of network traffic depends on the number and type of LAN interfaces you use on each node:

■ If there is a single primary interface card and no standby, then Cluster Manager, DLM, and application traffic (including any Oracle SQL*Net traffic) are all on one subnet. A serial (RS232) line can provide redundant heartbeat.

  In this situation if the LAN fails in a cluster with both nodes running, the cluster will survive as a one-node cluster (one node exits the cluster). However, clients will not be able to access the cluster if their route to the cluster is through the subnet that failed. Separate subnets and separate interface cards for client access are recommended.

  Note that a redundant serial (RS232) heartbeat does not prevent a failure of the only LAN card (providing heartbeat) from causing a two-node cluster to reform as a one-node cluster.

■ If there is one primary interface card and one standby, then there is still only one subnet, and thus all traffic is on that subnet, as above. A serial (RS232) line can provide redundant heartbeat. However, switching is now possible in the event of a failure on the primary interface card. Using a second LAN card as a standby will prevent a cluster reformation in the event of a failed primary LAN card.

■ If there are two primary interface cards and one standby, then there are two subnets available for network traffic. Heartbeat messages should either be sent over both LANs or one LAN should be dedicated to heartbeat only. A serial (RS232) line can provide redundant heartbeat. See "Heartbeat Messages" for more guidelines on handling heartbeat messages.

  Switching from either primary to the standby is possible in the event of a failure on one primary interface card.

Sample Network Configurations

Several additional possible network configurations are shown in the following figures.

Figure 1-5 shows a three-LAN configuration for additional redundancy using two bridges:
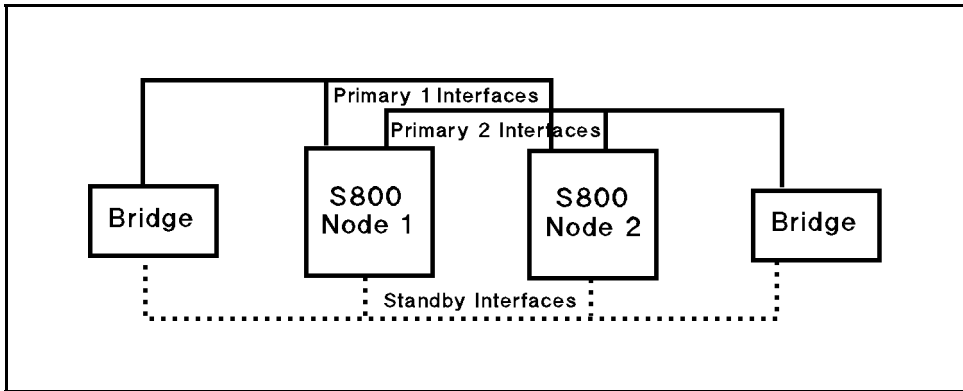
Figure 1-5. Three-LAN Ethernet Bridged Network

Figure 1-6 shows the state of the configuration following a failure of the cable on Primary interface 1:
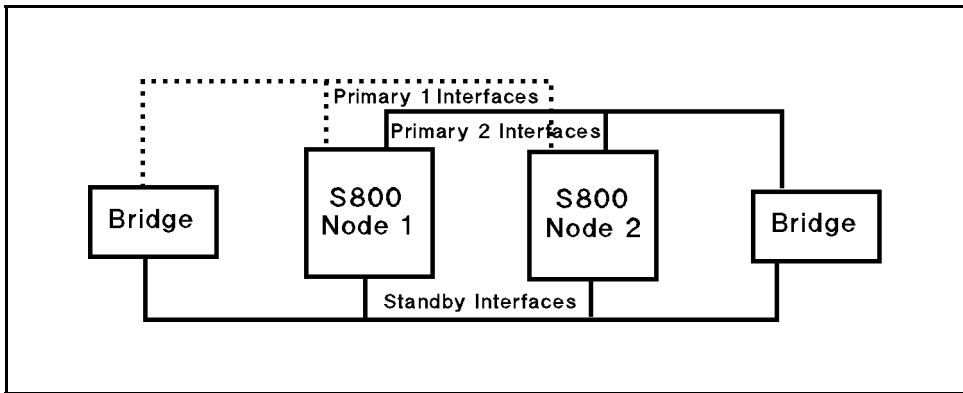
Figure 1-6. Three-LAN Ethernet Bridged Network after Failure of one LAN

Finally, Figure 1-7 shows an FDDI configuration:
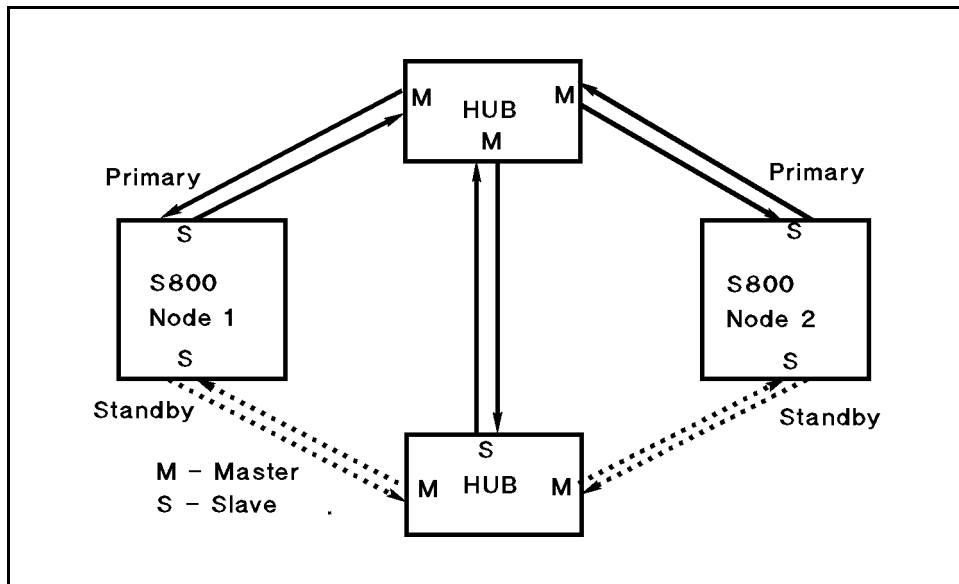
Figure 1-7. Two-Hub FDDI Bridged Network

I/O Busses

Disks using the Fast/Wide SCSI interface are supported as shared disks by MC/LockManager. Unshared disks can use any supported HP disk interface. A complete list of disks supported for OPS clusters appears in the *Release Notes* that accompany the software distribution media for MC/LockManager.

Members of the cluster should also be equipped with a sufficient number of I/O slots to configure an appropriate number of redundant device adapters for disk and network connections. Different mirror copies of a disk should be on different busses.

Disk Drives

Oracle Parallel Server uses shared database disks for data, log files, and control files. If you have configured your cluster to run packages that access data, they will also require disks, separate from those that contain OPS database files.

The disks are placed in volume groups whose logical volumes can be mirrored to a disk on a different bus. Disks that will be used in volume groups that are accessible by more than one node in the cluster must be one of the following types:

- F/W SCSI Disk
- F/W SCSI Disk Array

In particular, single-ended SCSI is not supported as a shared database disk. Check the *MC/LockManager Release Notes* for the list of supported shared disks. The use of unsupported disk devices may lead to unpredictable results.

Data Protection

It is strongly recommended that you provide data protection for your highly available cluster, using one of two methods:

- Disk Mirroring
- High Availability Disk Arrays

Disk Mirroring

Software disk mirroring is one method for providing data protection. MC/LockManager does not provide protection of data on your disks; protection is provided by HP's MirrorDisk/UX product, which operates in conjunction with the Logical Volume Manager. The following mirroring guidelines are suggested:

- Each database disk should be mirrored by at least one other disk on a separate bus.
- Mirroring is accomplished by identifying different physical volume groups within each shared volume group, then specifying PVG-strict mirror allocation.
- Although different types of disks may be used in a configuration, mirroring should be between disks of the *same type* for best performance.

Your HP representative can provide additional information about limits on cable length.

High Availability Disk Arrays

An alternate method of achieving protection for your data is to use an HP High Availability Disk Array, which supports RAID modes 0 (striping), 1 (mirroring), 0/1 (striping and mirroring), and 5 (rotating parity).

High Availability Disk Arrays have the following advantages:

- Redundant power supplies and fans.

- Hot-swap capability. The repair of a failed component in an High Availability Disk Array does not require any scheduled downtime for maintenance, since all the major components are hot-swappable, that is, they can be replaced online. Hot-swap is available in RAID modes 1, 0/1, and 5.

- A global spare for the disk volumes. This means that one spare drive unit can be used as a backup of all of the RAID volumes that have been configured in the array. If one drive fails in any of the defined volumes, the global spare is used to re-establish full redundancy.

For High Availability Disk Arrays that are configured for concurrent access by more than one node, redundancy of controllers is *not* currently supported.

To achieve the greatest degree of high availability, you can combine software disk mirroring and High Availability Disk Arrays. In this way you can take advantage of software disk mirroring to eliminate a possible single point of failure, and of the High Availability Disk Arrays' redundant capabilities and online maintenance features.

Power Supplies

You can extend the availability of your hardware by providing battery backup to your nodes and disks. HP's PowerTrust Uninterruptible Power Source (UPS) can provide this protection from momentary power loss.

Disks should be attached to power circuits in such a way that mirror copies are attached to different power sources. The boot disk should be powered from the same circuit as its corresponding node.

The cluster lock disk (used as a tie-breaker when re-forming a cluster) should have its own power supply, separate from the power supply used by the nodes in the cluster. Your HP representative can provide more details about the layout of power supplies, disks, and LAN hardware for clusters.

Software Components

OPS on HP-UX requires the following software components *on each node*:

- Oracle Parallel Server RDBMS.
- MC/LockManager, including cluster manager (CM), package manager (PM), network manager (NM), and the distributed lock manager (DLM).
- HP-UX operating system
- Mirror Disk/UX (for mirrored disk configurations).

A diagram of the software components *on each node* is presented in Figure 1-8.
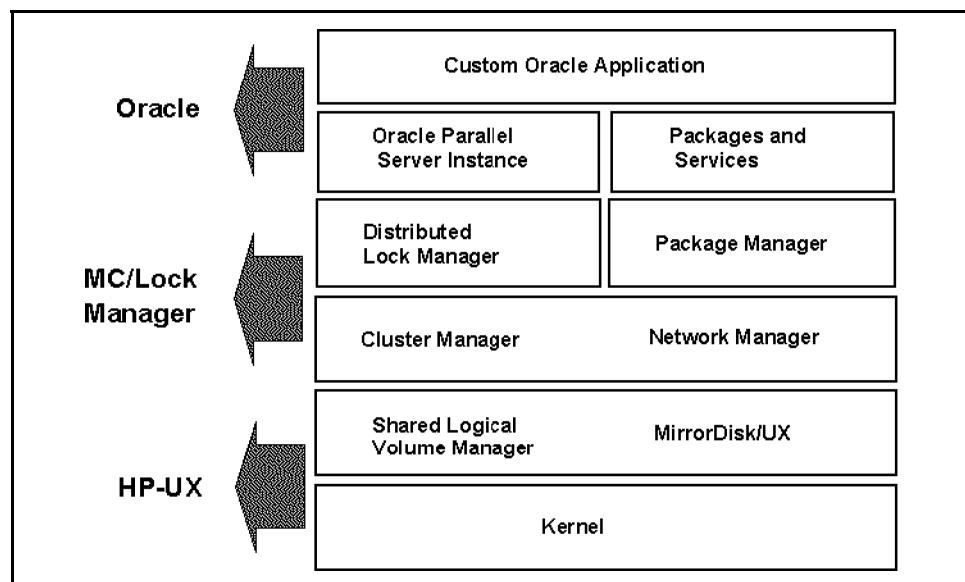
Figure 1-8. Oracle Parallel Server Software Configuration

Oracle Parallel Server RDBMS

The Oracle RDBMS provides greater database access and higher throughput by allowing parallel instances of the Oracle software to access the same database

image. At installation time, the Oracle Server software is linked with the Distributed Lock Manager (DLM) component of MC/LockManager. Refer to the *Oracle Parallel Server Administrator's Guide* for details about setting up Oracle database instances.

All Oracle instances have concurrent shared access to the same data files, log files, and control files. End user access to the Oracle instances might be through a host application program or a transaction monitor.

MC/LockManager

The key enabling software for Oracle Parallel Server on HP-UX is MC/LockManager. MC/LockManager coordinates the transaction activity that takes place on different nodes in the cluster and manages cluster configuration and reconfiguration tasks.

MC/LockManager is composed of four software components:

- Cluster Manager (CM) - initializes and monitors the cluster.

- Package Manager (PM) - monitors and controls packages containing highly available applications.

- Network Manager (NM) - detects and recovers from card and cable failures.

- Distributed Lock Manager (DLM) - provides parallel cache management for OPS.

More information on how these software components work is given later in this chapter.

HP-UX Operating System

The MC/LockManager product requires the HP-UX operating system, release 10.10. If you have a version of HP-UX that is earlier than 10.10, you will need to load a newer release. It is suggested that you review the use of Logical Volume Manager software before installing and configuring OPS on HP-UX. Refer to *System Administration Tasks* manual for HP-UX 10.0 Series 800.

For mirrored disk configurations, your system must have the HP Mirror Disk/UX software product installed on each node.

Logical Volume Manager (LVM)

The Logical Volume Manager (a subsystem of HP-UX) enables the use of **logical volumes**, which are collections of pieces of disk space from one or more disks. Each collection is put together so that it appears to the operating system like a single disk. Like disks, logical volumes can be used to hold file systems, raw data areas, dump areas, or swap areas. Unlike disks, the size of a logical volume can be chosen when the logical volume is created, and can later be expanded or reduced. Also, logical volumes may be mirrored singly or doubly. Mirroring to another disk device (located on a different I/O bus) provides higher availability.

Although both the Oracle Parallel Server and application packages use the Logical Volume Manager, there are some key differences:

- OPS uses *shared* volume groups in raw mode that can be accessed concurrently by different nodes in the cluster.

- Packages use *exclusive* volume groups, which can contain file systems, and which can be activated by only one node at a time.

The details for each case are explained in the following sections.

Shared Logical Volume Manager (SLVM) for OPS

Oracle Parallel Server uses disk files that are accessed by both nodes concurrently. This means two things:

- The configuration uses disk drives that are physically connected to both nodes.
- The OPS database files (data, log, and control files) are specifically configured as *shared raw logical volumes*.

The use of shared files means that each node can access the data directly, rather than requesting some other node to read the data and then send it across the network. Networking comes into play only to synchronize accesses to the disks rather than to gain access to the data itself.

Data that will be shared between the OPS nodes is stored on shared HP-UX volume groups, which are configured using the commands of the HP-UX Shared Logical Volume Manager (SLVM). OPS uses *raw access* to shared logical volumes, rather than going through a file system. Therefore, OPS itself provides concurrency control for data, as well as transaction logging and

recovery facilities, as appropriate. File systems are *not supported* on shared volume groups.

Volume groups for use with OPS are defined on the node where they are created, then the volume group structure is imported to the second node that will also share the data; the volume groups are then activated in shared mode by all nodes before the OPS instances are started. Details of importing and activating shared volume groups appear in Chapters 3 and 5. For general information about configuring logical volumes, refer to the *System Administration Tasks* manual for HP-UX 10.0 Series 800. In addition, consult the man page for the `vgchange` command *after you install MC/LockManager*. This command uses options that are not available in non-shared Logical Volume Manager.

The use of logical volumes allows the database administrator (DBA) to manage disk storage by adding capacity as needed to an already configured logical volume. Adding additional capacity to a volume in a shared volume group must be done when the volume group is in non-shared mode. A technique for adding capacity is given in the "Maintenance" chapter.

Logical Volume Manager (LVM) for Packages

Package volume groups can be accessed by both nodes at different times. This means two things:

- The configuration uses disk drives that are physically connected to both nodes.
- Unlike OPS, packages activate volume groups exclusively. At any given time only one node can access the disks in the package volume group. Also, package volume groups contain logical volumes that can have mounted file systems.

Package volume groups are defined on the node where they are created, then imported to the second node that can run the package in the event of a failure. Once cluster-bound, package volume groups are only activated in exclusive mode. Details of importing and activating package volume groups appear in Chapters 4 and 5. For general information about configuring logical volumes, refer to the *System Administration Tasks* manual for HP-UX 10.0 Series 800. In addition, consult the man page for the `vgchange` command.

How the Cluster Manager Works (CM)

The **cluster manager** is used to initialize a cluster, to monitor the health of the cluster, to recognize node failure if it should occur, and to regulate the re-formation of the cluster when a node joins or leaves the cluster. The cluster manager operates as a daemon process that runs on each node. During cluster startup and re-formation activities, one node is selected to act as the **cluster coordinator**. Although all nodes perform some cluster management functions, the cluster coordinator is the central point for heartbeat messages.

Configuration of the Cluster

The system administrator sets up cluster configuration parameters and does an initial cluster startup; thereafter, the cluster regulates itself without manual intervention in normal operation. Configuration parameters for the cluster include the cluster name and nodes, networking parameters for the cluster heartbeat, cluster lock disk information, and timing parameters (discussed in detail in the "Planning" chapter). Cluster parameters are entered using SAM or by editing an ASCII cluster configuration template file. The parameters you enter are used to build a binary configuration file which is propagated to all nodes in the cluster. This binary cluster configuration file must be the same on all the nodes in the cluster.

Manual Startup of Entire Cluster

A manual startup forms a cluster out of all the nodes in the cluster configuration. Manual startup is normally done the first time you bring up the cluster, after cluster-wide maintenance or upgrade, or after changing cluster parameters.

Before startup, the same binary cluster configuration file must exist on all nodes in the cluster. The system administrator starts the cluster in SAM or with the `cmruncl` command issued from one node. The `cmruncl` command can only be used when the cluster is not running, that is, when none of the nodes is running the *cmcld* daemon.

Warning	MC/LockManager cannot guarantee data integrity if you try to start a cluster with the `cmruncl` command while a subset of the cluster's nodes are already running a cluster.

During startup, the cluster manager software checks to see if all nodes specified in the startup command are valid members of the cluster, are up and running, are attempting to form a cluster, and can communicate with each other. If they can, then the cluster manager forms the cluster.

Automatic Cluster Startup on Each Node

Automatic startup is the process in which each node individually joins a cluster. If a cluster already exists, the node attempts to join it; if no cluster is running, the node attempts to form a cluster consisting of all configured nodes. Automatic cluster start is the preferred way to start a cluster. No action is required by the system administrator. To enable automatic cluster start, set the flag `AUTOSTART_CMCLD` to 1 in the `/etc/rc.config.d/cmcluster` file.

Heartbeat Messages

Central to the operation of the cluster manager is the sending and receiving of **heartbeat messages** among the nodes in the cluster.

Each node in the cluster sends a heartbeat message over the LAN or a serial (RS232) line to the cluster coordinator.

The cluster coordinator looks for this message from each node and if it is not received within the prescribed time it will re-form the cluster. At the end of the re-formation, if a new set of nodes form a cluster, that information is passed to the package manager. Control of the packages which were running on nodes that are no longer in the new cluster are transferred to the adoptive nodes in the new configuration.

You can separate the subnet carrying heartbeat messages from the subnet that carries user data. If heartbeat and data are sent over the same subnet, data congestion may cause MC/LockManager to miss heartbeats during the period of the heartbeat timeout and initiate a cluster re-formation that would not be needed if the congestion had not occurred.

To prevent this situation, do one or more of the following

- Run heartbeat on a dedicated or low-traffic LAN.

- In addition to a heartbeat LAN, run a serial (RS232) heartbeat line to provide redundancy.

- Run heartbeat on all available LANs. (If the LAN traffic is very heavy, you may still have a problem with heartbeat misses in this situation.)

Each node sends its heartbeat message at a rate specified by the cluster heartbeat interval. The cluster heartbeat interval is set in the cluster configuration file, which you create as a part of cluster configuration, described fully in Chapter 3.

Automatic Cluster Restart

An automatic cluster restart occurs when all nodes in a cluster have failed. This is usually the situation when there has been an extended power failure and all nodes were down. In order for an automatic cluster restart to take place, all nodes specified in the cluster configuration file must be up and running, must be trying to form a cluster, and must be able to communicate with one another. Automatic cluster restart will take place if the flag `AUTOSTART_CMCLD` is set to 1 in the `/etc/rc.config.d/cmcluster` file.

Dynamic Cluster Re-formation

A dynamic re-formation is a temporary change in cluster membership that takes place as nodes join or leave a running cluster. Re-formation differs from reconfiguration, which is a permanent modification of the configuration files. Re-formation of the cluster occurs under the following conditions:

- An node or network failure was detected on an active node.
- A software failure was detected on an active node.
- An inactive node wants to join the cluster. The cluster manager daemon has been started on that node.
- A node halts because of a package failure.
- A node halts because of a service failure.
- The system administrator halted a node.
- Heavy network traffic prohibited the heartbeat signal from being received by the cluster.
- The heartbeat network failed.

Typically, re-formation results in a cluster with a different composition. The new cluster may contain fewer or more nodes than in the previous incarnation of the cluster.

However, if there is a local standby LAN card, the same set of nodes will re-form a new cluster.

Cluster Quorum for Re-formation

The algorithm for cluster re-formation generally requires a cluster quorum of a strict majority (that is, more than 50%) of the nodes previously running. However, exactly 50% of the previously running nodes may re-form as a new cluster provided there is a guarantee that the other 50% of the previously running nodes do not also re-form. In these cases, a tie-breaker is needed. For example, if there is a communication failure between the nodes in a two-node cluster, and each node is attempting to re-form the cluster, then MC/LockManager only allows one node to form the new cluster. This is ensured by using a **cluster lock**.

Use of the Cluster Lock

The cluster lock is a disk area located in a volume group that is shared by all nodes in the cluster. The cluster lock volume group and physical volume names are identified in the cluster configuration file. The cluster lock is used as a tie-breaker only for situations in which a running cluster fails and, as MC/LockManager attempts to form a new cluster, the cluster is split into two sub-clusters of equal size. Each sub-cluster will attempt to acquire the cluster lock. The sub-cluster which gets the cluster lock will form the new cluster preventing the possibility of two sub-clusters running at the same time.

If you have a two node cluster, you are required to configure the cluster lock. Without a cluster lock, a failure of either node in the cluster will cause the other node, and therefore the cluster, to halt. If communications are lost between these two nodes, the node with the cluster lock will take over the cluster and the other node would shut down.

You can choose between two cluster lock options—a single or dual cluster lock—based on the kind of high availability configuration you are building. In both cases, it is important that the cluster lock disk be available even if one node loses power; thus, the choice of a lock configuration depends partly on the number of power circuits available. Regardless of your choice, all nodes in the cluster must have access to the cluster lock to maintain high availability.

Single Cluster Lock

When possible, it is highly recommended to use three power circuits for a two-node cluster, with a single, separately powered disk for the cluster lock. For two-node clusters, this disk may not share a power circuit with either node, and it must be an external disk.

Dual Cluster Lock

When it is not possible to use three power circuits—for example, when you are using two nodes and no external disks—use a dual cluster lock, with two cluster lock disks. The disks must not share either a power circuit or a node chassis with one another. In this case, if there is a power failure affecting one node and disk, the other node and disk remain available, so cluster re-formation can take place on the remaining node.

Note	Only configure a dual cluster lock when it is required by your cluster configuration (when you only have two power circuits). When possible, a single cluster lock is recommended.

No Cluster Lock

Normally, you should not configure a cluster without a cluster lock. In two-node clusters, a cluster lock is required for use in all cases involving the failure of a node or of intra-cluster communications. If you do not configure a lock, and if a failure necessitates tie-breaking, the entire cluster will become unavailable.

How the Distributed Lock Manager (DLM) Works

The distributed lock manager (DLM) provides parallel cache management for OPS. Each node in an OPS cluster starts an instance of the DLM process when the node joins the cluster, and the instances then communicate with each other over the network. DLM timing and other parameters for an OPS cluster are stored in the cluster configuration file and in a specific DLM configuration file.

DLM Configuration Files

One set of DLM parameters is stored along with the cluster configuration data in the cluster configuration file. Known as **cluster interface parameters**, they are a specific set of interval and timing parameters which ensure that the cluster manager can recognize the failure of the DLM when it occurs.

A second set of DLM configuration parameters is located in a binary DLM configuration file which is stored on all nodes in the cluster. These parameters, known as **internal parameters**, allow the DLM to operate smoothly with a particular OPS configuration. The Oracle database administrator and the HP-UX system administrator together decide on the values for the internal parameters. Details are given in Chapter 2.

The DLM configuration file may be created or modified using SAM or using HP-UX commands. Details are given in Chapter 3.

How the Package Manager Works

Packages provide the software support which enables and controls the transfer of applications (other than OPS instances) to another node or network after a node or network failure. For software failures, an application can be restarted on the same node or another node with minimum disruption.

Packages also give you the advantage of easily transferring control of your application to another node in order to bring the original node down for system administration, maintenance, or version upgrades.

The **package manager** is used to coordinate package activities among the nodes of the cluster. Each node in the cluster runs an instance of the package

manager; the package manager residing on the cluster coordinator is known as the **package coordinator**.

The package manager does the following:

- Decides when and where to run, halt or move packages.
- Executes the user-defined control script to run and halt packages and package services.
- Reacts to changes in the status of monitored resources.

The package manager monitors the health of the packages running on individual nodes. Any node running in the MC/LockManager cluster is called an **active node**. When you create the package, you specify a **primary node** and one or more **adoptive nodes** for the package. When a node or its network communications fails, MC/LockManager can transfer control of the package to the next available adoptive node.

After this transfer, the package remains on the adoptive node as long the adoptive node continues running, even if the primary node comes back online. In situations where the adoptive node continues running successfully, you must manually transfer control of the package back to the primary node at the appropriate time. In certain circumstances, in the event of an adoptive node failure, a package that is running on an adoptive node will switch back automatically to its primary node (assuming it is back online).

Deciding When and Where to Run and Halt Packages

Each package is separately configured by means of a package configuration file, which can be edited manually or through SAM. This file assigns a name to the package and identifies the nodes on which the package can run, in order of priority. It also indicates whether or not switching is enabled for the package, that is, whether the package should switch to another node or not in the case of a failure. There may be many applications in a package. Package configuration is described in detail in the chapter "Configuring Packages and their Services."

A typical OPS cluster using packages is shown in Figure 1-9.
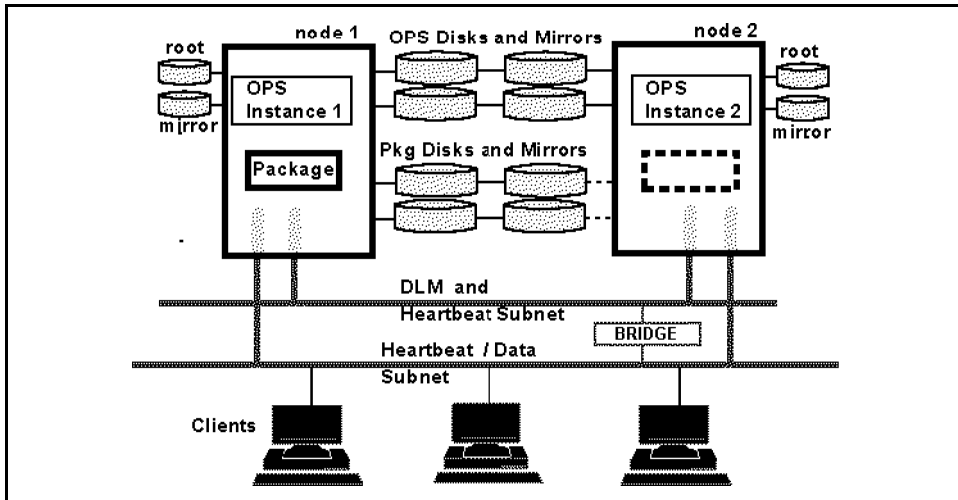
Figure 1-9. Cluster with Packages

In the figure, node 1 is running OPS instance 1, and node 2 is running instance 2. A package is shown running on node 1. Separate groups of disks are shown for the OPS instances and for the package. Node 1 is shown containing a dotted box, which indicates that the package can switch to node 2 if there is a failure on node 1.

Each package has a separate group of disks associated with it, containing data needed by the package's applications, and a mirror copy of the data. Note that both nodes are physically connected to the disks and their mirrors. However, only one node at a time may access the data for a given group of disks. In the figure, node 1 is shown with exclusive access to the package's disks (solid line), and node 2 is shown as connected without access to the disks (dotted line).

Mirror copies of data provide redundancy in case of disk failures. In addition, a total of four data busses are shown for the disks that are connected to node 1 and node 2—two for OPS data and two for package data. This configuration provides the maximum redundancy and also gives optimal I/O performance, since the package and the OPS instances are using different busses.

Starting the Package and Running Services

After a cluster has formed, the package manager on each node starts up the packages for which that node is primary. Starting the package means running the package control script with the 'start' parameter. This script performs the following tasks:

- uses Logical Volume Manager (LVM) commands to activate (in exclusive mode) volume groups needed by the package.
- mounts filesystems from the activated volume groups to the local node.
- uses `cmmodnet` to add the package's IP address to the current network interface running on a configured subnet. This allows clients to connect to the same address regardless of the node the service is running on.
- uses the `cmrunserv` command to start up each service configured in the package. This command also initiates monitoring of the service.

While the package is running, services are continuously monitored. The **Service Monitor** checks the PIDs of services started by the package control script. If it detects a PID failure, the package halt instructions are executed as part of a recovery process. A failure of any part of the package may result in simple loss of the service, a restart of the service, transfer of the package to an adoptive node, or transfer of all packages to an adoptive node, depending on the package configuration. The package configuration file and control script are described in detail in the chapter "Configuring Packages and their Services."

Stopping the Package

The package manager is notified when a command is issued to shut down a package. In this case, the package control script is run with the 'stop' parameter. For example, if the system administrator chooses "Halt Package" from the "Package Administrations" menu in SAM, the package manager will stop the package. Similarly, when a command is issued to halt a cluster node, the package manager will shut down all the packages running on the node, executing each package control script with the 'stop' parameter. When run with the 'stop' parameter, the control script:

- uses `cmhaltserv` to halt each service.
- unmounts filesystems from the activated volume groups to the local node.
- uses Logical Volume Manager (LVM) commands to deactivate volume groups used by the package.

- uses `cmmodnet` to delete the package's IP address from the current network interface.

How the Network Manager Works

The purpose of the network manager is to detect and recover from network card and cable failures so that network services remain highly available to clients. If your cluster is configured to run packages, this means assigning IP addresses for each package to the primary LAN interface card on the node where the package is running and monitoring the health of all interfaces, switching them when necessary.

Node and Package IP Addresses

Each node in the cluster should have an IP address for each active network interface. This address, known as a **stationary IP address**, is configured in the node's /etc/rc.config.d/netconf file. A stationary IP address is not transferrable to another node, but it is transferrable to a standby LAN interface card. The stationary IP address should *not* be associated with packages.

If your cluster is configured to use packages, in addition to the stationary IP address, you normally assign one or more unique IP addresses to each package. The package IP address is assigned to the primary LAN interface card by the `cmmodnet` command in the package control script when the package starts up. The IP addresses associated with a package are called **floating IP addresses** or **relocatable IP addresses** because the addresses can actually move from one cluster node to another.

Both stationary and package IP addresses will switch to a standby LAN interface in the event of a LAN card failure. In addition, package addresses (but not stationary addresses) can be taken over by an adoptive node if control of the package is transferred. This means that applications can access the package via its reloctable address without knowing the current node's stationary IP address or hostname.

Adding and Deleting Package IP Addresses

When a package is started, a package IP address can be added to a specified IP subnet. If an IP address is to be added, the network manager checks the status of the specified subnet; if the subnet is up, the network manager locates the primary interface card for the subnet and adds the IP address to it.

When the package is stopped, the package IP address is deleted from the specified subnet. In this case, the network manager locates the primary interface card on the specified subnet and deletes the IP address from it.

Adding and deleting package IP addresses is handled through the package control script, which is described in detail in the chapter "Configuring Packages and their Services."

IP addresses are configured only on each primary network interface card; standby cards are not configured with an IP address. Multiple IP addresses on the same network card must belong to the same IP subnet.

Load Sharing

It is possible to have multiple services on a node associated with the same IP address. If one service is moved to a new system, then the other services using the IP address will also be migrated. Load sharing can be achieved by making each service its own package and giving it a unique IP address. This gives the administrator the ability to move selected services to less loaded systems.

Limitations on Configuration

Two subnetworks can not be configured concurrently for the same network interface. This is a current limitation of HP's networking code. Also, one system can not have two active network interfaces using the same subnetwork. The networking product does not support that configuration.

Monitoring LAN Interfaces and Detecting Failure

MC/LockManager polls all the network interface cards specified in the cluster configuration file. Network failures are detected in the following manner. One interface in a bridged net is assigned to be the poller. The poller will poll the other primary and standby interfaces in the bridged net to see whether they are still healthy. Normally, the poller is a standby interface; if there are

no standby interfaces in a bridged net, the primary interface is assigned the polling task.

The polling interface sends link-level messages to all other interfaces in a bridged net and receives link-level messages back from all other interfaces in the bridged net. If an interface cannot receive or send a message, and when the numerical count of packets sent and received on an interface does not increment for an amount of time, the interface is considered DOWN.

Local Switching

A local network switch involves the detection of a local network interface failure and a failover to the local backup LAN card. The backup LAN card must not have any IP addresses configured. In the case of a local network switch, TCP/IP connections are not lost. During the transfer, IP packets will be lost, but TCP (Transmission Control Protocol) will retransmit the packets. In the case of UDP (User Datagram Protocol), the packets will not be retransmitted automatically by the protocol. However, since UDP is an unreliable service, UDP applications should be prepared to handle the case of lost network packets and handle this case appropriately. Note that a local switchover is supported only between two LANs of the same type. For example, a local switchover between Ethernet and FDDI interfaces is not supported.

Remote Switching

A remote switch involves moving packages and their associated IP addresses to a new system. The new system must already have the same subnetwork configured and working properly, otherwise the packages will not be started. With remote switching, TCP connections are lost. TCP applications must reconnect to regain connectivity; this is not handled automatically. Note that if the package is dependent on multiple subnetworks, all subnetworks must be available on the target node before the package will be started.

ARP Messages after Switching

When a floating IP address is moved to a new interface, either locally or remotely, an ARP message is broadcast to indicate the new mapping between IP address and link layer address. An ARP message is sent for each IP address

that moved. All systems receiving the broadcast should update the associated ARP cache entry to reflect the change.

Currently, the ARP messages are sent at the time the IP address is added to the new system. An ARP message is sent in the form of an ARP request. The sender and receiver protocol address fields of the ARP request message are both set to the same floating IP address. This ensures that nodes receiving the message will not send replies.

Responses to Failures

MC/LockManager responds to different kinds of failures in specific ways. For most hardware failures, the response is not user-configurable, but for package and service failures, you can choose the system's response, within limits.

Transfer of Control (TOC) When a Node Fails

If the daemon, *cmcld*, fails, the node is halted immediately using an HP-UX TOC (Transfer of Control), which is an immediate halt without a graceful shutdown. A system dump is performed and the following message is sent to the console:

```
Unable to maintain contact with cmcld daemon.
Performing TOC to ensure data integrity.
```

If the Package Failfast or Service Failfast parameter is Enabled in the package configuration file, the entire node will fail with a TOC whenever there is a failure of that specific package or service. A node-level failure may also be caused by events independent of a package and its services. Loss of the heartbeat, loss of the MC/LockManager or other critical daemons, or other events, will cause a node to fail even when its packages and their services are functioning.

Responses to Hardware Failures

If a serious system problem occurs, such as a panic or physical disruption of the node's circuits, MC/LockManager recognizes a node failure and transfers any packages currently running on that node to an adoptive node elsewhere in the cluster. The new location for each package is determined by that package's configuration file, which lists primary and alternate nodes for the package. Transfer of a package to another node does not transfer the program counter. Processes in a transferred package will restart from the beginning. In order for an application to be expeditiously restarted after a failure, it must be "crash-tolerant"; that is, all processes in the package must be written so that they can detect such a restart.

This is the same design required for restart after a normal system crash.

In the event of a LAN interface failure, a local switch is done to a standby LAN interface if one exists; otherwise, the node fails with a TOC.

MC/LockManager does not respond to disk failure; disk protection is provided by the separate product MirrorDisk/UX or HP High Availability Disk Arrays. Packages do not switch as a result of disk failures.

MC/LockManager does not respond directly to power failures, although a loss of power to an individual cluster component may appear to MC/LockManager like the failure of that component, and will result in the appropriate switching behavior.

Power protection is provided by PowerTrust, HP's uninterruptible power supply.

Responses to Package and Service Failures

In the default case, the failure of the package or of a service within a package causes the package to shut down by running the control script with the 'stop' parameter, and the package starts up on an alternate node.

If you wish, you can modify this default behavior by specifying that the node should crash (TOC) before the transfer takes place. In cases where package shutdown may take a long time but the package is crash-tolerant and can recover quickly on restart, this option can make the package and its associated

applications available to users more quickly. Remember, however, that when the node crashes, *all* packages on the node are halted abruptly.

The settings of package and service Failfast parameters during package configuration will determine the exact behavior of the package and the node in the event of failure. The section on "Package Configuration Parameters" in the "Planning" chapter contains details on how to choose an appropriate failover behavior.

Service Restarts

You can allow a service to restart locally following a failure. To do this, you indicate a number of restarts for each service in the package control script. When a service starts, the variable $RESTART\_COUNT$ is set in the service's environment. The service, as it executes, can examine this variable to see whether it has been restarted after a failure, and if so, it can take appropriate action such as cleanup.

Network Communication Failure

An important element in the cluster is the health of the network itself. As it continuously monitors the cluster, each node listens for heartbeat messages from the other nodes confirming that all nodes are able to communicate with each other. If a node does not hear these messages within the configured amount of time, a node timeout occurs, resulting in a TOC.

2

Planning and Documenting an OPS Cluster

Building the OPS configuration on HP-UX starts with a planning phase in which you gather and record information about all the hardware and software components of the configuration. Planning begins with a simple list of hardware and network components. As the installation and configuration continue, the list is extended and refined.

During the actual creation of the cluster, the planning worksheets provide the values that are input into SAM or the configuration file. Note that the same high availability options in SAM can be used both for the actual cluster creation as well as the planning stage.

After hardware installation, you can use SAM or a variety of HP-UX commands to obtain more information about the cluster you are building. After installing MC/LockManager, you can step through the SAM high availability configuration options to obtain a list of legal values to use in filling out the worksheets on cluster configuration. To do this without actually building the OPS cluster, use the following procedure *after the software has been installed*:

- Log on to one system as root.
- Invoke SAM.
- Select Clusters from the main menu, then choose the High Availability Clusters option.
- Step through the configuration process as if you were building an actual OPS cluster.
- SAM will provide you with lists of legal values to use in filling out the "Cluster Configuration", "DLM Configuration", and "Package Configuration" worksheets.
- When prompted to verify that you want to copy the configuration to all the nodes in the cluster, reply No to cancel the configuration.

When planning is complete, you can use SAM to actually implement the configuration. In this case, when prompted to verify that you want to copy the configuration to all the nodes in the cluster, reply Yes.

This chapter assists you in the following areas:

- Hardware Planning
- Power Supply Planning
- Shared Logical Volume Planning
- Physical Volume Planning
- Cluster Manager Planning
- Distributed Lock Manager Planning
- Package Configuration Planning

To assist in record-keeping, the description of each planning step in this chapter is accompanied by a worksheet on which you can optionally record the parameters and other data relevant for successful Oracle Parallel Server setup and maintenance. As you go through each step, record all the important details of the configuration so as to document your production system. Subsequent chapters describe installation, configuration, and maintenance tasks in detail. During the actual configuration of the OPS system, you will use the information from these worksheets. A complete set of blank worksheets is in Appendix B.

| **Note** | Planning and installation overlap considerably, so you may not be able to complete the worksheets entirely before you proceed to the actual configuration. In cases where the worksheet is incomplete, fill in the missing elements to document the system as you proceed with the configuration. |

Refer also to the chapter entitled "Planning your Installer Session" in the *Oracle Parallel Server for HP 9000 Series 800 Installation and Configuration Guide* for specific tips on installing your Oracle software and setting up the Oracle database.

Hardware Planning

Hardware planning requires examining the physical hardware itself. One useful procedure is to sketch the hardware configuration in a diagram that shows adapter cards and busses, cabling, disks and peripherals. Indicate which device adapters occupy which slots, and calculate the bus address for each adapter. Update the details as you do the actual configuration (described in Chapters 3 and 4).

Note	The process of configuring a cluster is easier if nodes as far as possible have identical hardware configuration (i.e, interface cards on different nodes have the same hardware I/O path, and a given disk on a shared bus has the same device file name on different nodes).

In addition to creating a diagram like the one described above, be sure to record the characteristics of the hardware on the Hardware Worksheet. *Use one form for each node.* This form has three parts:

- Node Information
- LAN and RS232 Information
- Disk I/O Information

Node Information

Node information includes the basic characteristics of the S800 systems you are using in the cluster. On the worksheet, include the following items:

S800 Series Number	Enter the series number, e.g., 892.
Host Name	Enter the name to be used on the system as the host name.
Memory Capacity	Enter the memory in MB.
Number of I/O slots	Indicate the number of slots.

LAN and RS232 Information

MC/LockManager monitors LAN interfaces as well as serial lines (RS232), which run heartbeat only.

LAN Information

Select LAN hardware that meets MC/LockManager requirements while providing the desired degree of availability. A minimum of one LAN interface per node is required, but additional interfaces may be used to provide greater availability and throughput. For each IP subnet to which the node is to be connected, obtain an IP address from your network administrator.

On the worksheet, enter the following for each LAN interface:

Subnet Name   Enter the name of the subnet. E.g, Blue, Green.

Interface Name   Enter the name of the LAN card used by Node 1 to access the subnet. This name is shown by `lanscan` after you install the card.

IP Address   Enter this node's host IP address intended to be used on this interface. The IP address is a string of digits separated with periods in the form 'nnn.nnn.nnn.nnn'. If the interface is a standby, enter 'Standby.'

Kind of LAN Traffic   Identify the purpose of the subnet. Valid types include the following:

- Heartbeat
- DLM
- Client Traffic
- Standby

Label the list to show the subnets that belong to a bridged net.

Information from this section of the worksheet is used in creating the subnet groupings and identifying the IP addresses in the configuration steps for the cluster manager and distributed lock manager.

RS232 Information

If you plan to configure a serial line (RS232) to carry heartbeat, you need to determine the serial device file that corresponds with the serial port on each node.

1. If you are using a MUX panel, make a note of the system slot number that corresponds to the MUX and also note the port number that appears next to the selected port on the panel.
2. On each node, use `ioscan` to display hardware addresses and device file names. For example:

       # `ioscan -fnC tty`

   This lists all the device files associated with each RS232 device on a specific node.
3. Select the file name corresponding to the slot number and port number you used when connecting the serial (RS232) cable to the node. For example, if you used port 0 on the MUX panel on this node, the device file might be /dev/tty0p0.
4. Once you have identified the device files, verify your connection as follows. Assume that node 1 uses /dev/tty0p0, and node 2 uses the same device file name, /dev/tty0p0.
   - Test the connection as follows. From a terminal on node 1, issue the following command:

         # `cat < /dev/tty0p0` (Return)
   - From a terminal on node 2, issue the following command:

         # `cat /etc/passwd > /dev/tty0p0` (Return)

     The contents of the password file should be displayed on the terminal on node 1.
5. On the worksheet, enter the following:

Node Name	Name of the node
RS232 Device File	Enter the device file name corresponding to a serial interface on each node. This parameter is known as SERIAL_DEVICE_FILE in the cluster ASCII configuration file.

Setting SCSI Addresses

SCSI standards define priority according to SCSI address. To prevent controller starvation on the node, the SCSI interface cards must be configured at the highest priorities. Therefore, when configuring a highly available cluster, you should give nodes the highest priority SCSI addresses, and give disks addresses of lesser priority.

High priority starts at seven, goes down to zero, and then goes from 15 to eight. Therefore, seven is the highest priority and eight is the lowest priority. For example, if there are two nodes in the cluster, and the two systems will share a string of disks, then the SCSI address must be uniquely set on the interface cards in both systems, and must be high priority addresses. So the addressing for the systems and disks would be as follows:

Table 2-1. SCSI Addressing in Cluster Configuration

System or Disk	Host Interface SCSI Address
Primary System A	7
Primary System B	6
Disk #1	5
Disk #2	4
Disk #3	3
Disk #4	2
Disk #5	1
Disk #6	0
Disk #7	15
Disk #8	14
Etc	13 - 8

Disk I/O Information for Shared Disks

This part of the worksheet lets you indicate where disk device adapters are installed. Use the same I/O slot on both systems for the disk adapter card if this is possible.

Enter the following items on the worksheet for each disk connected to each disk device adapter on the node:

Bus Type   Indicate the type of bus. F/W SCSI is supported.

Slot Number   Indicate the slot number in which the card is inserted. Use the even number printed at the bottom of the slot.

Hardware Path   Enter the hardware path, which will be seen on the system later when you use `ioscan` to display hardware.

Device File   Enter the device file name for the disk, as displayed in the output of the `ioscan` command.

Information from this section of the worksheet is used in creating the mirrored disk configuration using Logical Volume Manager. In addition to the information on the worksheet, you should attach printouts of the output from the following commands:

```
lanscan
netstat -i -n
ioscan -fnC disk
diskinfo disk
```

The commands should be issued from *both nodes* after installing the hardware and rebooting the system. The information will be useful when doing cluster and logical volume configuration.

Hardware Planning Worksheet

```
            HARDWARE WORKSHEET                           Page ___ of ____
==============================================================================
Node Information:

S800 Host Name ____manatee_____     S800 Series No ____892_____

Memory Capacity ____48 MB_____     Number of I/O Slots __12_____
==============================================================================
LAN Information:

Name of           Name of             IP                  Traffic
Subnet __Blue___  Interface __lan0____ Addr__35.12.16.10__ Type   __DLM, HB__

Name of           Name of             IP                  Traffic
Subnet __Red____  Interface __lan1____ Addr__35.12.15.12__ Type   __Clients, HB__

Name of           Name of             IP                  Traffic
Subnet _____  Interface __lan2____ Addr__Standby_____ Type   __Standby__


==============================================================================
Serial Heartbeat Interface Information:

Node Name _____node1_____  RS232 Device File _____/dev/tty0p0_____

Node Name _____node2_____  RS232 Device File _____/dev/tty0p0_____


==============================================================================
Disk I/O Information for Shared Disks:

                  Hardware              Device
Bus Type _SCSI_   Path    __32.1.0_____ File Name _/dev/rdsk/c0t1d0_

                  Hardware              Device
Bus Type _SCSI_   Path    __32.2.0_____ File Name _/dev/rdsk/c0t2d0_

                  Hardware              Device
Bus Type _____   Path    _____ File Name _____



Attach a printout of the output from "ioscan -fnC disk" after installing
disk hardware and rebooting the system.  Mark this printout to indicate
which physical volume group each disk belongs to.
```

Figure 2-1. Sample Worksheet for Hardware Planning

```
Class     I  H/W Path  Driver S/W State H/W Type  Description
=============================================================
disk      1  32.1.0     disc4  CLAIMED    DEVICE     HP 7937
                       /dev/dsk/c0t1d0   /dev/rdsk/c0t1d0
disk      2  32.2.0     disc4  CLAIMED    DEVICE     HP 7937
                       /dev/dsk/c0t2d0   /dev/rdsk/c0t2d0
disk      4  32.4.0     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t4d0   /dev/rdsk/c0t4d0
disk      5  32.4.1     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t4d1   /dev/rdsk/c0t4d1
disk      6  32.4.2     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t4d2   /dev/rdsk/c0t4d2
disk      7  32.4.3     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t4d3   /dev/rdsk/c0t4d3
disk      8  32.4.4     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t4d4   /dev/rdsk/c0t4d4
disk      9  32.5.0     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t5d0   /dev/rdsk/c0t5d0
disk     10  32.5.1     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t5d1   /dev/rdsk/c0t5d1
disk     11  32.5.2     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t5d2   /dev/rdsk/c0t5d2
disk     12  32.5.3     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t5d3   /dev/rdsk/c0t5d3
disk     13  32.5.4     disc4  CLAIMED    DEVICE     HP 2251
                       /dev/dsk/c0t5d4   /dev/rdsk/c0t5d4
```

Figure 2-2. Sample Output from `ioscan -fnC disk`

Power Supply Planning

To provide a high degree of availability in the event of power failure, systems should be equipped with uninterruptible power supplies (UPS). In order to eliminate single points of failure, use a separate UPS at least for each node and for the cluster lock disk, which is a physical volume in the cluster lock volume group, and ensure that disks which are members of different physical volume groups are connected to different power supplies. This last rule ensures that disk mirroring is between physical disks that are connected to different power supplies as well as being on different I/O busses.

To prevent confusion, it is suggested that you label each hardware unit and power supply unit clearly with a different unit number. Indicate on the Power Supply Worksheet the specific hardware units you are using and the power supply to which they will be connected. Enter the following items on the worksheet:

S800 Host Name   Enter the host name for each node.

Disk Unit   Enter the disk drive unit number for each disk.

Tape Unit   Enter the tape unit number for each backup
                        device.

Other Unit   Enter the number of any other unit.

Power Supply   Enter the power supply unit number of the UPS to
                        which the host or other device is connected.

Use this worksheet to ensure that:

■ Node circuits are different from disk power circuits.
■ mirrored disks are on different power circuits.
■ the physical volume for the lock volume group is on a different power circuit
  than the node's.

Be sure to follow UPS and cabinet power limits as well as node power limits.

Power Supply Worksheet

```
            POWER SUPPLY WORKSHEET                      Page ___ of ____
================================================================================
SPU Power:

S800 Host Name __node1_____      Power Supply _____1_____

S800 Host Name __node2_____      Power Supply _____2_____


================================================================================
Disk Power:

Disk Unit _____A_____      Power Supply _____1_____

Disk Unit _____B_____      Power Supply _____2_____

Disk Unit _____C_____      Power Supply _____3_____

Disk Unit _____D_____      Power Supply _____4_____

Disk Unit _____      Power Supply _____

Disk Unit _____      Power Supply _____


===============================================================================
Tape Backup Power:

Tape Unit _____      Power Supply _____

Tape Unit _____      Power Supply _____


===============================================================================
Other Power:

Unit Name _____      Power Supply _____

Unit Name _____      Power Supply _____
```

Figure 2-3. Sample Worksheet for Power Supplies

Shared Logical Volume Planning

Storage capacity for the Oracle database must be provided in the form of logical volumes located in shared volume groups. The Oracle software requires an Oracle control file, several log files for each Oracle instance, and files for the database itself. For all these files, OPS uses HP-UX *raw logical volumes*, which are located in volume groups that are shared between the nodes in the cluster. High availability is achieved by mirroring all the logical volumes that are created within a volume group to a different disk on a separate I/O bus. The technique used to achieve this mirroring is called the PVG-strict mirroring policy, which uses *physical volume groups* to divide a volume group's disk resources into separate sets for mirroring.

The following paragraphs show how to plan appropriate volume groups, physical volume groups, and logical volumes for your OPS demo database, which is created by the Oracle *installer* software. If you do not wish to install the demo database, use the same worksheets to define an appropriate set of volume groups, physical volume groups, and logical volumes for your development or production system.

Note   If you are planning to run packages, you will need to plan a volume group infrastructure for those volume groups that will be used by packages. These volume groups must be separate from volume groups that contain the OPS files. The basic methodology for planning volume groups for packages is the same as for shared volumes, but there are a few differences to note. Please see the following discussion "Package Configuration Planning" for details.

Planning Volume Groups

You should plan the number of OPS volume groups based on the availability of disk resources and on your desire to subdivide your disk resources for ease of maintenance or for other reasons. Although the examples shown in this section use a single volume group, /dev/vg_ops, you may wish to create more than one volume group. For example, you may want to use one volume group per tablespace.

The default number of volume groups allowed is 10. If your planned configuration will exceed this number, you need to change the MAX_VGS parameter in the /stand/system file. For the changed parameter to take effect, you need to regen the kernel and reboot the system. See the *HP-UX System Administration Tasks* manual for information on changing kernel parameters.

Planning Physical Volumes and Physical Volume Groups

In order to create a volume group, you must identify the physical volumes that will hold its data. To do this, examine the list of disks in the output of the `ioscan -fnC disk` command (attached to the hardware worksheet). Assuming you have disks attached to two different busses, identify which disks to be used for OPS are connected to which different I/O bus. Assign all the disks from one bus to one physical volume group, and assign all the disks from the other bus to a second physical volume group.

Note   If you are using more than one volume group, each volume group should have its own physical volume groups. A disk can only belong to one volume group; therefore, it can only belong to one physical volume group.

Planning Logical Volumes

A single volume group can hold up to 255 logical volumes, and the largest logical volume that can be employed as a raw file for OPS data is 4 GB. Thus, if your data is larger than 4 GB, you *must* use more than one logical volume. However, you *may* use as many as 255 logical volumes per volume group, even when the total size of your data is much less than 4 GB. For the OPS configuration, define enough logical volumes in appropriate sizes for the data and logs you need.

OPS Physical Volume Planning Worksheet

Fill out the OPS Physical Volume worksheet (Figure 2-4) to assist in adding the correct physical volumes to particular volume groups in the `vgcreate` command.

Logical Volume Planning Worksheet

Fill out the Logical Volume worksheet (Figure 2-5) to provide logical volume names for OPS logical volumes that you will create with the `lvcreate` command. The Oracle DBA and the HP-UX system administrator should prepare this worksheet together. Create entries for shared volumes only. For each logical volume, enter the full pathname of the raw logical volume device file. Be sure to include the desired size in MB.

```
OPS PHYSICAL VOLUME WORKSHEET                          Page ___ of ____
==================================================================================

Volume Group Name:        ____/dev/vg_ops_____

Name of First Physical Volume Group:  _____pvgops1_____

Physical Volume Name:     _____/dev/dsk/c3t2d0_____

Physical Volume Name:     _____/dev/dsk/c6t2d0_____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Name of Second Physical Volume Group: _____pvgops2_____

Physical Volume Name:     _____/dev/dsk/c4t2d0_____

Physical Volume Name:     _____/dev/dsk/c7t2d0_____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____

Physical Volume Name:     _____
```

Figure 2-4.
Sample Worksheet for OPS Physical Volumes and Physical Volume Groups

```
            OPS LOGICAL VOLUME WORKSHEET                 Page ___ of ____
==============================================================================
                          RAW LOGICAL VOLUME NAME           SIZE (MB)

Oracle Control File 1:  ___/dev/vg_ops/ropsctl1.ctl_____1_____

Oracle Control File 2:  ___/dev/vg_ops/ropsctl2.ctl_____1_____

Oracle Control File 3:  ___/dev/vg_ops/ropsctl3.ctl_____1_____

Instance 1 Redo Log 1:  ___/dev/vg_ops/rops1log1.log_____1_____

Instance 1 Redo Log 2:  ___/dev/vg_ops/rops1log2.log_____1_____

Instance 1 Redo Log 3:  ___/dev/vg_ops/rops1log3.log_____1_____

Instance 1 Redo Log:    _____

Instance 1 Redo Log:    _____

Instance 2 Redo Log 1:  ___/dev/vg_ops/rops2log1.log_____1_____

Instance 2 Redo Log 2:  ___/dev/vg_ops/rops2log2.log_____1_____

Instance 2 Redo Log 3:  ___/dev/vg_ops/rops2log3.log_____1_____

Instance 2 Redo Log:    _____

Instance 2 Redo Log:    _____

Data: System           ___/dev/vg_ops/rsystem.dbf_____25_____

Data: Rollback         ___/dev/vg_ops/rrollback.dbf_____4_____

Data: Temp             ___/dev/vg_ops/rtemp.dbf_____1_____

Data: Users            ___/dev/vg_ops/rusers.dbf_____1_____

Data: Tools            ___/dev/vg_ops/rtools.dbf_____15_____

Data: User data        ___/dev/vg_ops/opsdata1.dbf_____200_____

Data: User data        ___/dev/vg_ops/opsdata2.dbf_____200_____

Data: User data        ___/dev/vg_ops/opsdata3.dbf_____200_____
```

Figure 2-5. Sample Worksheet for Logical Volumes in Shared Volume Groups

Cluster Manager Planning

For the operation of the cluster manager (CM), you need to define a set of cluster parameters. These are stored in the binary cluster configuration file, which is located on all nodes in the cluster. These parameters can be entered by using SAM or by editing the cluster configuration template file created by issuing the `cmquerycl` command, as described in the chapter "Building an OPS Cluster Configuration." Cluster planning also includes establishing the parameters for the cluster manager software. The following parameters must be identified:

Cluster Name	The name of the cluster as it will appear in the output of `cmviewcl` and other commands, and as it appears in the cluster configuration file.
Cluster Nodes	The hostname of each system that will be a node in the cluster.
Volume Group	The name of a volume group that will be activated by packages and whose disks are attached to two nodes in the cluster. Such disks are considered cluster bound. In the ASCII cluster configuration file, this parameter is VOLUME_GROUP. Volume groups listed under this parameter are marked for activation in exclusive mode.
DLM Volume Group	The name of a volume group whose disks are attached to at least two nodes in the cluster; the disks will be accessed by more than one node at a time with concurrency control provided by the Distributed Lock Manager. Such disks are considered cluster bound. In the ASCII cluster configuration file, this parameter is DLM_VOLUME_GROUP. Volume groups listed under this parameter are marked for activation in shared mode.
Heartbeat Subnet Address	The IP address of the subnet that will carry the cluster heartbeat. Note that heartbeat addresses must be on the same subnet on each node. Up to seven subnets can be identified for heartbeats.

RS232 Heartbeat Network	The name of the device file that corresponds to serial (RS232) port that you have chosen on each node. Specify this parameter when you are using RS232 as a heartbeat line.
	In the ASCII cluster configuration file, this parameter is SERIAL_DEVICE_FILE.
Monitored Non-Heartbeat Subnet	The IP address of each monitored subnet that does not carry the cluster heartbeat. You can identify any number of subnets to be monitored. If you want to separate heartbeat messages from DLM messages, define a monitored non-heartbeat subnet here, then choose it when entering DLM Internal Parameters.
Lock Volume Group	The volume group on which a cluster lock is written. Identifying a cluster lock volume group is essential in a two-node cluster. If you are creating two cluster locks, enter the volume group name or names for both locks.
Physical Volumes	The name of the physical volume within the Lock Volume Group that will have the cluster lock written on it. Enter the physical volume name as it appears on both nodes in the cluster (the same physical volume may have a different name on each node). If you are creating two cluster locks, enter the physical volume names for both locks.
Disk Unit No.	Enter the number of the disk drive unit on which the physical volume is located.
Power Supply No.	Enter the number of the power supply to which the physical volume is connected.
Heartbeat Interval	The normal interval between the transmission of heartbeat messages from one node to the other in the cluster. Enter a number of seconds. Default: 1 second.

Node Timeout	The time after which a node may decide that the other node has become unavailable and initiate reconfiguration. Increasing or decreasing this value will impact failover time. Enter a number of seconds. Default: 2 seconds. Minimum is 2 * (Heartbeat Interval). If your node is configured for local Ethernet switching, you may need to increase the value to 8 seconds. See the following section for details.
Network Polling Interval	The frequency at which the networks configured for LockManager are checked. The current default is 2 seconds. Thus every 2 seconds, the cluster manager polls each network interface to make sure it can still send and receive information. Changing this value can effect how quickly a network failure is detected.
Autostart Delay	The time during which a node may join the cluster during an automatic cluster startup. (This occurs typically after a site-wide power failure.) All nodes wait this amount of time for other nodes to begin startup before the cluster completes the operation. The time should be selected based on the slowest boot time in the cluster. Enter a number of seconds. Default: 600 seconds.

If Your Node has Local Ethernet Switching

If your node is configured for local switching with Ethernet cards, you may need to change the default setting for the *Node Timeout* parameter in the cluster configuration file. With the default of 2 seconds, the node may timeout before the local switching is completed; this can cause an unnecessary re-formation of the cluster. To avoid this situation, increase the *Node Timeout* parameter to 8 seconds. (Note that this is only necessary for Ethernet, and not for other types of LAN.)

Cluster Manager Worksheet

Fill out this worksheet in cooperation with your LAN administrator prior to installing OPS software. The LAN administrator may suggest timing values that differ from the defaults.

```
              CLUSTER MANAGER CONFIGURATION WORKSHEET
================================================================================
Name and Nodes:
================================================================================
Cluster Name: ___opscluster_____

Node Names: ____node1_____    ____node2_____

DLM Volume Groups: _____

Volume Groups (for packages):_____
================================================================================
Subnets:
================================================================================
Heartbeat Subnet: ___15.13.168.0_____

Monitored Non-heartbeat Subnet: _____15.12.172.0___

Monitored Non-heartbeat Subnet: _____
================================================================================
Cluster Lock Volume Groups and Volumes:
================================================================================
First Lock Volume Group:    |        Physical Volume:
                            |
    __/dev/vg_ops___        |   Name on Node 1: _/dev/dsk/c1t2d0__
                            |
                            |   Name on Node 2: __/dev/dsk/c1t2d0_
                            |
                            |   Disk Unit No: ___1_____
                            |
                            |   Power Supply No: ___1_____
                            |
================================================================================
Timing Parameters:
================================================================================
Heartbeat Interval: _1 sec_
================================================================================
Node Timeout: _2 sec_
================================================================================
Network Polling Interval: _15 sec_
================================================================================
Autostart Delay: _600 sec_
```

Figure 2-6. Sample Worksheet for Cluster Manager Configuration

Distributed Lock Manager Planning

For operation of the distributed lock manager (DLM) software, you must define
two sets of parameters—cluster interface parameters and DLM lock database
parameters. The DLM lock database parameters relate to the number of
Oracle resources and the number of Oracle processes in the OPS configuration;
these values should be chosen in consultation with the Oracle DBA.

Cluster Interface Specific DLM Parameters

Cluster-specific DLM parameters are stored along with other cluster
information in the binary cluster configuration file, which is located on all
nodes in the cluster. These parameters can be entered by using SAM or
by editing the cluster configuration template file created by issuing the
`cmquerycl` command, as described in the chapter "Building an OPS Cluster
Configuration."

Appropriate values must be identified for the following DLM cluster interface
parameters:

DLM Enabled   When set to YES, the DLM starts in the cluster
                             when the cluster starts or reboots. Set to NO if
                             you wish not to start up OPS/DLM when the
                             cluster is started. Default: YES.

Reconfiguration Timeout   The number of seconds that the Cluster Manager
                             should wait for the DLM to start or reconfigure
                             before assuming the failure of the DLM. Default:
                             60 seconds.

Ping Interval   Interval at which the cluster manager sends
                             messages to the DLM to check the status of its
                             health. Default: 10 seconds.

Ping Timeout   Time after which the cluster manager assumes that
                             the DLM is not active. Default: 30 seconds.

DLM Connect Timeout   The upper bound on time available for the DLM to
                             initialize its shared memory on startup. Default:
                             30 seconds.

DLM Halt Timeout	The upper bound on time available to execute the OPS halt scripts. Default: 180 seconds.
Communication Fail Timeout	Time after which the cluster manager assumes that no reconfiguration will take place and takes action on a DLM communication failure. Default: 120 seconds.

Distributed lock manager planning can be done using the SAM high availability options, which let you display defaults or lists of acceptable values for the parameters listed above. Enter the appropriate values shown in the SAM display onto your DLM configuration worksheet (Figure 2-7). If you are unsure of what value to use for a parameter, start with the default.

DLM Internal Parameters

DLM internal parameters are stored in the binary DLM configuration file, which is located on all nodes in the cluster. These parameters can be entered by using SAM or by editing the DLM configuration template file created by issuing the `dlmquery` command, as described in the chapter "Building an OPS Cluster Configuration." The defaults are sufficient for the Oracle demo database, but you should adjust these parameters according to the size of your development or production system.

Appropriate values must be identified for the following:

Cluster Name	This is the same as the cluster name you use in cluster manager configuration.
Node Name(s)	These are the same as the node names you use in cluster manager configuration.
Resources	This is the total number of distributed locks for which memory must be allocated in an OPS on HP-UX system. The default is 6000.
	Use the following formula (which includes Oracle parameters) to approximate the number of resources required for your system. Refer to the *Oracle Parallel Server Administrator's Guide* for a description of these parameters.

```
      GC_DB_LOCKS + (GC_SEGMENTS*9)
   + (GC_ROLLBACK_LOCKS+1)*GC_ROLLBACK_SEGMENTS
   + (ENQUEUE_RESOURCE)
```

Set the Resources parameter to the greater of this value or the default.

Locks
The size of the DLM lock database. This is a value based on an Oracle configuration parameter known as DLM_RSRCS. The default is 2*(*Resources*) or 12000.

Processes
The maximum number of DLM processes that may run on the cluster. This is roughly equivalent to the number of Oracle processes that run concurrently on both nodes. The parameter has to be changed if the sum of the Oracle PROCESSES parameters in the two instances (one for each node) exceeds 2200. The DLM default, which is 2400, will have to be increased to leave some margin for miscellaneous additional processes.

Deadlock Detection Interval
Interval at which the DLM sends messages to determine whether deadlock has occurred between nodes. Default: 3 seconds.

DLM Monitor Interval
The interval at which the DLM monitors client processes. On discovering a dead client process, the DLM carries out lock recovery. The default interval is 3 seconds. In the ASCII DLM configuration file, this parameter is known as the PROCESS_MONITORING_INTERVAL.

Subnet Address
The subnet address of the LAN used for inter-DLM messages passed between OPS instances. By default, this is the same as the subnet used for the cluster manager heartbeat messages. If you wish to separate heartbeat traffic from DLM message traffic, select a different monitored subnet address for DLM than you chose for heartbeats in the basic cluster configuration. This value is used in configuring the DLM with SAM.

DLM Node 1 IP Address   The IP address of the Node 1 interface to the DLM subnet. Must be consistent with the subnet address. By default, this is the same as the IP address used for cluster heartbeat on this node, but if you wish to separate DLM message traffic from cluster heartbeat traffic, you can specify an IP address on a different monitored subnet than the one you chose for heartbeats in the basic cluster configuration. This value is used only when configuring the DLM by editing DLM configuration files.

DLM Node 2 IP Address   The IP address of the Node 2 interface to the DLM subnet. Must be consistent with the subnet address. By default, this is the same as the IP address used for cluster heartbeat on this node, but if you wish to separate DLM message traffic from cluster heartbeat traffic, you can specify an IP address on a different monitored subnet than the one you chose for heartbeats in the basic cluster configuration. This value is used only when configuring the DLM by editing DLM configuration files.

Distributed lock manager planning can be done using the SAM high availability options, which let you display defaults or lists of acceptable values for the parameters listed above. Enter the appropriate values shown in the SAM display onto your DLM configuration worksheet (Figure 2-7). If you are unsure of what value to use for a parameter, start with the default.

Distributed Lock Manager (DLM) Configuration Worksheet

Fill out this worksheet in cooperation with your HP-UX system administrator
and Oracle database administrator.

```
                      DLM CONFIGURATION WORKSHEET
     ===============================================================================
     Cluster-Specific Parameters:

           DLM Enabled:  ____YES_____

           Reconfig Timeout: __60 sec_____

           Ping Interval: __10 sec_____

           Ping Timeout: __30 sec_____

           DLM Connect Timeout: _ 30 sec_____

           DLM Halt Timeout: __180 sec_____

           Communication Fail Timeout: _120 sec_____

     ===============================================================================
     Internal DLM Parameters:

           Cluster Name: ___opscluster_____

           Node Name(s): ___node1, node2_____

           Resources: __6000_____

           Locks: __12000_____

           Processes: _2400_____

           Deadlock Detection Interval _3 sec_____

           DLM Monitor Interval _3 sec_____

           Subnet Address: _192.6.143.0_____

           Node 1 IP Address: __192.6.143.30____

 .         Node 2 IP Address: __192.6.143.31____
```

Figure 2-7. Sample Worksheet for DLM Configuration

Package Configuration Planning

Planning the package involves assembling information about each group of highly available services. Some of this information is package configuration data, and some is package control script data.

Logical Volume and Filesystem Planning for Packages

Like OPS, packages are configured to use cluster-bound volume groups (those accessible by more than one node in the cluster). Packages, which contain high availability applications, services, and data use separate volume groups from OPS. When a node fails, the volume group containing the data for the package of the failed node is deactivated on the failed primary node and activated in *exclusive* mode on the adoptive node. In order to do this, you have to configure the volume groups so that they can be transferred from the failed node to the adoptive node.

Volume groups configured for packages can contain file systems. When the package moves from one node to another, it must be able to access data residing on the same disk as on the previous node. This is accomplished by activating the volume group in exclusive mode and mounting the file system that resides on it.

As part of planning, you need to answer the following:

- What volume groups are needed?
- How much disk space is required, and how should this be allocated in logical volumes?
- What is the relocatable IP address of each package?
- What file systems need to be mounted for each package?
- If a package moves to an adoptive node, what effect will its presence have on performance?

Create a list by package of volume groups, logical volumes, and file systems. Indicate which nodes need to have access to the same filesystems at different times. Enter the information in the package configuration worksheet.

It is recommended that you use volume group and logical volume names other than the default names (vg01, vg02 or lvol1, lvol2, etc.). Choosing names that represent the high availability applications that they are associated with will simplify cluster administration.

Details about creating, exporting, and importing volume groups in MC/LockManager are given in the chapter on "Building an OPS Cluster Configuration."

Choosing Switching and Failover Parameters

Table 2-2 describes different types of package failover behavior and the parameters that determine each behavior as set in SAM or in the ASCII package configuration file.

Table 2-2. Package Failover Behavior

Switching Behavior	Parameters in SAM	Parameters in Package Configuration File
A package IP address switches to a standby LAN card transparently on LAN card failure	■ Automatic Switching set to Enabled for the package (Default)	■ `NET_SWITCHING_ENABLED` set to YES for the package (Default)
A package switches normally after detection of a failure. The package's halt script is run before the switch takes place (Default).	■ Package Failfast set to Disabled. (Default) ■ Service Failfast set to Disabled for all services. (Default) ■ Automatic Switching set to Enabled for the package. (Default)	■ `NODE_FAIL_FAST_ENABLED` set to NO. (Default) ■ `SERVICE_FAIL_FAST_ENABLED` set to NO for all services. (Default) ■ `PKG_SWITCHING_ENABLED` set to YES for the package. (Default)
All packages on the node switch following a TOC on the node when a specific service fails. The packages' halt scripts are not run.	■ Package Failfast set to Disabled ■ Service Failfast set to Enabled for a specific service ■ Automatic Switching set to Enabled for all packages.	■ `NODE_FAIL_FAST_ENABLED` set to NO ■ `SERVICE_FAIL_FAST_ENABLED` set to YES for a specific service. ■ `PKG_SWITCHING_ENABLED` set to YES for all packages.
All packages switch following a TOC on the node when any service fails.	■ Package Failfast set to Disabled. ■ Service Failfast set to Enabled for *all* services. ■ Automatic Switching set to Enabled for all packages.	■ `NODE_FAIL_FAST` set to NO. ■ `SERVICE_FAIL_FAST_ENABLED` set to YES for *all* services. ■ `PKG_SWITCHING_ENABLED` set to YES for all packages.
All packages switch following a TOC on the node when the run or halt script fails, that is, exits with an error other than 0 or 1.	■ Package Failfast set to Enabled. ■ Automatic Switching set to Enabled for all packages.	■ `NODE_FAIL_FAST` set to YES. ■ `PKG_SWITCHING_ENABLED` set to YES for all packages.

Package Configuration File Parameters

Prior to generation of the package configuration file, assemble the following package configuration data. The parameter names given below are the names that appear in SAM. The names coded in the ASCII cluster configuration file appear at the end of each entry. The following parameters must be identified and entered on the worksheet *for each package*:

Package Name   The name of the package. The package name must be unique in the cluster. It is used to start, stop, modify, and view the package.

Node Name   The names of primary and alternate nodes for the package, e.g., node1 and node2. The order in which you specify the node names is important. First list the primary node name, and then the adoptive node name. Transfer of control of the package will occur to the next adoptive node name listed in the package configuration file.

Control Script Pathname   Enter the full pathname of the package control script. It is recommended that you use the same script as both the run and halt script. This script will contain both your package run instructions and your package halt instructions. When the package starts, its run script is executed and passed the parameter 'start'; similarly, at package halt time, the halt script is executed and passed the parameter 'stop'.

In the ASCII package configuration file, this parameter maps to the two separate parameters named RUN_SCRIPT and HALT_SCRIPT. Use the name of the single control script as the name of the RUN_SCRIPT and the HALT_SCRIPT in the ASCII file.

If you wish to separate the package run instructions and package halt instructions into separate scripts, the package configuration file allows allows you to do this by naming two

separate scripts. However, under most conditions, it is simpler to combine your run and halt instructions into a single package control script and repeat its name for both the `RUN_SCRIPT` and the `HALT_SCRIPT`.

Note	If you choose to write separate package run and halt scripts, be sure to include identical configuration information (such as node names, IP addresses, etc.) in both scripts.

Ensure that this script is executable.

Run Script Timeout and Halt Script Timeout

If the script has not completed by the specified timeout value, LockManager will terminate the script. The default is 0, or no timeout.

If the timeout is exceeded:

- Control of the package will not be transferred.
- The run or halt instructions will not be run.
- Global switching will be disabled.
- The current node will be disabled from running the package.

In the ASCII package configuration file, this parameter is called `RUN_SCRIPT_TIMEOUT` and `HALT_SCRIPT_TIMEOUT`. The default for both is 0 or `NO_TIMEOUT`.

Service Name

Enter a unique name for each service.

In the ASCII package configuration file, this parameter is called `SERVICE_NAME`. Define one `SERVICE_NAME` entry for each service.

Service Fail Fast

Enter Enabled or Disabled for each service. This parameter indicates whether or not the failure of a service results in the failure of a node. If the parameter is set to Enabled, in the event of a service failure, LockManager will halt the node

on which the service is running with a TOC. The default is Disabled.

In the ASCII package configuration file, this parameter is `SERVICE_FAIL_FAST_ENABLED`, and possible values are YES and NO. The default is NO. Define one `SERVICE_FAIL_FAST_ENABLED` entry for each service.

Service Halt Timeout

In the event of a service halt, LockManager will first send out a SIGTERM signal to terminate the service. If the process is not terminated, LockManager will wait for the specified timeout before sending out the SIGKILL signal to force the process termination. Default is 300 seconds (5 minutes).

In the ASCII package configuration file, this parameter is `SERVICE_HALT_TIMEOUT`. Define one `SERVICE_HALT_TIMEOUT` entry for each service.

Subnet

Enter the IP subnet that is to be monitored for the package.

In the ASCII package configuration file, this parameter is called `SUBNET`.

Automatic Switching

Enter Enabled or Disabled. In the event of a failure, this permits LockManager to transfer the package to an adoptive node. The default is Enabled.

In the ASCII package configuration file, this parameter is called `PKG_SWITCHING_ENABLED`, and possible values are YES and NO. The default is YES.

If your cluster is using packages containing applications that access the OPS database, this parameter should be set to NO. See "Configuring Packages that Access the OPS Database" in Chapter 4 following for more information.

Local Switching	Enter Enabled or Disabled. In the event of a failure, this permits LockManager to switch LANs locally, that is, transfer to a standby LAN card. The default is Enabled.
	In the ASCII package configuration file, this parameter is called `NET_SWITCHING_ENABLED`, and possible values are YES and NO. The default is YES.
Package Fail Fast Enabled	In the event of the failure of the control script itself or the failure of a subnet, if this parameter is set to Enabled, MC/LockManager will issue a TOC on the node where the control script fails. The default is Disabled.
	In the ASCII package configuration file, this parameter is called `NODE_FAIL_FAST_ENABLED`, and possible values are YES and NO. The default is NO.

Package Control Script Variables

The control script that accompanies each package must also be edited to assign values to a set of variables. The following variables must be set:

Volume Groups, Logical Volumes, and File Systems	Determine the filesystems and corresponding logical volumes within the volume groups required. Example:
	pkg1 requires /dev/vg_pkg1/lvol1 mounted on /pkg1
	Indicate the names of volume groups that are to be activated and deactivated, together with the logical volumes and file systems that are to be mounted. On starting the package, the script activates a volume group in exclusive mode, and it may mount logical volumes onto file systems. At halt time, the script unmounts the file systems and deactivates each volume group. All volume groups must be accessible on each target node.

	In the ASCII package control script, these variables are arrays, as follows: `VG`, `LV`, and `FS`. For each file system (FS), you must identify a logical volume (LV). Include as many volume groups (VG's) as needed. If you are using raw files, the `LV` and `FS` entries are not needed.
IP Addresses and SUBNETs	These are the IP addresses by which a package is mapped to a LAN card. Indicate the IP addresses and subnets for each IP address you want to add to an interface card.
	In the ASCII package control script, these variables are entered in pairs. Example `IP[0]=192.10.25.12` and `SUBNET[0]=192.10.25.0`. (In this case the subnet mask is 255.255.255.0.)
Service Name	Enter a unique name for each specific service within the package. All services are monitored by the package manager. The service name, service command, and service restart parameters are entered in the package control script in groups of three. You may specify as many service names as you need. Each name must be unique within the cluster. The service name is the name used by `cmrunserv` and `cmhaltserv` inside the package control script.
	In the ASCII package control script, enter values into an array known as `SERVICE_NAME`. Enter one service name for each service.
Service Command	For each named service, enter a service command. This command will be executed through the control script by means of the `cmrunserv` command.
	In the ASCII package control script, enter values into an array known as `SERVICE_CMD`. Enter one service command string for each service.

Service Restart Parameter   Enter a number of restarts. One valid form of the parameter is `-r` *n* where *n* is a number of retries. A value of "-r 0" indicates no retries. A value of "-R" indicates an infinite number of retries. The default is 0, or no restarts.

In the ASCII package control script, enter values into an array known as `SERVICE_RESTART`. Enter one restart value for each service.

The package control script will clean up the environment and undo the operations in the event of an error.

Package Configuration Worksheet

Assemble your package configuration and control script data in a separate worksheet for each package.

```
          PACKAGE CONFIGURATION WORKSHEET                Page ___ of ___
================================================================================
Package Configuration File Data:
================================================================================

Package Name: _____pkg1_____

Nodes: _____node1_____ (Primary)

        _____node2_____

        _____

Package Run Script: __/etc/cmcluster/pkg1/control.sh____Timeout: _NO_TIMEOUT__

Package Halt Script: __/etc/cmcluster/pkg1/control.sh____Timeout: _NO_TIMEOUT__

Package Switching Enabled?  __YES_____ Network Switching Enabled?  __YES_____

Node Failfast Enabled?       ____NO_____

Control Script Data:
================================================================================

VG[0]___/dev/vg_pkg1_____LV[0]___/dev/vg_pkg1/lvol1__FS[0]__/mnt1_____

VG[1]_____LV[1]_____FS[1]_____

VG[2]_____LV[2]_____FS[2]_____


IP[0] ____15.13.171.14_____ SUBNET ___15.13.168.0_____

IP[1] _____ SUBNET _____


Service Name: __Svc1____ Run Command: __/usr/sbin/MySvc -f____Retries: "_-r 2"_

Service Fail Fast Enabled? __NO_____Service Halt Timeout ___NO_TIMEOUT_____

Service Name: _____ Run Command: _____ Retries: _____

Service Fail Fast Enabled? _____Service Halt Timeout _____
```

3

Building an OPS Cluster Configuration

The process of building an OPS cluster on HP-UX involves installing and configuring all the components on both cluster nodes in a consistent way. In brief, you install the software on both nodes, then you configure the cluster on one node (called the configuration node in this chapter) and propagate the configuration to the other node in the cluster. (It's a good idea to do all configuration on one node to simplify cluster administration.) When the cluster is running, you install the Oracle software and build the Oracle database. All the tasks described in this chapter require root permission.

Some configuration tasks can be completed using SAM (System Administration Manager). You can use the SAM High Availability options to configure the cluster manager and distributed lock manager as well as to start up the cluster on all nodes or on individual nodes. When you employ the SAM high availability options, you should be aware of the following user interface characteristics of SAM:

- SAM uses an object-action model. You select an object, then perform an action on it. The menu of actions available may vary depending on whether an object is selected or not.
- You must always deliberately select an item when choosing it from a list. Either click on the item or tab to it and press ⟨Return⟩, then select ⟨OK⟩ to choose the item or items. An item is not selected by default even though it may appear to be highlighted.
- To make more than one selection from a list with a mouse, click on the first item with the left mouse button, then click on subsequent items by holding down ⟨Ctrl⟩ and pressing the left mouse button. Finally, select ⟨OK⟩ to choose the items.

The rest of this chapter describes the following specific phases in installing and configuring your system:

- Installing the Hardware
- Preparing Your Systems
- Installing MC/LockManager
- Creating the Logical Volume Infrastructure for OPS
- Creating the Logical Volume Infrastructure for Packages
- Configuring the Cluster Manager Software
- Configuring the Distributed Lock Manager Software
- Testing the Configuration
- Creating OPS Startup and Shutdown Scripts
- Installing Oracle Parallel Server
- Starting Up Oracle Instances

Each of these phases is described in a separate section below. Some phases of configuration can only be carried out using HP-UX commands, since SAM does not support some of the options that MC/LockManager requires. Configuration steps that cannot be done in SAM are clearly marked "HP-UX Commands Only." In what follows, it is assumed that you have filled out the planning worksheets presented in the chapter "Planning and Documenting an OPS Cluster." The planning worksheets ensure that all the parameters you have chosen for the cluster configuration will be in front of you during the following steps. It is further assumed that you have already installed HP-UX 10.10.

Note	If you do not wish to install the Oracle *demo* database, you can defer cluster manager configuration, setting DLM parameters, and sharing of OPS volume groups until after the Oracle software is installed.

Refer also to the *Oracle Parallel Server for HP 9000 Series 800 Installation and Configuration Guide* for details about Oracle installation.

Installing the Hardware

The hardware installation should be sketched in a diagram like that shown in Figure 1-3. You should install both LAN and disk hardware using the appropriate cabling and using identical I/O slot configuration (if possible) in the card cages of both HP 9000 systems. You may wish to install both LAN and disk hardware before rebooting.

Installing LAN Hardware

In cooperation with the LAN administrator, install the primary and backup LAN hardware, and perform LAN configuration, specifying IP addresses for each non-standby LAN. After rebooting, use `lanscan` to ensure that LAN cards are configured correctly, and use `netstat -i -n` to display the LAN configuration on each node.

Installing the OPS Disks and Disk Interfaces

First, install the disk adapter cards, using the same I/O slots on both systems if this is possible. Then connect the OPS disks to the cables.

In the case of F/W SCSI, on a given bus, the interface cards must have the highest addresses on the bus, and they must be different from each other. See "Setting SCSI Addresses" in the "Planning" chapter for details.

For complete information on installing and configuring your disks, refer to the installation guide for the type of adapter card you are using.

After rebooting each system, use `ioscan -fnC disk` to identify the device files that are associated with the disks on both systems. Ensure that all disks are being seen by each node. A device will fail to show up in the output of `ioscan -fnC disk` if the address is set incorrectly, or if there are bad cables. Be sure to note the differences between the two nodes in the device file names and hardware paths associated with the shared disks. For example, if the same disk appears as /dev/dsk/c0t0d0 on one node and /dev/c2t0d0 on the other node, make a careful note of the correspondence on your planning worksheet.

Preparing Your Systems

Before configuring your cluster, ensure that all cluster nodes possess the appropriate security files and NTP (network time protocol) configuration.

Editing Security Files

MC/LockManager makes use of ARPA services to ensure secure communication among cluster nodes. Before installing MC/LockManager, you must identify the nodes in the cluster that permit access by the root user on other nodes. You can use SAM, or you can directly edit the /.rhosts file in the root home directory to include the names of all cluster nodes and the *root* user. The completed /.rhosts file will contain entries like the following:

```
node1 root
node2 root
```

where *node1* and *node2* are the names of the cluster nodes. The /.rhosts file should be copied to all cluster nodes.

You can also use the /etc/hosts.equiv and /var/adm/inetd.sec files to provide other levels of cluster security. For more information, refer to the HP 9000 guide *Administering ARPA Services*.

Enabling the Network Time Protocol

It is strongly recommended that you enable network time protocol (NTP) services on each node in the cluster. The use of NTP, which runs as a daemon process on each system, ensures that the system time on all nodes is consistent. The NTP services daemon, *xntpd*, should be running on all nodes before you begin cluster configuration. The NTP configuration file is /etc/ntp.conf.

For information about configuring NTP services, refer to the chapter "Configuring NTP," in the HP-UX manual, *Installing and Administering Internet Services*.

Installing MC/LockManager

Installing MC/LockManager includes updating the software and rebuilding the kernel to support high availability cluster operation for OPS. It is assumed that you have already installed HP-UX 10.10 with the separate MirrorDisk/UX product, if you are planning on doing software mirroring.

Use the following steps *for each node*:

1. Create a LockManager user account with the login name *dlm*, group *other*, and the home directory in /var/opt/dlm. This *dlm* home directory must not be in an NFS-mounted file system. The directory is used to store DLM-related log files and core dumps. Ensure that at least 20 MB of disk space is available for these files on the local file system.
2. Mount the distribution media in the tape drive or CD ROM reader.
3. Run Software Distributor, using the `swinstall` command.
4. Specify the correct input device.
5. Choose the following bundle from the displayed list:

   ```
   B5158AA                      MC/LOCKMANAGER
   ```
6. After choosing the bundle, select OK. The software is loaded.
7. Run `ioscan` on each node to validate that disks and drivers have been configured correctly.

For details about running `swinstall` and for creating new user accounts, refer to the *System Administration Tasks* manual for HP-UX 10.0 Series 800.

Creating the Logical Volume Infrastructure for OPS (HP-UX Commands Only)

After installing software components, it is necessary to create the appropriate logical volume infrastructure to support the use of shared, mirrored files within OPS. This is done by issuing Logical Volume Manager commands. The procedure described in this section uses physical volume groups to specify disk mirroring.

(If you are *not* using software mirroring on your cluster, you still need to create a logical volume infrastructure for the OPS database, but you can skip those steps or options that set up mirroring.)

While it is possible for experienced users of Logical Volume Manager to avoid the creation of physical volume groups in an OPS shared disk configuration, this guide describes their use to ensure that each logical volume is mirrored to a disk on a different I/O bus. The subject of logical volume management is discussed in detail in the section "Using HP-UX Commands to Manage Mirrors" in the "Logical Volume Manager" chapter of the *System Administration Tasks* manual for HP-UX 10.0 Series 800.

The steps described in the present section are carried out on the configuration node only.

Note	If you are planning to run packages, you will need to create a volume group infrastructure for those volume groups that are used by packages. These volume groups must be separate from volume groups that contain the OPS files. The basic methodology for creating volume groups for packages is the same as for OPS, but there are a few significant differences. Please see the following section "Creating the Logical Volume Infrastructure for Packages" for details.

Creating a Root Mirror

It is highly recommended that you use a mirrored root disk. If you are using software mirroring, use the following procedure to create a LVM root mirror. This procedure cannot be carried out with SAM. In this example and in the following commands, x and y should be replaced with the card instance and target device address of the disk you are using to mirror the root.

1. Create a bootable LVM disk to be used for the mirror.

    # `pvcreate -B /dev/rdsk/c`x`t`y`d`0 [Return]

2. Add this disk to the current root volume group.

    # `vgextend /dev/vg00 /dev/dsk/c`x`t`y`d`0 [Return]

3. Make the new disk a boot disk.

```
# mkboot /dev/rdsk/cxtyd0  [Return]
```

4. Copy the correct AUTO file into the new LIF area.

```
# mkboot -a "hpux (disk;0)/stand/vmunix" \  [Return]
    /dev/rdsk/cxtyd0  [Return]
```

5. Mirror the root and primary swap logical volumes to the new bootable disk. Ensure that all devices in vg00, such as /usr, /swap, etc., are mirrored.

The following is an example of mirroring the root logical volume:

```
# lvextend -m 1 /dev/vg00/lvol1 /dev/dsk/cxtyd0  [Return]
```

The following is an example of mirroring the primary swap logical volume:

```
# lvextend -m 1 /dev/vg00/lvol2 /dev/dsk/cxtyd0  [Return]
```

Note The root logical volume *must* be done first to ensure that it occupies the first contiguous set of extents on the new disk.

---

6. Update the boot information contained in the BDRA for the mirror copies of root and primary swap.

```
# /usr/sbin/lvlnboot -v -r /dev/vg00/lvol1  [Return]
```

```
# /usr/sbin/lvlnboot -s /dev/vg00/lvol2  [Return]
```

7. Check if the BDRA is correct.

```
# /usr/sbin/lvlnboot -R /dev/vg00  [Return]
```

8. Verify that the mirror was properly created.

```
# lvlnboot -v  [Return]
```

Building Volume Groups for OPS with LVM Commands

For each volume group you wish to create, you must first create a directory under /dev. The following is an example, using the volume group name *vg_ops* (you should use the volume group names you have entered on the OPS logical volume planning worksheet):

1. On the configuration node, issue the following command:

```
# mkdir /dev/vg_ops  [Return]
```

2. Create a control file named *group* in the directory /dev/vg_ops, as follows:

```
# mknod /dev/vg_ops/group c 64 0xhh0000  [Return]
```

The major number is always 64, and the hexadecimal minor number has the form

```
0xhh0000
```

where *hh* must be unique to the volume group you are creating. Use the next hexadecimal number that is available on your system, after the volume groups that are already configured. Use the following command to display a list of existing volume groups:

```
# ls -l /dev/*/group  [Return]
```

3. Mark the disks you wish to use as physical volumes using the pvcreate command. Be sure to use the character device file names. The following example initializes two physical disks that are on different busses:

```
# pvcreate -f /dev/rdsk/c0t2d0  [Return]
# pvcreate -f /dev/rdsk/c1t2d0  [Return]
```

The -f option is only necessary if the physical volume was previously used in some other volume group.

4. Create the volume group and specify one *physical volume group* as belonging to it. Specify also the block device file name of a physical volume on one I/O bus as belonging to the physical volume group:

```
# vgcreate -g pvgops1 /dev/vg_ops /dev/dsk/c0t2d0  [Return]
```

5. Extend the volume group to add a second physical volume group with the other physical volume (on the other I/O bus) belonging to it:

```
# vgextend -g pvgops2 /dev/vg_ops /dev/dsk/c1t2d0  [Return]
```

6. Use the vgextend command to add additional disks to the volume group, specifying the appropriate physical volume group name for each mirror copy.

Note   Logical Volume Manager records the names of physical volumes and physical volume groups in the ASCII file /etc/lvmpvg. Although each node in the cluster must have a copy of this file, the names of the physical volumes associated with the physical

volume group may be different, since the device file names for the disks may be different on different nodes.

Repeat this process for each distinct volume group you wish to create. For ease of system administration, you may wish to use different volume groups to separate logs from data and control files. Remember that each different mirrored volume group requires at least two disks.

Note The default maximum number of volume groups in HP-UX is 10. If you intend to create enough new volume groups that the total exceeds ten, you must increase the *maxvgs* system parameter and then re-gen the HP-UX kernel. In SAM, select the Kernel Configuration Area, then choose "Configurable Parameters." Maxvgs appears on the list.

Building Mirrored Logical Volumes for OPS with LVM Commands

After you create volume groups and define physical volumes for use in them, you define mirrored logical volumes for data, logs, and control files. It is recommended that you use a shell script to issue the commands described in the next sections. The commands you use for creating logical volumes vary slightly, depending on whether you are creating logical volumes for OPS redo log files or for use with Oracle data.

Creating Mirrored Logical Volumes for OPS Redo Log and Control Files

Create logical volumes for use as redo log files by selecting mirror consistency recovery. Use the same options as in the following example:

```
# lvcreate -m 1 -M n -c y -s g -n redo1.log -L 4 /dev/vg_ops  (\ Return)
```

The `-m 1` option specifies single mirroring; The `-M n` option ensures that mirror write cache recovery is set off; the `-c y` means that mirror consistency recovery is enabled; the `-s g` means that mirroring is PVG-strict, that is, it occurs between different physical volume groups; the `-n redo1.log` option lets you specify the name of the logical volume; and the `-L 4` option allocates 4 megabytes.

| **Note** | It is important that you use these the -M n and -c y options for redo log and control files. These options allow the redo log files to be resynchronized by SLVM following a system crash before Oracle recovery proceeds. If these options are not set correctly, you may not be able to continue with database recovery. |

Creating Mirrored Logical Volumes for OPS Data Files

For data files other than the redo logs or control files, choose a mirror recovery policy of "none" by disabling both mirror write caching and mirror consistency recovery. The following example shows how to create a singly mirrored logical volume for one Oracle data file.

- Issue the following command to create a logical volume for OPS system data:

  ```
  # lvcreate -m 1 -M n -c n -s g -n system.dbf  -L 28 /dev/vg_ops (Return)
  ```

  The **-m 1** option specifies single mirroring; The **-M n** option ensures that mirror write cache recovery is set off; the **-c n** means that mirror consistency recovery is disabled; the **-s g** means that mirroring is PVG-strict, that is, it occurs between different physical volume groups; the **-n system.dbf** option lets you specify the name of the logical volume; and the **-L 28** option allocates 28 megabytes.

  If the creation command is successful, the system will display messages like the following:

  ```
  Logical volume "/dev/vg_ops/system.dbf" has been successfully created
  with character device "/dev/vg_ops/rsystem.dbf"
  Logical volume "/dev/vg_ops/system.dbf" has been successfully extended
  ```

  Note that the *character* device file name (also called the raw logical volume name) is used by the Oracle DBA in building the OPS database.

Use the same procedure to create all Oracle files for all user data, and rollback segments. Use the procedure in the previous section for redo log files and control files.

Oracle Demo Database Files

The following set of files is required for the Oracle demo database which you can create during the installation process.

Required Oracle File Names for Demo Database

Logical Volume Name	LV Size (MB)	Raw Logical Volume Path Name	Oracle File Size (MB)*
opsctl1.ctl	4	/dev/vg_ops/ropsctl1.ctl	1
opsctl2.ctl	4	/dev/vg_ops/ropsctl2.ctl	1
opsctl3.ctl	4	/dev/vg_ops/ropsctl3.ctl	1
system.dbf	28	/dev/vg_ops/rsystem.dbf	25
ops1log1.log	4	/dev/vg_ops/rops1log1.log	1
ops1log2.log	4	/dev/vg_ops/rops1log2.log	1
ops1log3.log	4	/dev/vg_ops/rops1log3.log	1
rollback.dbf	8	/dev/vg_ops/rrollback.dbf	4
temp.dbf	4	/dev/vg_ops/rtemp.dbf	1
users.dbf	4	/dev/vg_ops/rusers.dbf	1
tools.dbf	16	/dev/vg_ops/rtools.dbf	15
ops2log1.log	4	/dev/vg_ops/rops2log1.log	1
ops2log2.log	4	/dev/vg_ops/rops2log2.log	1
ops2log3.log	4	/dev/vg_ops/rops2log3.log	1
opsdata1.dbf		/dev/vg_ops/ropsdata1.dbf	
opsdata2.dbf		/dev/vg_ops/ropsdata2.dbf	
opsdata3.dbf		/dev/vg_ops/ropsdata3.dbf	

* The size of the logical volume is larger than the Oracle file size because Oracle needs extra space to allocate a header in addition to the file's actual data capacity.

Create these files if you wish to build the demo database. The three logical volumes at the bottom of the table are included as additional data files, which you can create as needed, supplying the appropriate sizes. If your naming conventions require, you can include the Oracle SID and/or the database name to distinguish files for different instances and different databases.

If you are using the ORACLE_BASE directory structure, create symbolic links to the ORACLE_BASE files from the appropriate directory. Example:

```
# ln -s /dev/vg_ops/ropsctl1.ctl /u01/ORACLE/db001/ctrl01_1.ctl
```

(For more information about Oracle directories, refer to the *Oracle Server for HP 9000 Installation and Configuration Guide*.

After creating these files, set the owner to *oracle* and the group to *dba* with a file mode of 660. The logical volumes are now available on the primary node, and the raw logical volume names can now be used by the Oracle DBA.

Displaying the Logical Volume Infrastructure

To display the volume group, use the **vgdisplay** command:

```
# vgdisplay -v /dev/vg_ops (Return)
```

Review the output of this display to ensure that your volume groups and all logical volumes within them were created as you planned them. The last part of the output from **vgdisplay** shows the physical volume groups you defined. This information comes from the file /etc/lvmpvg on the configuration node. Make a copy of this file as follows for later use:

```
# cp /etc/lvmpvg /tmp/lvmpvg (Return)
```

Edit /tmp/lvmpvg, removing references to any physical volume groups other than the groups created for pvgops1 and pvgops2. Do *not* edit /etc/lvmpvg on the configuration node.

Exporting the Logical Volume Infrastructure

Before the OPS volume groups can be shared, their configuration data must be exported to other nodes in the cluster. This is done with the **vgexport** command, which you issue from the configuration node where the volume groups initially exist, and with the **vgimport** command, which you issue from the other node in the cluster. Use the following steps for each volume group that is to be shared between two nodes:

1. On the configuration node, use the **vgchange** command to deactivate the volume group:

   # vgchange -a n /dev/vg_ops [Return]

2. Create a map file containing the logical volume names for the volume group, using the following **vgexport** command. Be sure to use the **-p** option in addition to the **-m** option:

   # vgexport -p -m /tmp/vg_ops.map /dev/vg_ops [Return]

   The use of the map file ensures that the logical volumes on both nodes will have the same logical volume names.

3. Copy the map file (/tmp/vg_ops.map) to the same path (/tmp/vg_ops.map) on the other node. At the same time, copy the physical volume group file (/tmp/lvmpvg) to the same path (/tmp/lvmpvg) on the other node. This file was created during an earlier step.

4. On the other node, issue the following command:

   # mkdir /dev/vg_ops [Return]

5. Create a control file named *group* in the directory /dev/vg_ops, as in the following:

   # mknod /dev/vg_ops/group c 64 0x*hh*0000 [Return]

   The major number is always 64, and the hexadecimal minor number has the form

   0xhh0000

   where *hh* must be unique to the volume group you are creating. (If possible, use the same hexadecimal number for the volume group on both nodes.) Use the following command to display a list of existing volume groups:

   # ls -l /dev/*/group [Return]

6. Examine the /etc/lvmpvg file on the second node. If the file does not exist, create it. Insert the content of /tmp/lvmpvg copied over from the configuration node. You should be inserting information only about the two physical volume groups pvgops1 and pvgops2. If the block device names copied from the configuration node are not the same as the block devce names for the same disks on node 2, change them to the correct names for node 2. The planning worksheet should indicate what names are associated with the same disks on the different nodes.

7. Use the **vgimport** command on the second node for the same volume group that was created on the configuration node:

   # `vgimport -v -m /tmp/vg_ops.map /dev/vg_ops /dev/dsk/c0t2d0` \
   `/dev/dsk/c1t2d0` `Return`

   # `strings /etc/lvmtab` `Return`

8. Exit from the second node.

9. On the configuration node, use the **vgchange** command to reactivate the lock volume group:

   # `vgchange -a y /dev/vg_ops` `Return`

   This step ensures that the cluster lock volume group will be active on the configuration node during the cluster configuration process.

Note A volume group using an HP High Availability Disk Array cannot be designated as a cluster lock volume group. You will need to use another volume group containing an independent disk for the cluster lock.

If you are configuring your cluster to use packages, go on to the next section "Creating the Logical Volume Infrastructure for Packages". Otherwise, go to the following section "Creating Addtional Volume Groups."

Creating the Logical Volume Infrastructure for Packages

If you are creating packages, you must configure separate volume groups that can be activated each time the package starts up on a particular node. Packages activate volume groups in exclusive mode rather than shared mode, which is used for OPS volume groups. Another difference is that with package volume groups, you can define file systems that are mounted when the package starts up. By contrast, OPS data is stored on raw logical volumes, which do not use mounted file systems.

You create volume groups for packages in the same way you create volume groups for use by OPS, using the same LVM commands. The following example uses disks /dev/dsk/c5t2d0 and /dev/dsk/c6t2d0 to create a mirrored file system for use by a package. The example shows all the commands, including the commands to create a file system to be mounted on /mnt1:

1. On the configuration node, create physical volumes from the disks:

   ```
   # pvcreate -f /dev/rdsk/c5t2d0 [Return]
   # pvcreate -f /dev/rdsk/c6t2d0 [Return]
   ```

2. Use the following commands to create a volume group called vg_pkg1:

   ```
   # mkdir /dev/vg_pkg1 [Return]
   # mknod /dev/vg_pkg1/group c 64 0xhh0000 [Return]
   ```

   Replace the minor number hh with the next available hexadecimal minor group number available on your system.

3. Create the volume group with PVG strict mirroring of disks:

   ```
   # vgcreate -g bus1 /dev/vg_pkg1 /dev/dsk/c5t2d0 [Return]
   # vgextend -g bus2 /dev/vg_pkg1 /dev/dsk/c6t2d0 [Return]
   ```

4. Repeat this procedure for additional volume groups.

5. Create a PVG-strict mirrored logical volume named lvol1 with 12 MB:

   ```
   # lvcreate -L 12 -m 1 -s g /dev/vg_pkg1 [Return]
   ```

6. To display the volume group, use the vgdisplay command:

   ```
   # vgdisplay -v /dev/vg_pkg1 [Return]
   ```

   Make a copy of /etc/lvmpvg on the configuration node for later use:

   ```
   # cp /etc/lvmpvg /tmp/lvmpvg [Return]
   ```

Edit /tmp/lvmpvg, removing references to any physical volume groups other than the groups created for bus1 and bus2. Do *not* edit /etc/lvmpvg on the configuration node.

7. Create a file system on the logical volume and mount it on /mnt1.

```
# mkdir /mnt1 [Return]
# newfs -F vxfs /dev/vg_pkg1/rlvol1 [Return]
# mount /dev/vg_pkg1/lvol1 /mnt1 [Return]
```

8. Repeat this process for any other volume groups you need to create.

9. Before setting up the volume group for use on other nodes, you must first unmount any file systems that reside on the volume group, then deactivate it. At run time, volume group activation and file system mounting are done through the package control script.

```
# umount /mnt1 [Return]
# vgchange -a n /dev/vg_pkg1 [Return]
```

10. Use the following commands to set up the same volume group on another cluster node. In this example, the commands set up a new volume group on node 2 which will hold the same physical volume that was available on node 1. To set up the volume group on node 2, use the following steps:

a. On node 1, copy the mapping of the volume group group to a specified file.

```
# vgexport -p -m  /tmp/vg_pkg1.map  /dev/vg_pkg1 [Return]
```

b. Still on node 1, copy the map file (/tmp/vg_pkg1.map) to the same path (/tmp/vg_pkg1.map) on node 2. At the same time, copy the physical volume group file (/tmp/lvmpvg) to the same path (/tmp/lvmpvg) on the other node. This file was created during an earlier step.

c. On node 2, create the volume group directory:

```
# mkdir /dev/vg_pkg1 [Return]
```

d. Still on node 2, create a control file named *group* in the directory /dev/vg_pkg1, as follows:

```
# mknod /dev/vg_pkg1/group c 64 0xhh0000 [Return]
```

Replace the minor number *hh* with a hexadecimal minor group number available on your system. (If possible, use the same hexadecimal number on both nodes.)

e. Examine the /etc/lvmpvg file on the second node. If the file does not exist, create it. Insert the content of /tmp/lvmpvg copied over from the configuration node. You should be inserting information only about the two physical volume groups bus1 and bus2. If the block device names copied from node 1, are not the same as the block device names for the same disks on node 2, change them to the correct names for node 2. The planning worksheet should indicate what names are associated with the same disks on the different nodes.

f. Import the volume group data using the map file from node 1. On node 2, enter:

```
# vgimport -m /tmp/vg_pkg1.map /dev/vg_pkg1  \
  /dev/dsk/c1t2d0 /dev/dsk/c0t2d0 Return
```

g. Enable the volume group on node 2:

```
# vgchange -a y /dev/vg_pkg1 Return
```

h. Create a directory to mount the disk:

```
# mkdir /mnt1 Return
```

i. Mount and verify the volume group on node 2:

```
# mount /dev/vg_pkg1/lvol1 /mnt1 Return
```

j. Unmount the volume group on node 2:

```
# umount /mnt1
```

k. Deactivate the volume group on node 2:

```
# vgchange -a n /dev/vg_pkg1 Return
```

Creating Additional Volume Groups

The preceding sections show in general how to create volume groups and logical volumes for use with MC/LockManager. Repeat the procedure for as many volume groups as you need to create, substituting other volume group names, logical volume names, and physical volume names.

Final Steps Before Cluster Configuration

Before configuring the cluster, make sure that all volume groups are activated on the configuration node. Doing this ensures that MC/LockManager will have all the information it needs to proceed with configuration. Use the following command on node 1:

<u>vgchange -a y /dev/vg_ops</u> ⌐Return¬

Repeat the command for each volume group.

Preventing Automatic Activation of Volume Groups

It is important to prevent both OPS and package volume groups from being activated at system boot time by the /etc/lvmrc file. To ensure that this does not happen, edit the /etc/lvmrc file by setting AUTO_VG_ACTIVATE to 0, then include all volume groups that are *not* cluster-bound (like root) in the custom_vg_activation function. Volume groups that will be used by OPS or by packages should *not* be included in this file, since they will be activated and deactivated by the control scripts.

A completed example of a /etc/lvmrc file appears below.

```
# "@(#)/etc/lvmrc          $Revision: 72.2 $$Date: 94/05/20 17:46:54 $"
#
# This file is sourced by /sbin/lvmrc. This file contains the flags
# AUTO_VG_ACTIVATE and RESYNC which are required by the script in /sbin/lvmrc.
# These flags must be set to valid values (see below).
#

#
# The activation of Volume Groups may be customized by setting the
# AUTO_VG_ACTIVATE flag to 0 and customizing the function
# custom_vg_activation()
#

#
#       To disable automatic volume group activation,
#       set AUTO_VG_ACTIVATE to 0.
#

AUTO_VG_ACTIVATE=0

#
#       The variable RESYNC controls the order in which
#       Volume Groups are resyncronized. Allowed values
```

```
#       are:
#               "PARALLEL"       - resync all VGs at once.
#               "SERIAL"         - resync VGs one at a time.
#
#       SERIAL will take longer but will have less of an
#       impact on overall I/O performance.
#

RESYNC="SERIAL"


#
#       Add customized volume group activation here.
#       A function is available that will synchronize all
#       volume groups in a list in parallel. It is
#       called parallel_vg_sync.
#
#       This routine is only executed if AUTO_VG_ACTIVATE
#       equals 0.
#

custom_vg_activation()
{
    # e.g. /sbin/vgchange -a y -s
    #       parallel_vg_sync "/dev/vg00 /dev/vg01"
    #       parallel_vg_sync "/dev/vg02 /dev/vg03"

    if [ -r /etc/lvmtab ]
    then

    # The following assumes that /dev/vg00, /dev/vg01 and
    # /dev/vg02 are non-shared disks.  Include all non-
    # shared disks that are to be activated at boot time.

        VOLUME_GROUPS="/dev/vg00 /dev/vg01 /dev/vg02"

        for VG in ${VOLUME_GROUPS}
        do
            /sbin/vgchange -a y -s ${VG}
        done

        if [ -f /sbin/vgsync ]
        then
            {
            for VG in $VOLUME_GROUPS
            do
                {
                if /sbin/vgsync $VG > /dev/null
                then
```

```
                        echo "Resynchronized volume group $VG"
                fi
                }                   #
                # RESYNC is set in /etc/lvmrc
                #
                if [ $RESYNC = "SERIAL" ]
                then
                        wait
                fi
            done
            }           fi
        fi

        return 0
}


#
# The following functions should require no additional customization:
#

parallel_vg_sync()
{
        for VG in $*
        do
                {
                if /sbin/vgsync $VG > /dev/null
                then
                        echo "Resynchronized volume group $VG"
                fi
                }           done
}
```

Configuring the Cluster Manager Software

This section describes how to define the basic cluster configuration. To do
this in SAM, read the next section. If you want to use HP-UX commands
for cluster configuration, skip ahead to the section entitled "Using HP-UX
Commands to Configure the Cluster Manager."

Using SAM to Configure the Cluster Manager

To configure a high availability cluster for use with Oracle Parallel Server, use the following steps on the configuration node:

1. In SAM, select the High Availability Clusters option.
2. Choose the Cluster Configuration option. SAM displays a Cluster Configuration screen. If no clusters have yet been configured, the list area will be empty. If there are one or more HA clusters already configured on your local network, you will see them listed.
3. Select the Actions menu, and choose Create Cluster Configuration. A step menu appears.
4. Choose each required step in sequence, filling in the dialog boxes with required information, or accepting the default values shown. For information about each step, choose Help.
5. When finished with all steps, select (OK) at the Step Menu screen. This action creates the cluster configuration file and then copies the file to all the nodes in the cluster. When the file copying is finished, you return to the Cluster Configuration screen.
6. Exit from the Cluster Configuration screen, returning to the High Availability Clusters menu.

Skip ahead to the section entitled "Configuring the Distributed Lock Manager Software."

Using HP-UX Commands to Configure the Cluster Manager

The file containing cluster configuration data is known as
/etc/cmcluster/cmclconfig. This file is not editable, so you must
create and edit an ASCII file first, then convert it into binary form. First, on
the configuration node, use the following command to generate an editable
template file:

```
# cmquerycl -n node1 -n node2 -v  -C /etc/cmcluster/cluster.asc Return
```

The command specifies that you want to create an ASCII cluster configuration
template file for a two-node cluster consisting of *node1* and *node2*. The
command creates a file known as */etc/cmcluster/cluster.asc*, which you should
edit to incorporate specific cluster data. An example appears following.

Editing the ASCII Cluster Configuration File

Use the data from the cluster manager worksheet and the distributed lock manager worksheet when editing the template file.

```
# ************************************************************************
# ********* HIGH AVAILABILITY CLUSTER CONFIGURATION FILE ***************
# ***** For complete details about cluster parameters and how to    ****
# ***** set them, consult the cmquerycl(1m) manpage or your manual. ****
# ************************************************************************

# Enter a name for this cluster.  This name will be used to identify the
# cluster when viewing or manipulating it.

CLUSTER_NAME  cluster1

# Cluster Lock Device Parameters.  This is the volume group that
# holds the cluster lock which is used to break a cluster formation
# tie.  This volume group should not be used by any other cluster
# as cluster lock device.

FIRST_CLUSTER_LOCK_VG  /dev/vg_ops

# Definition of nodes in the cluster.
# Repeat node definitions as necessary for additional nodes.

NODE_NAME  node1
  NETWORK_INTERFACE lan0
    HEARTBEAT_IP  15.13.171.43
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c1d0s2
NODE_NAME  node2
  NETWORK_INTERFACE lan0
    HEARTBEAT_IP  15.13.171.44
  FIRST_CLUSTER_LOCK_PV /dev/dsk/c1d0s2

# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE /dev/tty0p0


# Cluster Timing Parmeters (microseconds).

HEARTBEAT_INTERVAL  1000000
NODE_TIMEOUT  2000000

# Configuration/Reconfiguration Timing Parameters (microseconds).

AUTO_START_TIMEOUT 600000000
NETWORK_POLLING_INTERVAL 2000000
```

```
# List of cluster aware Volume Groups. These volume groups will
# be used by clustered applications via the vgchange -a e command.
# For example:
# VOLUME_GROUP   /dev/vg_pkg1
# VOLUME_GROUP   /dev/vg02


# List of cluster aware Volume Groups.  These volume groups
# will be used by clustered applications via the vgchange -a s command
# For example: # DLM_VOLUME_GROUP /dev/vg_database
# DLM_VOLUME_GROUP /dev/vg02
DLM_VOLUME_GROUP /dev/vg_ops

# DLM parameters.

DLM_ENABLED    YES
DLM_CONNECT_TIMEOUT  30000000
DLM_PING_INTERVAL  20000000
DLM_PING_TIMEOUT  60000000
DLM_RECONFIG_TIMEOUT  120000000
DLM_COMMFAIL_TIMEOUT  150000000
DLM_HALT_TIMEOUT  240000000
```

Using an editor, review and complete the file, supplying appropriate values
as needed for the fields. In most cases, the defaults are correct, but it is
important to check them.

Identifying the Cluster Lock Volume Group and Disk

A cluster lock disk is required for OPS clusters. The disk must be accessible to
both systems and powered separately from the systems.

The default FIRST_CLUSTER_LOCK_VG and
FIRST_CLUSTER_LOCK_PV given in the template file are the
volume group and physical volume name of a disk chosen based on minimum
failover time calculations. You should ensure that this disk meets your power
wiring requirements. If necessary, choose a different disk.

If necessary, you can configure a second cluster lock. Enter the following
parameters in the cluster configuration file:

SECOND_CLUSTER_LOCK_VG */dev/volume-group*

SECOND_CLUSTER_LOCK_PV */dev/dsk/special-file*

The */dev/volume-group* is the name of the second volume group and *special-file* is the physical volume name of a lock disk in the chosen volume group. These lines should be added to the information for each node.

Note	Only configure a second cluster lock when it is required by your cluster configuration. When possible, a single cluster lock is recommended. See "Dual Cluster Lock" in chapter 1 for more information.

Identifying Serial Heartbeat Connections

If you are using a serial (RS232) line as a heartbeat connection, use the SERIAL_DEVICE_FILE parameter and enter the device file name that corresponds to the serial port you are using on each node. Be sure that the serial cable is securely attached during and after configuration.

Verifying Network Data

Verify that the IP addresses used for HEARTBEAT_IP on each node belong to the same bridged net. Verify that network interface entries for which no IP addresses appear are for standby use.

Identifying DLM Volume Groups

The template file will include an entry for all volume groups used by the Oracle Parallel Server that are accessed concurrently by the different nodes in the cluster. These volume groups are activated by the `vgchange -a s` command. A separate DLM_VOLUME_GROUP line should appear for each volume group that will be activated in shared mode. Volume groups that will be used by Oracle Parallel Server must be labelled DLM_VOLUME_GROUP.

Note	It's important that only volume groups used by OPS be listed with the DLM_VOLUME_GROUP parameter, since these volume groups will be marked for activation in *shared* mode. Volume groups used by packages should be listed with the VOLUME_GROUP parameter described following. You may need to change the default assignments in order to get this correct.

Identifying Volume Groups for Packages

The template file will include an entry for all volume groups used by packages. These volume groups are activated by the `vgchange -a e` command. A separate VOLUME_GROUP line should appear for each volume group that will be activated in exclusive mode.

Note	It's important that volume groups used by packages be listed with the VOLUME_GROUP parameter, since these volume groups will be marked for activation in *exclusive* mode. You may need to change the default assignments in order to get this correct.

Enabling DLM

When the DLM_ENABLED parameter is set to YES (the default), the DLM is started in the cluster when the cluster starts or reboots. If DLM has not yet been configured, and this parameter is set to YES, the cluster will not start successfully.

Verifying the Configuration

Use the following command to verify the configuration you enter into the template file:

```
# cmcheckconf -v -C /etc/cmcluster/cluster.asc  [Return]
```

This command checks the content of the ASCII template file and displays messages. If there are errors, edit the file again to correct them, then issue the `cmcheckconf` command again.

Activating the Lock Volume Group

On the configuration node only, activate the lock volume group with the following command:

```
# vgchange -a y /dev/vg_ops  [Return]
```

Activating the volume group allows the cluster configuration software to initialize the cluster lock during the next step.

Distributing the Configuration

Next, use the following command to distribute the configuration to all the nodes in the cluster:

    # cmapplyconf -v -C /etc/cmcluster/cluster.asc (Return)

The cmapplyconf command with these options creates a binary configuration file named /etc/cmcluster/cmclconfig and distributes it to all the nodes in the cluster. The cluster is not started, however, until you issue the cmrunnode command on each node. This is described in a later section, entitled "Using HP-UX Commands to Test the Configuration." Also, if you plan to run DLM, DLM must be configured before the cluster can start. See the following section "Configuring the Distributed Lock Manager Software."

Deactivating All Cluster-Bound Volume Groups

Finally, deactivate all cluster-bound volume groups:

    # vgchange -a n /dev/vg_ops (Return)

Repeat this command to deactivate each OPS and package volume group.

Setting up Autostart Features

In order to automate the startup of cluster nodes after a system boot, modify the /etc/rc.config.d/cmcluster file on each node. MC/LockManager provides this startup script to control the startup process:

```
#*************************** CMCLUSTER ************************
# Highly Available Cluster configuration
#
# @(#) $Revision: 72.2 $
#
# AUTOSTART_CMCLD:       If set to 1, the node will attempt to
#                        join it's CM cluster automaticly when
#                        the system boots.
#                        If set to 0, the node will not attempt
#                        to join it's CM cluster.
#
AUTOSTART_CMCLD=1
```

MC/LockManager also provides several commands for manual control of the cluster:

- `cmrunnode` is used to start a node.
- `cmhaltnode` is used to manually stop a running node.
- `cmruncl` is used to manually start a stopped cluster.
- `cmhaltcl` is used to manually stop a cluster.

Refer to the man pages for a complete description of these commands.

Automatic Shutdown

Cluster shutdown during a graceful system halt is automatic. The /sbin/init.d directory contains a script named *cmcluster*, which executes the `/usr/bin/cmhaltnode -f` command whenever the HP-UX `shutdown` command is run, thereby removing the node from the cluster gracefully before system halt.

Configuring the Distributed Lock Manager Software

This section describes DLM configuration. To do this in SAM, read the next section. If you want to use HP-UX commands for DLM configuration, skip ahead to the section entitled "Using HP-UX Commands to Configure the Distributed Lock Manager."

Using SAM to Configure the Distributed Lock Manager

Use the following steps on the configuration node:

1. From the High Availability Clusters menu in SAM, choose the DLM Configuration option. SAM displays a Distributed Lock Manager Configuration screen. If no cluster has yet been configured with the DLM, the list area will be empty.
2. Select the Actions menu, and choose Create DLM Configuration. A new screen appears, containing a list of clusters eligible for DLM configuration. The list contains only clusters that are not currently running. Select the cluster you wish to configure with the DLM, then select Specify DLM Parameters. A step menu appears.

3. Choose each step in sequence, filling in the dialog boxes with required information, or accepting the default values shown. For information about each step, choose Help.
4. When finished with all steps, select (OK) at the Step Menu screen. This action propagates the DLM configuration among all nodes and returns you to the Distributed Lock Manager Configuration screen.
5. Exit from the Distributed Lock Manager Configuration screen.

Skip ahead to the section entitled "Testing the Configuration."

Using HP-UX Commands to Configure the Distributed Lock Manager

This section describes how to set the the DLM's internal lock database parameters using HP-UX commands. DLM internal parameters are stored in a file known as */etc/opt/dlm/dlmconfig*. This file is not editable, so you must create an ASCII file first, then convert it into binary form. First, use the following command to generate an editable template file:

    # dlmquery -v -C /etc/opt/dlm/dlm.asc (Return)

Edit the file */etc/opt/dlm/dlm.asc* to incorporate the appropriate DLM internal parameters from the Distributed Lock Manager worksheet (defaults shown here are appropriate for the Oracle demo database, which can be installed at the same time you install Oracle software). The file looks like the following:

```
CLUSTER_NAME                    cluster1
MAXPROCESSES                    2400
MAXRESOURCES                    6000
MAXLOCKS                        12000
DEADLOCK_DETECTION_INTERVAL     300
PROCESS_MONITORING_INTERVAL     300

NODE_NAME                        node1
    IPADDR                      15.27.217.9

NODE_NAME                        node2
    IPADDR                      15.27.217.10
```

Review and complete the file, supplying approriate values as needed. Verify the cluster name, and ensure that the IP addresses used for IPADDR on each node belong to the same subnet. As necessary, you can substitute the values from your own Distributed Lock Manager worksheet for the defaults that are shown.

Use the following command to verify the configuration you enter into the template file:

> # `dlmcheckconf -v -C /etc/opt/dlm/dlm.asc` (Return)

This command checks the content of the ASCII template file and displays messages. If there are errors, edit the file again to correct them, then issue the `dlmcheckconf` command again. Use the following command to copy the configuration to all the nodes in the cluster:

> # `dlmapplyconf -v -C /etc/opt/dlm/dlm.asc` (Return)

Note   The `dlmquery` command should be used only for template creation; the output of the command does not necessarily reflect the current cluster configuration. For diagnostic information, refer to the section "DLM Diagnostic and Statistical Tools" in the chapter "Troubleshooting Your Cluster."

Testing the Configuration

After configuring the cluster manager and the Distributed Lock Manager, you must start up the cluster to verify proper operation. The next section shows how to do so using SAM. If you want to use HP-UX commands to test the configuration, skip ahead to the section entitled "Using HP-UX Commands to Test the Configuration."

Using SAM to Test the Configuration

Use the following steps to perform a sanity check on the cluster:

1. From the High Availability Clusters area in SAM, choose Cluster Administration. Select the cluster you have configured in previous steps. Then, from the Action list, choose Start Cluster. Also choose All Nodes, indicating that you want to start the cluster on all nodes configured for the cluster. Confirm that you want to start the cluster by selecting (Yes).
2. From the Action list, choose View Cluster Node States. Verify that both nodes are listed as Running.
3. From the Action list, choose View Syslog file to display the messages that have been logged during the configuration process.
4. After exiting from the High Availability Clusters area of SAM, you can enter the Process Management area and select Process Control to display a list of currently running processes. Verify that the following daemon processes are running:
   a. `cmcld` - CM daemon
   b. `cmdlmd` - DLM daemon
   c. `cmdlmmond` - DLM monitor daemon
   d. `cmlvmd` - SLVM daemon

Proceed to the section "Creating OPS Startup and Shutdown Scripts."

Using HP-UX Commands to Test the Configuration

Use `cmrunnode` on *both nodes* to start the cluster named in the cluster configuration file. In the following example, node 1 and node 2 are made into a cluster. From node 1:

    # cmrunnode -v (Return)

The following messages are displayed:

    Successfully started /etc/cmcld on node1.
    cmrunnode: Waiting for cluster to form......

Immediately change to node 2 and issue the same command:

    # cmrunnode -v (Return)

The following messages are displayed:

```
Successfully started /etc/cmcld on node2.
Cluster successfully formed.
```

Note that the node name is not required in this command. The command starts up the cluster manager daemon on each node, and one of the nodes becomes the cluster coordinator. Use the `cmviewcl -v` command to display information about the newly started cluster. This command will show whether or not the cluster has formed successfully.

Use the `ps -ef` command to display a list of currently running processes. Verify that the following daemon processes are running:

■ `cmcld` - CM daemon
■ `cmdlmd` - DLM daemon
■ `cmdlmmond` - DLM monitor daemon
■ `cmlvmd` - SLVM daemon

Testing Cluster Reconfiguration and Halt

For information on how to test cluster reconfiguration as nodes leave the cluster, see the chapter "Troubleshooting Your Cluster."

The next configuration steps must be completed with HP-UX commands.

Creating OPS Startup and Shutdown Scripts

To coordinate OPS startup and shutdown with cluster node startup and shutdown, you can create a DLM control script on each node. DLM control scripts perform three tasks:

■ Volume group activation and deactivation.
■ Oracle instance startup and shutdown.
■ Oracle application startup and shutdown.

Every time an OPS node or an entire OPS cluster starts up or shuts down, the DLM executes a control script named

```
/etc/opt/dlm/rc/runhalt.sh
```

You must customize this script for your cluster.

Both startup and shutdown use the same script called with different parameters. On starting up, the DLM will invoke this script with the parameter `start` and on shutting down, the DLM will invoke the script with the parameter `stop`. When called with the `start` parameter, the script will first activate shared volume groups and then start up OPS instances on the node. When called with the `stop` parameter, the script will first shut down OPS instances and then deactivate shared volume groups. While these tasks could be carried out on the command line, the use of scripts simplifies the process.

A template for this script is found in pathname /opt/dlm/newconfig/runhalt.sh. Copy the template to the /etc/opt/dlm/rc directory on the configuration node, and edit it to include the information relevant to your shared volume groups, Oracle instances and Oracle applications. Then change the permissions to 700, which permits execution by root. Copy the script to the same path on the other node. For initial testing, make sure that the SHARED_VGS parameter is defined to include all the shared volume groups you wish to activate when the DLM starts and deactivate when the DLM stops. Example:

```
SHARED_VGS="vg_ops"
```

After Oracle database software and applications are installed, set the Oracle parameters as suggested in the comments in the template file.

The time permitted for OPS startup and shutdown is regulated by two internal DLM parameters, DLM Connect Timeout and DLM Halt Timeout. The connect timeout is the upper bound on the time required to initialize DLM shared memory; the startup process runs in the background and is not bound by this limit. The halt timeout is the upper bound on the time required to run all OPS shutdown processes in the script. Be sure that this parameter is set to a value that allows all scripts and commands to complete before timeout occurs.

For more details about how to customize the startup/shutdown script, read the instructions that appear inside the template file.

Installing Oracle Parallel Server

Before installing the Oracle Parallel Server, make sure the cluster is running. Log in as the *oracle* user and then use the Oracle *installer* to install Oracle software and to build the correct Oracle runtime executables. The Oracle *installer* will also copy the executables to the other node in the cluster. Select the following installation option to install OPS software and to create the demo database:

```
COMPLETE SOFTWARE/DATABASE FRESH INSTALL
```

Refer to the *Oracle Parallel Server for HP 9000 Series 800 Installation and Configuration Guide* for details of the Oracle installation. As part of this installation, the Oracle installer builds the Oracle demo database on the primary node, using the character (raw) device file names for the logical volumes created earlier. For the demo database, create fourteen logical volumes as shown in the table "Required Oracle File Names" earlier in this chapter. As the installer prompts for database file names, enter the pathnames of the raw logical volumes instead of using the defaults. If you do not wish to install the demo database, select

```
SOFTWARE INSTALL ONLY
```

In this case, create an appropriate number of raw logical volumes to build your development or production system. Be sure to create enough log files for both instances.

Starting Up Oracle Instances

Once the Oracle installation is complete, ensure that the completed runhalt script is in place on each node and that each /etc/rc.config.d/cmcluster script contains the entry `AUTOSTART_CMCLD=1`. Then reboot each node. Within a couple of minutes following reboot, the cluster will reform, and the database instances and application programs will come up.

When Oracle has been started, you can use the SAM process management area or the `ps -ef` command on both nodes to verify that all OPS daemons and Oracle processes are running.

4

Configuring Packages and Their Services

If your cluster is going to run packages, you must identify your highly available applications and configure them into packages. This chapter describes the following *package configuration* tasks:

- Creating the Package Configuration
- Writing the Package Control Script
- Distributing the Binary Cluster Configuration File

Each of these tasks is described in a separate section below.

In configuring your own packages, use the Package Configuration Worksheet described in the "Planning" chapter. Package configuration data becomes part of the binary cluster configuration file on all nodes in the cluster. The control script data goes into an executable control script which runs specific package services and monitors their operation.

Creating the Package Configuration

You can create a package using SAM or using HP-UX commands and editors. The following section describes SAM configuration. If you are using HP-UX commands, skip ahead to the section entitled "Using HP-UX Commands to Create a Package."

Using SAM to Configure a Package

To configure a high availability package use the following steps on the configuration node (node 1):

1. In SAM, choose the "Clusters" area, then the High Availability Clusters option.
2. Choose the Package Configuration option. SAM displays a Package Configuration screen. If no packages have yet been configured, the list area will be empty. If there are one or more packages already configured on clusters in your network, you will see them listed.
3. Select the Actions menu, and choose Create/Add a Package. A step menu appears.
4. Choose each required step in sequence, filling in the dialog boxes with required information, or accepting the default values shown. For information about each step, choose Help.
5. When finished with all steps, select $\boxed{\text{OK}}$ at the Step Menu screen. This action creates the cluster configuration file and then copies the file to all the nodes in the cluster. When the file copying is finished, you return to the Package Configuration screen.
6. Exit from the Package Configuration screen, returning to the High Availability Clusters menu.

Skip ahead to the section on "Customizing the Package Control Script." This must be done with an editor, and cannot be done directly in SAM.

Using HP-UX Commands to Create a Package

Use the following procedure to create packages. The example shows the creation of two packages, pkg1 and pkg2 for a sample configuration.

1. First, create a subdirectory for each package you are configuring in the /etc/cmcluster directory:

    # mkdir /etc/cmcluster/pkg1 `Return`
    # mkdir /etc/cmcluster/pkg2 `Return`

    You can use any directory names you wish.

2. Next, generate a package configuration template for each package:

    # cmmakepkg -p /etc/cmcluster/pkg1/pkg1conf.asc `Return`
    # cmmakepkg -p /etc/cmcluster/pkg2/pkg2conf.asc `Return`

    You can use any file names you wish for the ASCII templates.

3. Edit these template files to reflect the configuration for each package. Include the information from the Package Configuration Worksheet.

The following is a sample package configuration file template customized for a typical package.

```
# ***********************************************************************
# ******* HIGH AVAILABILITY PACKAGE CONFIGURATION FILE (template) ******
# ***********************************************************************
# ******* Note: This file MUST be edited before it can be used. ********
# * For complete details about package parameters and how to set them, *
# ** consult the MC/ServiceGuard or MC/LockManager manpages or manuals.*
# ***********************************************************************

# Enter a name for this package.  This name will be used to identify the
# package when viewing or manipulating it.  It must be different from
# the other configured package names.

PACKAGE_NAME  pkg1


# Enter the names of the nodes configured for this package.  Repeat
# this line as necessary for additional adoptive nodes.
# Order IS relevant.  Put the second Adoptive Node AFTER the first
# one.
# Example : NODE_NAME  original_node
#           NODE_NAME  adoptive_node
```

```
NODE_NAME    node1
NODE_NAME    node2


# Enter the complete path for the run and halt scripts.  In most cases
# the run script and halt script specified here will be the same script,
# the package control script generated by the cmmakepkg command.  This
# control script handles the run(ing) and halt(ing) of the package.
# If the script has not completed by the specified timeout value,
# it will be terminated.  The default for each script timeout is
# NO_TIMEOUT.  Adjust the timeouts as necessary to permit full
# execution of each script.
# Note: The HALT_SCRIPT_TIMEOUT should be greater than the sum of
# all SERVICE_HALT_TIMEOUT specified for all services.

RUN_SCRIPT    /etc/cmcluster/pkg1/control.sh
RUN_SCRIPT_TIMEOUT  NO_TIMEOUT
HALT_SCRIPT    /etc/cmcluster/pkg1/control.sh
HALT_SCRIPT_TIMEOUT  NO_TIMEOUT


# Enter the name of the service, the SERVICE_FAIL_FAST_ENABLED value
# and the SERVICE_HALT_TIMEOUT value for this package.  Repeat these
# three lines as necessary for additional service names.  All service
# names MUST correspond to the service names used by cmrunserv and
# cmhaltserv commands inside the run and halt scripts.
# The default for SERVICE_FAIL_FAST_ENABLED is NO.  If set to YES,
# in the event of a service failure, the cluster software will halt
# the node on which the service is running.  Adjust as necessary.
# The default for SERVICE_HALT_TIMEOUT is 300 seconds (5 minutes).
# In the event of a service halt, the cluster software will first send out
# a SIGTERM signal to terminate the service.  If the process is not
# terminated, after waiting for the specified timeout, the process will
# be sent the SIGKILL signal to force its termination.
# Adjust the timeout as necessary to allow enough time for any
# cleanup process associated with the service to complete.

SERVICE_NAME    service1
SERVICE_FAIL_FAST_ENABLED NO
SERVICE_HALT_TIMEOUT 300


# Enter the network subnet name that is to be monitored for this package.
# Repeat this line as necessary for additional subnet names.

SUBNET   15.16.168.0


# Uncomment the following line and enter the name of additional resources
```

4-4   Configuring Packages and Their Services

```
# required by this package. As an example, OTS/9000 resource names consist
# of the X.25 CONS and CLNS subnet names configured in the ots_subnets file.
# Repeat the line as necessary for additional resource names.

RESOURCE_NAME

# The default for PKG_SWITCHING_ENABLED is YES. In the event of a
# failure, this permits MC/LockManager to transfer the package to an
# adoptive node.  Adjust as necessary.

PKG_SWITCHING_ENABLED  YES


# The default for NET_SWITCHING_ENABLED is YES.  In the event of a
# failure, this permits the cluster software to switch LANs locally
# (transfer to a standby LAN card).  Adjust as necessary.

NET_SWITCHING_ENABLED  YES


# The default for NODE_FAIL_FAST_ENABLED is NO.  If set to YES,
# in the event of a failure, the cluster software will halt the node
# on which the package is running.  Adjust as necessary.

NODE_FAIL_FAST_ENABLED NO
```

Package Configuration Template

Use the information on the Package Configuration worksheet to complete the file. You must include the following information:

- NODE_NAME. Enter the name of each node in the cluster on a separate line.
- RUN_SCRIPT and HALT_SCRIPT. Specify the pathname of the package control script (described in the next section). No default is provided.
- SERVICE_NAME, SERVICE_FAIL_FAST_ENABLED and SERVICE_HALT_TIMEOUT. Enter groups of these three for each service.
- SUBNET. Enter the subnet address used by the package.
- NODE_FAIL_FAST_ENABLED parameter. Enter YES or NO.

The package configuration file is later combined with the cluster configuration data in the binary cluster configuration file.

Note	If a package contains applications that will access the OPS database, you must configure the package startup in a specific way so that the package starts *after* OPS and the cluster manager (including DLM services) start up. Read the next section for more information.

Configuring Packages that Access the OPS Database

Use the following procedures for packages that contain applications which access the OPS database:

1. In the ASCII package configuration file, set the PKG_SWITCHING_ENABLED parameter to NO, or if you are using SAM to configure packages, set Automatic Switching to Disabled. This keeps the package from starting up immediately when the node joins the cluster.
2. You can then manually start the package using the `cmmodpkg -e` *packagename* command after OPS is started. Alternatively, you can choose to automate the process of package activation by writing your own script, and copying it to all nodes that can run the package. This script should contain the `cmmodpkg -e` command and activate the package after OPS and the cluster manager have started.

Writing the Package Control Script

The package control script contains all the information necessary to run all the services in the package, monitor them during operation, react to a failure, and halt the package when necessary. You can use either SAM or HP-UX commands to create the package control script. If you need to modify the script later, or if you wish to do extensive customizing, you may wish to use the command method.

Using SAM to Write the Package Control Script

Select the High Availability options in SAM, then choose "Package Configuration." From the Action menu, choose "Create/Add a Package." The step menu appears, showing a group of options. The last two steps on the menu are for creating the package control script. Select each option after you define the package itself. For more information, use the Help key.

When you create a package control script this way, you do not need to do any further editing, but you may customize the script if you wish.

Using Commands to Write the Package Control Script

Each package must have a separate control script. The control script is placed in the package directory and is given the same name that it has in the package configuration file. The package control script contains both the run instructions and the halt instructions for the package. It must be executable. Use the following procedure to create a control scripts for the sample package *pkg1*.

First, generate a control script template:

    # cmmakepkg -s /etc/cmcluster/pkg1/control.sh ⌷Return⌸

Next, make the script executable:

    # chmod +x /etc/cmcluster/pkg1/control.sh ⌷Return⌸

You may customize the script, as described in the next section.

Customizing the Package Control Script

Check the definitions and declarations at the beginning of the control script using the information in the Package Configuration worksheet.

Entries that Need to Be Customized

You need to customize as follows:

- Update the PATH statement to reflect any required paths needed to start your services.
- Enter the names of volume groups that will be activated by packages.

- Add the names of logical volumes and file systems that will be mounted on them.
- Define IP subnet and IP address pairs for your package.
- Add service name(s).
- Add service command(s)
- Add a service restart parameter, if desired.

Note   Use care in defining service run commands. Each run command is executed by the control script in the following way:

- The `cmrunserv` command executes each run command and then monitors the process id of the executing run command.
- When the command started by `cmrunserv` exits, MC/LockManager determines that a failure has occurred and takes appropriate action, which may include transferring the package to an adoptive node.
- If a run command is a shell script that runs some other command and then exits, MC/LockManager will consider this normal exit as a *failure*.

To avoid problems in the execution of control scripts, ensure that each run command is the name of an actual service and that its process remains alive until the actual service stops.

If you need to define a set of run and halt operations in addition to the defaults, create functions for them in the sections under the heading "CUSTOMER DEFINED FUNCTIONS."

Below is an excerpt from the control script for a sample package configuration.

```
#"(#) A.10.03        $Revision: 74.11 $ $Date: 95/05/11 14:59:55 $"
# **********************************************************************
# *                                                                    *
# *          HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)        *
# *                                                                    *
# *        Note: This file MUST be edited before it can be used.        *
# *                                                                    *
# **********************************************************************

# UNCOMMENT the variables as you set them.
```

```
# Set PATH to reference the appropriate directories.
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin

# VOLUME GROUP ACTIVATION:
# By default, volume groups are activated in exclusive mode. This
# assumes the volume groups have been initialized with 'vgchange -c y'
# at the time of creation. For the ability to recover from both Node and
# disk faults, if your disks are mirrored on separate physical paths,
# uncomment the first line and comment out the default.
#
# If you wish to use non-exclusive activation mode, uncomment the second
# line and comment out the default. Single node cluster configurations
# must use non-exclusive activation.
#
# VGCHANGE="vgchange -a e -q n"
# VGCHANGE="vgchange -a y"
VGCHANGE="vgchange -a e"  # Default

# VOLUME GROUPS
# Define the volume groups which are used by this package. The volume
# groups will be activated via the volume group activation method defined
# above. Filesystems associated with these volume groups are defined below.
#
# Example: VG[0]=vg01
#          VG[1]=pkg2
#
VG[0]=/dev/vg01

# FILESYSTEMS
# Define the filesystems which are used by this package. The filesystems
# are defined as pairs of entries specifying the logical volume and the
# mount point for the file system. Each filesystem will be fsck'd prior
# to being mounted. The filesystems will be mounted in the order specified
# during package startup and will be unmounted in reverse order during
# package shutdown. Ensure that volume groups referenced by the logical
# volume definitions below are included in volume group definitions
# above.
#
# Example: LV[0]=/dev/vg01/lvol1; FS[0]=/pkg1
#          LV[1]=/dev/pkg2/lvol1; FS[1]=/pkg2
#
# LV[0]=""; FS[0]=""
LV[0]=/dev/vg01/lvol1
FS[0]=/mnt1

# IP ADDRESSES
# IP/Subnet address pairs for each IP address you want to add to a subnet
# interface card.  Must be set in pairs, even for IP addresses on the same
# subnet.
```

```
#
# Hint: Run "netstat -i" to see the available subnets in the Network field.
# Example: IP[0]=192.10.25.12
# Example: SUBNET[0]=192.10.25.0 # (netmask=255.255.255.0)
#
IP[0]=15.13.171.23
SUBNET[0]=15.13.168.0

# SERVICE NAMES AND COMMANDS.
# Note: No environmental variables will be passed to the command, this
# includes the PATH variable. Absolute path names are required for the
# service command definition.  Default shell is /usr/bin/sh.
#
# Example: SERVICE_NAME[0]=pkg1a
# Example: SERVICE_CMD[0]="/usr/bin/X11/xclock -display 192.10.25.54:0"
# Example: SERVICE_RESTART[0]=""  # Will not restart the service.
# Example: SERVICE_NAME[1]=pkg1b
# Example: SERVICE_CMD[1]="/usr/bin/X11/xload -display 192.10.25.54:0"
# Example: SERVICE_RESTART[1]="-r 2"   # Will restart the service twice.
# Example: SERVICE_RESTART[2]="-R" # Will restart the service an infinite
#                                    number of times.
#
SERVICE_NAME[0]=service1
SERVICE_CMD[0]="/user/bin/X11/xclock -display displ1:0 -update 1"
SERVICE_RESTART[0]="-r 3"

# DTC manager information for each DTC.
# Example: DTC[0]=dtc_20
#DTC_NAME[0]=


# START OF CUSTOMER DEFINED FUNCTIONS

# This function is a place holder for customer define functions.
# You should define all actions you want to happen here, before the service is
# started.  You can create as many functions as you need.

function customer_defined_run_cmds
{
# ADD customer defined run commands.
: # do nothing instruction, because a function must contain some command.

 test_return 51
}

# This function is a place holder for customer defined functions.
# You should define all actions you want to happen here, before the service is
# halted.
```

```
function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
: # do nothing instruction, because a function must contain some command.
 test_return 52
}

# END OF CUSTOMER DEFINED FUNCTIONS
```

Package Control Script Template

This excerpt from the control script shows the assignment of values to a set of variables. The remainder of the script uses these variables to control the package by executing Logical Volume Manager commands and HP-UX commands, including `cmrunserv`, `cmmodnet`, and `cmhaltserv`. Examine a copy of the control script template to see the flow of logic. Use the following command:

    # cmmakepkg -s | more [Return]

The main function appears at the end of the script.

Verify and Distribute the Configuration

You can use SAM or HP-UX commands to verify and distribute the binary cluster configuration file among the nodes of the cluster.

Distributing the Configuration File And Control Script with SAM

When you have finished creating a package in the Package Configuration subarea in SAM, you are asked to verify the copying of the files to all the nodes in the cluster. When you respond OK to the verification prompt, MC/LockManager copies the binary configuration file and package control script to all the nodes in the cluster.

Copying Package Control Scripts with HP-UX commands

Use HP-UX commands to copy package control scripts from the configuration node to the same pathname on all nodes which can possibly run the package. Use your favorite method of file transfer (e. g., `rcp` or `ftp`). For example, from *node 1*, you can issue the `rcp` command to copy the package control script to *node 2*:

> # `rcp /etc/cmcluster/pkg1/control.sh node2:/etc/cmcluster/pkg1/control.sh` ⌐Return¬

Distributing the Binary Cluster Configuration File with HP-UX Commands

Use the following steps:

1. If your cluster is running, you need to halt it before proceeding. See the chapter "Maintaining an OPS Cluster" for an explanation of halting a cluster.
2. Verify that all the configuration scripts are correct. Use the following command:

> # `cmcheckconf -C /etc/cmcluster/cluster.asc -P \` ⌐Return¬
> `/etc/cmcluster/pkg1/pkg1conf.asc  -P \` ⌐Return¬
> `/etc/cmcluster/pkg2/pkg2conf.asc` ⌐Return¬

3. Generate the binary configuration file and distribute it across the nodes.

> # `cmapplyconf -v -C /etc/cmcluster/cluster.asc -P \` ⌐Return¬
> `/etc/cmcluster/pkg1/pkg1conf.asc -P \` ⌐Return¬
> `/etc/cmcluster/pkg2/pkg2conf.asc` ⌐Return¬

The `cmapplyconf` command creates a binary version of the cluster configuration file and distributes it to all nodes in the cluster. This action ensures that the contents of the file are consistent across all nodes.

Use the `cmapplyconf` command from the node on which the ASCII cluster and package configuration files exist. The cluster lock volume group must be activated on this node before issuing the command, so that the lock disk can be initialized. Be sure to deactivate this volume group after `cmapplyconf` is executed.

Note	While `cmcheckconf` and `cmapplyconf` must be used any time changes are made to cluster and package configuration scripts, they are generally executed after the package configuration file and the package control script are generated and modified during initial cluster configuration.

5

Maintaining an OPS Cluster

This chapter includes information about carrying out routine maintenance tasks on an Oracle Parallel Server configuration. Tasks include:

- Viewing the Status of the Cluster
- Starting and Stopping the Cluster
- Starting and Stopping Individual Nodes
- Administering Packages and Services
- Changing the Permanent Cluster Configuration
- Changing Oracle Parameters
- Making a Volume Group Sharable
- Activating a Volume Group in Shared Mode
- Making Changes to Shared Volume Groups
- Adding Additional Shared Volume Groups
- Adding Additional Disk Hardware

Refer to Appendix C for a complete list of man pages that relate to these tasks. The man pages contain additional useful information.

Note	You must be very careful to carry out all cluster management steps on only *one node at a time*. Be sure to exit from SAM on one node before running SAM on another node in the cluster to avoid any chance of one node's overwriting the cluster configuration that was defined on the other node. In general, it's simpler to carry out all configuration tasks on one "configuration node".

Viewing the Status of the Cluster

You can examine the status of all cluster nodes in SAM or by using HP-UX commands.

Using SAM to View Cluster Status

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, and open the Action Menu. Select an appropriate View option.

SAM provides three View options:

- View Cluster Network Configuration
- View Cluster Node States
- View Syslog File

Using HP-UX Commands to View Cluster Status

You can obtain information about the cluster by entering

    # cmviewcl -v (Return)

See Chapter 6 for examples and detailed information on reviewing cluster status.

Viewing the Status of Volume Groups

To display the current configuration of a shared volume group, use the `vgdisplay -v` command. An example is as follows:

    # vgdisplay -v /dev/vg_ops (Return)

The output includes a list of all volume groups, together with the logical volumes configured in them and all the physical volumes associated with them. Physical volume groups are also included.

Starting and Stopping the Cluster

During normal operation, the cluster functions continuously without intervention. When it becomes necessary to stop the entire cluster for such operations as replacing hardware or physically moving the nodes, you can manually halt the cluster and restart it at a later time. Stopping the cluster in this way has the effect of running the OPS halt scripts on each node to halt the operation of OPS and OPS applications as the cluster shuts down. Restarting the cluster has the effect of running the OPS start script on each node (if it is configured) to bring up Oracle Parallel Server and its applications.

Using SAM to Stop the Cluster

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, and select "Shut Down Cluster."
3. Respond Yes to the verification prompt.

Using HP-UX Commands to Stop the Cluster

To stop the entire cluster:

1. If you are running packages on your cluster, halt them.

    # cmhaltpkg pkg1 (Return)
2. Shutdown the OPS database.
3. If there are volume groups used by packages that have mounted file systems, unmount them.

    # umount /mnt1 (Return)
4. Deactivate each cluster-bound volume group.

    # vgchange -a n /dev/vg_ops (Return)
5. Use the `cmhaltcl` command from any one node to stop the entire cluster. `cmhaltcl` causes all nodes in a configured cluster to stop their MC/LockManager daemons.

    This command will halt all Oracle instances and stop all the MC/LockManager daemons on all currently running systems. If you only want to shut down a subset of nodes, the `cmhaltnode` command should be used instead.

Using SAM to Start the Cluster

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, and select "Start Cluster."
3. Select "Start Cluster on All Nodes."
4. Respond Yes to the verification prompt.

Using HP-UX Commands to Start the Cluster

Use the `cmruncl` command from any node to start the entire cluster. `cmruncl` causes all nodes in a configured cluster or all nodes specified to start their MC/LockManager daemons and form a new cluster. The command also runs OPS start scripts (if they are configured) to start up Oracle instances on each node.

Note	This command should only be run when the cluster is not active on any of the configured nodes. If a cluster is already running on a subset of the nodes, the `cmrunnode` command should be used to start the remaining nodes and force them to join the existing cluster.

If a node name is not specified in the `cmruncl` command line, the MC/LockManager daemons will be started on all the nodes in the cluster. All nodes in the cluster must be booted and available to run MC/LockManager.

Starting and Stopping Individual Nodes

At different times, you may need to remove a node from active cluster operation for maintenance or upgrade activities such as adding a peripheral, recabling or other activities. To do such maintenance, you remove the node from the cluster temporarily. This action does not change the cluster configuration. When the maintenance is finished, you return the node to cluster operation. During the maintenance of the inactive node, the cluster still operates using the other node.

Stopping a node in this way has the effect of running the OPS halt script (if it is configured) on that node to halt the operation of OPS and OPS

applications. Restarting the node has the effect of running the OPS start script (if it is configured) on that node to bring up Oracle Parallel Server and its applications.

Using SAM to Remove a Node from the Cluster Temporarily

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, and select "Specify Node to Leave the Cluster." From the select list, choose the node that is to be temporarily removed.
3. Respond Yes to the verification prompt.

This sequence of steps runs the OPS halt script to halt the Oracle instance on the specified node.

Using HP-UX Commands to Remove a Node from the Cluster Temporarily

1. Use the `cmhaltpkg` command to halt any running packages.
2. Stop OPS.
3. Use `vgchange -a n` *volumegroup* to deactivate volume groups used by OPS.
4. Use the `cmhaltnode` command on the node that is being removed.

The command also runs the OPS halt script to halt the Oracle instance on the specified node.

Using SAM to Return a Node to the Cluster

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, and select "Specify Node to Join the Cluster." From the select list, choose the node that is to be added back to the cluster.
3. Respond Yes to the verification prompt.

This sequence of steps runs the OPS start script to start the Oracle instance on the specified node.

Using HP-UX Commands to Return a Node to the Cluster

Use the `cmrunnode` command from the node that you wish to add back to the cluster. This command also runs the OPS start script (if it is configured) to start the Oracle instance on the specified node.

Administering Packages and Services

Administering packages and services involves the following tasks:

- Starting a Package
- Halting a Package
- Moving a Package
- Modifying a Package Configuration
- Reconfiguring a Package

You can use SAM or HP-UX commands to start, halt, and move packages. You can also modify package failover options without bringing down the cluster or package node.

Starting a Package

Ordinarily, a package configured as part of the cluster will start up on its primary node when the cluster starts up. You may need to start a package manually after it has been halted manually. You can do this either in SAM or with HP-UX commands.

Using SAM to Start a Package

In SAM, select "Package Administration," then choose the package you wish to start. From the "Actions" menu, choose "Start Package." If you wish to start the package on a specific node, choose "Start a Package on a Specific Node." Otherwise, choose "Start Package," and reply Yes to the verification prompt.

Using HP-UX Commands to Start a Package

Use the `cmrunpkg` command to run the package on a particular node, then use the `cmmodpkg` command to enable switching for the package. Example:

```
# cmrunpkg -n node1 pkg1  [Return]
# cmmodpkg -e node1 pkg1  [Return]
```

This starts up the package on *node 1*, then enables package switching. This
sequence is necessary when a package has previously been halted on some node,
since halting the package disables switching.

Halting a Package

You halt a package when you wish to bring the package out of use but wish the
node to continue in operation. You can halt a package using SAM or using
HP-UX commands. Halting a package has a different effect than halting the
node. When you halt the node, its packages may switch to adoptive nodes
(assuming that switching is enabled for them); when you halt the package, it is
disabled from switching to another node, and must be restarted manually on
another node or on the same node.

Using SAM to Halt a Package

In the SAM "Package Administration" area, choose a package from the list,
then select "Halt Package" from the Actions menu. Choose OK in response to
the verification prompt. When you halt the package in this way, it is disabled
from switching to another node.

Using HP-UX Commands to Halt a Package

Use the **cmhaltpkg** command to halt a package, as follows:

```
# cmhaltpkg pkg1  [Return]
```

This halts pkg1 and disables it from switching to another node.

Moving a Package

You can use SAM or HP-UX commands to move a package from one node to
another.

Using SAM to Move a Running Package

From the Package Administration screen in SAM, choose a package, then select "Move a Package" from the Actions menu. Choose the node you wish to move the package to, then select OK. Reply Yes to the verification prompt.

Using HP-UX Commands to Move a Running Package.

Before you move the package, halt it on its original node using the `cmhaltpkg` command. This action not only halts the package, but also disables switching the package back to the node on which it halts.

After you have moved the package you must restart it and enable switching. You can do this in SAM or by issuing the `cmrunpkg` command followed by `cmmodpkg -e` *package_name*. `cmmodpkg` can be used with the `-n` option to enable a package to run on a node if the package has been disabled from running on that node due to some sort of error. If no node is specified, the node the command is run on is the implied node.

Example:

```
# cmhaltpkg pkg1 -n node1 [Return]
# cmrunpkg -n node2 pkg1 [Return]
# cmmodpkg -e pkg1 [Return]
```

Reconfiguring the Package

To make a permanent change in package configuration, you must use the following steps:

1. Halt packages.
2. Stop OPS.
3. Deactivate volume groups used by OPS.
4. Halt the cluster on all nodes.
5. On one node, reconfigure the package as described in the chapter "Configuring the Package and Its Services." You can do this by editing the package ASCII file or by using the "Modify Package Configuration" options in the "High Availability Clusters" area in SAM.
6. To modify the package control script, edit the package control script directly or use the "Edit a Package Control Script" option in SAM. Any changes in service names will also require changes in the package configuration file.

7. Use SAM or HP-UX commands to copy the modified control script to all nodes that can run the package.
8. Use SAM or the `cmapplyconf` command to copy the binary cluster configuration file to all nodes. This file overwrites any previous version of the binary cluster configuration file.
9. Use SAM or the `cmruncl` command to start the cluster on all nodes or on a subset of nodes, as desired. The package will start up as nodes come online.

Responding to Cluster Events Affecting Packages

MC/LockManager does not require much on-going system administration intervention. As long as there are no failures, your cluster will be monitored and protected. In the event of a failure, those packages that you have designated to be transferred to another node will be transferred automatically. Your ongoing responsibility as the system administrator will be to monitor the cluster and determine if a transfer of package has occurred. If a transfer has occurred, you have to determine the cause and take corrective actions.

The typical corrective actions to take in the event of a transfer of package include:

■ Determining when a transfer has occurred.
■ Determining the cause of a transfer.
■ Repairing any hardware failures.
■ Correcting any software problems.
■ Restarting nodes.
■ Transferring packages back to their original nodes.

Changing the Permanent Cluster Configuration

If your network or LAN card configuration changes on the cluster, you may need to modify the basic cluster configuration. This can be done through SAM or with HP-UX commands. The basic process is as follows:

- Halt packages.
- Ensure that the OPS database is not active on either node.
- Deactivate and unshare any shared volume groups.
- Halt the cluster.
- Change the LAN configuration as needed.
- Using either SAM high availability options or HP-UX commands, create a new configuration file and propagate the new configuration to all the nodes in the cluster.
- Start up the cluster to see if it forms successfully.
- Reboot all nodes. The cluster should reform, and the OPS instances and packages should come up again.

For details about creating the configuration file, see the section "Configuring the Cluster Manager Software" in the chapter "Building an OPS Cluster Configuration."

Changing Lock Volume Group Configuration

If you decide to change the lock volume group in your configuration, you must issue the following commands on the lock volume group before reconfiguration will succeed:

```
# vgchange -S n -c n vg_ops  (Return)
# vgchange -a y vg_ops  (Return)
```

This process is needed when you modify the lock volume group configuration, although you can use the commands to convert any shared, cluster-bound LockManager volume group into a standard (non-shareable) LVM volume group.

Changing Oracle Parameters

When the Oracle DBA adjusts certain Oracle parameters, it may be necessary to adjust DLM parameters accordingly. For example, when the GC_DB_LOCKS parameter increases, you should change the internal DLM Resources parameter that is based on it. To make changed Oracle parameters effective, you need to bring down and restart each Oracle instance. To make DLM parameters effective, you must bring down and reconfigure the cluster. See "Configuring the Distributed Lock Manager Software" in Chapter 3 for details.

Making Volume Groups Shareable (HP-UX Commands Only)

Normally, volume groups are marked to be activated in shared mode when they are listed with the DLM_VOLUME_GROUP parameter in the cluster configuration file or in SAM. However, in some cases you may want to manually make a volume group sharable. For example, if you wish to add a new shared volume group without shutting down the cluster, you can use the manual method to do it online. However, when convenient, it's a good practice to bring down the cluster and reconfigure it to include the new volume group.

1. Use the `vgchange` command on each node to ensure that the volume group to be shared is currently inactive on all nodes. Example:

   `# vgchange -a n /dev/vg_ops` `Return`

2. On the configuration node, use the `vgchange` command to make the volume group shareable by members of the cluster:

   `# vgchange -S y -c y /dev/vg_ops` `Return`

   This command is issued from the configuration node only, and the cluster must be running on all nodes for the command to succeed. Note that both the `-S` and the `-c` options are specified. The `-S y` option makes the volume group shareable, and the `-c y` option causes the cluster id to be written out to all the disks in the volume group. In effect, this command specifies the cluster to which a node must belong in order to obtain shared access to the volume group.

Making a Volume Group Unshareable

If you wish to unmark a previously marked shared volume group:

1. Remove the volume group name from the ASCII cluster configuration file.
2. Enter

      vgchange -S n -c n */dev/volumegroup*

The above example marks the volume group as non-shared and not associated with a cluster.

Activating a Volume Group in Shared Mode (HP-UX Commands Only)

Activation and deactivation of shared volume groups can be done through the DLM control script (runhalt.sh). If you need to perform activation from the command line, you can issue the following command from each node to activate the volume group in shared mode. (The node on which you first enter the command becomes the server node.)

    # vgchange -a s -p /dev/vg_ops [Return]

The following message is displayed:

```
Activated volume group in shared mode.
This node is the Server.
```

When the same command is entered on the second node, the following message is displayed:

```
Activated volume group in shared mode.
This node is a Client.
```

Note	Do *not* share volume groups that are not part of the OPS configuration.

Deactivating a Shared Volume Group

Issue the following command from each node to deactivate the shared volume group:

    # vgchange -a n /dev/vg_ops (Return)

Remember that volume groups remain shareable even when nodes enter and leave the cluster.

Note   If you wish to change the capacity of a volume group at a later time, you must deactivate and unshare the volume group first. If you add disks, you must specify the appropriate physical volume group name and make sure the /etc/lvmpvg file is correctly updated on both nodes.

Making Changes to Shared Volume Groups (HP-UX Commands Only)

You may need to change the volume group configuration of OPS shared logical volumes to add capacity to the data files or to add log files. No configuration changes are allowed on shared volume groups while they are activated. The volume group must be deactivated first on all nodes, and marked as non-shareable. Use the following procedure (examples assume the volume group is being shared by node 1 and node 2, and they use the volume group vg_ops):

1. Ensure that the OPS database is not active on either node.
2. From node 2, use the **vgchange** command to deactivate the volume group:

    # vgchange -a n /dev/vg_ops (Return)

3. From node 2, use the **vgexport** command to export the volume group:

    # vgexport -m /tmp/vg_ops.map.old /dev/vg_ops (Return)

    This dissociates the volume group from node 2.
4. From node 1, use the **vgchange** command to deactivate the volume group:

    # vgchange -a n /dev/vg_ops (Return)

5. Use the `vgchange` command to mark the volume group as unshareable:

       # vgchange -S n -c n /dev/vg_ops [Return]

6. Prior to making configuration changes, activate the volume group in normal (non-shared) mode:

       # vgchange -a y /dev/vg_ops [Return]

7. Use normal LVM commands to make the needed changes. If you add physical disks to a volume group, make sure that mirror copies are added to the correct physical volume groups. Be sure to set the raw logical volume device file's owner to *oracle* and group to *dba*, with a mode of 660.

8. Next, still from node 1, deactivate the volume group:

       # vgchange -a n /dev/vg_ops [Return]

9. Use the `vgexport` command with the options shown in the example to create a new map file:

       # vgexport -p -m /tmp/vg_ops.map /dev/vg_ops [Return]

   Make a copy of /etc/lvmpvg in /tmp/lvmpvg, then copy the file to /tmp/lvmpvg on node 2. Copy the file /tmp/vg_ops.map to node 2.

10. Use the following command to make the volume group shareable by the entire cluster again:

        # vgchange -S y -c y /dev/vg_ops [Return]

11. On node 2, issue the following command:

        # mkdir /dev/vg_ops [Return]

12. Create a control file named *group* in the directory /dev/vg_ops, as in the following:

        # mknod /dev/vg_ops/group c 64 0x*hh*0000 [Return]

    The major number is always 64, and the hexadecimal minor number has the form

        0xhh0000

    where *hh* must be unique to the volume group you are creating. Use the next hexadecimal number that is available on your system, after the volume groups that are already configured.

13. Use the `vgimport` command, specifying the map file you copied from the configuration node. In the following example, the `vgimport` command is

issued on the second node for the same volume group that was modified on the first node:

```
# vgimport -v -m /tmp/vg_ops.map /dev/vg_ops /dev/dsk/c0t2d0 \
    /dev/dsk/c1t2d0 Return
```

14. Review the content of /etc/lvmpvg on the second node to ensure that all disks are correctly named and that any added disks are included in the correct physical volume groups. Use the /tmp/lvmpvg file copied from the first as a reference.
15. Activate the volume group in shared mode by issuing the following command on both nodes:

```
# vgchange -a s -p /dev/vg_ops  Return
```

Skip this step if you use the OPS runhalt script to activate and deactivate the shared volume group as a part of OPS startup and shutdown.

Adding Additional Shared Volume Groups

To add capacity or to organize your disk resources for ease of management, you may wish to create additional shared volume groups for your OPS databases. If you decide to use additional shared volume groups, they must conform to the following rules:

- Mirror copies of a logical volume must be between disks that are connected to different busses, and assigned to different physical volume groups.
- All nodes in the cluster must have an /etc/lvmpvg file that identifies the physical volume groups in use. The physical volumes in each physical volume group must be the same disks on all nodes, even if the physical volume name is different.
- Volume group names must be the same on all nodes in the cluster.
- Logical volume names must be the same on all nodes in the cluster.

Adding Additional Disk Hardware

As your system expands, you may need to add disk hardware. This also means modifying the logical volume structure. Changing the logical volume structure cannot be done with SAM; the procedure for adding disks to your system is as follows using HP-UX commands:

1. Halt packages.
2. Ensure that the OPS database is not active on either node.
3. Deactivate and mark as unshareable any shared volume groups.
4. Halt the cluster.
5. Deactivate automatic cluster startup.
6. Shutdown and power off system before installing new hardware.
7. Install the new disk hardware with connections on all nodes.
8. Reboot all nodes.
9. On the configuration node, add the new physical volumes to existing volume groups, or create new volume groups as needed. Use physical volume groups to ensure that mirroring is done between disks that are attached to different busses.
10. Start up the cluster.
11. Make the volume groups shareable, then import each shareable volume group onto the other nodes in the cluster.
12. Activate the volume groups in shared mode on all nodes.
13. Start up the OPS instances on all nodes.
14. Activate automatic cluster startup.

Note   As you add new disks to the system, update the planning worksheets (described in the chapter "Planning and Documenting an OPS Cluster") so as to record the exact configuration you are using.

6

Troubleshooting Your Cluster

This chapter describes how to review cluster status, and some approaches to troubleshooting. Topics are as follows:

- Troubleshooting Approaches
- Testing Cluster Halt and Reconfiguration
- Solving Package Problems

Troubleshooting Approaches

The following sections offer a few suggestions for troubleshooting by reviewing the state of the running system and by examining cluster status data, log files, and configuration files. Topics include:

- Reviewing Cluster and Package States.
- Reviewing RS232 Status
- Reviewing Package IP addresses
- Reviewing the System Log File.
- Reviewing Configuration Files.
- Reviewing the Package Control Script.
- Using `cmquerycl` and `cmcheckconf`
- Reviewing the LAN Configuration
- Reviewing the Status of Shared Volume Groups
- Using DLM Diagnostic Tools
- Understanding Messages and Message Logs

Reviewing Cluster and Package States

A cluster or its component nodes may be in several different states at different points in time. Status information for clusters, packages and other cluster elements is shown in the output of the `cmviewcl` command and in some displays in SAM. This section explains the meaning of many of the common conditions the cluster or package may be in.

You can examine the status of all cluster nodes in SAM or by using HP-UX commands.

Using SAM to View Cluster Status

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, and open the Action Menu. Select an appropriate View option.

SAM provides three View options:

- View Cluster Network Configuration
- View Cluster Node States
- View Syslog File

Using HP-UX Commands to View Cluster Status

Information about cluster status is stored in the status database, which is maintained on each individual node in the cluster. You can display information contained in this database by issuing the `cmviewcl` command:

> **#** <u>cmviewcl -v</u> Return

The command when issued with the `-v` option displays information about the whole cluster. See the man page for a detailed description of other `cmviewcl` options.

Cluster States

The *status* of a cluster may be one of the following:

- Up. At least one node has a running cluster daemon, and reconfiguration is not taking place.
- Down. No cluster daemons are running on any cluster node.

■ Reforming. The cluster is in the process of determining its active membership. At least one cluster daemon is ruuning.

Node States

The *status* of a node is either up or down, depending on whether its cluster daemon is running or not. A node may also be in one of the following states:

■ Initializing. A node sees itself in this state after its daemon has started, but before it is ready to communicate with other nodes' daemons. Other nodes never see a node in this state.
■ Failed. A node never sees itself in this state. Other active members of the cluster will see a node in this state if that node was in an active cluster, but is no longer, and is not halted.
■ Cluster Reforming. A node in this state is running the protocols which ensure that all nodes agree to the new membership of an active cluster. If agreement is reached, the status database is updated to reflect the new cluster membership.
■ Running. A node in this state has completed all required activity for the last re-formation and is operating normally.
■ Halted. A node never sees itself in this state. Other nodes will see it in this state after the node has gracefully left the active cluster, for instance with a `cmhaltnode` command.
■ Unknown. A node never sees itself in this state. Other nodes assign a node this state if it has never been an active cluster member.

Package States

The *status* of a package can be one of the following:

■ Up. The package control script is active.
■ Down. The package control script is not active.
■ Unknown.

The *state* of the package can be one of the following:

■ Starting. The start instructions in the control script are being run.
■ Running. Services are active and being monitored.
■ Halting. The halt instructions in the control script are being run.
■ Not Owned. A package has stopped, and has been disabled from switching to another node.

Packages also have the following switching attributes:

- Package Switching. Enabled means that the package can switch to another node in the event of failure.
- Switching Enabled for a Node. Enabled means that the package can switch to the referenced node. Disabled means that the package cannot switch to the specified node until the node is enabled for the package using the `cmmodpkg` command.

  Every package is marked Enabled or Disabled for each node that is either a primary or adoptive node for the package.

Service States

Services have only status, as follows:

- Up. The service is being monitored.
- Down. The service is not being monitored. It may have halted or failed.
- Unknown.

Examples of Cluster and Package States

The following sample output from the `cmviewcl -v` command shows status for the cluster in the sample configuration.

Normal Running Status. Everything is running normally; both nodes in the
cluster are running, and the packages are in their primary locations.

```
CLUSTER         STATUS
example         up
  NODE            STATUS        STATE
  node1           up            running

    Network_Parameters:
    INTERFACE     STATUS        PATH          NAME
    PRIMARY       up            56/36.1       lan0
    STANDBY       up            60/6          lan1

    PACKAGE        STATUS       STATE         PKG_SWITCH    NODE
    pkg1           up           running       enabled       node1

      Script_Parameters:
      ITEM           STATUS        NAME          MAX_RESTARTS    RESTARTS
      Service        up            service1                 0           0
      Subnet         up            15.13.168.0              0           0

      Node_Switching_Parameters:
      NODE_TYPE      STATUS        SWITCHING     NAME
      Primary        up            enabled       node1         (current)
      Alternate      up            enabled       node2
  NODE            STATUS        STATE
  node2           up            running

    Network_Parameters:
    INTERFACE     STATUS        PATH          NAME
    PRIMARY       up            28.1          lan0
    STANDBY       up            32.1          lan1

    PACKAGE        STATUS       STATE         PKG_SWITCH    NODE
    pkg2           up           running       enabled       node2

      Script_Parameters:
      ITEM           STATUS        NAME          MAX_RESTARTS    RESTARTS
      Service        up            service2                 0           0
      Subnet         up            15.13.168.0              0           0

      Node_Switching_Parameters:
      NODE_TYPE      STATUS        SWITCHING     NAME
      Primary        up            enabled       node2         (current)
      Alternate      up            enabled       node1
```

Status After Halting a Package. After halting pkg2 with the `cmhaltpkg` command, the output of `cmviewcl -v` is as follows:

```
CLUSTER         STATUS
example         up

  NODE            STATUS       STATE
  node1           up           running

    Network_Parameters:
    INTERFACE     STATUS       PATH          NAME
    PRIMARY       up           56/36.1       lan0
    STANDBY       up           60/6          lan1

    PACKAGE       STATUS       STATE         PKG_SWITCH     NODE
    pkg1          up           running       enabled        node1

      Script_Parameters:
      ITEM          STATUS       NAME            MAX_RESTARTS   RESTARTS
      Service       up           service1                  0          0
      Subnet        up           15.13.168.0               0          0

      Node_Switching_Parameters:
      NODE_TYPE     STATUS       SWITCHING     NAME
      Primary       up           enabled       node1        (current)
      Alternate     up           enabled       node2


  NODE            STATUS       STATE
  node2           up           running

    Network_Parameters:
    INTERFACE     STATUS       PATH          NAME
    PRIMARY       up           28.1          lan0
    STANDBY       up           32.1          lan1

UNOWNED_PACKAGES

    PACKAGE       STATUS       STATE         PKG_SWITCH     NODE
    pkg2          down         unowned       disabled       unowned

      Node_Switching_Parameters:
      NODE_TYPE     STATUS       SWITCHING     NAME
      Primary       up           enabled       node2
      Alternate     up           enabled       node1
```

Pkg2 now has the status "down," and it is shown as in the unowned state, with package switching disabled. Note that switching is enabled for both nodes, however. This means that once global switching is re-enabled for the package, it will attempt to start up on the primary node.

Status After Moving the Package to Another Node. After issuing the following command:

<u>cmrunpkg -n node1 pkg2</u> [Return]

the output of the cmviewcl -v command is as follows:

```
CLUSTER       STATUS
example       up

 NODE         STATUS      STATE
 node1        up          running

   Network_Parameters:
   INTERFACE   STATUS      PATH        NAME
   PRIMARY     up          56/36.1     lan0
   STANDBY     up          60/6        lan1


   PACKAGE      STATUS      STATE       PKG_SWITCH   NODE
   pkg1         up          running     enabled      node1

     Script_Parameters:
     ITEM        STATUS      NAME             MAX_RESTARTS    RESTARTS
     Service     up          service1.1             0               0
     Subnet      up          15.13.168.0            0               0

     Node_Switching_Parameters:
     NODE_TYPE   STATUS      SWITCHING    NAME
     Primary     up          enabled      node1       (current)
     Alternate   up          enabled      node2

   PACKAGE      STATUS      STATE       PKG_SWITCH   NODE
   pkg2         up          running     disabled     node1

     Script_Parameters:
     ITEM        STATUS      NAME             MAX_RESTARTS    RESTARTS
     Service     up          service2.1             0               0
     Subnet      up          15.13.168.0            0               0
```

```
Node_Switching_Parameters:
NODE_TYPE    STATUS        SWITCHING     NAME
Primary      up            enabled       node2
Alternate    up            enabled       node1      (current)

NODE           STATUS        STATE
node2          up            running

Network_Parameters:
INTERFACE      STATUS        PATH          NAME
PRIMARY        up            28.1          lan0
STANDBY        up            32.1          lan1
```

Now pkg2 is running on node 1. Note that it is still disabled from switching.

Status After Package Switching is Enabled. The following command changes package status back to Package Switching Enabled:

    # <u>cmmodpkg -e pkg2</u> [Return]

The result is now as follows:

```
CLUSTER        STATUS
example        up

NODE           STATUS        STATE
node1          up            running

  PACKAGE        STATUS        STATE         PKG_SWITCH     NODE
  pkg1           up            running       enabled        node1
  pkg2           up            running       enabled        node1

NODE           STATUS        STATE
node2          up            running
```

Both packages are now running on node 1 and pkg2 is enabled for switching. Node 2 is running the daemon and no packages are running on node 2.

Status After Halting a Node. After halting *node 2*, with the following command:

    # <u>cmhaltnode node2</u> (Return)

the output of `cmviewcl` is as follows on *node 1*:

```
CLUSTER        STATUS
example        up

  NODE          STATUS         STATE
  node1         up             running

    PACKAGE       STATUS        STATE       PKG_SWITCH   NODE
    pkg1          up            running     enabled      node1
    pkg2          up            running     enabled      node1

  NODE          STATUS         STATE
  node2         down           halted
```

This output is seen only on *node 1*; on *node 2*, the output of `cmviewcl` is as follows:

```
CLUSTER        STATUS
               down

cmviewcl: Local node node2 is not currently
an active member of a cluster.
```

Reviewing RS232 Status

If you are using a serial (RS232) line as a heartbeat connection, you will
see a list of configured RS232 device files in the output of the `cmviewcl -v`
command:

```
CLUSTER        STATUS
example        up
  NODE          STATUS        STATE
  node1         up            running

    Network_Parameters:
    INTERFACE    STATUS        PATH          NAME
    PRIMARY      up            56/36.1       lan0
    STANDBY      up            60/6          lan1

    Serial Heartbeat:
    DEVICE FILE NAME      CONNECTED TO:
    /dev/tty0p0           node2     /dev/tty0p0
```

Reviewing Package IP Addresses

The `netstat -in` command can be used to examine the LAN configuration.
The command, if executed on *node 1* after the halting of *node 2*, shows that
the package IP addresses are assigned to lan0 on node 1 along with the
heartbeat IP address.

```
Name   Mtu   Network       Address          Ipkts  Ierrs    Opkts Oerrs  Coll
ni0*   0     none          none                 0      0        0     0      0
ni1*   0     none          none                 0      0        0     0      0
lo0    4608  127           127.0.0.1        10114      0    10114     0      0
lan0   1500  15.13.168     15.13.171.14    959269      2   305189    47  30538
lan0   1500  15.13.168     15.13.171.23    959269      2   305189    47  30538
lan0   1500  15.13.168     15.13.171.20    959269      2   305189    47  30538
lan1*  1500  none          none            418623     27    41716     3   5149
```

Reviewing Configuration Files

Review the following configuration files:

- Cluster configuration file `/etc/cmcluster/cmclconf.asc`.
- Package configuration files. For each package, the file is called `/etc/cmcluster/`*package_name*`/`*package_name*`conf.asc`.

Reviewing the Package Control Script

Ensure that the package control script is found on all nodes where the package can run and that the file is identical on all nodes. Ensure that the script is executable on all nodes. Ensure that the name of the control script appears in the package configuration file, and ensure that all services named in the package configuration file also appear in the package control script.

Information about the starting and halting of each package is found in the package's control script log. This log provides the history of the operation of the package control script. It is found at `/etc/cmcluster/`*package_name*`/control.sh.log`. This log documents all package run and halt activities. If you have written a separate run and halt script script for a package, each script will have its own log.

Using `cmquerycl` and `cmcheckconf`

In addition, `cmquerycl` and `cmcheckconf` can be used to troubleshoot your cluster just as they were used to verify its configuration. The following example shows the commands used to verify the existing cluster configuration on *node 1* and *node 2*:

```
# cmquerycl -v -C /etc/cmcluster/verify.asc -n node1 -n node2 (Return)
# cmcheckconf -v -C /etc/cmcluster/verify.asc (Return)
```

The `cmcheckconf` command checks:

- The network addresses and connections.
- The cluster lock connectivity.
- The validity of configuration parameters of the cluster and packages for:
  - The uniqueness of names.
  - The existence and permission of scripts.

It doesn't check:

- The correct setup of the power circuits.
- The correctness of the package configuration script.

Reviewing the LAN Configuration

The following networking commands can be used to diagnose problems:

- `netstat -in` can be used to examine the LAN configuration. This command lists all IP addresses assigned to each LAN interface card.
- `lanscan` can also be used to examine the LAN configuration. This command lists the MAC addresses and status of all LAN interface cards on the node.
- `arp -a` can be used to check the arp tables.
- `landiag` is useful to display, diagnose, and reset LAN card information.
- `linkloop` verifies the communication between LAN cards at MAC address levels. For example, if you enter `looplink -i4 0x08000993AB72`, you should see displayed the message "Link Connectivity to LAN station: 0x08000993AB72—OK"
- `/usr/contrib/bin/cmgetconfig -f` can be used to verify that Primary and Standby LANs are on the same bridged net.
- `cmviewcl` -v shows the status of primary and standby LANs.

Use these commands on all nodes.

Reviewing the Status of Shared Volume Groups

To display the current configuration of a shared volume group, use the `vgdisplay -v` command. An example is as follows:

```
# vgdisplay -v /dev/vg_ops (Return)
```

The output includes a list of all volume groups, together with the logical volumes configured in them and all the physical volumes associated with them. Physical volume groups are also included.

Using DLM Diagnostic Tools

MC/LockManager software includes a group of diagnostic tools that may be helpful in troubleshooting. Use these tools in cooperation with your HP representative or technical consultant. Refer also to the man page for each command.

dlmdump

`dlmdump` is a tool that dumps DLM-related memory structures. `dlmdump` is used for debugging purposes. It allows the user to obtain a snapshot about an object given its handle. Since `dlmdump` only provides a "snapshot" of the current state, what is displayed might not be completely consistent with the actual data.

dlmstat

`dlmstat` is a tool that tracks DLM-related statistical information. `dlmstat` is used to acquire statistics for the process, resource, instance, and cluster objects from the DLM database.

For example,

```
dlmstat -i -q -t +1 -n 10 Return
```

Core Dump Locations

Core dumps for the *cmcld* and *cmlvmd* daemons are produced in the /var/adm/cmcluster and /etc/lvmconf directories, respectively. The DLM daemons deposit dumps in the *cores* subdirectory of the *dlm* home directory.

Understanding Messages and Message Logs

All the components of MC/LockManager produce messages at different times indicating the completion of a step or an error or warning condition. Messages generated by SAM are displayed to the user in a message box; messages from HP-UX commands are normally displayed on the standard output; some information may also be written to different log files, depending on which software component is generating the message. Messages from the cluster manager are found in the system log file, `/var/adm/syslog/syslog.log`.

Messages from the distributed lock manager are placed in files in a subdirectory of the home directory of the *dlm* user, as well as being sent to /var/adm/syslog/syslog.log.

Messages Written to the System Log File

Messages from the Cluster Manager and Package Manager are written to the system log file. Each message is accompanied by a timestamp showing the date and time the message was written out and the name of the process that generated the message. The default location of the log file is /var/adm/syslog/syslog.log.

You can distinguish messages from the following daemon processes:

■ cmcld - CM daemon
■ cmclconfd - CM cluster configuration daemon
■ cmlvmd - SLVM daemon
■ dlm - DLM daemons and clients

You can examine the syslog.log file periodically for messages relating to the configuration. In SAM, use the following steps:

1. Run SAM, and choose the High Availability options.
2. Choose Cluster Administration, then select "View Syslog File" from the Cluster Administration Actions menu.

You can also browse the syslog file directly:

   # more /var/adm/syslog/syslog.log (Return)

The cluster manager employs several types of messages to convey information about the running system. Each message is accompanied by a prefix that identifies the message type. The categories are as follows:

LOG_INTERNAL   This type of message is used to log ongoing processes
                   occurring within the MC/LockManager software or one
                   of its related commands.

LOG_EXTERNAL   This type of message indicates that there has been a
                   change in the condition of some piece of hardware or
                   software outside MC/LockManager itself. Examples: a
                   LAN card fails, or a node comes back into the cluster.

LOG_PERIODIC	This type of message is a special case of the LOG_INTERNAL category. Periodic messages report those events or actions which occur all the time, whether or not a problem or change is detected in the cluster.
LOG_ERROR	This type of message is used to report incorrect MC/LockManager behavior, which may be related to the inability to obtain system resources or other problems within MC/LockManager.
LOG_DEATH	This type of message accompanies the death of a daemon process.

Messages are intended to be self-explanatory, but occasionally it is necessary to study several messages together in context to determine the appropriate corrective action. In some cases, no action is required because the message is purely informative, as when a message reports successful completion of a task. In other cases, the only action may be to gather information from the running system for use in diagnosis of the problem by HP field personnel.

Messages Written to the DLM Log Directory

The following DLM daemons produce messages:

- cmlkmgrd - DLM configuration daemon
- cmdlmd - DLM daemon
- cmdlmmon - DLM monitor daemon

These daemon processes direct their messages to the *logs* directory inside the *dlm* home directory. There are two log files that contain messages produced by the DLM daemons (and client processes attached to the DLM):

- dlmstart.log. This file contains messages from the DLM daemons produced during startup.
- dlm.log. This file contains messages from the DLM daemons produced during normal operation, reconfiguration and shutdown.

Important DLM messages are also directed to /var/adm/syslog/syslog.log.

List of DLM Error Messages

See Appendix E for a listing of all DLM error messages, together with a probable cause for the error condition, and the action you should take to eliminate the problem.

Testing Cluster Halt and Reconfiguration

This section shows how to test the correct reconfiguration of the cluster after the loss of a node. To do this in SAM, read the next section. If you want to test reconfiguration using HP-UX commands, skip ahead to the section entitled "Using HP-UX Commands to Test Cluster Halt and Reconfiguration."

Using SAM to Test Cluster Halt and Reconfiguration

Perform the following steps:

1. From SAM, select the High Availability options, then choose Cluster Administration.
2. In the Cluster Administration area, choose Specify Node(s) to Leave the Cluster, and then select one node to halt. Reply Yes when asked for a verification.
3. When the Cluster Administration area screen reappears, make sure the selected node is no longer an active part of the cluster.
4. From the Action list, choose View Syslog. Read the messages to verify that the reconfiguration has taken place.
5. From the Action list, select Specify Node(s) to Leave the Cluster, and choose the other node. Reply Yes when asked for a verification.
6. In the Process Management area of SAM, observe that none of the following daemon processes is running:
   - `cmcld` - CM daemon
   - `cmdlmd` - DLM daemon
   - `cmdlmmond` - DLM monitor daemon
   - `cmlvmd` - SLVM daemon

   These are shown as children of the `init` process.

To start the cluster running again, in the Cluster Administration area, choose Start Cluster, and select All Nodes.

Using HP-UX Commands to Test Cluster Halt and Reconfiguration

To test the correct reconfiguration of the cluster following the loss of a node, issue the following sequence of commands on node 1:

```
# vgchange -a n /dev/vg_ops (Return)
# cmhaltnode -v (Return)
```

This stops node 1. Use the following command on node 2 after waiting about a minute for the reconfiguration to take place:

```
# cmviewcl -v (Return)
```

The output of the command should show that the cluster has reconfigured with only a single node.

Note   If you issue the `cmviewcl -v` command on node 1 after halting node 1, you will see the following message:

```
CLUSTER        STATUS
cluster1       down
```

If the cluster is running, always be sure to issue the `cmviewcl` command on a node that is an active participant in the running cluster.

Use the following command on node 2 to halt the second node:

```
# vgchange -a n /dev/vg_ops (Return)
# cmhaltnode -v (Return)
```

The use of `cmviewcl -v` on either node should now show that no cluster is active. At this point, you can use the `ps -ef` command on both nodes to show that the following processes no longer exist:

- `cmcld` - CM daemon
- `cmdlmd` - DLM daemon
- `cmdlmmond` - DLM monitor daemon
- `cmlvmd` - SLVM daemon

To start the cluster is running again. Use the following commands on each node:

```
# cmrunnode -v (Return)
# vgchange -a s -p /dev/vg_ops (Return)
```

Solving Package Problems

Problems with packages fall into three categories:

- System administration errors.
- Package movement errors.
- Node and network failures.

The first two categories of problems occur with the incorrect configuration of MC/LockManager. The last category contains "normal" failures to which MC/LockManager is designed to react and ensure the availability of packages containing your applications.

System Administration Errors

There are a number of errors you can make when configuring MC/LockManager that will not show up when you start the cluster. Your cluster can be running, and everything appears to be fine, until there is a hardware or software failure and control of your packages are not transferred to another node as you would have expected.

These are errors caused specifically by errors in the cluster configuration file and package configuration scripts. Examples of these errors include:

- Volume groups not defined on adoptive node.
- Mount point does not exist on adoptive node.
- Network errors on adoptive node (configuration errors).
- User information not correct on adoptive node.

You can use the following commands to check the status of your disks:

- bdf - to see if your package's volume group is mounted.
- vgdisplay -v - to see if all volumes are present.
- lvdisplay -v - to see if the mirrors are synchronized.

- `strings /etc/lvmtab` - to ensure that the configuration is correct.
- `ioscan -fnC disk` - to see physical disks.
- `diskinfo -v /dev/rdsk/cxtydz` - to display information about a disk.

Package Movement Errors

These errors are similar to the system administration errors except they are caused specifically by errors in the package control script. The best way to prevent these errors is to test your package control script before putting your high availability application on line.

Running your script with the `-x` shell option will give you details on where your script may be failing.

Node and Network Failures

Node and network failures cause MC/LockManager to transfer control of a package to another node. This is the normal action of MC/LockManager, but you have to be able to recognize when a transfer has taken place and decide to leave the cluster is its current condition or to restore it to its original condition.

Possible node failures can be caused by the following conditions:

- HPMC.
- TOC.
- Panics.
- Hangs.
- Power failures.

In the event of a TOC, a system dump is performed on the failed node and numerous messages are also displayed on the console.

You can use the following commands to check the status of your network and subnets:

- `netstat -in` - to display LAN status and check to see if the package IP is stacked on the LAN card.
- `lanscan` - to see if the LAN is on the primary interface or has switched to the standby interface.
- `arp -a` - to check the arp tables.
- `landiag` - to display, test, and reset the LAN cards.

Since your cluster is unique, there are no cookbook solutions to possible problems. But if you apply these checks and commands and work your way through the log files, you will be successful in identifying and solving problems.

A

Moving from HP-UX 9.04 to HP-UX 10.10

If you are currently running an OPS cluster on the HP-UX 9.04 release, this appendix describes the procedure for moving forward to the HP-UX 10.10 release.

Before Converting . . .

Before carrying out the conversion to MC/LockManager 10.10, the following are suggested:

- Make copies of all OPS/DLM scripts and configuration files.
- Halt OPS instances on both nodes.
- Halt the cluster on both nodes.
- Create a complete system backup.
- Review the hardware configuration to ensure that all hardware is fully supported in the MC/LockManager 10.10 environment.
- Review and update all the planning worksheets created for the existing cluster.
- Mark the volume groups as not shared and not associated with a cluster. Example:

      # vgchange -c n -S n /dev/vg_ops ⌊Return⌋

Upgrading the Operating System

Before you can convert the cluster to the 10.10 version, you must migrate the HP-UX operating system forward to HP-UX 10.10. This must be done in several stages *on both nodes*:

1. Install the HP-UX 10.0 software
2. Run the upgrade software, which modifies files and scripts for the 10.0 operating system. Reboot as needed.
3. Update to HP-UX 10.01. Reboot as needed.
4. Update to HP-UX 10.10. Reboot as needed.
5. Ensure that security files are in place.
6. Ensure that NTP has been configured.

Conversion Process

The following steps are carried out after the HP-UX 10.10 system is up and running correctly in all respects *on both nodes*.

1. Install MC/LockManager release A.10.00.
2. Apply any required HP-UX SLVM patches. Rebuild the kernel and reboot as needed.
3. Install Oracle 7.3 using the Oracle *installer*.
4. Apply any required Oracle patches.
5. Use the `cmquerycl` command to create a new cluster configuration file. Edit the file to include DLM parameters.
6. Use the following steps to prepare your converted OPS volume groups for cluster use. On the configuration node:

   ```
   # vgchange -a y /dev/vg_ops  Return
   # vgexport -p -m /tmp/vg_ops.map /dev/vg_ops  Return
   ```

   Make a copy of /etc/lvmpvg in /tmp/lvmpvg, then copy the file to /tmp/lvmpvg on node 2. Copy the file /tmp/vg_ops.map to node 2. On node 2:

   ```
   $ vgimport -v -m /tmp/vg_ops.map /dev/vg_ops /dev/dsk/c0t2d0  \
       /dev/dsk/c1t2d0  Return
   ```

   On the configuration node:

```
# vgchange -a n /dev/vg_ops [Return]
# vgchange -c y -S y /dev/vg_ops [Return]
```
7. Use the `cmapplyconf` command to complete the configuration and copy the binary cluster configuration file to the other node.
8. If you wish the node to join the cluster automatically at bootup, edit the /etc/rc.config.d/cmcluster file to set AUTOSTART_CMCLD to 1:

```
#************************** CMCLUSTER **************************

# Highly Available Cluster configuration
#
# @(#) $Revision: 76.1 $
#
# AUTOSTART_CMCLD:      If set to 1, the node will attempt to
#                       join its CM cluster automatically when
#                       the system boots.
#                       If set to 0, the node will not attempt
#                       to join its CM cluster.
#
AUTOSTART_CMCLD=1
#
# exit
```

B

Blank Planning Worksheets

This appendix reprints blank versions of the planning worksheets described in the "Planning" chapter. You can duplicate any of these worksheets that you find useful and fill them in as a part of the planning process.

```
            HARDWARE WORKSHEET                      Page ___ of ____
================================================================================
Node Information:


S800 Host Name _____     S800 Series No _____


Memory Capacity _____     Number of I/O Slots _____
================================================================================
LAN Information:

Name of         Name of         IP              Traffic
Subnet _____  Interface _____  Addr_____  Type  _____


Name of         Name of         IP              Traffic
Subnet _____  Interface _____  Addr_____  Type  _____


Name of         Name of         IP              Traffic
Subnet _____  Interface _____  Addr_____  Type  _____


================================================================================


Serial Heartbeat Interface Information:

Node Name _____     RS232 Device File _____

Node Name _____     RS232 Device File _____


================================================================================
Disk I/O Information:


                Hardware              Device
Bus Type _____  Path   _____  File Name  _____


                Hardware              Device
Bus Type _____  Path   _____  File Name  _____


                Hardware              Device
Bus Type _____  Path   _____  File Name  _____


Attach a printout of the output from ioscan -f and lssf /dev/*dsk/*s2
after installing disk hardware and rebooting the system.  Mark this
printout to indicate which physical volume group each disk belongs to.
```

Figure B-1. Blank Worksheet for Hardware Planning

B-2   Blank Planning Worksheets

```
        POWER SUPPLY WORKSHEET                        Page ___ of ____
===============================================================================
SPU Power:

S800 Host Name _____        Power Supply _____

S800 Host Name _____        Power Supply _____


===============================================================================
Disk Power:

Disk Unit _____        Power Supply _____

Disk Unit _____        Power Supply _____

Disk Unit _____        Power Supply _____

Disk Unit _____        Power Supply _____

Disk Unit _____        Power Supply _____

Disk Unit _____        Power Supply _____


===============================================================================
Tape Backup Power:

Tape Unit _____        Power Supply _____

Tape Unit _____        Power Supply _____


===============================================================================
Other Power:

Unit Name _____        Power Supply _____

Unit Name _____        Power Supply _____
```

Figure B-2. Blank Worksheet for Power Supplies

```
OPS PHYSICAL VOLUME WORKSHEET                    Page ___ of ____
================================================================================

Volume Group Name:          ------------------------------------------------

Name of First Physical Volume Group:  ------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Name of Second Physical Volume Group: ------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------

Physical Volume Name:       ------------------------------------------------
```

Figure B-3.
Blank Worksheet for Physical Volumes and Physical Volume Groups

B-4   Blank Planning Worksheets

```
OPS LOGICAL VOLUME WORKSHEET                    Page ___ of ____
==============================================================================
                         NAME                              SIZE

Oracle Control File 1:   ----------------------------------------------------

Oracle Control File 2:   ----------------------------------------------------

Oracle Control File 3:   ----------------------------------------------------

Instance 1 Redo Log 1:   ----------------------------------------------------

Instance 1 Redo Log 2:   ----------------------------------------------------

Instance 1 Redo Log 3:   ----------------------------------------------------

Instance 1 Redo Log:     ----------------------------------------------------

Instance 1 Redo Log:     ----------------------------------------------------

Instance 2 Redo Log 1:   ----------------------------------------------------

Instance 2 Redo Log 2:   ----------------------------------------------------

Instance 2 Redo Log 3:   ----------------------------------------------------

Instance 2 Redo Log:     ----------------------------------------------------

Instance 2 Redo Log:     ----------------------------------------------------

Data: System            ----------------------------------------------------

Data: Rollback          ----------------------------------------------------

Data: Temp              ----------------------------------------------------

Data: Users             ----------------------------------------------------

Data: Tools             ----------------------------------------------------

                        ----------------------------------------------------

                        ----------------------------------------------------

                        ----------------------------------------------------
```

Figure B-4. Blank Worksheet for Logical Volumes

```
                CLUSTER MANAGER CONFIGURATION WORKSHEET
===============================================================================
Name and Nodes:
===============================================================================


    Cluster Name: _____


    Node Names: _____


                _____

    DLM Volume Groups: _____

    Volume Groups (for packages):_____


===============================================================================
Subnets:
===============================================================================
Heartbeat Subnet: _____

Monitored Non-heartbeat Subnet: _____
===============================================================================
Cluster Lock Volume Groups and Volumes:
===============================================================================
First Lock Volume Group:   |        Physical Volume:
                           |
      _____      |    Name on Node 1: _____
                           |
                           |    Name on Node 2: _____
                           |
                           |    Disk Unit No: _____
                           |
                           |    Power Supply No: _____
                           |
===============================================================================
Timing Parameters:

Heartbeat Interval: _____

Node Timeout: _____

Network Polling Interval: _____

Autostart Delay: _____
```

Figure B-5. Blank Worksheet for Cluster Manager Configuration

```
                    DLM CONFIGURATION WORKSHEET
===============================================================================
Cluster-Specific Parameters:

        DLM Enabled:  _____

        Reconfiguration Timeout: _____

        Ping Interval: _____

        Ping Timeout: _____

        DLM Connect Timeout: _____

        DLM Halt Timeout: _____

        Communication Fail Timeout: _____

==============================================================================
Internal DLM Parameters:

        Cluster Name: _____

        Node Name(s): _____

        Resources: _____

        Locks: _____

        Processes: _____

        Deadlock Detection Interval: _____

        DLM Monitor Interval: _____

        Subnet Address: _____

        Node 1 IP Address: _____

.       Node 2 IP Address: _____
```

Figure B-6. Blank Worksheet for DLM Configuration

```
                 PACKAGE CONFIGURATION WORKSHEET              Page ___ of ___
================================================================================
Package Configuration File Data:
================================================================================

Package Name: _____

Nodes: _____ (Primary)

       _____

       _____

Package Run Script: _____Timeout: _____

Package Halt Script: _____Timeout: _____

Package Switching Enabled?  _____ Network Switching Enabled?  _____

Node Failfast Enabled?      _____

Control Script Data:
================================================================================

VG[0]_____LV[0]_____FS[0]_____

VG[1]_____LV[1]_____FS[1]_____

VG[2]_____LV[2]_____FS[2]_____


IP[0] _____ SUBNET _____

IP[1] _____ SUBNET _____


Service Name: _____ Run Command: _____Retries: _____

Service Fail Fast Enabled? _____Service Halt Timeout _____

Service Name: _____ Run Command: _____ Retries: _____

Service Fail Fast Enabled? _____Service Halt Timeout _____
```

Figure B-7. Blank Worksheet for Package Manager Configuration

C

Man Pages for MC/Lock Manager Configuration

The following is a list of man pages for the commands and files used for MC/Lock Manager configuration. These pages are available on your system *after installation* (described in the first part of Chapter 3).

Man Pages for Cluster Manager Commands:

- cmapplyconf(1m)
- cmcheckconf(1m)
- cmhaltcl(1m)
- cmhaltnode(1m)
- cmhaltpkg(1m)
- cmmodpkg(1m)
- cmquerycl(1m)
- cmruncl(1m)
- cmrunnode(1m)
- cmrunpkg(1m)
- cmviewcl(1m)

Man Pages for DLM Commands:

- dlmquery(1m)
- dlmapplyconf(1m)
- dlmcheckconf(1m)
- dlmstat(1m)
- dlmdump(1m)

Man Pages for SLVM Commands:

- vgchange(1m)
- vgdisplay(1m)
- vgexport(1m)
- vgimport(1m)

D

Designing Highly Available Cluster Applications

This appendix describes how to create or port applications for high availability, with emphasis on the following topics:

- Automating Application Operation
- Controlling the Speed of Application Failover
- Designing Applications to Run on Multiple Systems
- Restoring Client Connections
- Handling Application Failures
- Minimizing Planned Downtime

Designing for high availability means reducing the amount of unplanned and planned downtime that users will experience. Unplanned downtime includes unscheduled events such as power outages, system failures, network failures, or disk crashes. Planned downtime includes scheduled events such as scheduled backups or hardware replacements.

Two key strategies should be kept in mind:

1. Design the application to handle a system reboot or panic. If you are modifying an existing application for a highly available environment, determine what happens currently with the application after a system panic. In a highly available environment there should be defined (and scripted) procedures for restarting the application. Procedures for starting and stopping the application should be automatic, with no user intervention required.
2. The application should not use any system-specific information such as the following that would prevent it from failing over to another system:
   a. The application should not refer to `uname()` or `gethostname()`.
   b. The application should not refer to the SPU ID.
   c. The application should not refer to the MAC (link-level) address.

Automating Application Operation

Can the application be started and stopped automatically or does it require operator intervention?

This section describes how to automate application operations to avoid the need for user intervention. One of the first rules of high availability is to avoid manual intervention. If it takes a user at a terminal, console or GUI interface to enter commands to bring up a subsystem, the user becomes a key part of the system. It may take hours before a user can get to a system console to do the work necessary. The hardware in question may be located in a far-off area where no trained users are available, the systems may be located in a secure datacenter, or in off hours someone may have to connect via modem.

There are two principles to keep in mind for automating application relocation:

■ Insulate users from outages.

■ Applications must have defined startup and shutdown procedures.

You need to be aware of what happens currently when the system your application is running on is rebooted, and whether changes need to be made in the application's response for high availability.

Insulate Users from Outages

Wherever possible, insulate your end users from outages. Issues include the following:

■ Do not require user intervention to reconnect when a connection is lost due to a failed server.
■ Where possible, warn users of slight delays due to a failover in progress.
■ Minimize the reentry of data.
■ Engineer the system for reserve capacity to minimize the performance degradation experienced by users.

Define Applications' Startup and Shutdown

Applications must be restartable. If the application requires a switch to be flipped on a piece of hardware, then automated restart is impossible. Procedures for application startup, shutdown and monitoring must be created so that the HA software can perform these functions automatically.

To ensure automated response, there should be defined procedures for starting up the application and stopping the application. In MC/LockManager these procedures are placed in the package control script. These procedures must check for errors and return status to the HA control software. The startup and shutdown should be command-line driven and not interactive unless all of the answers can be predetermined and scripted.

In an HA failover environment, HA software restarts the application on a surviving system in the cluster that has the necessary resources, like access to the necessary disk drives. The application must be restartable in two aspects:

- It must be able to restart and recover on the backup system (or on the same system if the application restart option is chosen).
- It must be able to restart if it fails during the startup and the cause of the failure is resolved.

Application administrators need to learn to startup and shutdown applications using the appropriate HA commands. Inadvertently shutting down the application directly will initiate an unwanted failover. Application administrators also need to be careful that they don't accidently shut down a production instance of an application rather than a test instance in a development environment.

A mechanism to monitor whether the application is active is necessary so that the HA software knows when the application has failed. This may be as simple as a script that issues the command `ps -ef | grep xxx` for all the processes belonging to the application.

To reduce the impact on users, the application should not simply abort in case of error, since aborting would cause an unneeded failover to a backup system. Applications should determine the exact error and take specific action to recover from the error rather than, for example, aborting upon receipt of any error.

Controlling the Speed of Application Failover

What steps can be taken to ensure the fastest failover?

If a failure does occur causing the application to be moved to another node, there are many things the application can do to speed up the amount of time it takes to get the application back up and running. The topics covered are as follows:

- Replicate Non-Data File Systems
- Use Raw Volumes
- Evaluate the Use of JFS
- Minimize Data Loss
- Use Restartable Transactions
- Use Checkpoints
- Design for Multiple Servers
- Design for Replicated Data Sites

Replicate Non-Data File Systems

Non-data file systems should be replicated rather than shared. There can only be one copy of the application data itself. It will be located on a set of disks that is accessed by the system that is running the application. After failover, if these data disks are filesystems, they must go through filesystems recovery (`fsck`) before the data can be accessed. To help reduce this recovery time, the smaller these filesystems are, the faster the recovery will be. Therefore, it is best to keep anything that can be replicated off the data filesystem. For example, there should be a copy of the application executables on each system rather than having one copy of the executables on a shared filesystem.

Use Raw Volumes

If your application uses data, use raw volumes rather than filesystems. Raw volumes do not require an `fsck` of the filesystem, thus eliminating one of the potentially lengthy steps during a failover.

Evaluate the Use of JFS

If a file system must be used, a JFS offers significantly faster file system recovery as compared to an HFS. However, performance of the JFS may vary with the application.

Minimize Data Loss

Minimize the amount of data that might be lost at the time of an unplanned outage. It is impossible to prevent some data from being lost when a failure occurs. However, it is advisable to take certain actions to minimize the amount of data that will be lost, as explained in the following discussion.

Minimize the Use and Amount of Memory-Based Data

Any in-memory data (the in-memory context) will be lost when a failure occurs. The application should be designed to minimize the amount of in-memory data that exists unless this data can be easily recalculated. When the application restarts on the standby machine, it must recalculate or reread from disk any information it needs to have in memory.

One way to measure the speed of failover is to calculate how long it takes the application to start up on a normal system after a reboot. Does the application start up immediately? Or are there a number of steps the application must go through before an end-user can connect to it? Ideally, the application can start up quickly without having to reinitialize in-memory data structures or tables.

Performance concerns might dictate that data be kept in memory rather than written to the disk. However, the risk associated with the loss of this data should be weighed against the performance impact of posting the data to the disk.

Data that is read from a shared disk into memory, and then used as read-only data can be kept in memory without concern.

Keep Logs Small

Some databases permit logs to be buffered in memory to increase online performance. Of course, when a failure occurs, any in-flight transaction will be lost. However, minimizing the size of this in-memory log will reduce the amount of completed transaction data that would be lost in case of failure.

Keeping the size of the on-disk log small allows the log to be archived or replicated more frequently, reducing the risk of data loss if a disaster were to occur. There is, of course, a trade-off between online performance and the size of the log.

Eliminate Need for Local Data

When possible, eliminate the need for local data. In a three-tier, client/server environment, the middle tier can often be dataless (i.e., there is no local data that is client specific or needs to be modified). This "application server" tier can then provide additional levels of availability, load-balancing, and failover. However, this scenario requires that all data be stored either on the client (tier 1) or on the database server (tier 3).

Use Restartable Transactions

Transactions need to be restartable so that the client does not need to re-enter or back out of the transaction when a server fails, and the application is restarted on another system. In other words, if a failure occurs in the middle of a transaction, there should be no need to start over again from the beginning. This capability makes the application more robust and reduces the visibility of a failover to the user.

A common example is a print job. Printer applications typically schedule jobs. When that job completes, the scheduler goes on to the next job. If, however, the system dies in the middle of a long job (say it is printing paychecks for 3 hours), what happens when the system comes back up again? Does the job restart from the beginning, reprinting all the paychecks, does the job start from where it left off, or does the scheduler assume that the job was done and not print the last hours worth of paychecks? The correct behavior in a highly available environment is to restart where it left off, ensuring that everyone gets one and only one paycheck.

Another example is an application where a clerk is entering data about a new employee. Suppose this application requires that employee numbers be unique, and that after the name and number of the new employee is entered, a failure occurs. Since the employee number had been entered before the failure, does the application refuse to allow it to be re-entered? Does it require that the partially entered information be deleted first? More appropriately, in a highly

available environment the application will allow the clerk to easily restart the entry or to continue at the next data item.

Use Checkpoints

Design applications to checkpoint complex transactions. A single transaction from the user's perspective may result in several actual database transactions. Although this issue is related to restartable transactions, here it is advisable to record progress locally on the client so that a transaction that was interrupted by a system failure can be completed after the failover occurs.

For example, suppose the application being used is calculating PI. On the original system, the application has gotten to the 1,000th decimal point, but the application has not yet written anything to disk. At that moment in time, the node crashes. The application is restarted on the second node, but the application is started up from scratch. The application must recalculate those 1000 decimal points. However, if the application had written to disk the decimal points on a regular basis, the application could have restarted from where it left off.

Balance Checkpoint Frequency with Performance

It is important to balance checkpoint frequency with performance. The trade-off with checkpointing to disk is the impact of this checkpointing on performance. Obviously if you checkpoint too often the application slows; if you don't checkpoint often enough, it will take longer to get the application back to its current state after a failover. Ideally, the checkpointing frequency is customized by the end-user. The customer should be able to decide how often to checkpoint. Provide customizable parameters so the end-user can tune the checkpoint frequency.

Design for Multiple Servers

If you use multiple active servers, multiple service points can provide relatively transparent service to a client. However, this capability requires that the client be smart enough to have knowledge about the multiple servers and the priority for addressing them. It also requires access to the data of the failed server or replicated data.

For example, rather than having a single application which fails over to a second system, consider having both systems running the application. After a failure of the first system, the second system simply takes over the load of the first system. This eliminates the start up time of the application. There are many ways to design this sort of architecture, and there are also many issues with this sort of design. This discussion will not go into details other than to give a few examples.

The simplest method is to have two applications running in a master/slave relationship where the slave is simply a hot standby application for the master. When the master fails, the slave on the second system would still need to figure out what state the data was in (i.e., data recovery would still take place). However, the time to fork the application and do the initial startup is saved.

Another possibility is having two applications that are both active. An example might be two application servers which feed a database. Half of the clients connect to one application server and half of the clients connect to the second application server. If one server fails, then all the clients connect to the remaining application server.

Design for Replicated Data Sites

Replicated data sites are a benefit for both fast failover and disaster recovery. With replicated data, data disks are *not* shared between systems. There is no data recovery that has to take place. This makes the recovery time faster. However, there may be performance trade-offs associated with replicating data. There are a number of ways to perform data replication, which should be fully investigated by the application designer.

Many of the standard database products provide for data replication transparent to the client application. By designing your application to use a standard database, the end-user can determine if data replication is desired.

Designing Applications to Run on Multiple Systems

If an application can be failed to a backup machine, how will it work on a different system?

The previous sections discussed methods to ensure that an application can be automatically restarted. This section will discuss some ways to ensure the application can run on multiple systems. Topics are as follows:

- Avoid Node Specific Information
- Assign Unique Names to Applications
- Use `Uname(2)` With Care
- Bind to a Fixed Port
- Bind to a Relocatable IP Addresses
- Give Each Application its Own Volume Group
- Use Multiple Destinations for SNA Applications
- Avoid File Locking

Avoid Node Specific Information

Typically, when a new system is installed, an IP address must be assigned to each active network interface. This IP address is always associated with the node and is called a **stationary** IP address.

The use of highly available applications, or in the case of MC/LockManager, packages containing highly available applications, adds the requirement for an additional set of IP addresses, which are assigned to the applications themselves. These are known as **relocatable** application IP addresses. MC/LockManager packages monitor these relocatable application IP addresses. When packages are configured in MC/LockManager, the associated subnetwork address is specified as a package dependency, and a list of nodes on which the package can run is also provided. When failing a package over to a remote node, the subnetwork must already be active on the target node.

Each application or package should be given a unique name as well as a relocatable IP address. Following this rule separates the application from the system on which it runs, thus removing the need for user knowledge of which system the application runs on. It also makes it easier to move the application among different systems in a cluster for for load balancing or other reasons. If

two applications share a single IP address, they must move together. Instead, using independent names and addresses allows them to move separately.

For external access to the cluster, clients must know how to refer to the application. One option is to tell the client which relocatable IP address is associated with the application. Another option is to think of the application name as a host name, and configure the name to address mapping in the Domain Name System (DNS). In either case, the client will ultimately be communicating with the application relocatable IP address. If the application moves to another node, the IP address will move with it, allowing the client to use the application without knowing its current location. Remember that each network interface must have a stationary IP address associated with it. This IP address does *not* move to a remote system in the event of a network failure.

Obtain Enough IP Addresses

Each application receives a *relocatable* IP address that is separate from the stationary IP address assigned to the system itself. Therefore, a single system might have many IP addresses, one for itself and one for each of the applications that it normally runs. Therefore, IP addresses in a given subnet range will be consumed faster than without high availablity. It might be necessary to acquire additional IP addresses.

Multiple IP addresses on the same network interface are supported only if they are on the same subnetwork.

Allow Multiple Instances on Same System

Applications should be written so that multiple instances, each with its own application name and IP address, can run on a single system. It might be necessary to invoke the application with a parameter showing which instance it is. This allows distributing the users among several systems under normal circumstances, while allowing all of the users to be serviced in case of failure on a single system.

Avoid Using SPU IDs or MAC Addresses

Design the application so that it does not rely on the SPU ID or MAC (link-level) addresses. The SPU ID is a unique hardware ID contained in non-volatile memory, which cannot be changed. A MAC address is a link-specific address associated with the LAN hardware. The use of these addresses is a common problem for license servers, since for security reasons they want to use hardware to ensure the license isn't copied to multiple nodes. One workaround is to have multiple licenses; one for each node the application will run on. Another way is to have a cluster-wide mechanism that lists a set of SPU IDs or nodenames. If your application is running on a machine in the specified set, then the license is approved.

Previous generation HA software would move the MAC address of the network card along with the IP address when services were moved to a backup system. This is no longer allowed in MC/LockManager.

There were a couple of reasons for using a MAC address, which have been addressed below:

- Old network devices between the source and the destination such as routers had to be manually programmed with MAC and IP address pairs. The solution to this problem is to move the MAC address along with the IP address in case of failover.
- Up to 20 minute delays could occur while network device caches were updated due to timeouts associated with systems going down. This is dealt with in current HA software by broadcasting a new ARP translation of the old IP address with the new MAC address.

Assign Unique Names to Applications

A unique name should be assigned to each application. This name should then be configured in DNS so that the name can be used as input to `gethostbyname()`, as described in the following discussion.

Use DNS

DNS provides an API which can be used to map hostnames to IP addresses and vice versa. This is useful for BSD socket applications such as telnet which are first told the target system name. The application must then map the

name to an IP address in order to establish a connection. However, some calls should be used with caution.

Applications should *not* reference official hostnames or IP addresses. The official hostname and corresponding IP address for the hostname refer to the primary LAN card and the *stationary IP address* for that card. Therefore, any application that refers to, or requires the hostname or primary IP address will not work in an HA environment where the network identity of the system that supports a given application moves from one system to another, but the hostname does not move.

One way to look for problems in this area is to look for calls to `gethostname(2)` in the application. HA services should use `gethostname()` with caution, since the response may change over time if the application migrates. Applications that use `gethostname()` to determine the name for a call to `gethostbyname(2)` should also be avoided for the same reason. Also, the `gethostbyaddr()` call may return different answers over time if called with a relocatable IP addresses.

Instead, the application should always refer to the application name and relocatable IP address rather than the hostname and stationary IP address. It is appropriate for the application to call `gethostbyname(2)`, specifying the application name rather than the hostname. `gethostbyname(2)` will pass in the IP address of the application. This IP address will move with the application to the new node.

However, `gethostbyname(2)` should be used to locate the IP address of an application only if the application name is configured in DNS. It is probably best to associate a different application name with each independent HA service. This allows each application and its IP address to be moved to another node without affecting other applications. Only the stationary IP addresses should be associated with the hostname in DNS.

Use uname(2) With Care

Related to the hostname issue discussed in the previous section is the application's use of `uname(2)`, which returns the official system name. The system name is unique to a given system whatever the number of LAN cards in the system. By convention, the `uname` and `hostname` are the same, but they do not have to be. Some applications, after connection to a system, might call

`uname(2)` to validate for security purposes that they are really on the correct system. This is not appropriate in an HA environment, since the service is moved from one system to another, and neither the `uname` nor the `hostname` are moved. Applications should develop alternate means of verifying where they are running. For example, an application might check a list of hostnames that have been provided in a configuration file.

Bind to a Fixed Port

When binding a socket, a port address can be specified or one can be assigned dynamically. One issue with binding to random ports is that a different port may be assigned if the application is later restarted on another cluster node. This may be confusing to clients accessing the application.

The recommended method is using fixed ports that are the same on all nodes where the application will run, instead of assigning port numbers dynamically. The application will then always return the same port number regardless of which node is currently running the application. Application port assignments should be put in `/etc/services` to keep track of them and to help ensure that someone will not choose the same port number.

Bind to Relocatable IP Addresses

When sockets are bound, an IP address is specified in addition to the port number. This indicates the IP address to use for communication and is meant to allow applications to limit which interfaces can communicate with clients. An application can bind to INADDR_ANY as an indication that messages can arrive on any interface.

Network applications can bind to a stationary IP address, a relocatable IP address, or INADDR_ANY. If the stationary IP address is specified, then the application will fail when restarted on another node, because the stationary IP address is not moved to the new system.

If an application binds to the relocatable IP address, then the application will behave correctly when moved to another system.

Many server-style applications will bind to INADDR_ANY, meaning that they will receive requests on any interface. This allows clients to send to the stationary or relocatable IP addresses. However, in this case the networking

code cannot determine which source IP address is most appropriate for responses, so it will always pick the stationary IP address.

For TCP stream sockets, the TCP level of the protocol stack resolves this problem for the client since it is a connection-based protocol. On the client, TCP ignores the stationary IP address and continues to use the previously bound relocatable IP address originally used by the client.

With UDP datagram sockets, however, there is a problem. The client may connect to multiple servers, transmit to the relocatable IP address and sort out the replies based on the source IP address in the message. However, the source IP address will be the stationary IP address rather than the relocatable application IP address. Therefore, when creating a UDP socket for listening, the application must always call `bind(2)` with the appropriate relocatable application IP address rather than `INADDR_ANY`.

If the application cannot be modified as recommended above, a workaround to this problem is to not use the stationary IP address at all, and only use a single relocatable application IP address on a given LAN card. Limitations with this workaround are as follows:

■ Local LAN failover will not work.
■ There has to be an idle LAN card on each backup node that is used to relocate the relocatable application IP address in case of a failure.

Call bind() before connect()

When a application initiates its own connection, it should first call `bind(2)`, specifying the application IP address before calling `connect(2)`. Otherwise the connect request will be sent using the stationary IP address of the system's outbound LAN interface rather than the desired relocatable application IP address. The client will receive this IP address from the `accept(2)` call, confusing the client software and preventing it from working correctly.

Give Each Application its Own Volume Group

Use a separate volume group for each application that uses data. If the application doesn't use disk, it is not necessary to assign it a separate volume group. A volume group (group of disks) is the unit of data that can move between nodes. The greatest flexibility for load balancing exists when each application is confined to its own volume group, i.e., two applications do not

share the same set of disk drives. If two applications do use the same disk drives to store their data, then the applications must move together. If the applications are in separate volume groups, they can switch to different nodes in the event of a failover.

The application data should be set up on different disk drives and if applicable, different mount points. The application should be designed to allow for different disks and separate mount points. If possible, the application should not assume a specific mount point.

To prevent one system from inadvertently accessing disks being used by the application on another system, HA software uses a disk locking mechanism to enforce exclusive access. This lock applies to a volume group as a whole.

Use Multiple Destinations for SNA Applications

SNA is point-to-point link-oriented; that is, the *services* cannot simply be moved to another system, since that system has a different point-to-point link which originates in the mainframe. Therefore, backup links in a node and/or backup links in other nodes should be configured so that SNA does not become a single point of failure. Note that only one configuration for an SNA link can be active at a time. Therefore, backup links that are used for other purposes should be reconfigured for the primary mission-critical purpose upon failover.

Avoid File Locking

In an NFS environment, applications should avoid using file-locking mechanisms, where the file to be locked is on an NFS Server. File locking should be avoided in an application both on local and remote systems. If local file locking is employed and the system fails, the system acting as the backup system will not have any knowledge of the locks maintained by the failed system. This may or may not cause problems when the application restarts.

Remote file locking is the worst of the two situations, since the system doing the locking may be the system that fails. Then, the lock might never be released, and other parts of the application will be unable to access that data. In an NFS environment, file locking can cause long delays in case of NFS client system failure and might even delay the failover itself.

Restoring Client Connections

How does a client reconnect to the server after a failure?

It is important to write client applications to specifically differentiate between the loss of a connection to the server and other application-oriented errors that might be returned. The application should take special action in case of connection loss.

One question to consider is how a client knows after a failure when to reconnect to the newly started server. The typical scenario is that the client must simply restart their session, or relog in. However, this method is not very automated. For example, a well-tuned hardware and application system may fail over in 5 minutes. But if the users, after experiencing no response during the failure, give up after 2 minutes and go for coffee and don't come back for 28 minutes, the perceived downtime is actually 30 minutes, not 5! Factors to consider are the number of reconnection attempts to make, the frequency of reconnection attempts, and whether or not to notify the user of connection loss.

There are a number of strategies to use for client reconnection:

■ Design clients which continue to try to reconnect to their failed server.

Put the work into the client application rather than relying on the user to reconnect. If the server is back up and running in 5 minutes, and the client is continually retrying, then after 5 minutes, the client application will reestablish the link with the server and either restart or continue the transaction. No intervention from the user is required.

■ Design clients to reconnect to a *different* server.

If you have a server design which includes multiple active servers, the client could connect to the second server, and the user would only experience a brief delay.

The problem with this design is knowing when the client should switch to the second server. How long does a client retry to the first server before giving up and going to the second server? There are no definitive answers for this. The answer depends on the design of the server application. If the application can be restarted on the same node after a failure (see "Handling Application Failures" following), the retry to the current server should continue for the amount of time it takes to restart the server locally. This

will keep the client from having to switch to the second server in the event of a application failure.

■ Use a transaction processing monitor or message queueing software to increase robustness.

Use transaction processing monitors such as Tuxedo or DCE/Encina, which provide an interface between the server and the client. Transaction processing monitors (TPMs) can be useful in creating a more highly available application. Transactions can be queued such that the client does not detect a server failure. Many TPMs provide for the optional automatic rerouting to alternate servers or for the automatic retry of a transaction. TPMs also provide for ensuring the reliable completion of transactions, although they are not the only mechanism for doing this. After the server is back online, the transaction monitor reconnects to the new server and continues routing it the transactions.

■ Queue Up Requests

As an alternative to using a TPM, queue up requests when the server is unavailable. Rather than notifying the user when a server is unavailable, the user request is queued up and transmitted later when the server becomes available again. Message queueing software ensures that messages of any kind, not necessarily just transactions, are delivered and acknowledged.

Message queueing is useful only when the user does not need or expect response that the request has been completed (i.e, the application is not interactive).

Handling Application Failures

What happens if part or all of an application fails?

All of the preceding sections have assumed the failure in question was not a failure of the application, but of another component of the cluster. This section deals specifically with application problems. For instance, software bugs may cause an application to fail or system resource issues (such as low swap/memory space) may cause an application to die. The section deals with how to design your application to recover after these types of failures.

Create Applications to be Failure Tolerant

An application should be tolerant of failures in a single component. Many applications have multiple processes running on a single node. If one process fails, what happens to the other processes? Do they also fail? Can the failed process be restarted on the same node without affecting the remaining pieces of the application?

Ideally, if one process fails, the other processes can wait a period of time for that component to come back online. This is true whether the component is on the same system or a remote system. The failed component can be restarted automatically on the same system and rejoin the waiting processing and continue on. This type of failure can be detected and restarted within a few seconds, so the end user would never know a failure occurred.

Another alternative is for the failure of one component to still allow bringing down the other components cleanly. If a database SQL server fails, the database should still be able to be brought down cleanly so that no database recovery is necessary.

The worse case is for a failure of one component to cause the entire system to be "bounced". If one component fails and all other components need to be restarted, the downtime will be high.

Be Able to Monitor Applications

All components in a system, including applications, should be able to be monitored for their health. A monitor might be as simple as a display command or as complicated as a SQL query. There must be a way to ensure that the application is behaving correctly. If the application fails and it is not detected automatically, it might take hours for an user to determine the cause of the downtime and recover from it.

Minimizing Planned Downtime

Planned downtime (as opposed to unplanned downtime) is scheduled; examples include backups, systems upgrades to new operating system revisions, or hardware replacements. For planned downtime, application designers should consider:

- **Providing for online application reconfiguration**.

  Can the configuration information used by the application be changed without bringing down the application?

- **Documenting maintenance operations**.

  Does an operator know how to handle maintenance operations?

The following sections discuss ways of handling the different types of planned downtime.

Providing Online Application Reconfiguration

Most applications have some sort of configuration information that is read when the application is started. If to make a change to the configuration, the application must be halted and a new configuration file read, downtime is incurred.

To avoid this downtime use configuration tools that interact with an application and make dynamic changes online. The ideal solution is to have a configuration tool which interacts with the application. Changes are made online with little or no interruption to the end-user. This tool must be able to do everything online, such as expanding the size of the data, adding new machine into the system, adding new users to the application, etc. Every task that an administrator needs to do to the application system can be made available online.

Documenting Maintenance Operations

Standard procedures are important. An application designer should make every effort to make tasks common for both the highly available environment and the normal environment. If an administrator is accustomed to bringing down the entire system after a failure, he or she will continue to do so even if the application has been redesigned to handle a single failure. It is important that

application documentation discuss alternatives with regards to high availability for typical maintenance operations.

E

Distributed Lock Manager Error Messages

This appendix lists DLM error messages with cause and action text. The messages are grouped in the following categories:

- DLM-1 - Startup Errors
- DLM-2 - Normal Runtime Errors
- DLM-3 - Runtime Errors and Alerts
- DLM-7 - Reconfiguration Timing Errors
- DLM-8 - DLM Internal Errors
- DLM-9 - CM-DLM interface errors

DLM Startup Errors

DLM-1001 MESSAGE   [DLM-1001] Could not open DLM log file <*log file path*>.

CAUSE   Unable to create and open the DLM log file name specified by <*log file path*>. Possible reason include no permission for DLM daemons (user root) to create the log file name specified by path.

ACTION   Change permissions as appropriate and try to startup cluster again.

DLM-1002 MESSAGE   [DLM-1002] Could not open DLM trace file <*trace file path*>.

CAUSE   Unable to create and open the DLM trace file name specified by <*trace file path*>. Possible reason include no permission for DLM daemons (user root) to create the trace file name specified by <*trace file path*>.

ACTION   Create or change dlm user home diretory or change permissions as appropriate and try to startup cluster again.

DLM-1003 MESSAGE   [DLM-1003] Not enough memory for DLM trace file <*trace file name*> initialization.

CAUSE   There is not enough memory on your system to allocate memory for trace file initialization.

ACTION   Contact your system administrator to add more memory to your system(s) or contact your database administrator to reduce the lock database parameters to values that are more appropriate for your system. The OPS administration guide and the MC/LockManager User manual has information on how to adjust DLM lock database parameter values.

DLM-1004	MESSAGE	[DLM-1004] DLM shared memory allocation failed (size is *<lock database size>*). Lookup DLM startup log *<DLM startup log file path>* for details.
	CAUSE	There are three possible causes:
		■ There is not sufficient lockable memory on your system to create and lock a shared memory segment for a lock database of this size.
		■ The *<lock database size>* is less than the system-imposed minimum or greater than the system-imposed maximum.
		■ A shared memory identifier is to be created but the system imposed limit on the maximum number of allowed shared memory identifiers system wide would be exceeded.
	ACTION	Contact your system administrator to add more memory to your system(s) or increase the system-imposed limits for shared memory segments or contact your database administrator to change the lock database parameters to values that are more appropriate for your system. The OPS administration guide and the MC/LockManager User manual has information on how to adjust DLM lock database parameter values.
DLM-1005	MESSAGE	[DLM-1005] Unable to create UNIX domain socket pipe *<pipe directory>*.
	CAUSE	DLM is unable to create the UNIX domain socket special files under the specified dlm home directory.
	ACTION	Check for appropriate permissions on the dlm home (or log root) directory.

DLM-1006	MESSAGE	[DLM-1006] DLM daemon unable to create child processes: *<error string>*
	CAUSE	DLM is unable to fork its child processes. Possible reasons include:

- The system-imposed limit on the total number of processes under execution would be exceeded.
- There is insufficient swap space and/or physical memory available in which to create the new process. *<error string>* explains the error received by DLM while invoking the fork() call to create child processes.

	ACTION	Contact your system administrator to increase the system-imposed limits on the maximum number of runnable process OR to increase the swap space and/or physical memory.

DLM-1007	MESSAGE	[DLM-1007] Illegal lock database parameter values in DLM configuration file. Lookup DLM startup log *<DLM startup log file path>* for details.
	CAUSE	One or more of the lock database parameter values specified is less than the minimum DLM-imposed limit. The DLM startup log specified by *<DLM startup log file path>* will contain information about the DLM-imposed minimum limits when this error occurs.
	ACTION	Increase the DLM lock database parameter values to a value greater than the DLM-imposed minimum.

DLM-1008	MESSAGE	[DLM-1008] Possible mismatch between cluster configuration file and DLM configuration file.
	CAUSE	You may have changed the cluster configuration but not your DLM configuration.
	ACTION	Redo the DLM configuration step either using DLM configuration GUI or DLM configuration commands to create a new DLM binary configuration file that correctly reflects the current cluster.

DLM-1009	MESSAGE	[DLM-1009] Illegal DLM configuration file.
	CAUSE	The DLM binary configuration file maybe corrupted or is old.
	ACTION	Redo the DLM configuration step either using DLM configuration GUI or DLM configuration commands to create a new and valid DLM binary configuration file.

DLM-1010	MESSAGE	[DLM-1010] Error trying to read DLM configuration file *<binary configuration file>*.
	CAUSE	The DLM binary configuration file does not exist on this node.
	ACTION	Redo the dlm configuration step either using DLM configuration GUI or DLM configuration commands to create the dlm binary configuration file.

DLM-1011	MESSAGE	[DLM-1011] getservbyname fails looking up hacl-dlm/tcp. Check /etc/services.
	CAUSE	Your /etc/services file does not contain the dlm service entry
	ACTION	Re-install the MC/LockManager product or add the following line to your /etc/services file:

hacl-dlm 5408/tcp # HA Cluster distributed lock manager

NOTE: The port number can be any valid value not used by other services

Normal Runtime Errors

DLM-2001   MESSAGE   [DLM-2001] Unable to lock DLM daemons in memory.

                CAUSE   Not enough lockable memory on your system.

                ACTION   Shutdown cluster and increase memory on your system or reduce the lock database size, because this error can affect OPS performance.

DLM-2002   MESSAGE   [DLM-2002] DLM daemon cannot run with realtime priority: $<error\ string>$

                CAUSE   Unable to set realtime priority. $<error\ string>$ specifies the reason for the failure.

                ACTION   DLM continues to run with timeshare priority levels. However, this error can affect performance. So shutdown the cluster and restart the cluster after making sure that DLM has the appopriate priveleges to change to real time priority.

Runtime Errors and Alerts

DLM-3001   **MESSAGE**   [DLM-3001] DLM resource structure usage exceeded configured value. Increase MAXRESOURCES parameter.

        CAUSE   DLM clients (OPS) has tried to open and use more resources than what was configured during DLM configuration.

        ACTION   Contact your database administrator. The DBA should recalculate the number of DLM resources needed for the database application and either increase the MAXRESOURCES parameter in the dlm configuration file and reconfigure your cluster or reduce the databases' DLM resource usage.

The OPS administration guide and the MC/LockManager user manual together explains how to configure the lock database parameter values for your database.

DLM-3002	MESSAGE	[DLM-3002] DLM lock structure usage exceeded configured value. Increase MAXLOCKS parameter.
	CAUSE	DLM clients (OPS) has tried to open and use more dlm locks than what was configured during DLM configuration.
	ACTION	Contact your database administrator. The DBA should recalculate the number of DLM locks needed for the database and either increase the MAXLOCKS parameter in the dlm configuration file and reconfigure your cluster or adjust the database applications' DLM lock usage.
		The OPS administration guide and the MC/LockManager user manual together explains how to configure the lock database parameter values for your database.

DLM-3003	MESSAGE	[DLM-3003] DLM process structure usage exceeded configured value. Increase MAXPROCESSES parameter.
	CAUSE	More DLM clients (OPS instance and servers) has tried to attach to DLM than what was configured during DLM configuration.
	ACTION	Contact your database administrator. The DBA should either reduce the number of lck processes for the OPS instance or increase the MAXPROCESSES lock database parameter value to correctly reflect the number of DLM clients that will be attaching to DLM.
		The OPS administration guide and the MC/LockManager user manual together explains how to configure the lock database parameter values for your database.

Reconfiguration Time Errors

DLM-7001   MESSAGE   [DLM-7001] DLM clients not responding to DLMs'
                        reconfiguration request in time. DLM Aborting.
           CAUSE   One or more of the dlm clients (OPS instance
                        and OPS servers) is not responding to DLM's
                        reconfiguration signals. When this happens, the
                        clients that are not responding may be hung inside
                        the kernel. This event CAUSEs the node to halt.
           ACTION   Contact support personnel.

DLM Internal Errors

DLM-8001   MESSAGE   [DLM-8001] Fatal DLM internal error : *<more
                        information>*
           CAUSE   DLM has encountered an internal error.
           ACTION   Contact support personnel.

Cluster Manager-DLM Interface Errors

DLM-9001   **MESSAGE**   [DLM-9001] Unable to complete handshaking to start DLM. DLM failed to start.

          CAUSE   DLM failed to start up.

          ACTION   Check the syslog for additional MESSAGEs which explain why DLM failed to start up.

DLM-9002   **MESSAGE**   [DLM-9002] Failed to send halt MESSAGE to DLM.

          CAUSE   The socket connection between cluster monitor and DLM is broken. A possible reason is abnormal DLM termination caused by either killing the DLM daemons or an internal error.

          ACTION   Check the syslog and the DLM logs for additional MESSAGEs explaining why DLM may have terminated abnormally. Contact a support representative, if DLM has reported an internal error as the reason for its termination.

DLM-9003   **MESSAGE**   [DLM-9003] DLM daemon file *<dlm daemon path>* could not be executed: *<error string>*

          CAUSE   No execute permission for DLM daemon file specified by *<dlm daemon path>* or DLM startup failed.

          ACTION   Check the *<error string>* in the MESSAGE and check for appropriate permissions for the DLM daemon file *<dlm daemon path>*. Permission should be 0544.

DLM-9004	MESSAGE	[DLM-9004] Cluster Manager cannot communicate with DLM.
	CAUSE	Local communication between cluster monitor and DLM is broken. A possible CAUSE is abnormal DLM termination. This error can occur if dlm daemons are killed or if the dlm daemons reported an internal error and aborted.
	ACTION	Check the syslog for additional MESSAGEs explaining why DLM may have terminated abnormally. Contact a support representative, if DLM has reported an internal error.

DLM-9005	MESSAGE	[DLM-9005] DLM daemon ($<pid>$) has terminated.
	CAUSE	Cluster monitor has detected a problem with DLM daemons and has terminated the daemon.
	ACTION	Check the syslog for additional MESSAGEs explaining the reason for terminating DLM.

DLM-9006	MESSAGE	[DLM-9006] Failure checking for DLM termination: $<error\ string>$
	CAUSE	Cluster monitor detects a problem with the DLM daemon and tries to terminate the daemon. However, DLM has not yet terminated within a given time. In this situation the DLM core image may not be complete.
	ACTION	Check the syslog for additional MESSAGEs explaining the reason for terminating DLM.

DLM-9007	MESSAGE	[DLM-9007] Halting node. Timeout during DLM reconfiguration.
	CAUSE	DLM encountered either an internal error or a communication problem during its reconfiguration or DLM simply took too long to reconfigure, possibly due to a DLM clients failure to respond to DLM reconfiguration request (see error DLM-7001).
	ACTION	If DLM has encountered an internal error (error DLM-8001) or if DLM clients have failed to respond to DLM reconfiguration request (error DLM-7001), contact a support representative. Otherwise, check if DLM communication network is operating correctly and check DLM timing parameters in the cluster configuration file.

DLM-9008	MESSAGE	[DLM-9008] Halting node. DLM is not responding to ping from Cluster Manager.
	CAUSE	DLM was dangerously slow in responding to Cluster Manager ping requests. The system may not be able to handle the load applied.
	ACTION	This is a DLM internal error. Check DLM timing parameters in the cluster configuration file. Otherwise, save DLM logs and cores and contact a support person.

DLM-9009	MESSAGE	[DLM-9009] Halting node $<node\_name>$ due to DLM communication failure.
	CAUSE	DLM or the cluster manager encountered a communication failure on the DLM communication network.
	ACTION	Check if DLM communication network is operating correctly.

Index

DLM processes parameter, 2-24
DLM reconfig timeout parameter,
    2-22
DLM resources parameter, 2-23
filled in planning worksheet, 2-26
node IP address parameter, 2-25
node name parameter, 2-23
planning the configuration, 2-22
use in OPS cluster, 1-22
distributing the cluster and package
    configuration, 4-11
DLM
    blank planning worksheet, B-7
    cluster name parameter, 2-23
    component of MC/LockManager,
        1-13
    configuring with HP-UX commands,
        3-29
    deadlock detection interval parameter,
        2-24
    DLM commfail timeout parameter,
        2-23
    DLM connect timeout parameter,
        2-22
    DLM enabled parameter, 2-22
    DLM halt timeout parameter, 2-22
    DLM locks parameter, 2-24
    DLM monitor interval parameter,
        2-24
    DLM ping interval parameter, 2-22
    DLM ping timeout parameter, 2-22
    DLM processes parameter, 2-24
    DLM reconfig timeout parameter,
        2-22
    DLM resources parameter, 2-23
    DLM subnet address parameter, 2-24
    node IP address parameter, 2-25
    node name parameter, 2-23
    planning the configuration, 2-22
    use in OPS cluster, 1-22
DLM commfail timeout

parameter in cluster manager
    configuration, 2-23
DLM configuration file
    editing the ASCII template, 3-29
    on all nodes, 1-22
DLM connect timeout
    parameter in cluster manager
        configuration, 2-22
DLM enabled
    parameter in cluster manager
        configuration, 2-22
DLM halt timeout
    parameter in cluster manager
        configuration, 2-22
DLM locks
    parameter in distributed lock manager
        configuration, 2-24
DLM node IP address
    parameter in distributed lock manager
        configuration, 2-25
DLM ping interval
    parameter in cluster manager
        configuration, 2-22
DLM ping timeout
    parameter in cluster manager
        configuration, 2-22
DLM processes
    parameter in distributed lock manager
        configuration, 2-24
DLM reconfig timeout
    parameter in cluster manager
        configuration, 2-22
DLM resources
    parameter in distributed lock manager
        configuration, 2-23
DLM subnet address
    parameter in distributed lock manager
        configuration, 2-24
dual cluster locks
    choosing, 1-21