IBM

Implementing Oracle9*i* RAC with Linux on IBM @server xSeries Servers

- Set up Oracle9*i* Real Application Clusters with SuSE Linux 7.2
- Set up and configure IBM FAStT Storage Manager on Linux
- Configure Oracle9*i* RAC for high availability

Redpaper

Steve Russell
Stephen R. Poon
Gregory Verstraeten

ibm.com/redbooks

Redbooks

International Technical Support Organization

Implementing Oracle9*i* RAC with Linux on IBM @server xSeries Servers

October 2001

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page 111.

First Edition (October 2001)

This edition applies to Oracle9*i* Real Application Clusters for use with the SuSE V7.2 Linux operating system.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195

Contents

Preface

This Redpaper is aimed at technical staff within IBM, our business partners, and customers who have been charged with implementing Oracle9*i* Real Application Clusters (RAC) running on Linux. It is intended to augment the redbook *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449. Even though RAC is a new architecture for clustering with Oracle, and we are dealing with a different operating system, there is some commonality with the content of the book. You will see that the structure used in the paper is largely based on that of the book, and similar topics are covered. Duplication, however, has been kept to a minimum, with references to the original book being made where appropriate.

RAC is the immediate successor to Oracle8*i* Parallel Server (OPS), and offers many improvements in performance and availability in comparison. The basic purpose, however, has not changed, so RAC still offers the benefits gained by clustering multiple nodes, with each node running an instance of the database. Each node can access a common database, which offers benefits both in scalability and availability beyond that offered by a simple failover cluster.

In particular:

- Node failures only affect the subset of users attached to the failed node.
- Users attached to the failed node fail over to the surviving nodes in the cluster. Since Oracle is already running on the other nodes, this failover is fast and minimizes disruption to connected users.
- All nodes within the cluster are active, so system resources are better utilized and load balancing across nodes takes place, offering greater efficiency than an active - passive cluster environment.

The Oracle clusters discussed in this paper are based on the Linux operating system (SuSE 7.2) running on IBM @server xSeries servers. Enterprise-class xSeries 370 systems support up to eight processors in a symmetrical multiprocessing configuration to provide unprecedented processing power. Used in conjunction with the FAStT500 Fibre Channel disk subsystem, they provide an excellent hardware platform on which to implement Oracle clusters. The paper describes the correct way to configure these components to successfully implement your own Oracle RAC cluster.

It is assumed that readers of this paper are familiar with Oracle9*i* concepts and implementation, and the Linux operating system.

Important: IBM has certified a two-node IBM Netfinity 8500R configuration with Oracle Parallel Server V8.1.6 with patches 8.1.6.1.1 and 8.1.6.1.2 running Windows NT V4.0 Enterprise Edition with SP5. However, the IBM RAC certification for SuSE Linux 7.2 had not been completed at the time this Redpaper was written. Please check the following Web site to view solutions that are currently certified and supported:

http://www.pc.ibm.com/ww/eserver/xseries/clustering/parallel_server.html

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Steve Russell is a Certified Consulting IT Specialist at the International Technical Support Organization, Raleigh Center, where he manages residencies and produces Redbooks related to IBM xSeries and Netfinity servers. Before joining the ITSO, Steve worked in the Technical Marketing field in the UK as a member of the IBM Netfinity organization in EMEA. Prior to that, he spent nearly 15 years managing and developing PC-based hardware and software projects at the IBM Hursley laboratory in the UK. He holds a degree in Electrical and Electronic Engineering, and is a member of the Institution of Electrical Engineers and a Chartered Engineer.

Stephen R. Poon is an IT Specialist in EAS America's Advanced Technical Support and works within the IBM/Oracle International Competency Center (ICC) in San Mateo, California. He works on system-sizing guidelines and provides pre-sales technical support for the Oracle E-business Suite running on IBM xSeries and Netfinity servers. Before joining the ICC, Stephen's positions during his IBM career have included software product development, systems engineering and IT management consulting. He holds an MS degree in Mechanical Engineering from the University of California, Berkeley and has been with IBM for over 15 years.

Gregory Verstraeten is an Oracle products specialist at the EMEA PSSC in Montpelier, France. He works within the EMEA Oracle/IBM Joint Solutions Center, where his main responsibilities include designing and sizing architectures, supporting benchmarks and teaching workshops involving Oracle products (Oracle database, Oracle E-business Suite) on xSeries and pSeries platforms. He holds a degree in Computer Science Engineering.

Special thanks to the following people for their contributions to this project:

Francois Pons	Oracle, EMEA Partner Technical Services

Thanks to the following people for their contributions to this project:

Gail Christensen	ITSO Raleigh Center
Virginia Packer	Oracle Technical Relationship Manager, IBM Raleigh
Linda Robinson	ITSO Raleigh Center
Margaret Ticknor	ITSO Raleigh Center

The IBM/Oracle International Competency Center
The EMEA Oracle/IBM Joint Solutions Center

Notice

This publication is intended to help technical specialists to implement Oracle9*i* Real Application Clusters running on the Linux operating system. The information in this publication is not intended as the specification of any programming interfaces that are provided by Oracle9*i* or Linux. See the relevant product documentation for more information about what publications provide this information.

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)® 
IBM ®
EtherJet™
Netfinity®
pSeries™
Redbooks
Redbooks Logo 
SP™
xSeries™
z/OS™

Comments welcome

Your comments are important to us!

We want our Redpapers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to the address on page ii.

1

Introduction

The IBM Redbook, *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449, described the implementation of Oracle8*i* Parallel Server (OPS) in a Windows 2000 environment using Netfinity servers. It provided an example configuration that was used to illustrate and take the reader through the detailed procedures needed to configure OPS for load balancing and failover.

Since that time, Oracle has released the Oracle9*i* database for most major platforms, including z/OS, AIX, Linux, HP-UX, Solaris, and Microsoft Windows 2000. The Real Application Clusters (RAC) option in Oracle9*i* Enterprise Edition has replaced OPS. With RAC, Oracle has significantly improved OPS technology, including architectural enhancements to provide better performance, easier management, and enhanced availability.

This Redpaper extends the afore-mentioned IBM Redbook to Oracle9*i* RAC. It takes the reader through the implementation of Oracle9*i* Enterprise Edition and Real Application Clusters on Linux and IBM @server xSeries servers, using the latest IBM Storage Area Network (SAN) technology in the form of the FAStT500 Fibre Channel storage subsystem.

In the chapters that follow, we take you through the implementation and installation of an Oracle9*i* Real Application Clusters configuration, based on real-life experience gained as we installed the example configuration in our laboratory.

We begin by discussing the architectural improvements found in Oracle9*i* RAC. Using an example configuration, we provide a detailed procedure for implementing RAC with Linux, IBM xSeries servers, and the IBM FAStT Storage Server.

Planning is an important factor in a successful implementation and Chapter 3, “Planning for Oracle9i RAC” on page 9 is devoted to this topic. Chapter 4, “Setting up Linux” on page 17 provides suggestions for ensuring that Linux is correctly installed on the servers of the cluster. Details about the storage subsystem are provided in Chapter 5, “FAStT controller and Storage Manager setup” on page 25.

After these introductory chapters, the Redpaper covers the fine details of installing Oracle9*i* RAC, starting with Chapter 6, “Pre-installation tasks” on page 35, and continuing with Chapter 7, “Oracle9i RAC Installation” on page 57. To complete the story, Chapter 8, “Creating and validating a database” on page 75 and Chapter 9, “Configuring RAC for High Availability and Load Balancing” on page 101 show you how to create the first database and how to maximize the availability of your Oracle9*i* RAC system respectively.

Finally, Appendix A, “Useful commands” on page 109 provides a list of useful commands for managing the Oracle9*i* database.

2

Oracle9*i* RAC architecture overview

Oracle9*i* Real Application Clusters (RAC) is the successor product to Oracle8*i* Parallel Server (OPS). The new features introduced with RAC are supported by evolutionary changes to the underlying cluster architecture. This chapter briefly introduces the three most significant changes introduced in RAC. These are the Cache Fusion concept, those architecture components in charge of database management for concurrent access, and the major processes that implement the architecture.

The architectural changes in RAC are designed to increase the product's scalability and high-availability features beyond those found in OPS. In particular, RAC provides the following:

- Virtually unlimited scalability

  Oracle9*i* RAC provides the way to run an Oracle9*i* database across multiple servers while presenting applications and users with a single view of the database.

- Enhanced availability

  If any server in a cluster fails, other servers in the cluster will handle the workload from the failed server with a minimum of I/O recovery. Users on a failed machine will be transparently moved over to one of the remaining servers and, along with other affected users, continue to operate unimpeded by the failure.

- It can be managed as easily as a single system

  With improved management tools (such as srvctl, oem, dbca, and netca), a multi-instance database with RAC is managed as easily as a single instance database.

- It supports all customers' applications without modification

  Applications can benefit from the availability and scalability features of *Oracle9i* RAC even if they have not been designed specifically for RAC.

2.1 Cache Fusion improvement

The concept of *Cache Fusion* (introduced in Oracle8*i* OPS) is now fully implemented in *Oracle9i* RAC. Cache Fusion is designed to eliminate forced disk writes and re-reads when an instance requests a database block that has already been used by another instance. This is sometimes referred to as block pinging, and, without Cache Fusion, has a detrimental effect on system performance.

There are four cases we have to consider:

- Read-Read

  An instance requests a database block for read when this block has previously been read by another instance.

- Write-Read

  An instance requests a database block for read when this block has previously been updated by another instance.

- Read-Write

  An instance requests a database block for write when this block has previously been read by another instance.

- Write-Write

  An instance requests a database block for write when this block has previously been updated by another instance.

Cache Fusion as implemented on Oracle8*i* Parallel Server eliminates pinging that would occur in read-read situations, but not for write-read, read-write, or write-write cases. In the first case, the block being requested is transferred from cache to cache through the interconnect link. In the remaining situations, however, disk access is required in order to access the block.

In contrast, Oracle9*i* RAC transfers the block from cache to cache through the interconnect link in all four cases. Since a disk access is not required and a block transfer through the interconnect is typically 100 times faster than a transfer with pinging, scalability is enhanced.

2.2 Resource management components

An Oracle cluster comprises a number of different components, all of which have to be managed. Resources such as database blocks, data dictionary blocks, the library cache, and so on, must all be kept in order. Figure 2-1 provides an overview diagram of the way a two-node implementation of Oracle9*i* Real Application Clusters is structured:

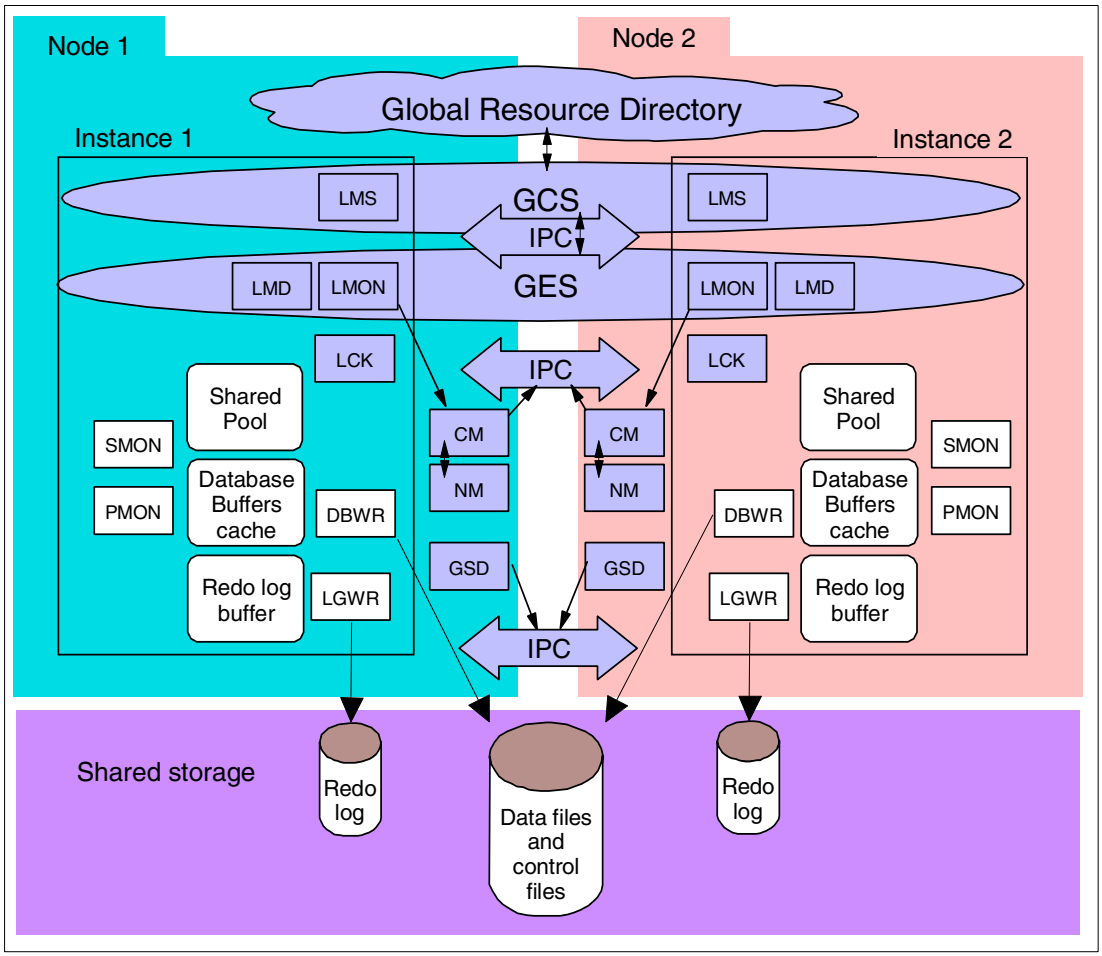

Figure 2-1 Oracle9i RAC architecture

Three major new components have been introduced in the Oracle9*i* RAC architecture. They are in charge of the database resources management and are called the *Global Resource Directory*, the *Global Cache Service* and the *Global Enqueue Service*. A brief description of these components follows:

- The Global Resource Directory contains the current status of resources shared by the instances.
- The Global Cache Service (GCS) is the controlling process that implements the Cache Fusion concept. It maintains the status of database blocks and is responsible for block transfers between instances. Keeping the instances informed of the obsolescence of their cached database blocks is another important function of this service.
- The Global Enqueue Service (GES) serializes access to database resource for all instances.

The Global Resource Directory replaces the lock database of OPS, while the GCS and GES together replace its Distributed Lock Manager.

2.3 Background processes

Referring again to Figure 2-1, the items in white rectangles are standard components for Oracle instances, while the remaining items are unique to RAC.

Several background processes implement the Oracle9*i* RAC architecture and we briefly describe them below:

- Cluster Manager (CM)

  The Cluster Manager is unique to each supported operating system. It discovers and tracks the membership state of nodes, providing a common view of cluster membership across the cluster. It monitors process health, specifically the health of the database instance.

- Node Monitor (NM)

  The Node Monitor provides the Cluster Manager with the status of the node, the interconnect link, the shared disks, and the Oracle instances.

The Cluster Manager and Node Monitor are hardware vendor-supplied components unique to each platform, with the exception of Microsoft Windows and Linux, for which Oracle provide these components.

- Global Enqueue Service Daemon (LMD)

  The Global Enqueue Service (GES) Daemon is the resource agent process that manages GES resource requests. It also handles deadlock detection in the GES.

- Global Enqueue Service Monitor (LMON)

  The Global Enqueue Service Monitor monitors the entire cluster in order to manage global resources. It manages instance and process expirations and the associated recovery for the GCS and GES. In particular, LMON handles the part of recovery associated with global resources. LMON registers and de-registers from CM.

The LMON and LMD processes implement the Global Enqueue Service (GES).

- Global Cache Service (LMS)

  Each instance has one or more LMS processes. They implement the Global Cache Service.

- Lock process (LCK)

  The lock process manages global enqueue requests. OPS uses it to keep database blocks from being simultaneously changed by different instances, for consistency and data integrity. RAC does not use it for this purpose (though it is still possible to use this mechanism). In RAC, the Global Cache Service ensure consistency and data integrity.

- Global Services Daemon (GSD)

  The Global Services Daemon receives requests from the RAC control utility (SVRCTL) to execute administrative tasks, such as startup or shutdown of the database.

The Database Writer process (DBWR), the Log Writer process (LGWR), the System Monitor process (SMON) and the Process Monitor process (PMON) are the standard background processes also used in a single instance database. Their role is described in *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449.

2.4 Oracle9*i* RAC Guard

Oracle9*i* RAC Guard is a package that works with RAC within a two-node cluster configured in active/passive mode. Using RAC Guard increases the failover capability and enhances monitoring of cluster resources.

RAC Guard is designed to provide a lower-priced entry point into the use of Oracle9*i* RAC. It does not provide the load balancing features of standard RAC, but failover capabilities are enhanced. In an active/passive cluster, one node handles the entire workload while the second node stands by waiting for a failure of the active node. When the active node fails, the second node picks up the workload. With RAC Guard, the transfer to the second node takes place significantly faster than in a standard active/active Oracle RAC configuration.

The configuration of RAC Guard will not be dealt with in this Redpaper. For more information on it, refer to the *Oracle9i Real Application Clusters Guard Administration and Reference Guide*, which can be found at:

http://docs.oracle.com/

3

Planning for Oracle9*i* RAC

An *Oracle9i* Real Application Clusters installation comprises a number of servers with access to a common disk subsystem. In this chapter, we develop plans for a two-node cluster, which we then use for the remainder of the paper to illustrate the implementation of Oracle9*i* RAC.

The two-node cluster is described in overview initially. Further details are then given about each of the major hardware and software components, including:

- The major hardware components
- The xSeries servers
- The Fibre Channel disk subsystem
- SuSE Linux V7.2
- FAStT Storage Manager V7.10 for Linux

The cluster discussed was installed in our laboratory during the development of this paper.

Important: IBM has certified a two-node IBM Netfinity 8500R configuration with Oracle Parallel Server V8.1.6 with patches 8.1.6.1.1 and 8.1.6.1.2 running Windows NT V4.0 Enterprise Edition with SP5. However, the IBM RAC certification for SuSE Linux 7.2 had not been completed at the time this Redpaper was written. Please check the following Web site to view solutions that are currently certified and supported:

http://www.pc.ibm.com/ww/eserver/xseries/clustering/parallel_server.html

3.1 Solution overview

The basic topology of our two-node cluster implementation (see Figure 3-1) utilizes a pair of xSeries 370 servers, which are connected to the production network through Ethernet adapters. The two nodes are interconnected using a private high-speed Gigabit Ethernet link. This provides maximum performance for RAC, which relies heavily on inter-process communication between instances for its Cache Fusion implementation and resource management services.

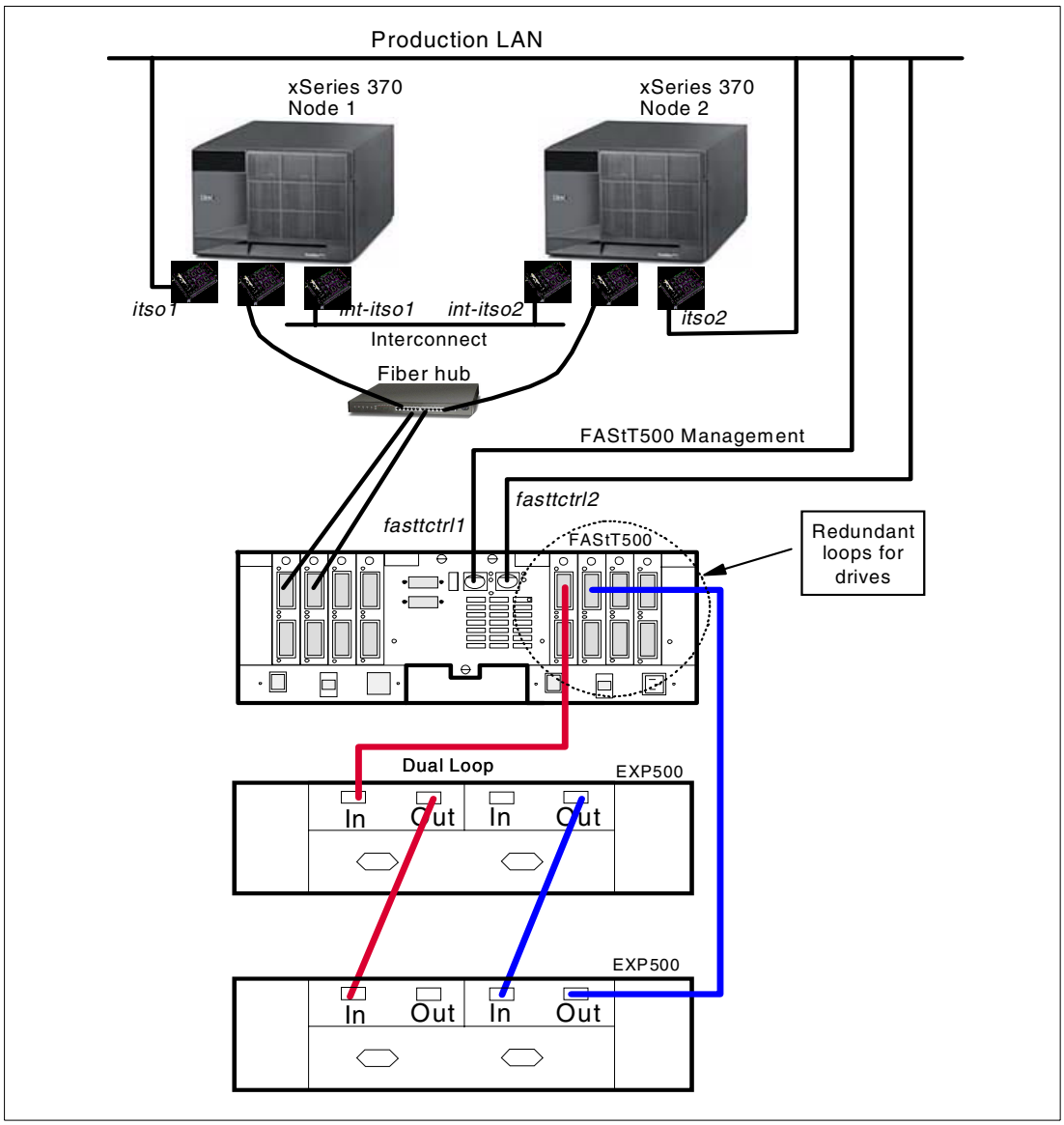

Figure 3-1 Sample two-node Oracle9i Real Application Clusters topology

The FAStT500 Storage Server provides an external common storage subsystem, accessible by both nodes of the cluster. Each server is connected through a fiber hub to the host side of both of the two FAStT500 RAID controllers located in the control unit. The fiber hub is required to provide connectivity from each node to both FAStT500 RAID controllers.

On the disk side of the controller, the FAStT500 is connected to two EXP500 Fibre Channel disk enclosures through a redundant pair of Fibre Channel (FC) loops.

The Storage Manager agent was not available for Linux at the time of writing this paper, so host-attached management was not possible. We therefore used direct-attached management for storage configuration, with both controllers in the FAStT500 RAID control unit connected to the production Ethernet network. This enabled us to use node 1 of the cluster as our management workstation.

The Storage Manager agent will be part of the FAStT Management Suite Java for Linux. If you use the host-attached management with the agent, you will not have to go through the relevant steps outlined in 5.1, “FAStT controller setup” on page 25.

3.2 Hardware components

In this section, we briefly describe the major hardware components and their key features. The components used in our solution are listed in Table 3-1:

Table 3-1 Solution hardware components

Component	Quantity
xSeries 370 server	2
IBM FAStT500 Storage Server	1
IBM FAStT EXP500 Storage Expansion Unit	2
IBM FAStT Host Bus Adapter	2
IBM EtherJet 10/100 PCI Adapter	2
Netfinity Gigabit Ethernet SX Adapter	2
36.4 MB hard disk	20
Fiber hub	1
Fibre Channel Gigabit Interface Converters (GBIC)	10
Fibre Channel cable	8
Ethernet cable	4
Gigabit Ethernet cable (crossover)	1
Rack and mounting hardware	

3.2.1 xSeries servers

Our Oracle9*i* RAC implementation has two nodes, each one being an xSeries 370 system with the following configuration:

- Eight Pentium III Xeon CPUs, running at 700 MHz, with 2048 KB cache.
- 8 GB of main memory.
- 4 MB of video memory.
- Two internal SCSI 17 GB hard drives, used to hold the Linux operating system and the Oracle9*i* executable code.
- Redundant power supplies.
- One FAStT Host Adapter.
- One IBM EtherJet 10/100 PCI Adapter, used to connect to the production LAN. This network provides access to the Oracle9*i* database by applications and users.

- One Netfinity Ethernet Gigabit SX Adapter, used to provide a private interconnect between the two nodes of the cluster. Using Gigabit network adapters ensures maximum performance. Alternatively, EtherJet 10/100 PCI adapters could have been used, in which case they should be set to 100 Mbps, full duplex mode.

Further information on the x370 and other IBM xSeries servers can be found at the IBM xSeries Web site:

http://www.pc.ibm.com/us/eserver/xseries

3.2.2 FAStT500 Storage Server

In the redbook *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449, an IBM 3526 Fibre Channel RAID controller, driving EXP200 external SCSI disk enclosures, was used for external shared storage. Our configuration uses the FAStT500 Storage Server, a newer, all Fibre Channel, subsystem that provides a higher level of performance, availability and expandability than its predecessor. Another Fibre Channel alternative is the entry-level FAStT200 HA Storage Server, which combines controller and disk enclosure in one physical unit.

The FAStT500 provides better scalability than the FAStT200. Eleven EXP500 expansion enclosures, each capable of holding 10 hard drives each, can be connected to each redundant Fibre Channel (FC) loop on the drive side. With two such fully redundant FC loops, this means that up to 220 hard disks can be attached to the FAStT500 without a single point of failure.

Each xSeries 370 is attached to the FAStT500 through a FAStT Host Adapter. A fiber hub is used to provide an access path from each x370 to both of the two controllers in the FAStT500 control unit. Fitting a second FAStT Host Adapter in each x370 provides two physical access paths from each server to the FAStT500. The fiber hub would not be needed in this case, since the connections would be made directly to the FAStT500's mini-hubs.

To take advantage of the additional redundancy offered by using two FAStT Host Adapters in each x370, two additional components are required to enable the redundant data paths. These are the Auto Volume Transfer (AVT) feature in the FAStT microcode, and a host multi-path I/O driver such as Redundant Disk Array Controller (RDAC) used in Windows environments. The Linux multi-path I/O driver was not available at the time of writing this paper. The FAStT Management Suite Java (MSJ) product, part of the FAStT Storage Manager package, will provide multi-path I/O drivers for Linux, and is scheduled for the fourth quarter of 2001.

For more detail on the FAStT500 Storage Server and other FAStT storage products, we recommend the IBM Redbook *Fibre Array Storage Technology, A FAStT Introduction,* SG24-6246.

3.2.3 EXP500 Fibre Storage Expansion Enclosure

In our example, two EXP500 Fibre Storage Expansion Enclosures are attached to the FAStT500 Storage Server. Each EXP500 enclosure contains ten 36.4 GB disks for a total external storage of 728 GB.

3.2.4 FAStT700 and EXP700 solutions

Although not available while this paper was being written, a new FAStT product, the FAStT700, was imminent and we felt it worthwhile to mention this solution.

FAStT700 Storage Server

The FAStT700 Storage Server provides a higher level of performance, availability and expandability than the FAStT500. It provides advanced storage function and incorporates the latest 2 Gbps Fibre Channel technology. FAStT700 features include:

- Dual 2 Gbps controllers
- 1 GB RAID cache
- It can attach up to 224 drives, yielding over 16 TB of storage
- Fibre Channel connection with small form factor pluggable (SFP) GBIC

FAStT FC-2 Host Adapter

The FAStT FC-2 Host Adapter incorporates the latest 1 and 2 Gbps Fibre Channel technology and provides full FC fabric and IP over Fibre Channel support. This adapter is a 64-bit PCI-X adapter, capable of running at 66 or 133 MHz and is fully backward compatible with previous hardware.

FAStT EXP700 Expansion Unit

THe FAStT EXP700 Expansion Unit is a fully redundant, high-capacity, high-performance, Fiber Channel expansion enclosure utilizing universal drive trays. Fourteen bays support any combination of 18, 36, or 73 GB disks, providing a maximum storage capacity of 1 TB. The EXP700 will support 1 Gbps data rates initially, with a later upgrade to 2Gbps.

THe EXP700 can be used with the existing FAStT200 and FAStT500 products and is an excellent choice for upgrading existing systems or for new SAN customers.

3.3 Software components

In this section, we briefly describe important aspects of the software used in our configuration. The software discussed is:

- SuSE Linux V7.2
- The FAStT Host Adapter driver
- Oracle9*i* Release 1 (9.0.1) for Linux Intel
- FAStT Storage Manager V7.10
- FAStT500 Storage Server firmware and NVSRAM

3.3.1 SuSE Linux V7.2

At the time of this writing, Oracle9*i* has been certified with the following distributions of Linux:

- SuSE Linux V7.2
- SuSE Linux V7.1, with Kernel 2.4.4 and GNU Library C 2.2 (glibc 2.2)
- Red Hat Linux V7.1

SuSE Linux V7.2 includes the driver for the FAStT Host Adapter, so this is the distribution we used in our lab configuration.

Note: For current information on product certification, you should check Oracle Metalink or call Oracle Support.

3.3.2 FAStT Storage Manager V7.10

The latest version of FAStT Storage Manager provides several major enhancements to the FAStT200 and FAStT500 Storage Server products. These enhancements include:

- Support for heterogeneous hosts

  With Version 7.10, hosts running different operating systems (heterogeneous hosts) can connect to the same FAStT Storage Server. The operating systems supported are:

  - Windows NT
  - Windows 2000
  - Linux
  - Novell NetWare
  - Sun Solaris
  - HP-UX

  AIX support is scheduled for 2001.

  Heterogeneous host support works in conjunction with storage partitioning. All host systems in the same storage partition must run the same operating system. At least one dedicated storage partition must be defined for each operating system that connects to the Storage Server.

  Note: FAStT Storage Manager does not provide any mechanism to prevent concurrent access to the same disk blocks. Access control must be provided by cluster or file-sharing software.

- Configuration replication

  A FAStT Storage Server configuration can be saved and used later by importing to another Storage Server with identical hardware. This feature provides the following capabilities:

  - Backup of a Storage Server configuration
  - Replication to multiple Storage Servers

- Event Monitoring

  Event Monitor is a background program that constantly checks the Storage Server for critical events such as disk drive failures, RAID controller faults and other similar problems. If a fault is detected, Event Monitor can alert remote systems by sending an e-mail or an SNMP trap. With previous versions of Storage Manager, the Enterprise Management Window had to be open at all times in order to constantly monitor the Storage Server.

- More storage partitions

  Version 7.10 supports up to 16 storage partitions. This is useful as the number of host systems and disk drives attached to the controller increases.

- Command line support

  Script engine commands can be executed using the operating system command-line interface. This allows creation of batch files and is particularly useful when used with systems management applications, such as IBM Director.

- Controller diagnostics

  Built-in diagnostic tests can be performed when troubleshooting the RAID controllers. FC loop connectivity on both the host and drive sides can be verified.

- Access Logical Drive mapping

  The Access Logical Drive is a special logical drive that allows the Storage Manager Agent to communicate with the Fibre Channel RAID controllers through the fiber connection for storage management services.

3.3.3 FAStT Storage Server Firmware and NVSRAM

The microcode of the FAStT Storage Server consists of two packages: the firmware and the NVSRAM package, which includes the settings for booting the FAStT Storage Server. The firmware and NVSRAM are not independent; you must install the correct combination of the two packages.

4

Setting up Linux

As mentioned previously, we based our lab implementation on SuSE Linux V7.2, which has the convenience of including the necessary drivers for the FAStT Host Adapter. In this chapter, we describe how we installed the Linux operating systems and the administrative tasks needed to prepare the servers for Oracle9*i*. We also describe the installation of the FAStT Storage Manager package.

4.1 Installing SuSE Linux

SuSE Linux V7.2 has to be installed on both nodes. The basic installation is straightforward and you should refer to the SuSE Linux documentation for installation procedures if necessary. You may also go to the SuSE Web site at:

http://www.suse.com

for installation support.

In our example, we selected **Default System** from the Software Selection window. From the Detailed Software Selection window, we selected the **Development** and **Network/Server** categories.

The *oracle* account (user ID) has been included in SuSE Linux 7.2. In later sections, this account will be used to install, configure and run Oracle9*i*. You should change the password for the oracle user ID to secure your system. For the configuration steps described later, log in to X-Windows using this account. When necessary, we indicate the relevant steps for which you need to switch to the root user ID.

4.1.1 Define network interfaces

During our installation of Linux, we encountered problems with the network configuration after attempting to configure our second network interface using YaST2. To circumvent this situation, we modified the configuration text files to define our Ethernet and Gigabit Ethernet interfaces and their respective device drivers and then used YaST2 to activate the interfaces and define other network parameters. These are the steps we used:

1. Open a terminal window and switch to root user.
2. Edit /etc/modules.conf, add the statements with bold highlighting in Example 4-1.

Example 4-1 Network device modules

```
#
# YaST2: Network card
#
alias eth0 pcnet32
alias eth1 e1000
alias tr0 off
alias scsi_hostadapter off
alias fb0 off
```

> **Note:** If you have different network adapters from those in this example, and do not know what device driver to specify, try running YaST2 to see if it can determine the correct driver. You can do this by adding the adapter in question. Delete any existing network adapter before making this attempt.

3. Next, edit the file /etc/rc.config. Find and modify the statements (using your own IP addresses) as shown with bold highlighting in Example 4-2:

Example 4-2 Network definitions in rc.config

```
# Number of network cards: "_0" for one, "_0 _1 _2 _3" for four cards
#
NETCONFIG="_0 _1"
#
# This variable contains all indices of active PCMCIA network devices
#
NETCONFIG_PCMCIA=""
#
# IP Adresses
#
IPADDR_0="9.24.105.147"
IPADDR_1="1.1.1.1"
IPADDR_2=""
IPADDR_3=""
#
# Network device names (e.g. "eth0")
#
NETDEV_0="eth0"
NETDEV_1="eth1"
NETDEV_2=""
NETDEV_3=""
#
# Parameters for ifconfig, simply enter "bootp" or "dhcpclient" to use the
# respective service for configuration.
# Sample entry for ethernet:
# IFCONFIG_0="192.168.81.38 broadcast 192.168.81.63 netmask 255.255.255.224"
#
IFCONFIG_0="9.24.105.147 broadcast 9.24.105.255 netmask 255.255.255.0"
IFCONFIG_1="1.1.1.1 broadcast 1.1.1.255 netmask 255.255.255.0"
IFCONFIG_2=""
IFCONFIG_3=""
```

4. Add the host names to /etc/hosts as shown in Example 4-3:

Example 4-3 /etc/hosts entries for cluster node host names

```
9.24.105.147    itso1                   # Oracle9i RAC node 1
9.24.105.148    itso2                   # Oracle9i RAC node 2

1.1.1.1         int-itso1               # Oracle9i RAC interconnect node 1
1.1.1.2         int-itso2               # Oracle9i RAC interconnect node 2

9.24.105.69     fasttctrl1              # FAStT500 controller 1 mgmt
9.24.105.66     fasttctrl2              # FAStT500 controller 2 mgmt
```

5. From the oracle user's shell prompt, enter **`xhost +`** to allow connections to X-windows.
6. From the root user shell prompt, check that the DISPLAY variable is set to "localhost:0.0". If not, set it, then enter **`yast2`**.
7. In the YaST2 window, select **Network/Basic** in the left pane and then **Network card configuration** in the right pane. The Network basic configuration window, shown in Figure 4-1, is displayed:

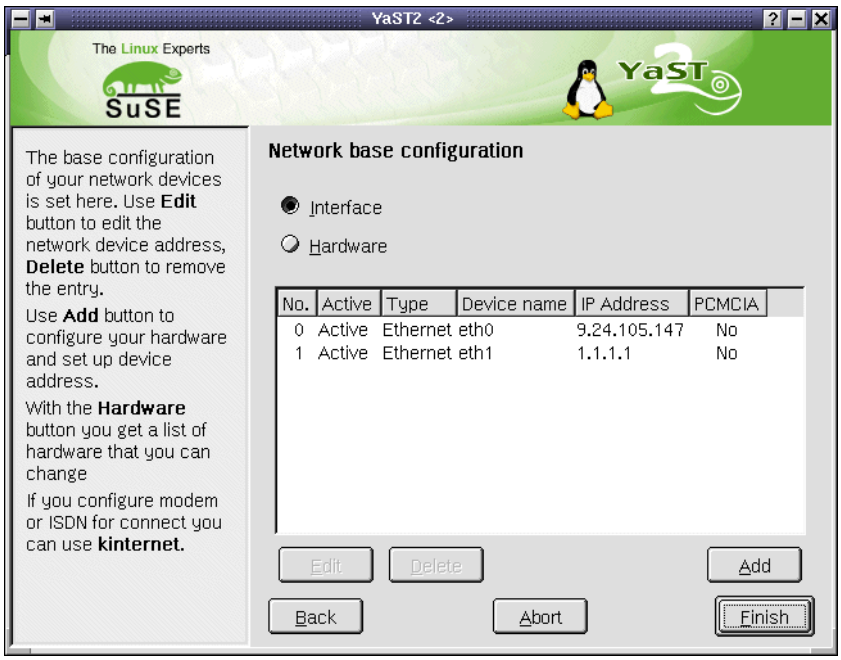

Figure 4-1 Network basic configuration window

8. Verify that the two network devices, eth0 and eth1, are displayed as shown. If so, click **Finish**. Otherwise, go back and make sure that steps 1and 3 have been done correctly.
9. Now we change the node's host name. From YaST2, select **Network/Basic** then **Hostname & DNS**. The Hostname & name server configuration window, shown in Figure 4-2, is displayed:

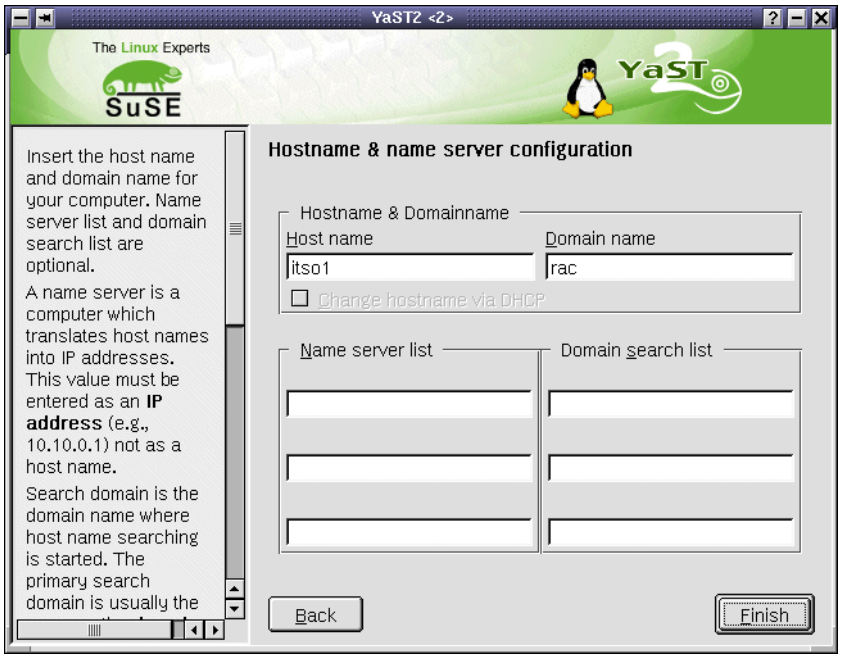

Figure 4-2 Hostname and name server configuration

10. Type in the appropriate entries for host name, domain name and name servers. Click **Finish** when done.

11. Now we will enable network services through the inetd server. From YaST2, select **Network/Basic**, then **Stop/start services**. The Network services window is displayed as shown in Figure 4-3:

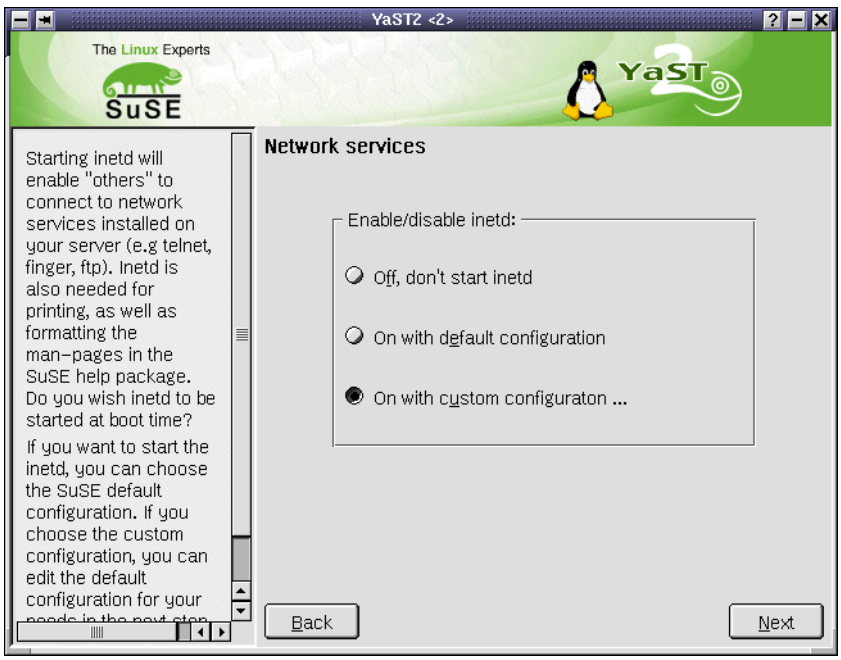

Figure 4-3 Network services window

12. Select **On with custom configuration ...** and click **Next**.

13. The **Enable/disable network services** window is displayed, as shown in Figure 4-4:

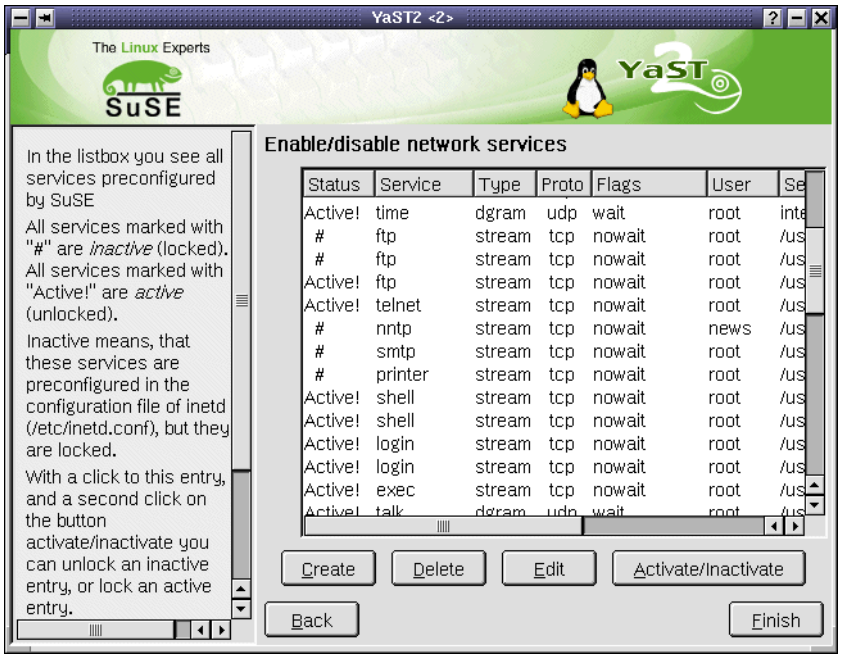

Figure 4-4 Enable/disable network services

14. Verify that the services shown in Table 4-1 are activated:

Table 4-1

Service	Server / Args
ftp	/usr/sbin/tcpd in.ftpd
telnet	/usr/sbin/tcpd in.telnetd
shell	/usr/sbin/tcpd in.rshd -L
login	/usr/sbin/tcpd in.rlogind

If **Active!** does not appear in the Status column for each of these services, select the service concerned and click **Activate/inactivate**.

15. Click **Finish** when done, then click **Close** to exit YaST2.

4.1.2 Set up host equivalence

In order to install and run Oracle9*i*, the oracle user ID must be allowed to use the r-commands (rsh, rlogin, rcp) without supplying a password. We achieve this on both nodes as follows:

1. On node 1 (itso1 in our case), switch to root user and edit /etc/hosts.equiv, adding the host names as shown in Example 4-4.

Example 4-4 /etc/hosts.equiv file

```
#
# hosts.equiv   This file describes the names of the hosts which are
#               to be considered "equivalent", i.e. which are to be
#               trusted enought for allowing rsh(1) commands.
#
# hostname

itso1
itso2
int-itso1
int-itso2
~~
```

2. Make the same changes to /etc/hosts.equiv on node 2 (itso2).
3. Verify the results by executing **rsh**, **rcp** and **rlogin** on both itso1 and itso2.

   Example 4-5 shows the output from these commands on itso1:

Example 4-5 Testing host equivalence on itso1

```
oracle@itso1:~ > rsh itso2 date
Mon Sep 17 10:35:11 EDT 2001
oracle@itso1:~ > rcp /etc/hosts itso2:/tmp
oracle@itso1:~ > rlogin itso2
Last login: Mon Sep 17 10:04:10 from itso1
Have a lot of fun...
oracle@itso2:~ > ls /tmp/hosts
/tmp/hosts
oracle@itso2:~ > exit
logout
rlogin: connection closed.
oracle@itso1:~ >
```

4.1.3 Install the FAStT kernel module

If the FAStT Host Adapter was installed prior to installing Linux, the kernel module for the adapter is automatically installed. However, if the adapter was not installed prior to installing Linux, you will need to install the kernel module. This is done as follows:

1. From the root shell prompt, enter **lsmod** to display loaded modules. An example result is shown in Example 4-6:

Example 4-6 List of kernel modules

```
itso1:~ # lsmod
Module                  Size  Used by
softdog                 1648   0  (unused)
mousedev                4160   0  (unused)
hid                    11728   0  (unused)
input                   3328   0  [mousedev hid]
usb-uhci               23040   0  (unused)
usbcore                48944   1  [hid usb-uhci]
nfsd                   68144   0  (autoclean)
ipv6                  157792  -1  (autoclean)
e1000                  34800   1  (autoclean)
pcnet32                11824   1  (autoclean)
ipchains               35424   0  (unused)
lvm-mod                43488  40  (autoclean)
qla2x00               174544   0
aic7xxx               101584   3
```

2. Examine the result from the **lsmod** command. If qla2x00 is listed (as shown in Example 4-6 in boldface), the kernel module for the FAStT adapter is loaded.
3. If qla2x00 is not loaded, install it by entering **insmod qla2x00**.

   To load it automatically at server startup, make these changes on each node:

   – Edit /etc/rc.config, find the line beginning with INITRD_MODULES and add qla2x00 to the list of modules, as shown in Example 4-7. Be sure to leave a blank between the module names.

   – As root, run **/sbin/mk_initrd**.

Example 4-7 /etc/rc.config

```
#
# This variable contains the list of modules to be added to the initial
# ramdisk by calling the script "mk_initrd"
# (like drivers for scsi-controllers, for lvm or reiserfs)
#
INITRD_MODULES="aic7xxx qla2x00"
```

4.1.4 Install Linux patches

Patches for SuSE Linux can be found on the SuSE Web site at:

http://www.suse.com

Select **Downloads**, then **Updates, Patches, Bugfixes**. Select the appropriate platform and Linux version for a list of available patches.

In our example, we installed the following patch:

```
java-1.1.8v3-0.src.rpm
```

using the command:

```
rpm -U java-1.1.8v3-0.src.rpm
```

4.2 Install FAStT Storage Manager client

Obtain this software from the IBM Web site at:

http://www.pc.ibm.com

1. Select **Storage**, then **Fibre Channel products and Storage Area Networks**.
2. Next, select **View download support for FAStT Storage Manager software** under **FAStT Storage Manager**, and download **IBM FAStT Storage Manager for RedHat Linux Version 7.10.**

The downloaded file is a gzip-compressed tar file. Transfer the file to node 1 and use the **gunzip** command to uncompress it. Then extract the package named SM7client-Linux-0710G506.rpm using the **tar** command.

Installation of the FAStT Storage Manager Client is required on one node of the cluster only. We installed it on itso1 using the following command from a root shell prompt:

```
rpm -U SM7client-Linux-0710G506.rpm
```

5

FAStT controller and Storage Manager setup

This section describes how to set up the FAStT Storage Manager software to configure the FAStT500 Storage Server. First we describe how to set up the FAStT Storage Server for direct-attached management and then we configure the FAStT Storage Manager so that it can manage the disk subsystem.

5.1 FAStT controller setup

There are two methods available for managing the FAStT Storage Server. These are:

- Direct-attached (or out-of-band)

  The direct-attached method uses an Ethernet link to connect to each RAID controller. This method requires that the Storage Manager Client be installed on the managing machine. By default, the FAStT Storage Server uses a DHCP/BOOTP server to obtain TCP/IP settings. Alternatively, you can configure the Storage Server with static IP addresses using a serial interface and an asynchronous terminal emulation utility.

  For security reasons, it is recommended that the Ethernet connection between the FAStT Storage Manager Client and the Storage Servers be a dedicated segment, separate from your production network.

- Host-attached (or in-band)

  This method uses the fiber connection itself for management traffic. For host-attached management, the FAStT Storage Manager Agent must be installed on the host system. The client may be installed on a networked workstation or the host system itself.

In our example, we use the direct-attached method to manage our FAStT500 Storage Server and set up the FAStT controllers with static IP addresses.

In order to set up static IP addresses, we connect to each FAStT controller using the serial ports. Node 1 (itso1) is used as the managing station and we use **`minicom`**, which is a serial communications program provided with SuSE Linux, for the terminal emulation.

The procedure we used to set up the FAStT500 is as follows:

1. Connect a null modem cable from the itso1 serial port to the serial port of FAStT500 controller A.
2. Open a terminal window and switch to the root user. Run **minicom** in setup mode with the command:

   ```
   minicom -s
   ```

   The configuration window (see Figure 5-1) is displayed:

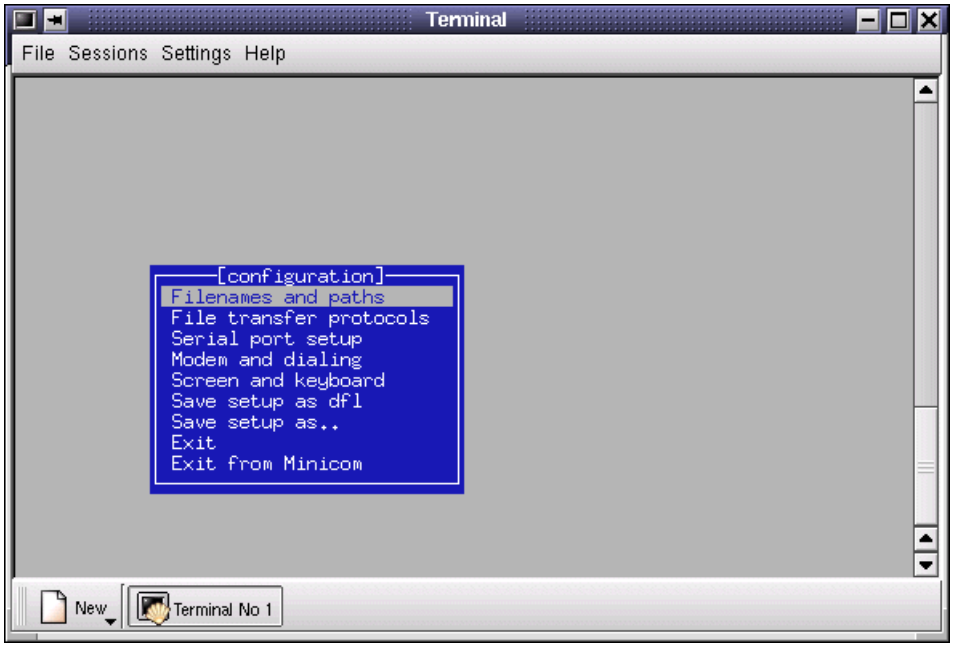

Figure 5-1 Minicom setup window

3. Use the cursor keys to select **Serial port setup** and press Enter to display the window that allows you to modify the serial port settings, as shown in Figure 5-2:

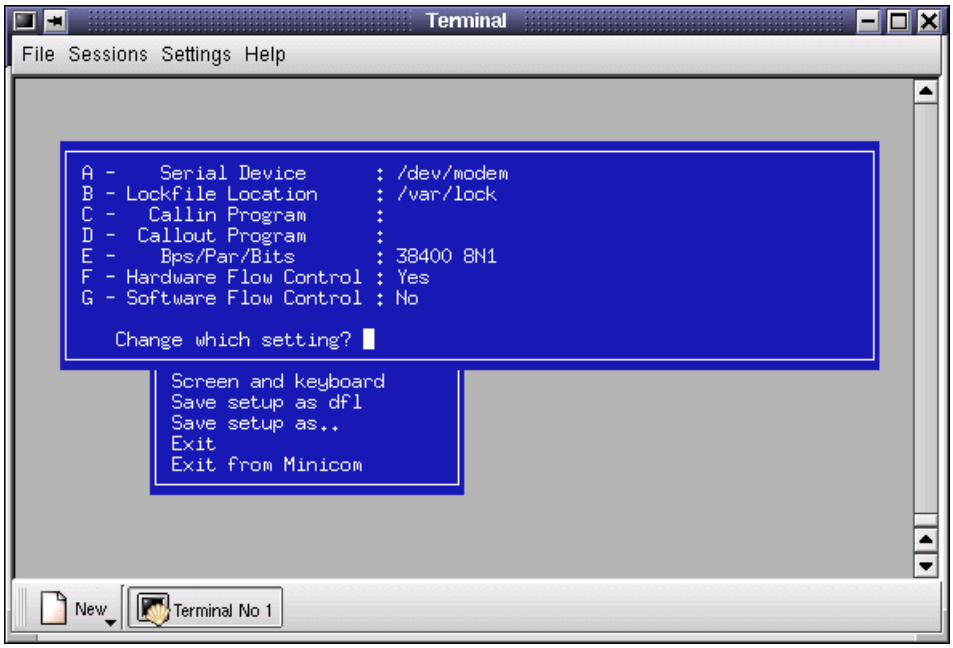

Figure 5-2 Minicom serial port setup

4. Press A to change the serial device. The cursor will be positioned in the Serial Device name field. Change the serial device name to /dev/ttyS0 (this is the device for the first serial port of the server) and press Enter.
5. We need the port to be set to 9600 bps, 8 bits, no parity and 1 stop bit, so choose **E** to change the communication settings, and another window appears as shown in Figure 5-3:

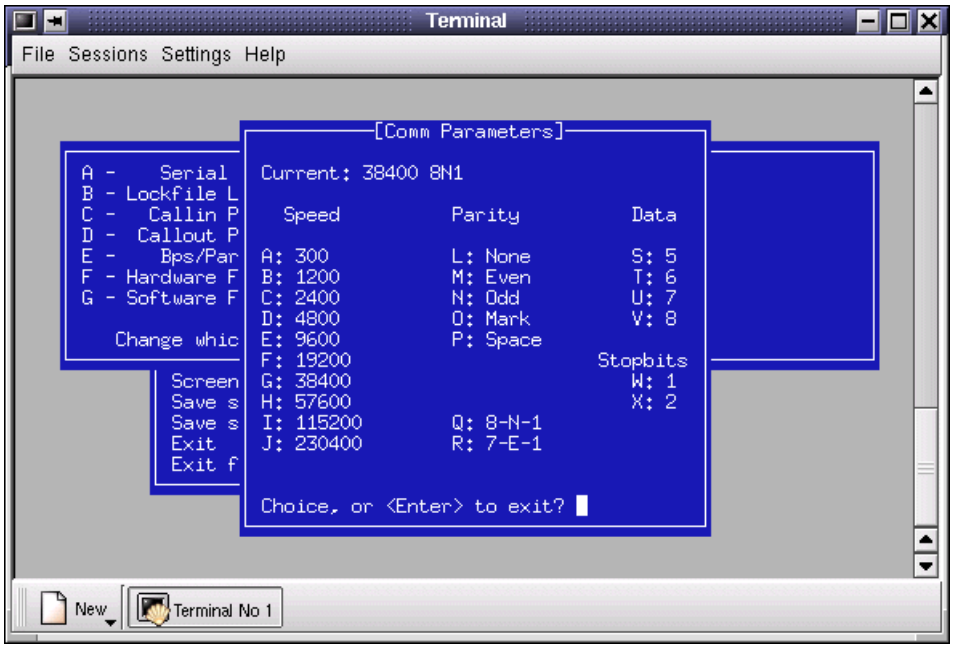

Figure 5-3 Minicom Comm Parameters

6. Use the menu to select the correct settings. For example, choose **E** to set the speed to 9600 bps.
7. When the current settings are correct (9600 8N1), press Enter.

8. Press Enter again to return to the main menu.
9. User the cursor keys to select **Save setup as dfl** and press Enter. A confirmation window is displayed, as shown in Figure 5-4:

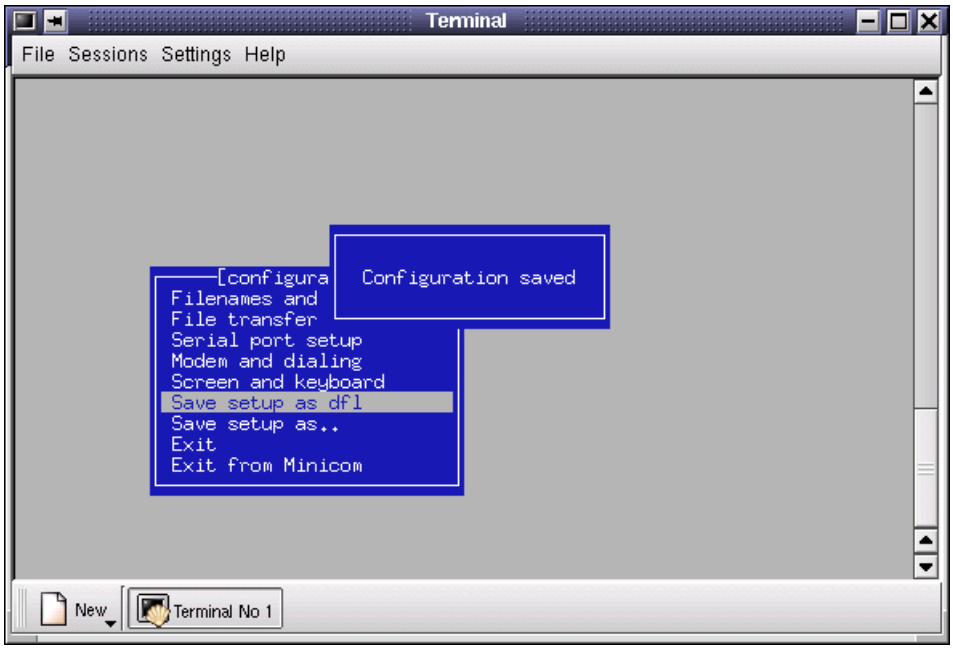

Figure 5-4 Saving setup

10. Select **Exit from Minicom** and press Enter.
11. Now run **`minicom`** in normal mode by omitting the -s switch:

```
minicom
```

12. Press Ctrl+A then Z to get the Minicom Command Summary menu (shown in Figure 5-5):

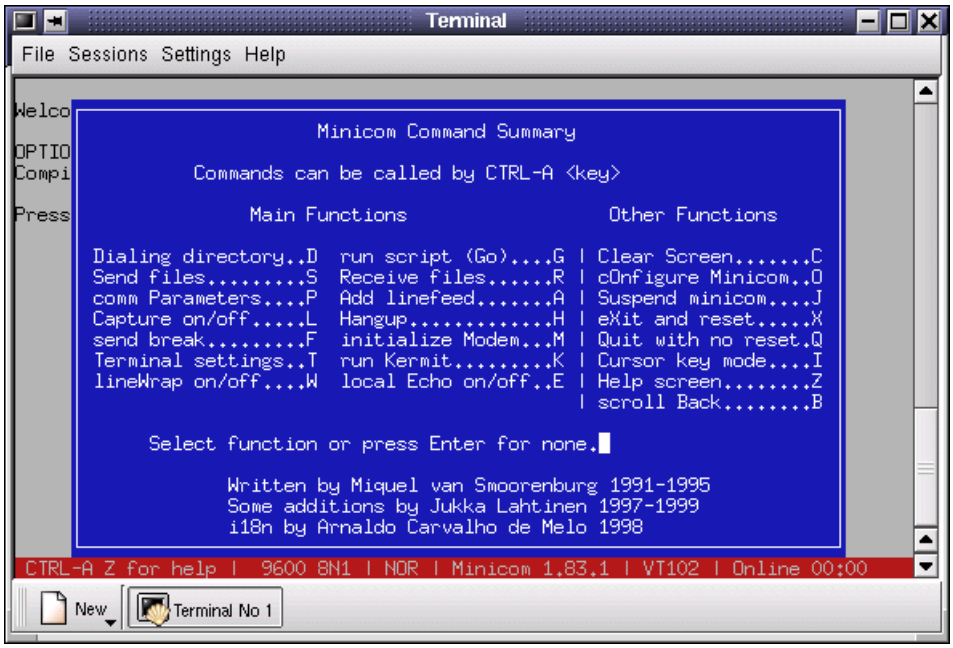

Figure 5-5 Minicom Command Summary pop-up menu

13. Send a break by choosing F. Then press the Esc key for the SHELL. The password prompt appears as shown in Figure 5-6.

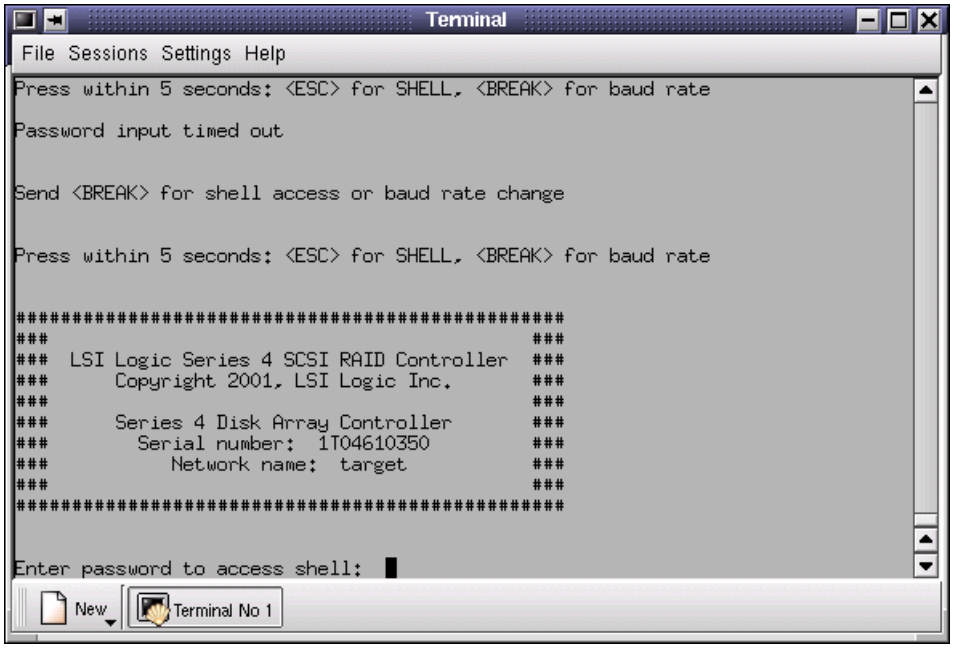

Figure 5-6 FAStT shell password prompt

14. Enter the FAStT500 password; the default is `infiniti`. You are now at the FAStT controller shell prompt, shown in Figure 5-7:

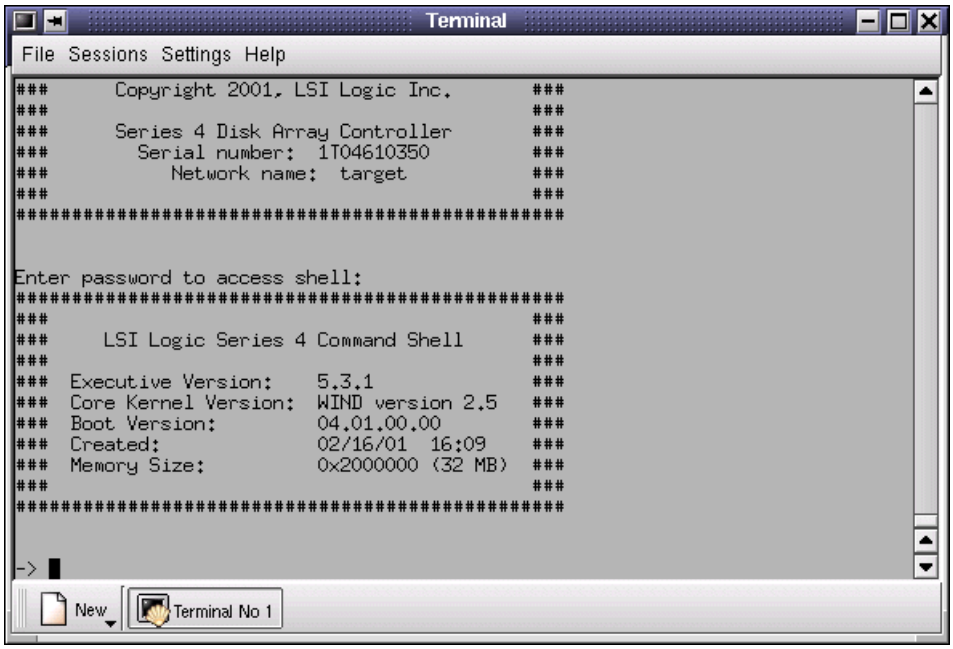

Figure 5-7 FAStT shell prompt

15. At the shell prompt, type `netCfgSet` and press Enter.

16. A series of options will be displayed in line mode. Press Enter for the options you don't want to change. At the `My Host Name` prompt, enter the controller host name (we used `fasttctrl1` in our example) then press Enter to continue.

17. At the `My IP Address` prompt, enter the controller IP address. In our example, the IP address we entered is 9.24.105.69, replacing the address obtained using DHCP. Press Enter to accept all of the remaining parameters. After the last parameter, the window appears as shown in Figure 5-8:

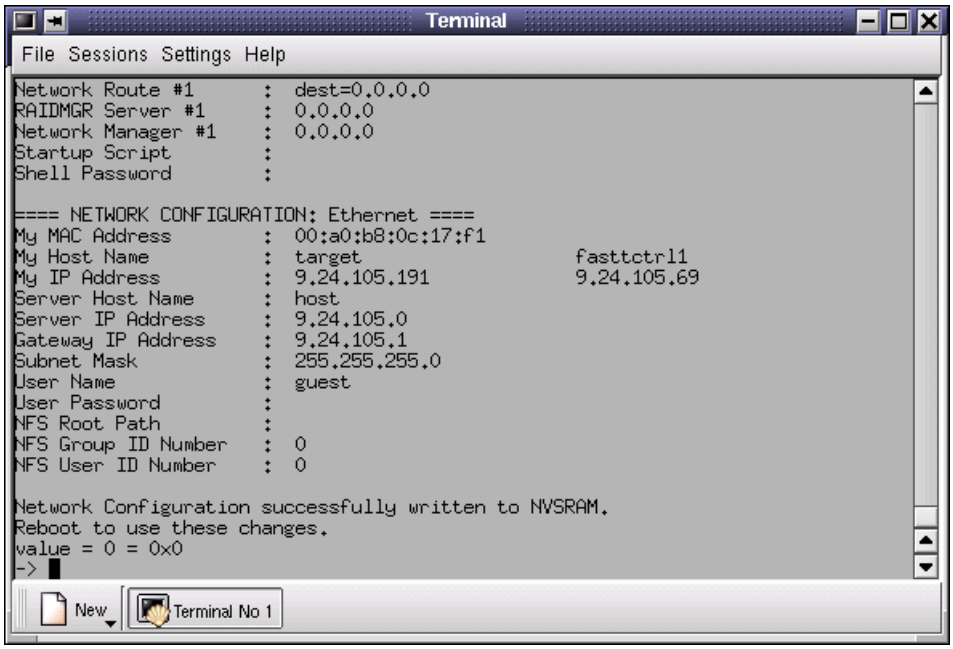

Figure 5-8 FAStT parameter settings

18. Reboot the controller by typing the command **`sysReboot`** and pressing Enter.
19. Connect the null modem cable to the serial port for controller B of the FAStT500. Repeat the above steps, this time using the host name and IP address for Controller B. In our example, Controller B's host name is `fasttctrl2` and its IP address is 9.24.105.66.

5.2 FAStT Storage Manager setup

We now describe the steps required to set up the FAStT Storage Manager. First we add the logical devices that represent the FAStT500 controllers, then we discuss firmware and NVSRAM versions, and finally we describe how to enable Premium Features, which is necessary to be able to use storage partitioning.

5.2.1 Adding FAStT controllers

Follow these steps to add the FAStT controllers to Storage Manager:

1. Launch (as root user) the Storage Manager client with the command:

   `/usr/bin/SM7client`

2. The FAStT Storage Manager window (see Figure 5-9) is displayed. If not, check that you have set the DISPLAY variable to "localhost:0.0" in the current shell, and that the **`xhost +`** command has been run by the user (oracle) that is running X-windows. This was discussed in 4.1.1, "Define network interfaces" on page 17.

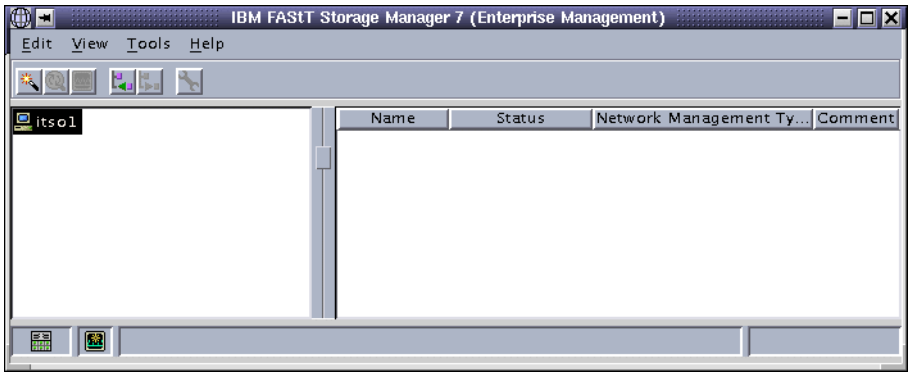

Figure 5-9 FAStT Storage Manager

3. Select **Edit -> Add Device** as shown in Figure 5-10:

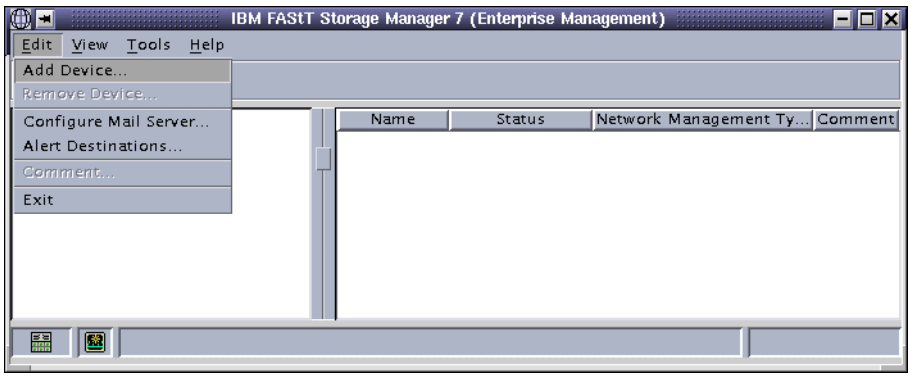

Figure 5-10 Adding a device

4. In the resulting window, enter the name of the first controller, `fasttctrl1` and click **Add** (see Figure 5-11):

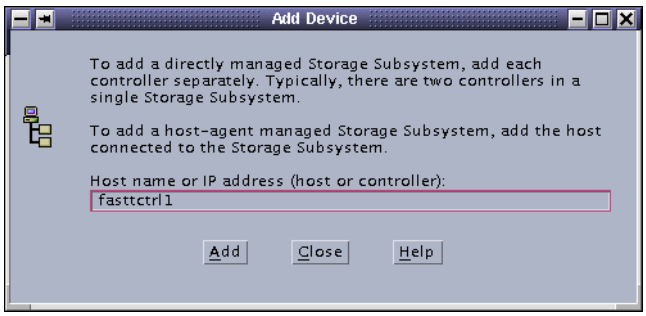

Figure 5-11 Add Device window

5. In the same window, enter the name of the second controller, `fasttctrl2` and click **Add** (see Figure 5-12):

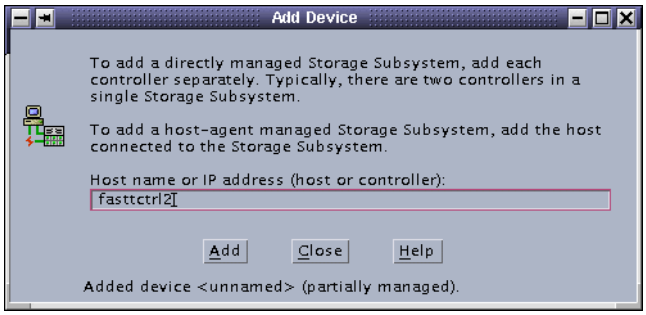

Figure 5-12 Adding the second controller

6. Click **Close** to return to the Storage Manager window. Figure 5-13 shows the entry **Storage Subsystem**, which represents the two controllers we have just added.

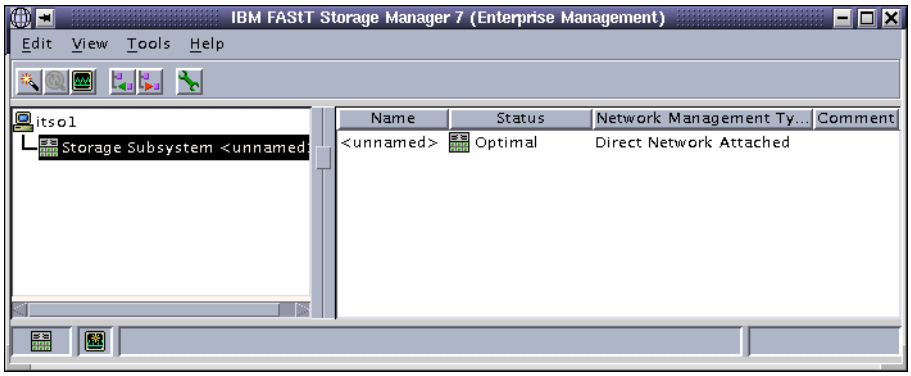

Figure 5-13 SM7client main window

5.2.2 Firmware and NVSRAM

Storage partitioning for a RAC implementation requires that FAStT Storage Manager Premium Features are enabled. To allow the use of storage partitioning and heterogeneous hosts, we updated the firmware and NVSRAM in our configuration. Table 5-1 shows the original and updated versions of firmware and NVSRAM.

Table 5-1 Firmware and NVSRAM versions

	Original	Upated	Filename
Firmware	04.00.02.02	04.01.02.07	FW_04010207_04010200.dlp
NVSRAM	NV4774WNT856004	NV3552R710NT004	NV3552R710NT004.dlp

Use a Windows workstation to download the self-extracting zipped file containing the firmware and NVSRAM files. These are available from the IBM xSeries Web site at:

`http://www.pc.ibm.com/us/eserver/xseries`

To locate the file, select **Storage**, then **Fibre Channel products and Storage Area Networks**. Next, select **View download support for FAStT Storage Manager software** under **FAStT Storage Manager**. Download **Servers - IBM FAStT Fibre Channel Controller Firmware version 04.01.02.07** or a later version, if it has been updated.

Run the self-extracting file on the Windows workstation. Two directories, called Firmware and NVSRAM, will be created. The firmware and NVSRAM files will be found in subdirectories corresponding to the FAStT model number. In our example, our FAStT Storage Server is a Model 3552.

Transfer the .dlp files to node 1. The procedure for downloading firmware and NVSRAM is as follows:

1. Select **Storage Subsystem <unnamed>** in the left pane. From the menu, select **Tools -> Manage Device**. The window shown in Figure 5-14 is displayed:

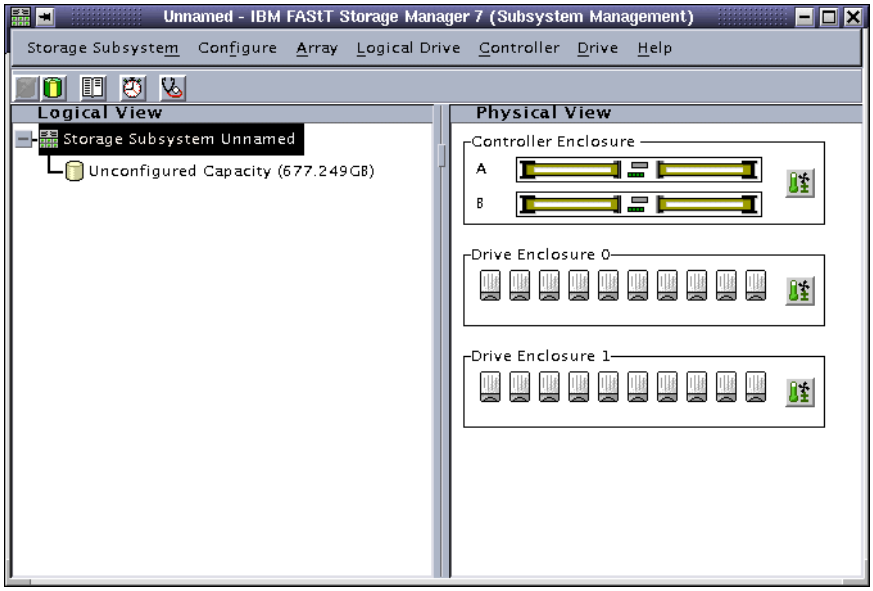

Figure 5-14 Subsystem Management window

2. From the menu, select **Storage Subsystem -> Download -> Firmware**. In the resulting window, point to the location of the firmware file, then click **Update** to refresh File Information. Click **OK** to continue.
3. A confirmation window will appear. Click **Yes** to perform the download.
4. To perform the NVSRAM download, select **Storage Subsystem -> Download -> NVSRAM**. As in the previous step, point to the location of the NVSRAM file, and click **Update** to refresh File Information. Click **OK** to continue, and click **Yes** when the confirmation window appears.

5.2.3 Enabling Premium Features

Storage Partitioning is a premium feature of the FAStT Storage Server which has to be enabled. To do so, follow these steps:

1. From the Subsystem Management window menu, select **Storage Subsystem -> Premium Features**. If **Enable** and **Disable** are greyed out, you need to request a key to be able to enable storage partitioning.

   The Feature Enable Identifier is needed to request a key. It can be obtained by selecting **Storage Subsystem -> Premium Features -> List**. A small window containing the Feature Enable Identifier is displayed (see Figure 5-15):

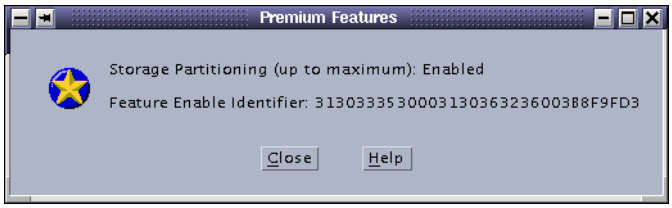

Figure 5-15 List Premium Features

Record the Feature Enable Identifier, which is a 32-character string.

You will also need the model number of the FAStT Storage Server. This is found on the serial number tag at the back of the FAStT Storage Server.

Call IBM Support and request a feature key file for enabling Premium Features.

2. Once you have obtained the key file, enable Premium Features by selecting **Storage Subsystem -> Premium Features -> Enable**. In the resulting window, point to the location where the key file is stored. A confirmation window will appear. Click **Yes** to continue.

   The change will take place immediately.

3. Exit the Storage Manager client.

6

Pre-installation tasks

Oracle9*i* RAC uses shared storage to store database files, information about those databases, and configuration data about the nodes in the cluster. In addition, data necessary to maintain the integrity of the cluster under failure conditions, such as the interconnect link being broken, is stored on the shared Cluster Manager (CM) disk.

In this chapter, we explain how to configure both the shared storage for the RAC configuration and the Cluster Manager disk, and the devices that access it. Specifically, we create a logical drive on a RAID-1 array that contains two logical volumes. These are used as raw devices for the CM disk and the configuration repository for the RAC system. Good performance is important for the CM disk, so it is inadvisable to use the same logical drive to store data that will be heavily accessed. The configuration repository for the RAC system can be stored on it, however, since it is not frequently accessed.

Storage configuration for Oracle databases is covered in Chapter 8, “Creating and validating a database” on page 75.

6.1 Storage configuration

Our goal is to create a logical drive on a RAID-1 array, and to configure partitioning to make this logical drive available to the nodes of the cluster. To do so, perform these steps on the node of the cluster on which FAStT Storage Manager is installed:

1. Start FAStT Storage Manager as root on one node of the cluster by running the command:

   ```
   SM7client
   ```

2. Double-click **Storage subsystem** to display the Subsystem Management window as shown in Figure 6-1:

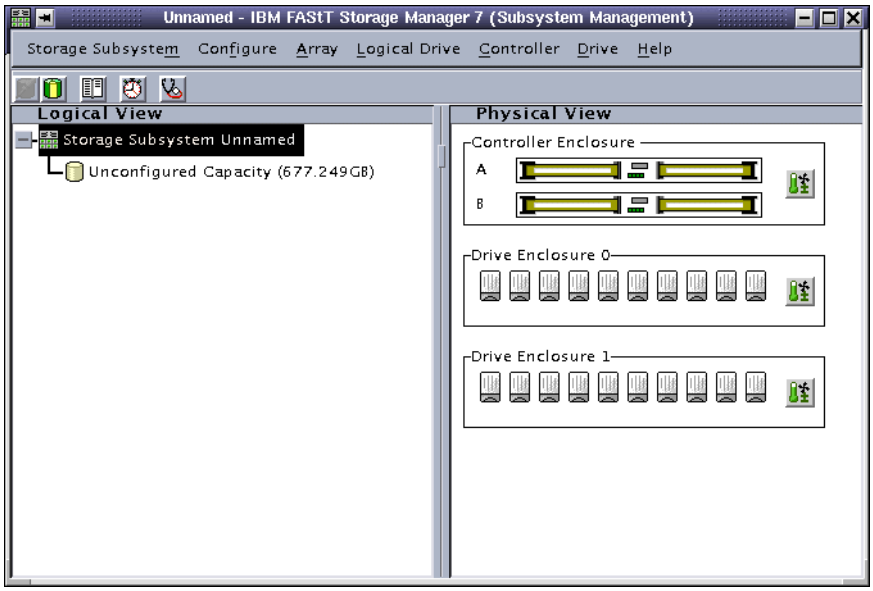

Figure 6-1 Subsystem Management window

3. To create the logical drive, highlight **Unconfigured Capacity**.
4. From the main menu, select **Configure -> Create Array/Logical Drive...** (see Figure 6-2):

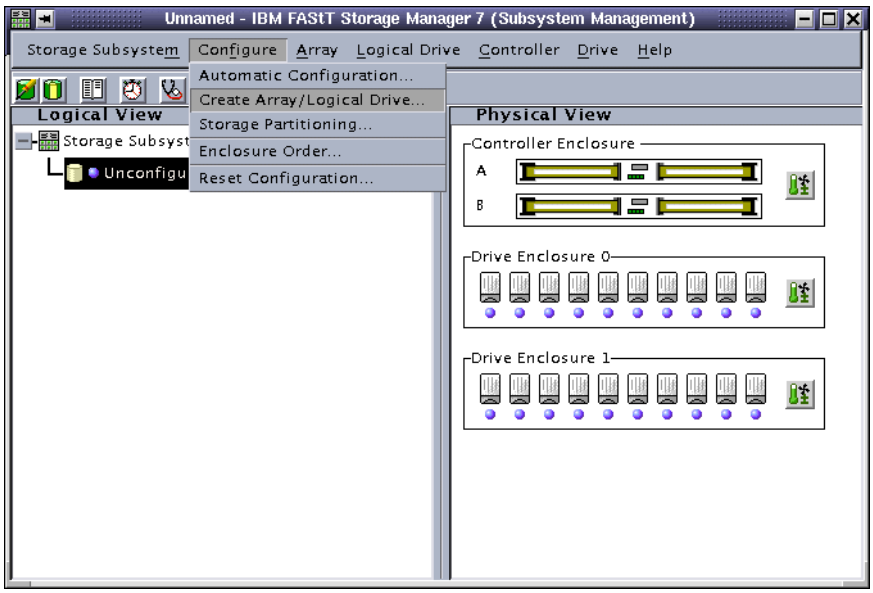

Figure 6-2 Create Array/Logical Drive...

5. The Create Logical Drive - Select Array window is displayed. In the RAID level selection pane, select **RAID 1**. In the Capacity selection pane, select **2** drives, as shown in Figure 6-3:

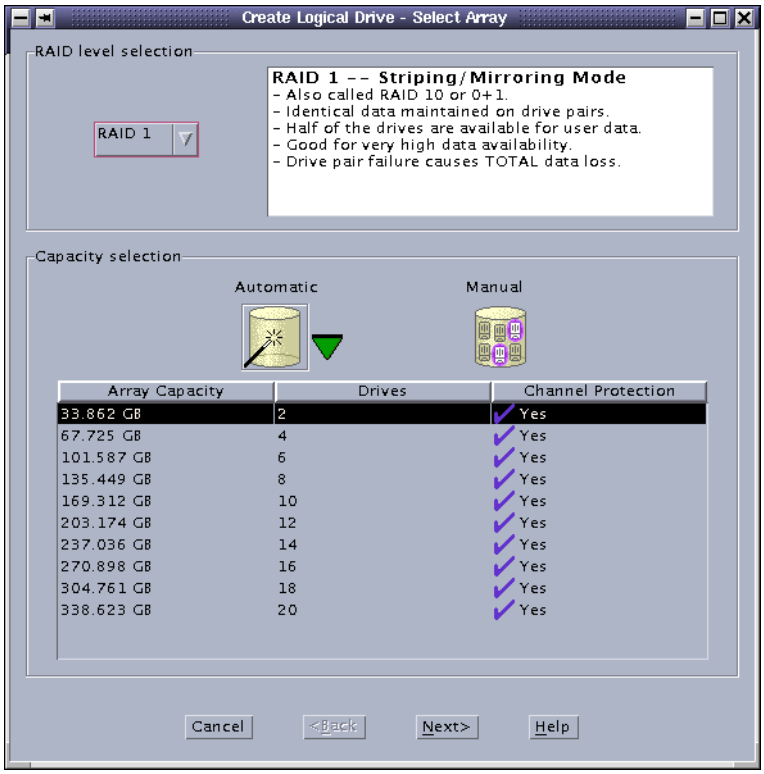

Figure 6-3 Create Logical Drive - Select Array window

6. Click **Next** to display the Create Logical Drive - Specify Logical Drive Parameters window.
   - We recommend that you assign the total capacity of the array to the logical drive, even though you may not need all of its capacity. Logical Volume Manager allows us to utilize only a portion of this logical drive, with the remainder reserved for future growth. We don't want these disks used for another purpose, since their performance is crucial for RAC.
   - Enter the name of the logical drive: `RAC`.
   - For the expected logical drive usage, select **Database**.
   - For the segment size, keep the value **Based on Expected Usage.**
   - Select a slot for the controller ownership, we chose **Slot A.**
   - Select **Map later with Storage Partitioning** in the Logical Drive-to-LUN mapping group.

   You can see our settings in Figure 6-4:

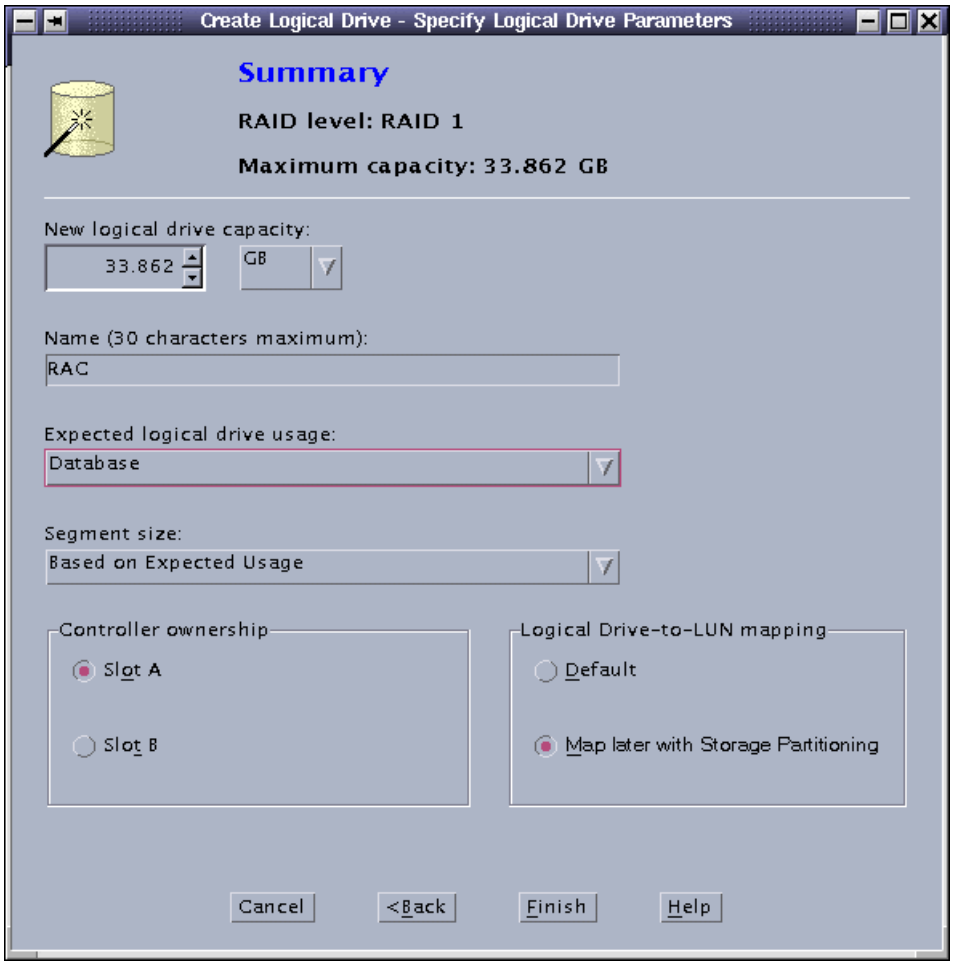

Figure 6-4 Specify Logical Drive Parameters window

7. Click **Finish** to display the Logical Drive Creation Successful window and click **OK** to complete this process.

The new array and logical drive now appear in the Subsystem Management window (see Figure 6-5):

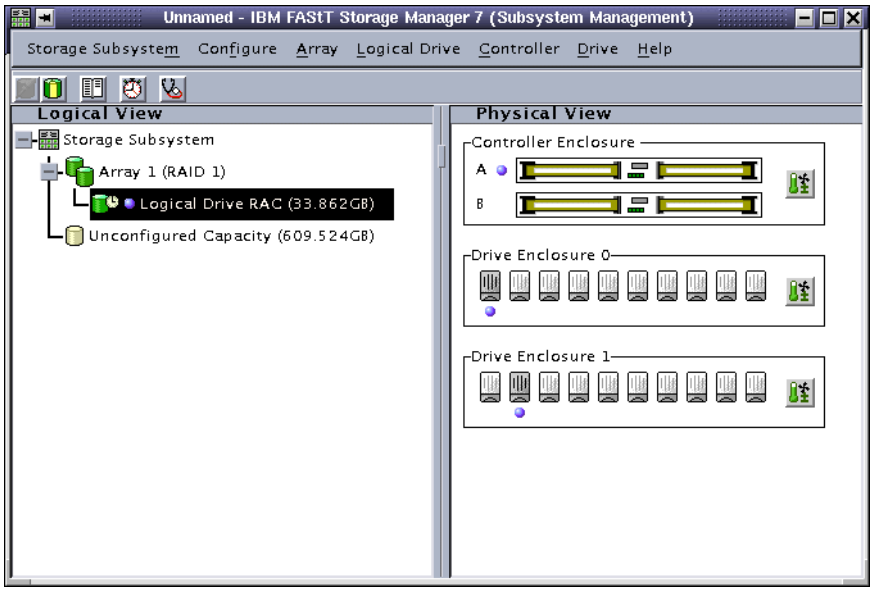

Figure 6-5 The new array and logical drive are now displayed

6.1.1 Storage partitioning

Next we have to configure the storage partitions. Storage Partitioning is a premium feature that must be licensed as already discussed in 5.2.3, “Enabling Premium Features” on page 33. To check that it has been licensed, from the menu select **Storage Subsystem -> Premium Features -> List**. If Storage Partitioning is licensed, the following window is displayed:

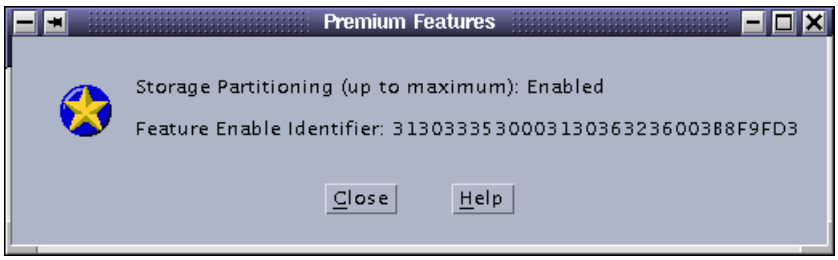

Figure 6-6 Storage Partitioning licensed

If Storage Partitioning is not licensed, then just change the host type for the storage subsystem to Linux, by selecting **Configure -> Change Storage Subsystem Host Type** and choosing the Linux option (see Figure 6-7) and then continue with 6.1.2, “Activating the Linux devices” on page 45.

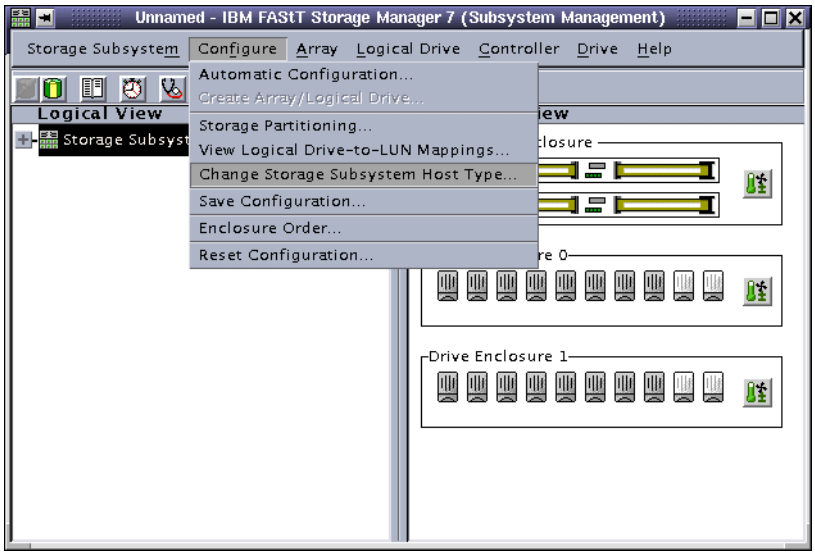

Figure 6-7 Change Storage Subsystem Host Type

If, however, you do have Storage Partitioning licensed, follow these steps:

1. Highlight the logical drive we have just created (RAC).
2. In the menu, select **Configure -> Storage Partitioning...** to display the Storage Partitioning Definitions window (see Figure 6-8):

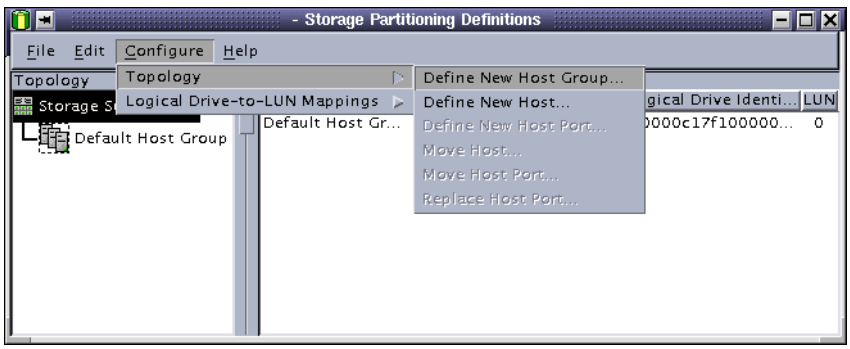

Figure 6-8 Define New Host Group

3. Now we define the group that will represent the cluster. Select **Configure -> Topology -> Define New Host Group** as shown in Figure 6-8 to display the Define New Host Group window (Figure 6-8):

Figure 6-9 Define New Host Group window

4. Enter the name of the host group that represents the entire cluster. We named our cluster `ITSO`. Click **Add**, then **Close**.

The next step is to define the nodes of the cluster, called hosts by Storage Manager.

5. Highlight the Host Group you have created, ITSO in our example.
6. Select **Configure -> Topology -> Define New Host...** (Figure 6-10):

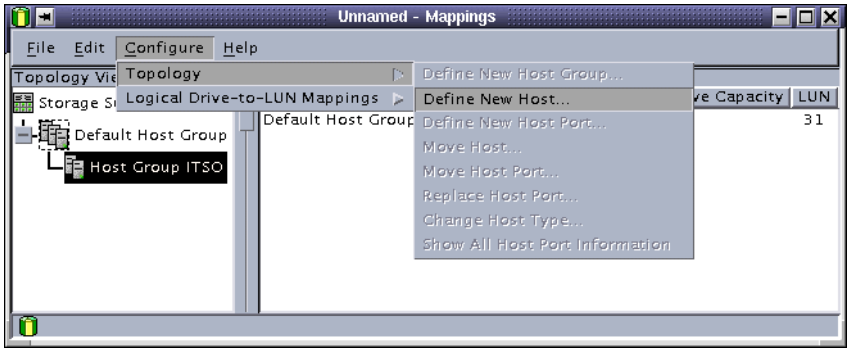

Figure 6-10 Define New Host

7. The **Define New Host** window is displayed (Figure 6-11):

Figure 6-11 Define New Host window

8. Enter the name of the host and click **Add**.

   This does not have to be the server's host name, but using it helps to keep things simple. You can see we have entered `ITSO1`.

9. Enter the name of the second host, and click **Add**. In our case, we entered `ITSO2` (see Figure 6-12):

Figure 6-12 Define New Host window

10. Repeat step 9 for all servers in the cluster and then click **Close**.

Assigning Host Ports

Now we define the adapters belonging to the nodes in the cluster. Storage Manager calls them Host Ports.

1. Select the first host.
2. Select **Configure -> Topology -> Define New host Port...** as shown in Figure 6-13:

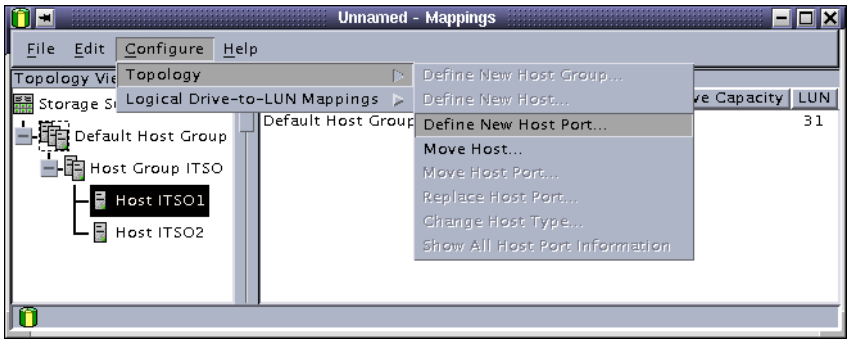

Figure 6-13 Define New Host Port

The **Define New Host Port** window is displayed (Figure 6-14):

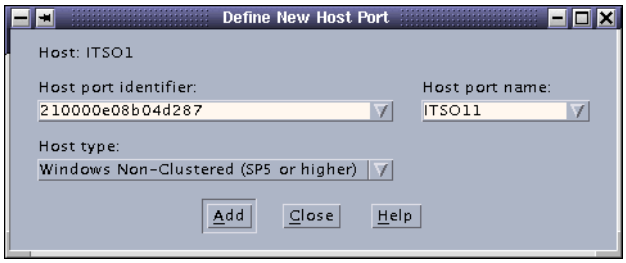

Figure 6-14 Define New Host Port window

3. In the **Host port identifier** list box, select the adapter identifier for the host.

Note: To find a node's adapter identifier:

- Log in to the node as root.
- Check the qla2x00 module is loaded with the **`lsmod`** command. If it is not, load it with the command (as root): **`insmod qla2x00`**.
- Check the last entry in the file /var/log/messages that contains scsi-qla0-adapter-port, using the command:

  `cat /var/log/messages | grep "scsi-qla0-adapter-port"`

  The number displayed is the adapter identifier. If multiple qlogic adapters are installed in the server, search for the string `scsi-qla?-adapter-port`, replacing the question mark with 0, 1, and so on, to locate each adapter in turn

An example is shown in Figure 6-15:

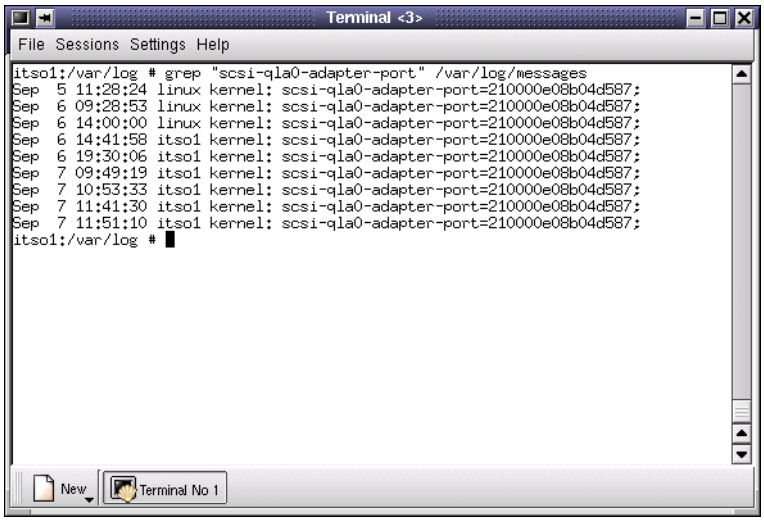

Figure 6-15 cat /var/log/messages | grep “scsi-qla0-adapter-port” output

4. In the Host port name field, enter a name for your adapter, we used `itso1 port`.
5. In the Host type list box, select **Linux** (see Figure 6-16).

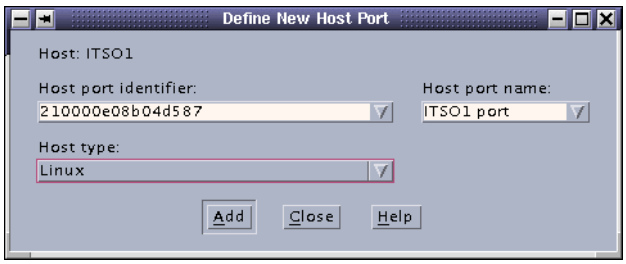

Figure 6-16 Define New Host Port window

6. Click **Add**, then click **Close**

Repeat steps 2 through 6 for the second host.

Mapping the logical drive

The next step is to make the logical drive we have created available to the nodes of the cluster. Follow these steps.

1. Select the host group **ITSO**.
2. From the menu, select **Configure -> Logical Drive-to-LUN Mapping -> Define New Mapping** (see Figure 6-17):

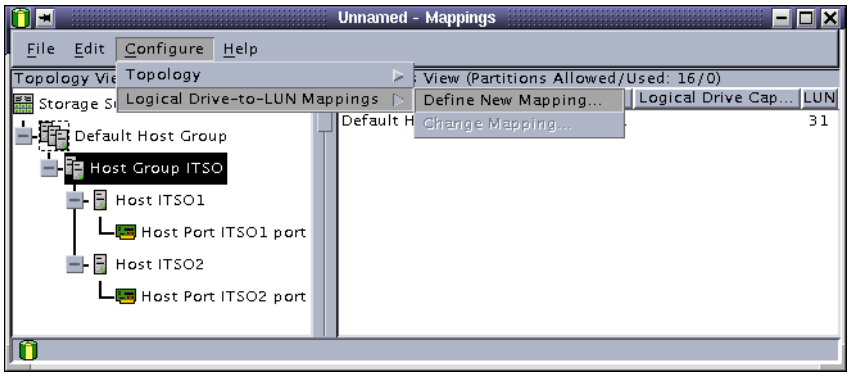

Figure 6-17 Define New Mapping

3. In the Define New Mapping window, select your logical drive name. In 6.1, “Storage configuration” on page 35, we used the name RAC for our logical drive (see Figure 6-18):

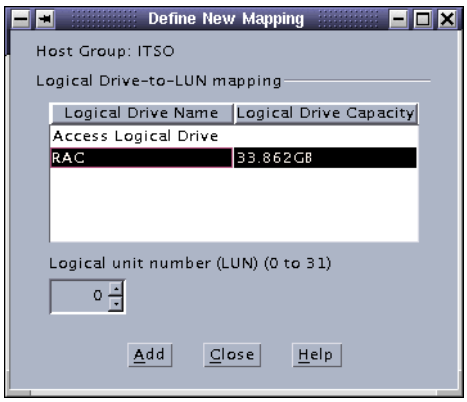

Figure 6-18 Define New Mapping window

4. Click **Add**, then **Close**. The Topology View is updated to reflect these changes, as shown in Figure 6-19:

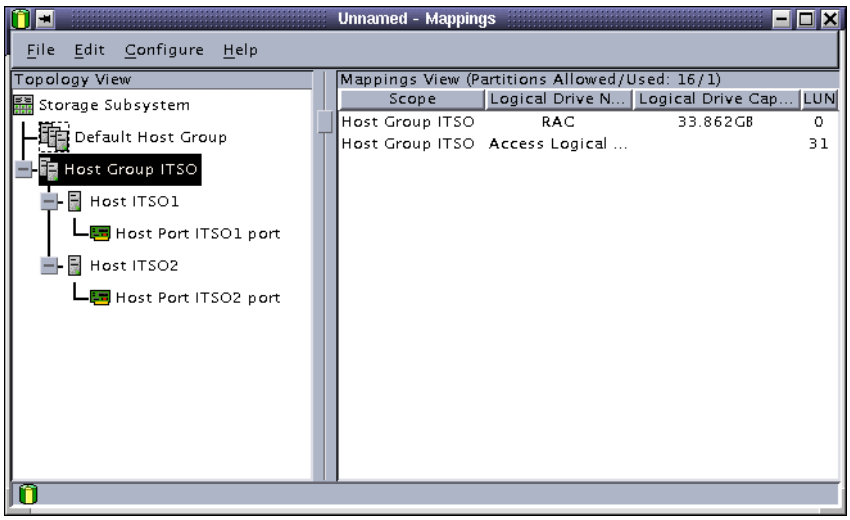

Figure 6-19 New topology view

Close the Storage Partitioning Definitions window and exit Storage Manager.

6.1.2 Activating the Linux devices

To enable the operating system to see the logical drive you have created, follow these steps on all nodes of the cluster:

1. As root, unload and reload the qlogic module using the **rmmod** and **insmod** commands, as shown in Figure 6-20:

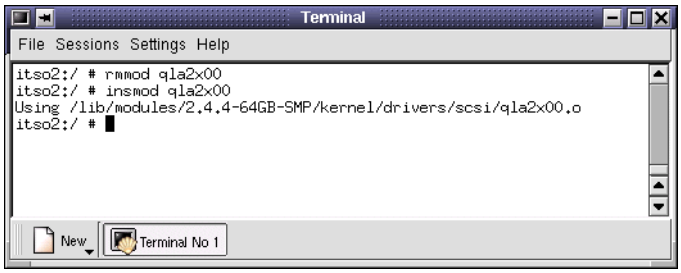

Figure 6-20 reloading the qlogic module

2. Check the last 30 lines of the log file /var/log/messages for errors.
3. If everything has worked as expected, the device that maps the logical drive for RAC appears in the next to the last line. In our case it is /dev/sdc, as you can see in Figure 6-21.

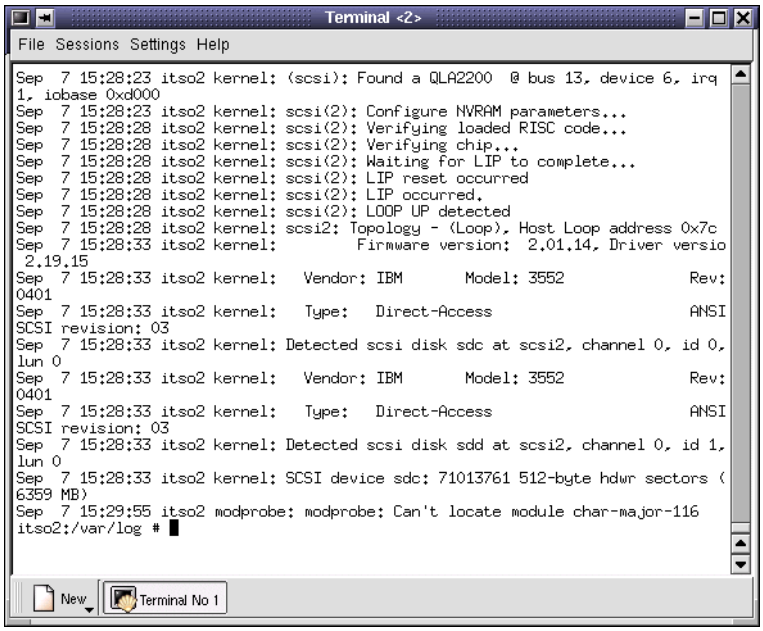

Figure 6-21 /var/log/messages

> **Note:** To see only the disk device for the logical drive, execute this command:
>
> **cat /var/log/messages | grep "SCSI device"**

6.2 Logical Volume Manager configuration

We use Logical Volume Manager (LVM) to create logical volumes for use as raw devices by RAC. We recommend that you review the SuSE white paper about Logical Volume Manager, which can be found at:

http://www.suse.de/en/support/oracle/docs/lvm_whitepaper.pdf

1. Launch YaST as root on one node of the cluster by executing the command **yast**.

> **Note:** LVM is not available in YaST2 (the graphical interface for YaST).

2. In the Installation settings menu, select **Configure the Logical Volume Manager** and press Enter (see Figure 6-22):

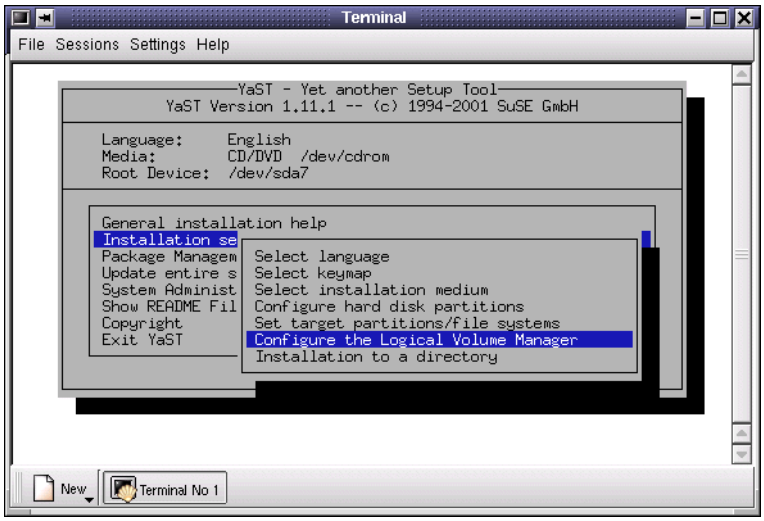

Figure 6-22 Launching LVM

3. You will be asked to confirm that you want to configure LVM. Select **Yes** to start LVM setup.

6.2.1 Creating the Volume Group

We are going to create the Volume Group (representing one or more drives). First we have to partition the drive we have created to make it usable by LVM. We are now viewing the Administration Of LVM Volume Groups window, shown in Figure 6-23:

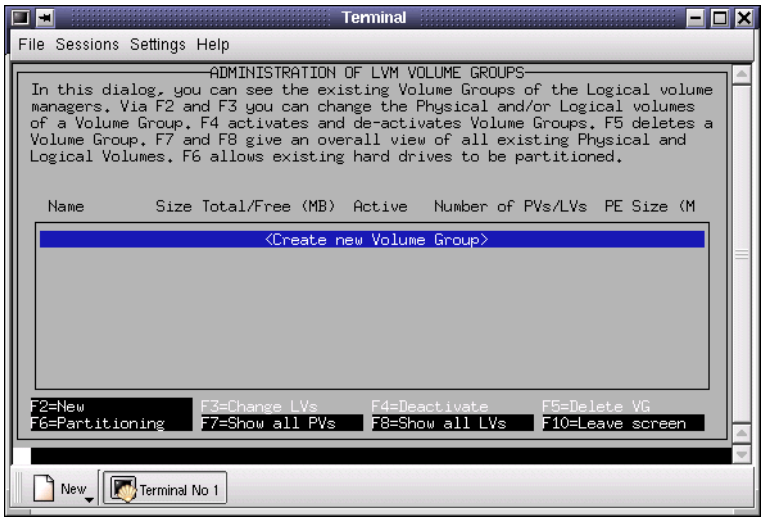

Figure 6-23 LVM administration

1. Press F6 for Partitioning. You will be asked for confirmation. Press Enter to continue.
2. Now select the device to be partitioned. In our case, as we found in step 3 on page 45, it is /dev/sdc (see Figure 6-24):

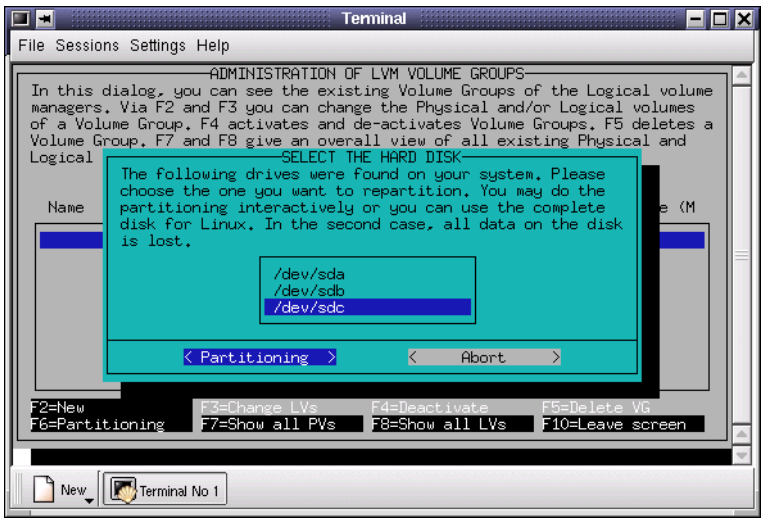

Figure 6-24 Select the hard drive

3. Select **Partitioning**, press Enter.
4. Press F5 to create a new partition.
5. In the Partition Type window (see Figure 6-25), select **Primary partition**, then select **Continue**, and press Enter.

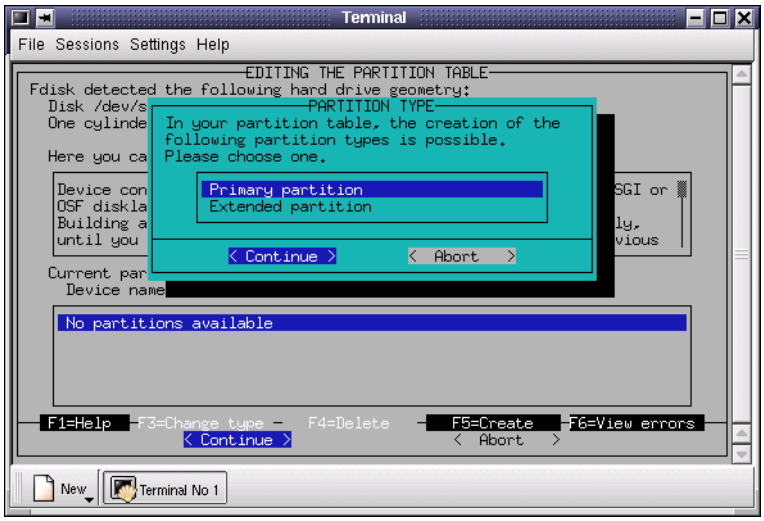

Figure 6-25 Partition type selection

6. In the Primary Partition window (Figure 6-26), select the first line (/dev/sdc1 in our system), then select **Continue**, and press Enter.

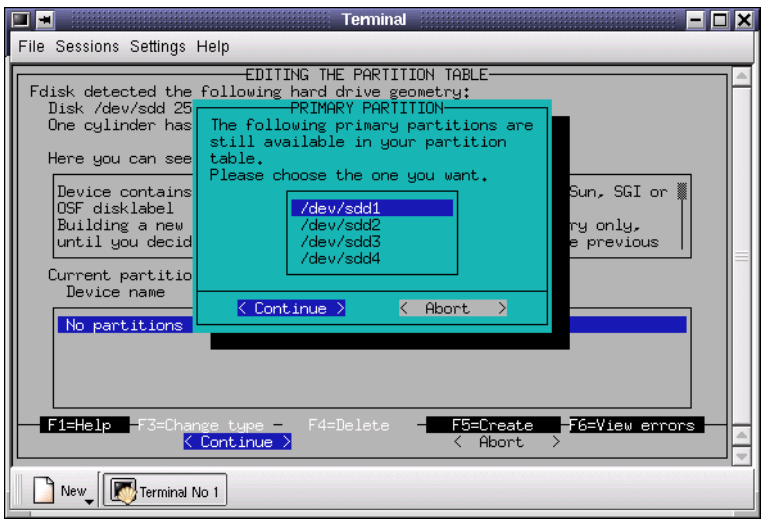

Figure 6-26 Partition number selection

7. In the Location of the Partition window (see Figure 6-27), accept the default entries to create a single partition for the whole disk. Select **Continue**, and press Enter.

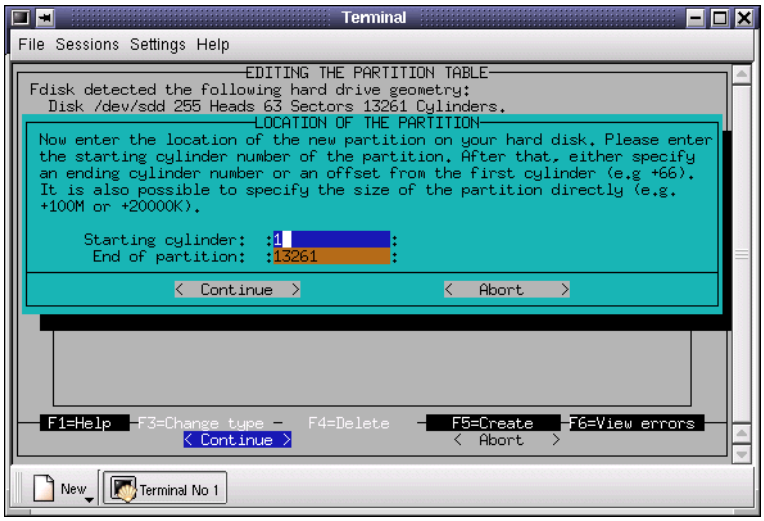

Figure 6-27 Location of the partition

8. The partition is now created. Press F5 to change its type.
9. In the Enter the Partition Type window (Figure 6-28), select **LVM Partition**, then select **Continue**, and press Enter.

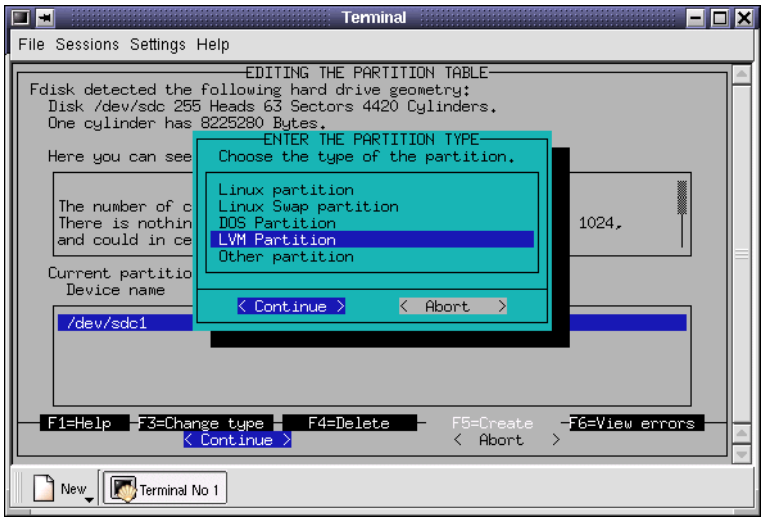

Figure 6-28 Partition type

10. The changes you have made are now reflected in the partition table as shown in Figure 6-29:

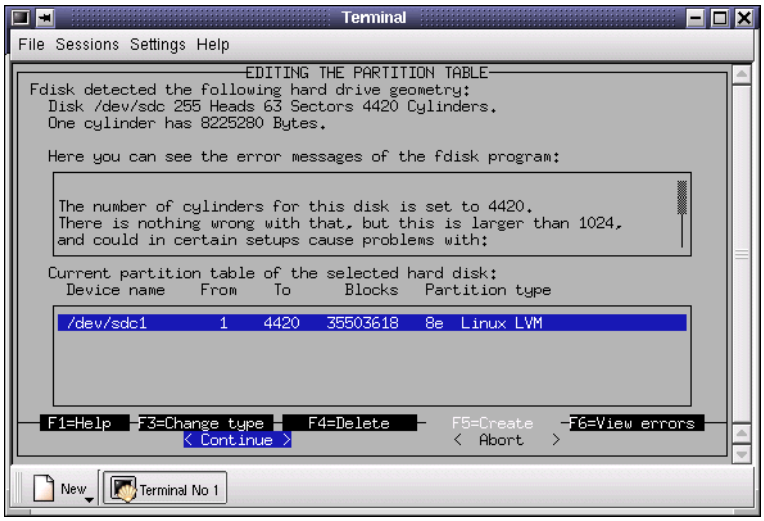

Figure 6-29 Partition table

11. Select **Continue** and press Enter. Press Esc to return to the LVM main window (see Figure 6-30):

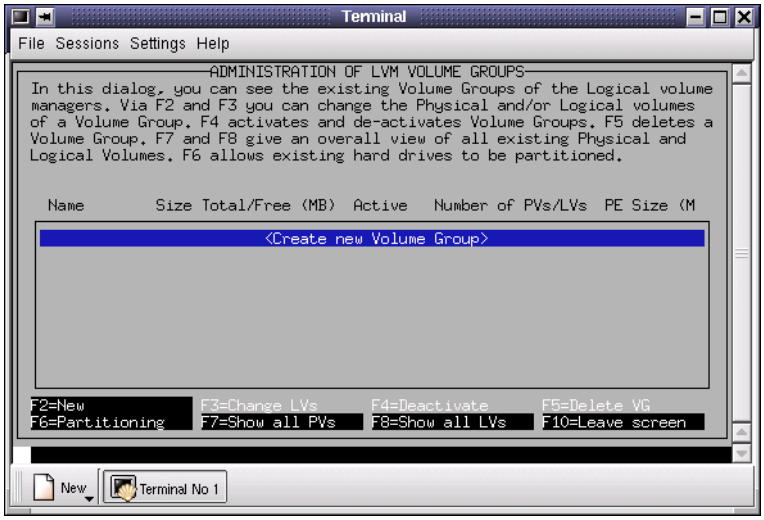

Figure 6-30 LVM main window

12. Press F2 to create a new Volume Group.
13. In the Physical Volume window (Figure 6-31), follow the following steps:
    - In the field **Name of the Volume Group**, enter the volume group name. We used `racvg`.
    - Accept the default, 4, as The size of a Physical extent.
    - Highlight the partition for the device RAC (/dev/sdc1), and press the Spacebar to select it.
    - Select **Create Volume Group**, and press Enter.

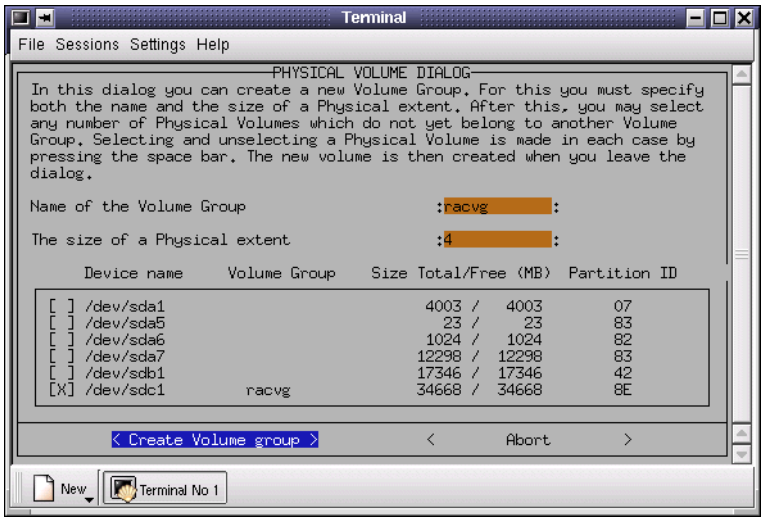

Figure 6-31 Physical Volume window

14. The new Volume Group is displayed in the LVM main window (Figure 6-32):

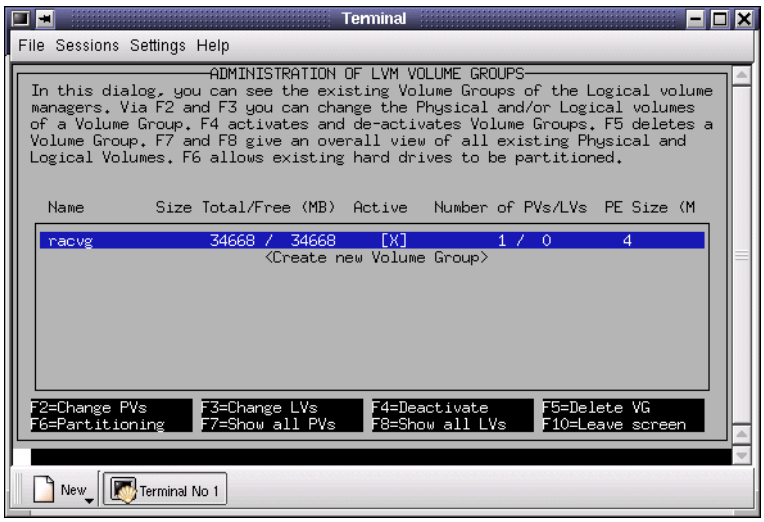

Figure 6-32 LVM main window

6.2.2 Creating the Logical Volumes

The next phase of the process is to create logical volumes within the volume group:

1. Select the **racvg** Volume Group on the LVM window. Press F3 to Change Logical Volumes. The Logical Volume window is displayed (Figure 6-33):

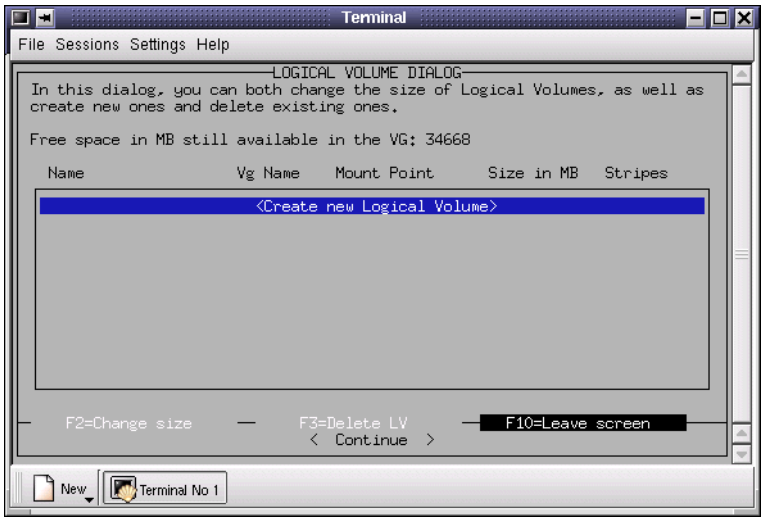

Figure 6-33 Logical Volume window

2. Now we create the logical volume for the Cluster Manager disk. Select **Create new Logical Volume**, and press Enter.
3. In the Create a New Logical Volume window (Figure 6-34), fill in the fields as follows:
   - In the Name of the Logical Volume field, enter the name for the logical volume that will handle the CM disk. Name it `cmlv`.
   - In the Size of the Logical Volume field, enter `100` for the CM disk (this number is in megabytes by default).
   - Accept the default number of stripes (`1`).

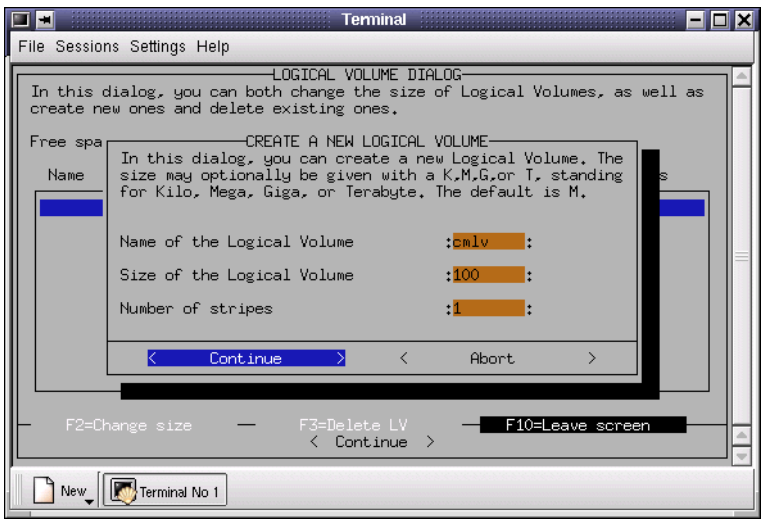

Figure 6-34 Create a new Logical Volume window

4. Select **Continue** and press Enter.
5. Repeat the previous steps to create the logical volume for the RAC configuration:
   - Logical Volume name: `svrcfglv`
   - Logical Volume size: `100`
   - Number of stripes: `1`
6. The Logical Volume window (Figure 6-35) displays the two logical volumes just created:

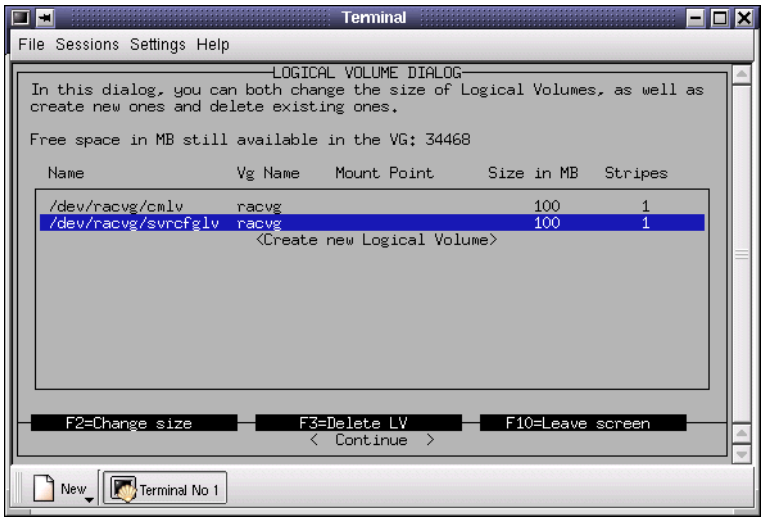

Figure 6-35 Logical Volume window

7. Press F10 to return to the LVM main window. Note that the number of Logical Volumes (LV) for racvg has changed to 2, as shown in Figure 6-36:

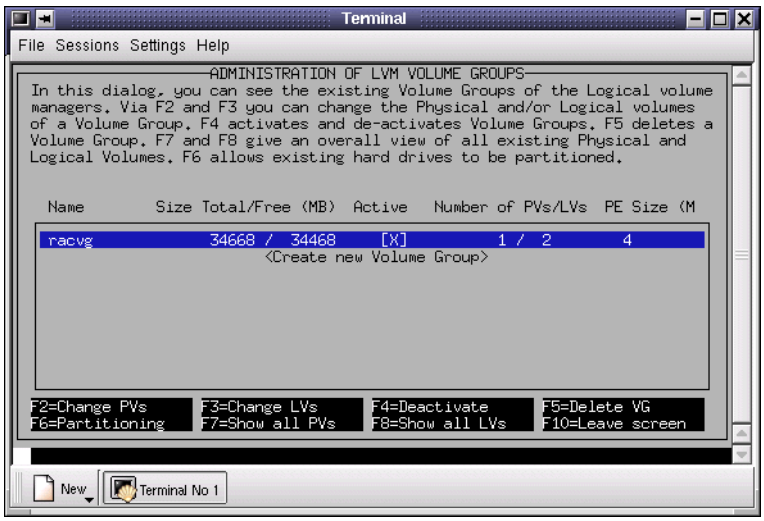

Figure 6-36 LVM main window

8. Press F10 to exit LVM.
9. Press Esc twice to leave YaST.

Note: Check the directory /dev/racvg. It represents the racvg volume group. In this directory, you will find the files cmlv and srvcfg. They represent the two logical volumes we have created.

6.2.3 Activating the LVM configuration on the other nodes

Until now, only the node that run LVM sees the LVM configuration. The LVM configuration must be seen by all nodes. To achieve this, follow the steps below on each of the remaining nodes in the cluster:

1. Reload the qla2x00 module
2. Execute (as root) the command: **/sbin/vgscan**
3. Execute (as root) the command: **/sbin/vgchange -a y racvg**

> **Note:** When you change a volume group definition (racvg for example), you have to deactivate it on each node, reload the module, discover it, and reactivate it, with the following commands
>
> ```
> vgscan -a n racvg
> rmmod qla2x00
> insmod qla2x00
> /sbin/vgscan
> /sbin/vgchange -a y racvg
> ```

Figure 6-37 shows the execution of these commands on one node:

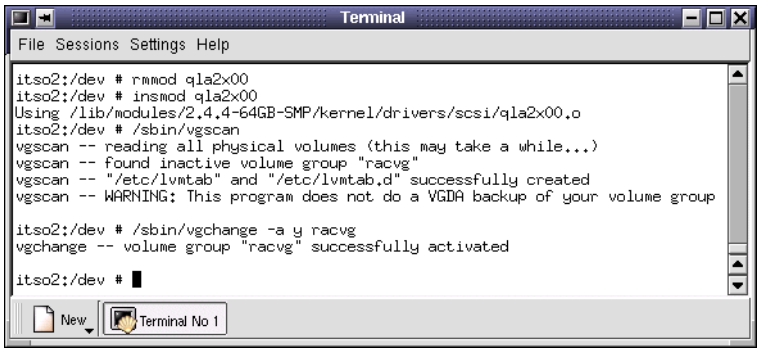

Figure 6-37 LVM configuration activation

6.3 RAC raw devices creation

In order to make the servers use the raw device to store the database files, these steps have to be performed on each node:

1. Create the script raw_devices in /etc/rc.d as shown in Example 6-1. This script maps the raw devices to the logical volumes we created earlier. The script has to be owned by root, and have permissions set to 744.

Example 6-1 /etc/rc.d/raw_devices

```
#! /bin/sh
#
echo "Raw device mapping"
#  racvg
raw /dev/raw1  /dev/racvg/cmlv
raw /dev/raw2 /dev/racvg/svrcfglv
```

2. We want the raw devices to be mapped at server startup, so we configure the system to execute the script at Runlevel 5. To do so, create a symbolic link to the script file in

/etc/rc.d/rc5.d. We made it the last script to be executed at start up by placing it at the 25th position in the start up sequence. Executing the following commands as root achieves this:

```
cd /etc/rc.d/rc5.d
ln -s ../raw_devices S25raw_devices
```

3. The raw devices have to be accessed by Oracle, so set permissions so that all of the raw devices are owned by root and the group oinstall, and then give them both read/write access. Executing the following commands as root achieves this:

```
chown root:oinstall /dev/raw*
chmod 660 /dev/raw*
```

4. Execute the /etc/rc.d/raw_devices script as root. The output from this command is shown in Figure 6-38:

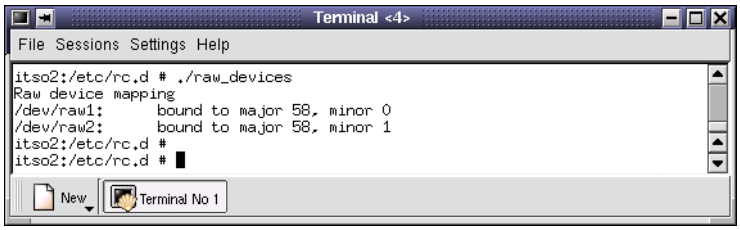

Figure 6-38 /etc/rc.d/raw_devices execution

6.4 Storage subsystem summary

There are several different layers of abstraction in the way that the storage subsystem has been configured. At the lowest level, we have physical drives, then move up through RAID arrays and logical drives, to LVM partitions, volume groups and logical volumes, and culminate in the raw devices that are used by Oracle9*i*.

To clarify the way all of these layers work together, we have included a diagram illustrating the relationships between them, including the numbers of entities involved, such as one-to-one, many (n)-to-one, and so on. This is shown in Figure 6-39:

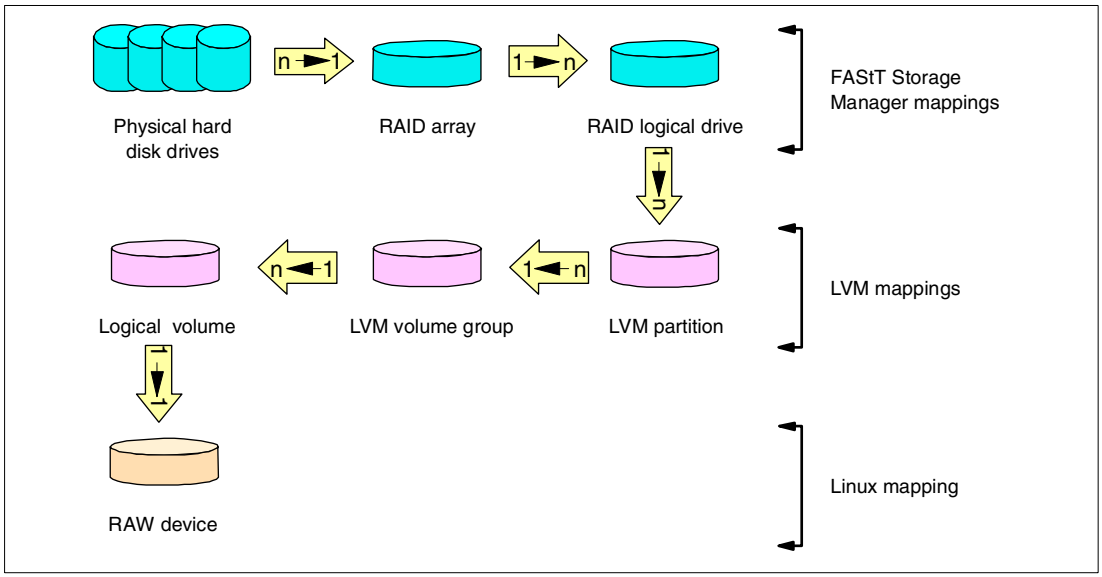

Figure 6-39 Storage subsystem relationships

The arrows in Figure 6-39 indicate the mappings that you can make between the different layers of the disk subsystem. For example, the arrow between RAID logical drive and LVM partition contains 1->n. This means that a single RAID logical drive can contain multiple LVM partitions, but you may choose to create only one if you wish.

7

Oracle9*i* RAC Installation

This section describes the procedure for installing Oracle9*i* RAC. It guides you through the pre-installation tasks, installation of Oracle9*i* using the Universal Installer, and the post-installation tasks necessary to complete the installation.

7.1 Prepare for Oracle9*i* installation

We started our installation by copying the Oracle9*i* product CDs to a system hard disk. This is not necessary, as you may install the product by inserting the CDs when the Oracle9*i* Universal Installer directs you to do so. Having the product CD data available on a hard disk is convenient, however, especially if installation has to be repeated for any reason.

In our example, the three product CDs were copied to three directories called /Redpaper/9icode/disk1, .../disk2 and .../disk3 respectively.

The pre-installation tasks described below have to be performed on both nodes. You must perform these tasks as root user.

1. First, we modify kernel parameters that affect resources used by Oracle9*i*.
   a. Review the kernel parameter recommendations in *Oracle9i Quick Installation Procedure, Release 1 (9.0.1) for Linux,* A90352-01. This document, *Oracle9i Database Quick Installation Procedure Release 9.0.1 for Linux*, and *Oracle9i Database Installation Checklist Release 9.0.1 for Linux* may be downloaded from:

      `http://otn.oracle.com/docs/products/oracle9i/content.html`

      Registration is required for access to this site.

   b. In our example, we increased the values of kernel parameters `SEMOPM` and `SHMMAX` to the minimum recommended. Example 7-1 shows a shell script, which we called rac, that contains the commands used to modify the kernel parameters.

Example 7-1 Kernel parameter modification script

```
# modify kernel parameters
echo 250 32000 100 128 > /proc/sys/kernel/sem
echo 4294967295 > /proc/sys/kernel/shmmax
```

The first command writes the settings for kernel semaphore parameters SEMMSL, SEMMNS, SEMOPM, and SEMMNI to the /proc/sys/kernel/sem file. The values for SEMMSL, SEMMNS and SEMMNI were the original values set by the Linux installation.

The second command modifies SHMMAX to reflect our system's physical memory of 8 GB. The recommended setting is one half of the physical memory.

c. These parameters need to be initialized during system startup, so we store the rac script in /etc/rc.d.

d. Add execute permission to this script with the command:

```
chmod u+x /etc/rc.d/rac
```

e. Now set it up to run at runlevel 5 with the commands:

```
itso1:~ # cd /etc/rc.d/rc5.d
itso1:/etc/rc.d/rc5.d # ln -s ../rac S26rac
```

f. Execute /etc/rc.d/rac to modify the current kernel settings.

2. As mentioned previously, the oracle user ID is defined in the basic installation of SuSE Linux. Also included are the oinstall and dba groups. In addition, we created the oper group (as recommended in *Oracle9i Database Quick Installation Procedure Release 9.0.1 for Linux*), with oracle as the only member of the group.

3. Create the mount point for the Oracle9*i* product. In our example, the mount point was /oracle/product/9.0.1. We changed ownership and access permission as follows:

```
chown -R oracle:oinstall /oracle
chmod -R ug=rwx,o=rx /oracle
```

4. Finally, a number of environmental variables have to be set up for the oracle user ID. Modify .bash_profile in oracle's home directory, adding the statements shown in Example 7-2.

Example 7-2 Setting environmental variables in .bash_profile

```
export DISPLAY=localhost:0.0

export TMPDIR=/tmp

export ORACLE_BASE=/oracle
export ORACLE_HOME=/oracle/product/9.0.1
export ORACLE_TERM=xterm
export TNS_ADMIN=/oracle/product/9.0.1/network/admin
export NLS_LANG=AMERICAN_AMERICA.US7ASCII
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/openwin/lib:/usr/local/lib
export THREADS_FLAG=native
export PATH=$ORACLE_HOME/bin:$PATH

export SRVM_SHARED_CONFIG=/dev/raw2
```

5. Verify that the environmental variables are set correctly by logging in using the oracle user ID and echoing a variable to the console. Here is an example:

```
oracle@itso1:~ > echo $ORACLE_BASE
/oracle
oracle@itso1:~ >
```

7.2 Run the Oracle9*i* Universal Installer

1. Log in as `oracle` on itso1 and invoke the Oracle9*i* Universal Installer as follows:

   ```
   cd /Redpaper/9icode/disk1
   runInstaller
   ```

   The Welcome window is displayed as shown in Figure 7-1.

Figure 7-1 Oracle9i Universal Installer Welcome window

2. Click **Next** to display the Inventory Location window shown in Figure 7-2:

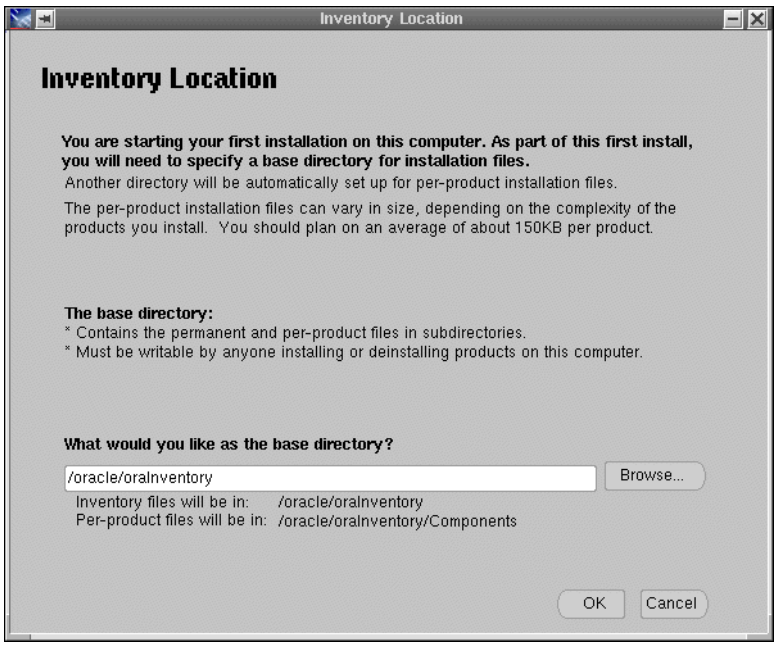

Figure 7-2 Inventory Location window

3. Verify the base directory is correct and click **OK** to display the UNIX Group Name window. Enter `oinstall` for the UNIX Group Name as shown in Figure 7-3, and click **Next** to continue.

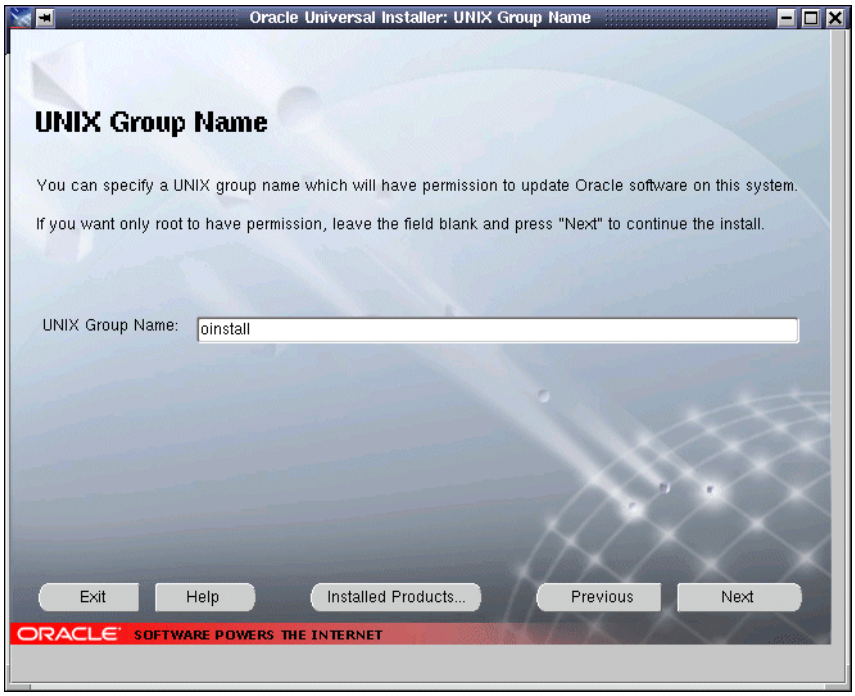

Figure 7-3 UNIX Group Name window

4. A pop-up window (shown in Figure 7-4) appears, which asks you to execute **`/tmp/orainstRoot.sh`** as root.

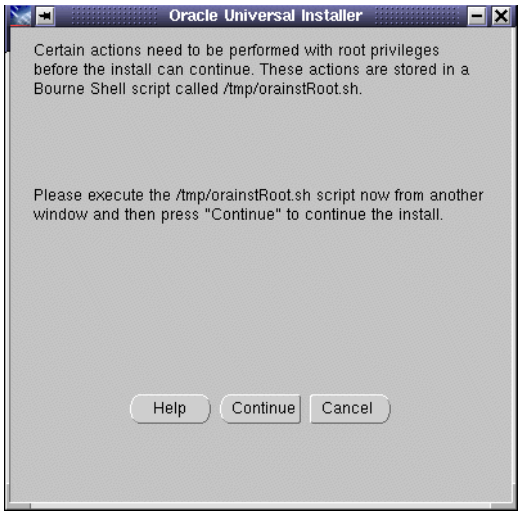

Figure 7-4 Pop-up window - Execute orainstRoot.sh

5. Open a new terminal window, switch to root user and execute the script as directed. See Figure 7-5 for the result of running this command:

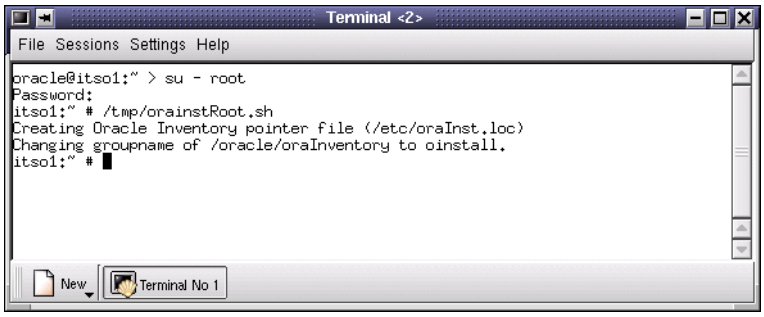

Figure 7-5 Execution of orainstRoot.sh

6. Leave this terminal window open at the root user prompt, because you will need it later in this procedure. Click **Continue** in the pop-up window in Figure 7-4 to display the File Locations window (see Figure 7-6):

Figure 7-6 File Locations window

7. Change or verify the fields as shown and click **Next** to display the Available Products window as shown in Figure 7-7:

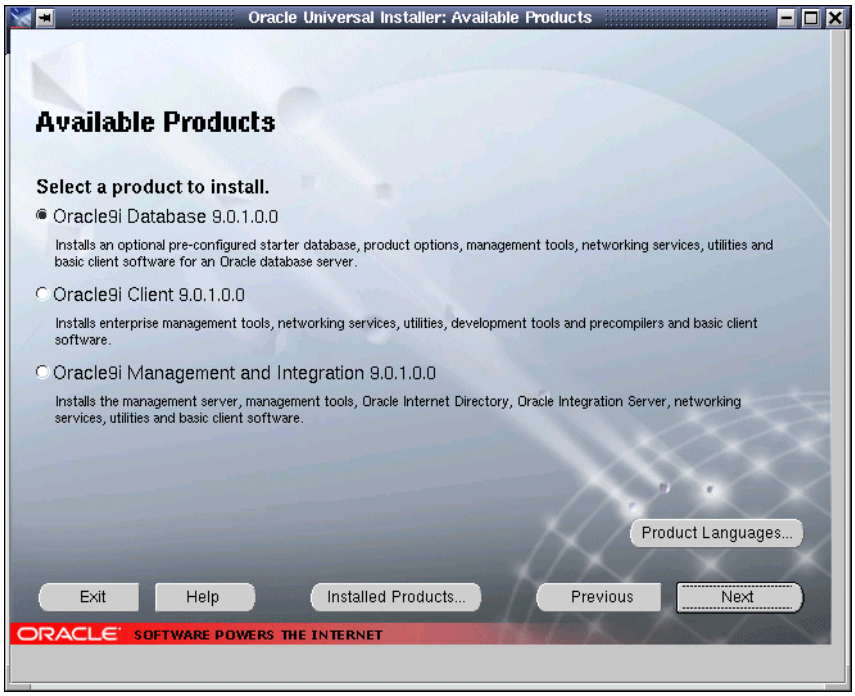

Figure 7-7 Available Products window

8. Verify that **Oracle9i Database 9.0.1.0.0** is selected and click **Next** to display the Installation Types window (Figure 7-8):

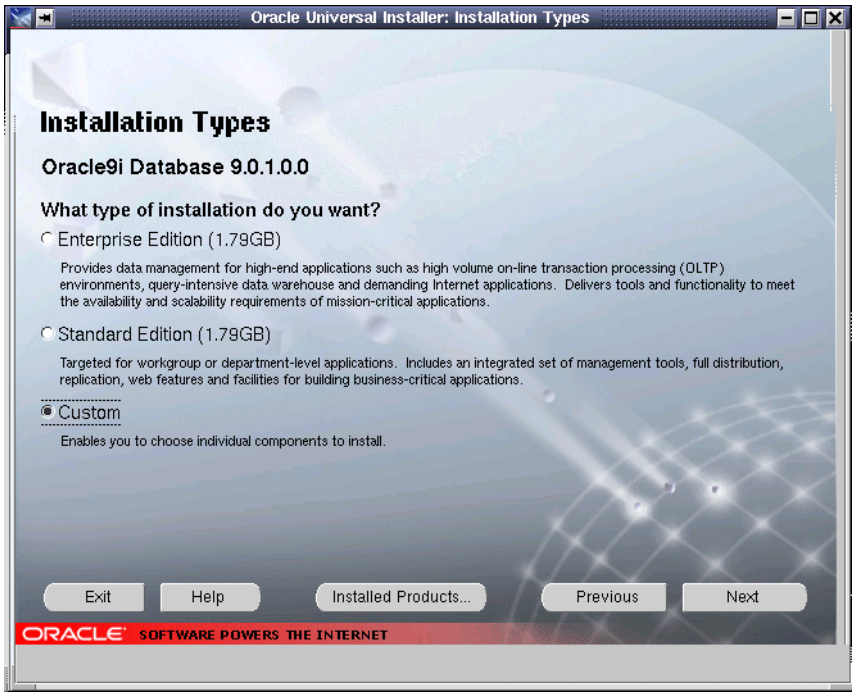

Figure 7-8 Installation Types window

9. Select **Custom** and click **Next** to display the Available Product Components window as shown in Figure 7-9:

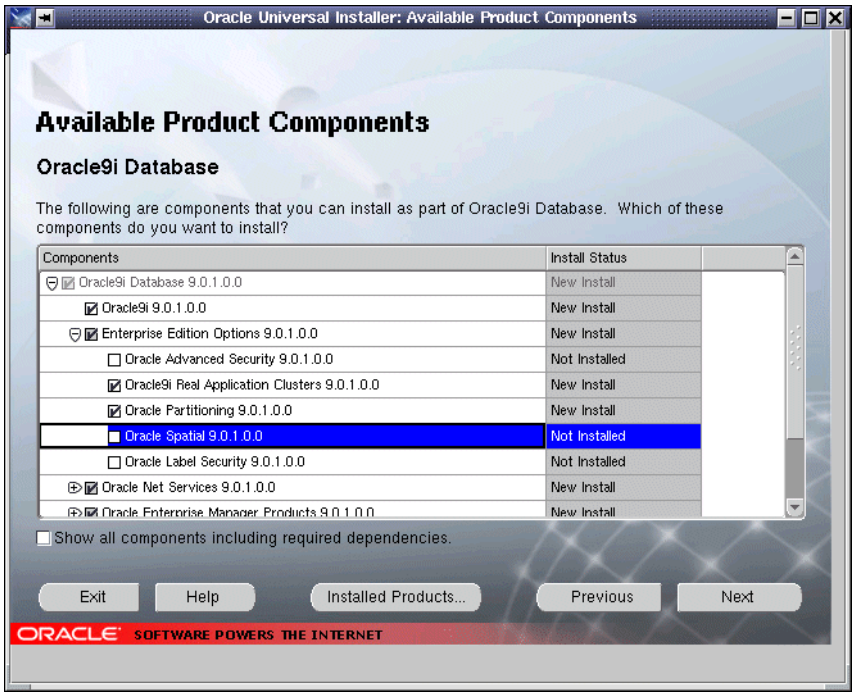

Figure 7-9 Available Product Components window

10. Select and deselect items in the list accordingly to leave only the following products selected:
    - Oracle 9.0.1.0.0
    - Enterprise Edition Options
      - Oracle Real Application Clusters 9.0.1.0.0
      - Oracle Partitioning 9.0.1.0.0
    - Oracle Net Services 9.0.1.0.0
      - Oracle Net Listener
    - Oracle Enterprise Manager Products 9.0.1.0.0
      - Oracle Intelligent Agent 9.0.1.0.0
11. When you have finished selecting the product components, click **Next** to display the Component Locations window (Figure 7-10):

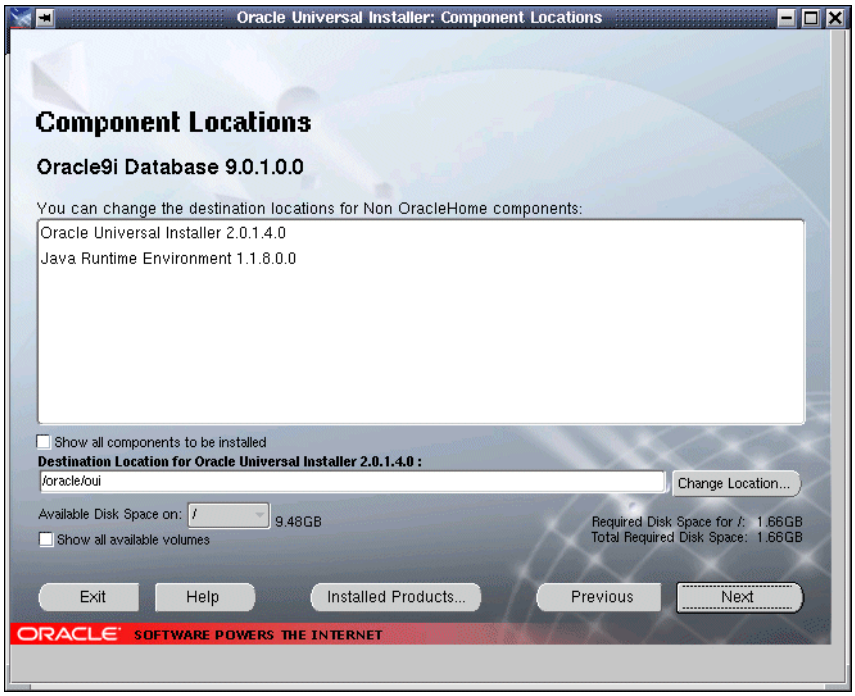

Figure 7-10 Component Locations window

12. Accept the defaults and click **Next**. The Cluster Nodes Selection window is displayed (Figure 7-11):

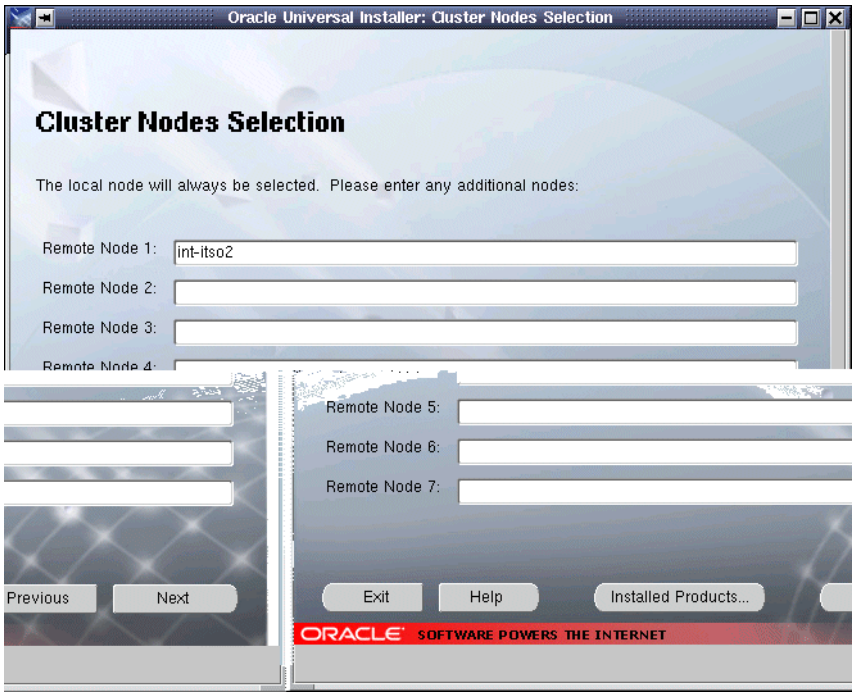

Figure 7-11 Cluster Nodes Selection window

13. Enter `int-itso2` in the Remote Node 1 field to use the private Gigabit LAN (this name was defined in 4.1.1, “Define network interfaces” on page 17). Click **Next** to display the Privileged Operating System Groups window (Figure 7-12):

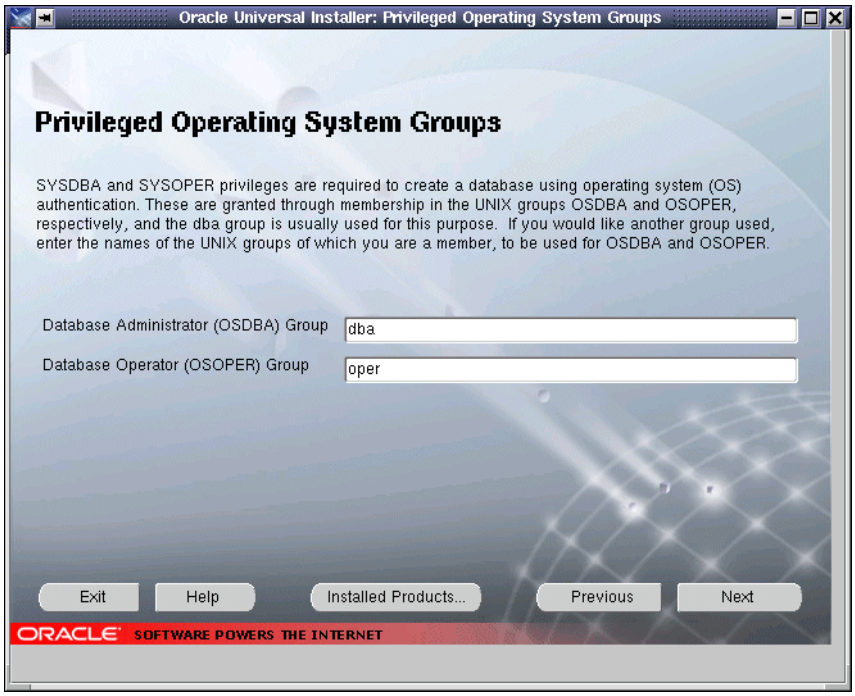

Figure 7-12 Privileged Operating System Groups window

14. Change Database Operator (OSOPER) Group to `oper` as shown. Click **Next** to continue and display the Summary window (Figure 7-13):

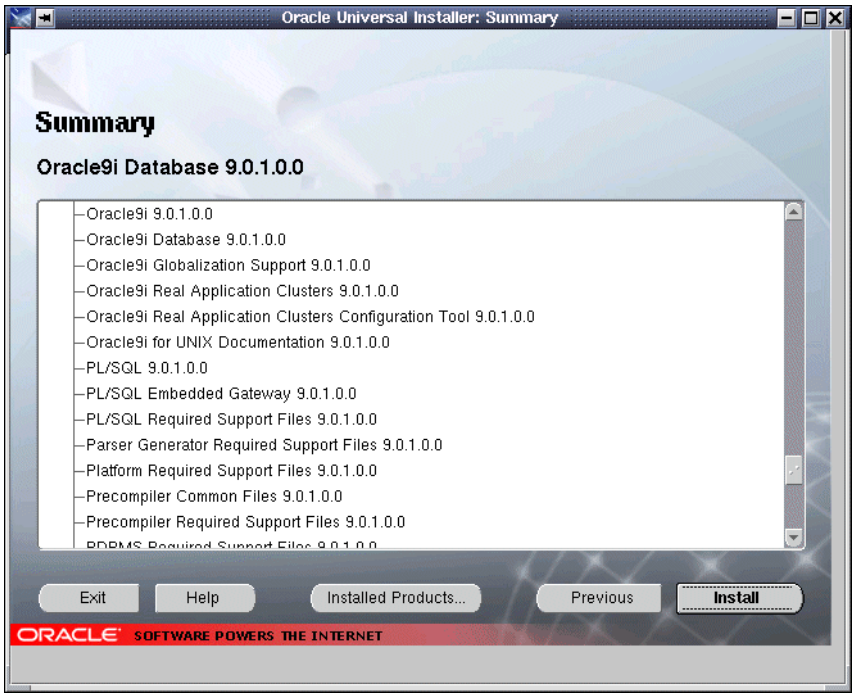

Figure 7-13 Summary window

15. Scroll down the list and verify that Oracle9i Real Application Clusters appears. Click **Next** to start the actual installation. A window entitled Install is displayed, showing progress of the installation (Figure 7-14):

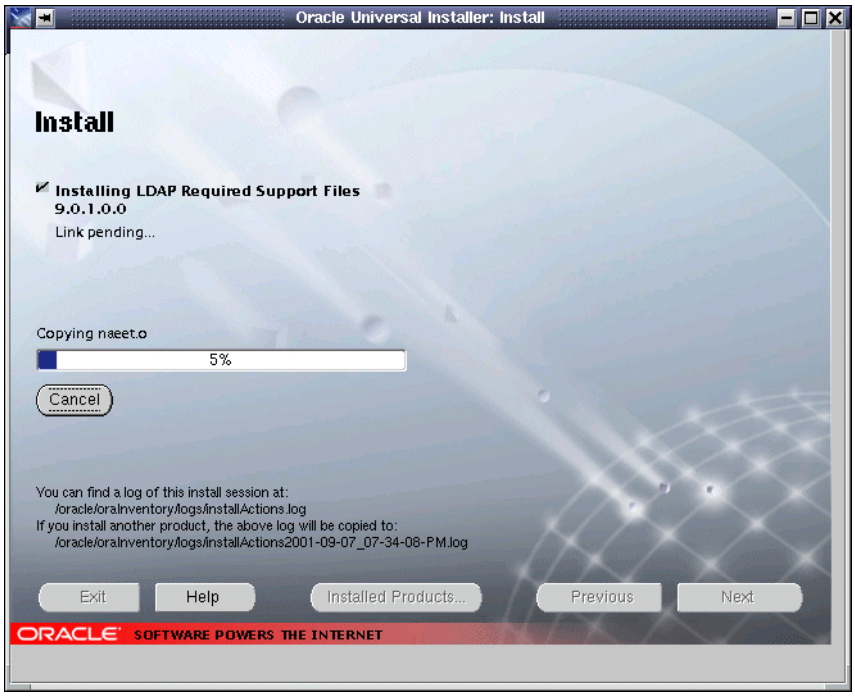

Figure 7-14 Install status window

> **Note:** The Oracle9*i* Universal Installer creates a log of the install session at /oracle/oraInventory/logs/installActions.log. Installation progress may be monitored in greater detail by examining this file.

16. After a few minutes, a pop-up window entitled Disk Location displays (Figure 7-15):

Figure 7-15 Disk Location prompt for disk2

17. Modify the path to indicate where the second CD's contents can be found (if you used a scheme similar to ours, change `disk1` to `disk2`) and click **OK**. Installation continues.
18. After another few minutes, another Disk Location pop-up window is displayed to ask for the location of CD 3's contents. Modify the path as necessary (by changing `disk2` to `disk3`, for example) and click **Next** to continue the installation.
19. During this last phase of Oracle9*i* installation, a pop-up window entitled Setup Privileges is displayed, asking you to execute **`root.sh`** (Figure 7-16):

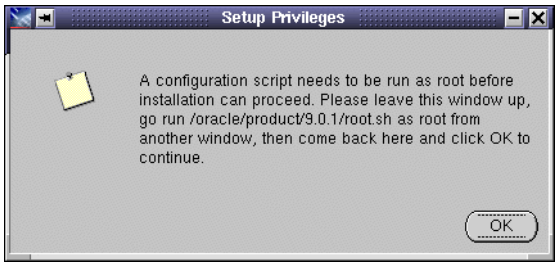

Figure 7-16 Setup Privileges window

20. Before executing **root.sh**, create the directory /var/opt/oracle and make `oracle` its owner on *both* nodes by executing these commands from the root shell prompt:

```
mkdir /var/opt/oracle
chown oracle:oinstall /var/opt/oracle.
```

21. From the root command prompt, execute **/oracle/product/9.0.1/root.sh.** The results of this are shown in Figure 7-17:

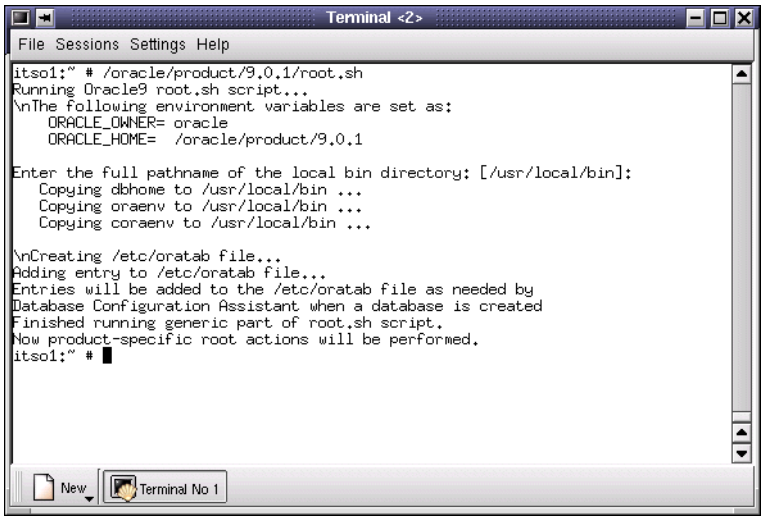

Figure 7-17 Execution of root.sh script

22. Click **OK** in the Setup Privileges window (Figure 7-16) to display the End of Installation window shown in Figure 7-18:

Figure 7-18 End of Installation window

23. You may browse the list of installed products by clicking **Installed Products**. Click **Exit** to leave the Oracle9*i* Universal Installer.

The Universal Installer installs Oracle9*i* on all members of the cluster in this one process. For our example, we verified that the installation had taken place correctly on the other node by logging in to itso2 and checking to see that the contents of /oracle/product/9.0.1 were the same as on itso1.

7.3 Post-installation

Before creating our database, we need to set up Oracle Cluster Management Software (OCMS) and the Global Services Daemon (GSD). OCMS is part of Oracle9i Enterprise Edition for Linux. It manages cluster membership, maintains a global view of clusters, monitors cluster nodes, and performs cluster reconfiguration. OCMS is installed automatically when RAC is selected and consists of the following components:

- Watchdog Daemon (watchdogd)
- Node Monitor (NM)
- Cluster Manager (CM)

The Global Services Daemon (GSD) is a background process that receives and carries out requests to perform system management tasks, such as startup or shutdown, from the **`srvctl`** utility. Running in each cluster node, GSD executes requests from **`srvctl`** locally, and sends the results back to **`srvctl`**.

For further information on OCMS, refer to the *Oracle9i Administrator's Reference, Release 1 (9.0.1) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel and Sun SPARC Solaris*.

For further information on GSD, refer to *Oracle9i Real Application Clusters, Installation and Configuration, Release 1 (9.0.1)*.

The following tasks must be performed to set up Cluster Manager (CM), Node Monitor (NM) and Global Services Daemon (GSD):

1. The Watchdog Daemon uses the Linux watchdog device to provide timer services to Node Monitor and Cluster Manager. Oracle RAC requires that the flag `CONFIG_WATCHDOG_NOWAYOUT=Y` be passed to the watchdog device. This is done by editing /etc/lilo.conf. On each node, add the statement shown in bold type in Example 7-3.

Example 7-3 Changes to /etc/lilo.conf

```
boot    = /dev/sda
vga     = 771
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32
prompt
timeout = 80
message = /boot/message

  image  = /boot/vmlinuz
  label  = linux
  root   = /dev/sda7
  initrd = /boot/initrd
  append = "CONFIG_WATCHDOG_NOWAYOUT=Y"

  image  = /boot/vmlinuz.suse
  label  = suse
  root   = /dev/sda7
  initrd = /boot/initrd.suse
  optional

  other  = /dev/sda1
  label  = nt
```

2. From the root shell prompt, enter **`lilo`** to install the boot loader and reboot the node.
3. Create the configuration file $ORACLE_HOME/oracm/admin/nmcfg.ora for the NM on each node. This configuration file is shown in Example 7-4 for node itso1. The file for itso2 is the same, except for the last line which should be:

   `CmHostName=int-itso2`

   The `DefinedNodes` parameter is the list of nodes in the cluster. `CmDiskFiles` is the raw device for the Cluster Manager disk, and `CmHostName` is the host name of the local node.

Example 7-4 $ORACLE_HOME/oracm/admin/nmcfg.ora on itso1

```
DefinedNodes=int-itso1 int-itso2
CmDiskFiles=/dev/raw1
CmHostName=int-itso1
```

4. Modify $ORACLE_HOME/oracm/admin/ocmargs.ora, adding the parameters shown in boldface in Example 7-5. This prevents NM from sending a message to watchdogd to reboot, in case of failure.

Example 7-5 $ORACLE_HOME/oracm/admin/ocmargs.ora

```
# Sample configuration file $ORACLE_HOME/oracm/admin/ocmargs.ora
watchdogd -g dba -l 0 -d /dev/null
oranm
oracm /a:0
norestart 1800
```

Note: Prior to production, do not forget to edit ocmargs.ora to remove the statement that was added in Example 7-5 (shown in boldface). This will take effect after NM is restarted.

5. Start `watchdogd`, `oranm`, and `oracm` by executing the command:

   `$ORACLE_HOME/oracm/bin/ocmstart.sh`

   from the root shell prompt.

   Note: You need the oracle user environment for this command, but you have to be root to run it. A good solution is to log on as oracle (command: **`su - oracle`**) and then log in as root without the minus (-) option (command: **`su root`**).

6. Verify that Oracle Cluster Manager is running. Do this with the **`ps -ef`** command as shown in Example 7-6.

Example 7-6 Checking that Cluster Manager is running

```
itso1:~ # ps -ef |egrep "watchdog|oracm|oranm"
root       799     1  0 19:22 ?        00:00:04 watchdogd -g dba -l 0 -d /dev/nu
ll
root       801     1  0 19:22 ?        00:00:00 oranm
root       802     1  0 19:22 ?        00:00:00 oracm /a:0
root       804   801  0 19:22 ?        00:00:00 oranm
root       805   804  0 19:22 ?        00:00:01 oranm
root       807   804  0 19:22 ?        00:00:00 oranm
root       808   802  0 19:22 ?        00:00:00 oracm /a:0
root       809   808  0 19:22 ?        00:00:00 oracm /a:0
root       810   804  0 19:22 ?        00:00:00 oranm
root       811   804  0 19:22 ?        00:00:00 oranm
root       812   804  0 19:22 ?        00:00:01 oranm
root       813   808  0 19:22 ?        00:00:00 oracm /a:0
root       824   808  0 19:22 ?        00:00:00 oracm /a:0
root       847   808  0 19:22 ?        00:00:00 oracm /a:0
root      1246   808  0 19:31 ?        00:00:00 oracm /a:0
root      1269   808  0 19:31 ?        00:00:02 oracm /a:0
root      1270   808  0 19:31 ?        00:00:00 oracm /a:0
root      1271   808  0 19:31 ?        00:00:00 oracm /a:0
root      1272   808  0 19:31 ?        00:00:00 oracm /a:0
root      1273   808  0 19:31 ?        00:00:00 oracm /a:0
root      1274   808  0 19:31 ?        00:00:00 oracm /a:0
root      1275   808  0 19:31 ?        00:00:00 oracm /a:0
root      1278   808  0 19:31 ?        00:00:00 oracm /a:0
root      1997  1963  0 21:57 pts/2    00:00:00 egrep watchdog|oracm|oranm
itso1:~ #
```

Note: If the Cluster Manager does not start:

- Check the log files $ORACLE_HOME/oracm/log/oracm.log and $ORACLE_HOME/oracm/log/oranm.log
- Check that /etc/hosts is correct and that all nodes are referenced
- Check the volume group, logical volumes are active
- Check the permissions on the raw devices
- Check the file $ORACLE_HOME/oracm/admin/nmcfg.ora is correct
- Remove the file $ORACLE_HOME/oracm/log/ocmstart.ts

7. Since this is the first Oracle9*i* database on this cluster, you must initialize the cluster-wide SRVM configuration. This is done with the following command:

   ```
   srvconfig -init
   ```

8. Ensure that GSD, the Global Services Daemon is *not* running. Do this by issuing the command:

   ```
   ps -ef |grep OPSMDaemon |grep -v grep
   ```

   If GSD is *not* running, no processes should be listed by this command.

9. On both nodes, verify that /var/opt/oracle/srvConfig.loc exists and that its contents match the oracle user's environmental variable `SRVM_SHARED_CONFIG`.

10. Verify that the listener nodes are running. Execute:

    ```
    lsnodes
    ```

    at the oracle user's shell prompt. The expected results are shown in Example 7-7.

Example 7-7 Checking listener nodes

```
oracle@itso1:~ > lsnodes
int-itso1
int-itso2
oracle@itso1:~ >
```

11. Test the watchdog device by executing **$ORACLE_HOME/oracm/bin/wdttest1** and **$ORACLE_HOME/oracm/bin/wdttest2** from the oracle shell prompt. First execute **wdttest1** as shown in Example 7-8:

Example 7-8 Running wdttest1

```
# oracle@itso1:~ > su root
Password:
root@itso1:/opt/oracle > cd $ORACLE_HOME/oracm/bin
root@itso1:/oracle/product/9.0.1/oracm/bin > wdttest1
```

After 60 seconds, the system will reboot. Log in as root and start watchdogd, oranm and oracm as described in step 5 on page 70.

Now run **wdttest2** instead of **wdttest1**. After sixty seconds, the system will reboot.

If the system does not reboot, then:

- Check permissions on `/dev/watchdog`. The owner must be root:root and its permission settings must be rw-rw-rw. If not, change them.
- Make sure that the watchdog daemon is running. Do this with the command:

  ```
  ps -ef |grep watchdogd |grep -v grep
  ```

  If it is not running, start it as shown in step 5 on page 70.
- Check to see that the softdog kernel module is loaded. Execute:

```
lsmod
```

and examine the output for an entry for softdog. If it does not appear, install it by executing:

```
insmod softdog
```

and add the line `/sbin/insmod softdog soft_margin=60 > /dev/null` to `/etc/rc.d/rac`.

12. To test access to the raw devices, use **$ORACLE_HOME/oracm/bin/rdevtest**. From the oracle shell:

    a. Create the file rdevtest.conf, containing the lines shown in Example 7-9.

Example 7-9 rdevtest.conf file

```
int-itso1:/dev/raw1
int-itso2:/dev/raw1
int-itso1:/dev/raw2
int-itso2:/dev/raw2
```

    b. Run **$ORACLE_HOME/oracm/bin/rdevtest**, with `rdevtest.conf` as an argument. If the raw devices are accessible, the result is shown in Example 7-10.

Example 7-10 Running rdevtest

```
oracle@itso1:/oracle/product/9.0.1/oracm/bin > rdevtest rdevtest.conf
Ok
```

> **Note:** **rdevtest** does not work properly when you have more than four lines in `rdevtest.conf`.

If **rdevtest** returns the message `Cannot read /dev/raw_ from node`, do the following:

- Check permissions on the raw devices (/dev/raw1, /dev/raw2, and so on). Refer to 6.3, “RAC raw devices creation” on page 54 for further information.
- Check the volume group and logical volumes are active.
- Check `/etc/hosts` is correct and that all nodes are referenced.
- Check Cluster Manager is running.

13. Run **clustca** on one node. Type the command:

```
clustca -nodeinfo int-itso1,0 int-itso2,1
```

If this fails, check that the environmental variable `THREADS_FLAG` is set to `native` with the command:

```
echo $THREADS_FLAG
```

If not, make sure that it is being set in `.bash_profile`. If appropriate, execute **.bash_profile** in the current shell by typing:

```
. ~/.bash_profile
```

If you encounter a raw device access error:

- Check the file /var/opt/oracle/srvConfig.loc and make sure it matches the environmental variable `SRVM_SHARED_CONFIG`
- Check permissions on the raw devices (/dev/raw1, /dev/raw2, and so on).
- Check the volume group and logical volumes are active

14. On both nodes, check to see if the file $ORACLE_HOME/srvm/config/srvConfig.loc exists. If it does, delete it. Create a symbolic link from /var/opt/oracle/srvConfig.loc to $ORACLE_HOME/srvm/config/srvConfig.loc by executing:

```
ln -s /var/opt/oracle/srvConfig.loc $ORACLE_HOME/srvm/config/srvConfig.loc
```

15. Check to see that GSD is started on all nodes. Execute:

    ps -ef |grep OPSMDaemon |grep -v grep

    If you do not see output referring to OPSMDaemon, start GSD by executing (as oracle):

    gsd

16. Create a script that will be used to start CM, NM and GSD at server startup. We used the script named /etc/rc.d/rac created in 7.1, “Prepare for Oracle9i installation” on page 57, step 1c.

    The modified version is shown in Example 7-11. Note that we have enhanced /etc/rc.d/rac to handle stopping, restarting and status queries.

Example 7-11 Startup script - /etc/rc.d/rac

```
#! /bin/sh
#
. /opt/oracle/.bash_profile

case "$1" in
    start)
#       modify kernel parameters
       echo 250 32000 100 128 > /proc/sys/kernel/sem
       echo 4294967295 > /proc/sys/kernel/shmmax

       echo "Starting watchdog timer device"
       /sbin/insmod softdog soft_margin=60 > /dev/null

       echo "Starting Oracle Cluster Manager"
       $ORACLE_HOME/oracm/bin/ocmstart.sh > /dev/null

       sleep 15

       echo "Starting Global Services Daemon"
       su - oracle -c "gsd"
          ;;
    stop)
       echo "Stopping Global Services Daemon"
       kill `ps -ef |grep OPSMDaemon |grep -v grep |awk ' { print $2 } '`
       echo "Stopping Oracle Cluster Manager"
       killall -q oranm
       killall -q oracm
       echo "Stopping watchdog timer device"
       killall -q watchdogd
          ;;
    try-restart)
          ;;
    restart)
       $0 stop
       rm $ORACLE_HOME/oracm/log/ocmstart.ts
       $0 start
          ;;
    force-reload)
       $0 stop
       rm $ORACLE_HOME/oracm/log/ocmstart.ts
       $0 start
          ;;
    reload)
       . /opt/oracle/.bash_profile
       $0 stop
       rm $ORACLE_HOME/oracm/log/ocmstart.ts
```

```
      $0 start
         ;;
    status)
      ps -ef |grep watchdogd | grep -v grep  > /dev/null
      if test $? = 0; then echo "Watchdog is running"; else echo "Watchdog is not
      running"; fi
      ps -ef |grep oranm | grep -v grep  > /dev/null
      if test $? = 0; then echo "Oracle Node Monitor is running"; else echo "Oracle Node
      Monitor is not running"; fi
      ps -ef |grep oracm | grep -v grep > /dev/null
      if test $? = 0; then echo "Oracle Cluster Manager is running"; else echo "Oracle
      Cluster Manager is not running"; fi
      ps -ef |grep OPSMDaemon | grep -v grep > /dev/null
      if test $? = 0; then echo "Oracle Global Services Daemon is running"; else echo
      "Oracle Global Services Daemon is not running"; fi
         ;;
    probe)
         ;;
    *)
      echo "Usage: $0 {start|stop|status|try-restart|restart|force-reload|reload|probe}"
      exit 1
         ;;
esac
```

7.4 Oracle9*i* Patch Set 1

While writing this paper, Oracle9*i* Data Server 9.0.1 Patch Set Version 9.0.1.1.0 became generally available, but not in time for us to install the patch set on our example configuration. Based on the documentation, if the base release (9.0.1.0.0) is already installed, Oracle Universal Installer will detect that you are on a cluster node and prompt you for the nodes on which you would like to install the patch set. This means that you need only run the installation procedure on one node and the patch set will be installed on other cluster nodes where the base release (9.0.1.0.0) is already installed.

You can download this patch set from Oracle Metalink or you may contact Oracle Support.

We recommend that you regularly check Metalink and Oracle Support for availability of patch sets.

8

Creating and validating a database

In previous chapters we have been through the planning, system configuration, and software installation necessary to get a basic Oracle9*i* Real Application Clusters system up and running. Now we have reached the point where we can create a database with its associated storage, and this is described in this chapter.

8.1 Storage planning

Just as for the predecessor product, Oracle8*i* Parallel Server, the Oracle9*i* RAC system stores its database files in raw devices. Each database file has to be assigned to a specific raw device. Thanks to Logical Volume Manager, we are able to create multiple raw devices in a single logical drive as created by Storage Manager.

Table 8-1 summarizes the relations we established between Storage Manager logical drive names, database file names, and Logical Volume names, and indicates their recommended minimum size and the associated raw devices for each file of the database. We recommend you compile this type of table to plan your database with your own tablespaces and their sizes.

One design decision you have to make during the planning phase is the RAID level you wish to assign to the arrays of the disk subsystem used to store your databases. Further information on this topic is available in Chapter 8 of *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449 and is thus not repeated here. For the database in our lab configuration, we created all the logical drives on a number of RAID-1 arrays.

We defined four logical drives with Storage Manager:

- The Data drive in a RAID-1 array with 6 disks
- The Index drive in a RAID-1 array with 4 disks
- The Redo drive in a RAID-1 array with 2 disks
- The Undo drive in a RAID-1 array with 4 disks

Summing the size requirement for each file that resides on a single logical drive provides the minimum size of the array required. Remember to account for the space that has to be set aside for the particular RAID level you have chosen. In our system, we chose to use six 36 GB drives for the Data logical drive in a RAID-1 configuration, giving us a usable capacity of 108 GB (6 * 36 / 2 GB). This is significantly more that the required minimum capacity as indicated in Table 8-1, but using six physical drives offers considerably improved performance from the disk subsystem.

Table 8-1 Array and database correspondence and sizes

Drive name	Database file	Recommended minimum size	Logical Volume name	Raw device
Data	SPFILE	5 MB	spfilelv	/dev/raw3
Data	SYSTEM	400 MB	systemlv	/dev/raw4
Data	USERS	120 MB	userslv	/dev/raw5
Data	TOOLS	12 MB	toolslv	/dev/raw6
Data	TEMP	110 MB	templv	/dev/raw7
Data	Control file 1	110 MB	ctrl1lv	/dev/raw8
Data	Control file 2	110 MB	ctrl2lv	/dev/raw9
Index	INDEX	70 MB	indexlv	/dev/raw10
Redo	Redo log Thread 1 Group 1	120 MB	redo1_1lv	/dev/raw11
Redo	Redo log Thread 1 Group 2	120 MB	redo1_2lv	/dev/raw12
Redo	Redo log Thread 2 Group 1	120 MB	redo2_1lv	/dev/raw13
Redo	Redo log Thread 2 Group 2	120 MB	redo2_2lv	/dev/raw14
Undo	Undo tablespace Thread 1	312 MB	undo1lv	/dev/raw15
Undo	Undo tablespace Thread 2	312 MB	undo2lv	/dev/raw16

8.2 Storage configuration for the database

Logical drive creation is performed through the FAStT Storage Manager. You have already been through this process in 6.1, “Storage configuration” on page 35, so we do not repeat it in detail here. Refer to that section to review the storage configuration process.

1. Start Storage manager as root on one node of the cluster.
2. Double-click **Storage Subsystem** to display the Subsystem Management window.

We now create the logical drive we have called Data (these steps will be repeated for the remaining drives):

3. Highlight **Unconfigured capacity**.
4. Select **Configure -> Create Array/Logical Drive...**.
5. Select **RAID 1** in the RAID level selection list box.

6. Select **6** drives in the Capacity selection list.
7. Click **Next**.
8. In the Create Logical Drive - Specify Logical Drive Parameters window, we configured the logical drive in the array.
   a. Assign the total capacity of the array to the logical drive.
   b. Enter the name of the logical drive: `Data`.
   c. For the expected logical drive usage, select **Database**.
   d. For the segment size, select **Based on expected usage**.
   e. Select a slot for the controller ownership; we chose **Slot A**.
   f. For the Logical Drive-to-LUN mapping, select **Map later with Storage Partitioning**.
   g. Click **Finish** and then **OK**.

The next step is to enable the nodes in the cluster to use the logical drive. Since the topology has already been created in 6.1.1, “Storage partitioning” on page 39, we do not have to go through the entire process of storage partitioning. We simply need to map our Logical Volume to our cluster group:

9. In the menu select **Configure -> Storage Partitioning...**.
10. Select your Host Group, **ITSO** in our example.
11. From the menu, select **Configure -> Logical Drive-to-LUN Mapping -> Define New Mapping**.
12. In the window:
    a. Select the logical drive name: **Data**.
    b. Increment the logical unit number, which must be unique for each logical drive, as shown in Figure 8-1.

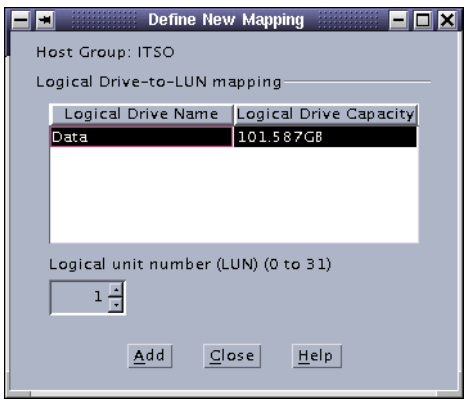

Figure 8-1 Define New Mapping window

    c. Click **Add,** then **Close**
13. The Mapping View should now look similar to Figure 8-2:

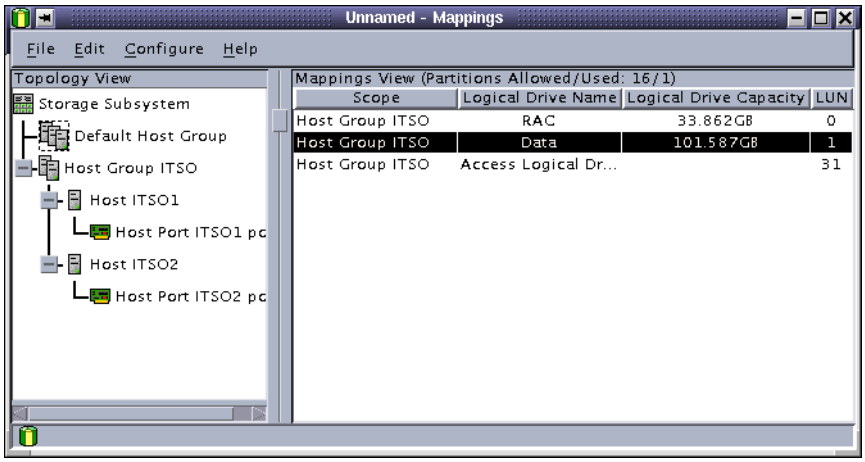

Figure 8-2 Mappings View

14. Repeat steps 3 through 13 for each logical drive you wish to create. In our configuration, these are Data (used above as the example), Index, Redo, and Undo.
15. Close the window.

8.2.1 Activating the devices

On each node, as root, reload the qla2x00 module using the commands shown in Figure 8-3:

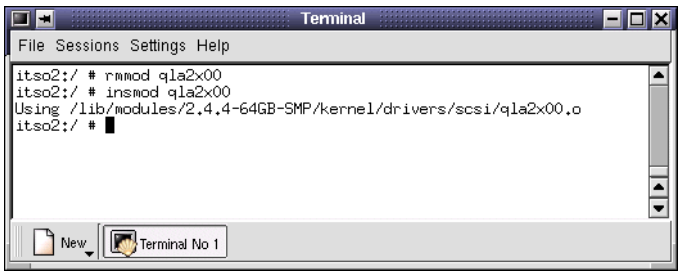

Figure 8-3 reload of the qla2x00 module

Check the last 30 lines of the log file /var/log/messages, to see the devices that map to the new logical drives. In our case it is /dev/sdd, /dev/sdj, /dev/sdk and /dev/sdl. Make a note of these device names, because they will be needed later.

8.3 LVM configuration for the database

We now once again use Logical Volume Manager (LVM) to create logical volumes for use as raw devices by RAC. Follow the process for LVM configuration, described in 6.2, “Logical Volume Manager configuration” on page 46, to create the following Volume Groups:

- **datavg** on the device mapped to the Data drive (for us, partition /dev/sdd1 on /dev/sdd)
- **indexvg** on the device mapped to the Index drive (for us, partition /dev/sdj1 on /dev/sdj)
- **redovg** on the device mapped to the Redo drive (for us, partition /dev/sdk1 on /dev/sdk)
- **undovg** on the device mapped to the Undo drive (for us, partition /dev/sdl1 on /dev/sdl)

Next, using the process already described in 6.2.2, “Creating the Logical Volumes” on page 51, create the following Logical Volumes, assigning the relevant sizes from your database planning table (Table 8-1):

- **spfilelv** in the Volume Group **datavg**
- **systemlv** in the Volume Group **datavg**
- **userslv** in the Volume Group **datavg**
- **toolslv** in the Volume Group **datavg**
- **ctrl1lv** in the Volume Group **datavg**
- **ctrl2lv** in the Volume Group **datavg**
- **indexlv** in the Volume Group **indexvg**
- **redo1_1lv** in the Volume Group **redovg**
- **redo1_2lv** in the Volume Group **redovg**
- **redo2_1lv** in the Volume Group **redovg**
- **redo2_2lv** in the Volume Group **redovg**
- **undo1lv** in the Volume Group **undovg**
- **undo2lv** in the Volume Group **undovg**

LVM configuration is performed on one node only. On each other nodes, log on as root and follow these steps:

1. Reload the qla2x00 module by issuing these commands:

   ```
   rmmod qla2x00
   insmod qla2x00
   ```

2. Execute the command:

   ```
   /sbin/vgscan
   ```

3. Execute the commands:

   ```
   /sbin/vgchange -a y datavg
   /sbin/vgchange -a y indexvg
   /sbin/vgchange -a y redovg
   /sbin/vgchange -a y undovg
   ```

8.4 Database Raw Devices creation

In order to make the servers use the raw device to store the database files, these steps have to be performed on each node:

1. Update the script, raw_devices in /etc/rc.d (created in 6.3, “RAC raw devices creation” on page 54) as shown in Example 8-1:

Example 8-1 /etc/rc.d/raw_devices

```
#! /bin/sh
#

echo "Raw device mapping"
#  racvg
raw /dev/raw1  /dev/racvg/cmlv
raw /dev/raw2 /dev/racvg/svrcfglv

#  datavg
```

```
raw /dev/raw3 /dev/datavg/spfilelv
raw /dev/raw4 /dev/datavg/systemlv
raw /dev/raw5 /dev/datavg/userslv
raw /dev/raw6 /dev/datavg/toolslv
raw /dev/raw7 /dev/datavg/templv
raw /dev/raw8 /dev/datavg/controlfile1lv
raw /dev/raw9 /dev/datavg/controlfile2lv

#  indexvg
raw /dev/raw10 /dev/indexvg/index1lv

#  redovg
raw /dev/raw11 /dev/redovg/redo1_1lv
raw /dev/raw12 /dev/redovg/redo1_2lv
raw /dev/raw13 /dev/redovg/redo2_1lv
raw /dev/raw14 /dev/redovg/redo2_2lv

#  undovg
raw /dev/raw15 /dev/undovg/undo1lv
raw /dev/raw16 /dev/undovg/undo2lv
```

2. Execute this script as root.

8.5 Oracle Net Services initial configuration

In this section, we configure Oracle Net Services. This will create the files listener.ora and tnsnames.ora. The DBCA utility will then update these files with the new database information. Follow the process below:

1. Start the Oracle Net Configuration Assistant by executing **`netca`** as root from a terminal window. The TOPSWelcome window is displayed as shown in Figure 8-4:

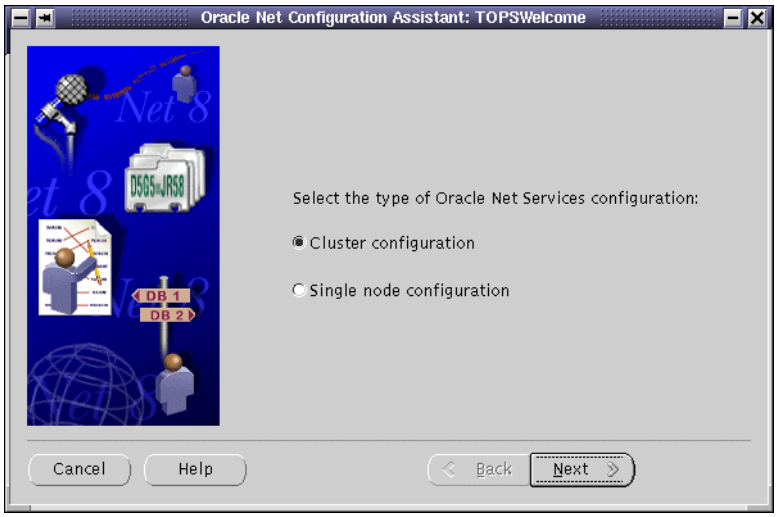

Figure 8-4 Oracle Net Configuration Assistant

2. Select **Cluster configuration** and click **Next**. The Node selection window is displayed (Figure 8-5):

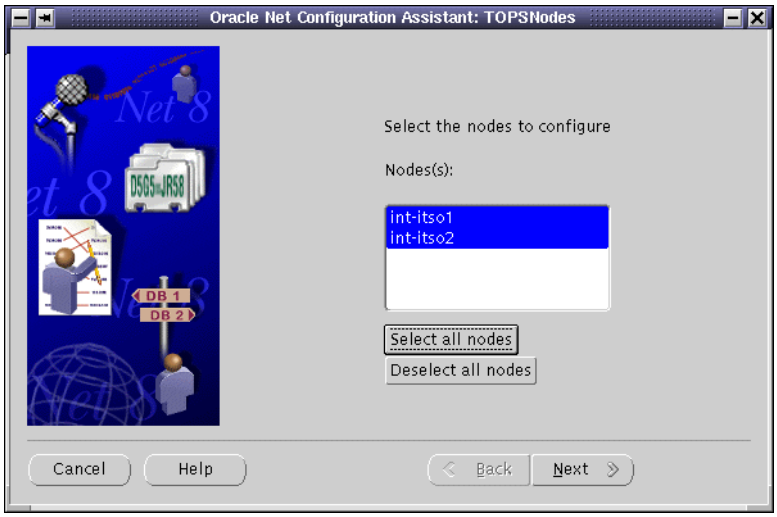

Figure 8-5 Node selection

3. Click **Select all nodes**, then click **Next** to display the Oracle Net Configuration Welcome window (Figure 8-6):

Figure 8-6 Oracle Net Configuration welcome window

4. Select **Listener configurator** and click **Next** to display the Listener Configuration window (Figure 8-7):

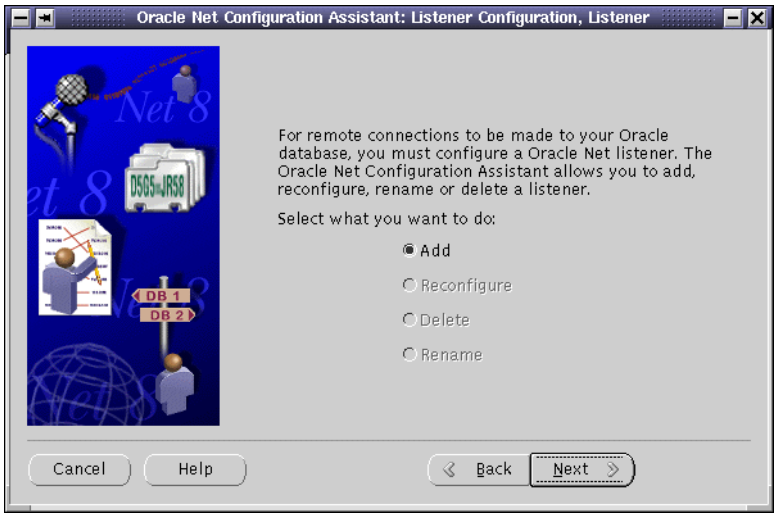

Figure 8-7 Listener configuration

5. Select **Add** and click **Next** to display the Listener Name window (Figure 8-8):

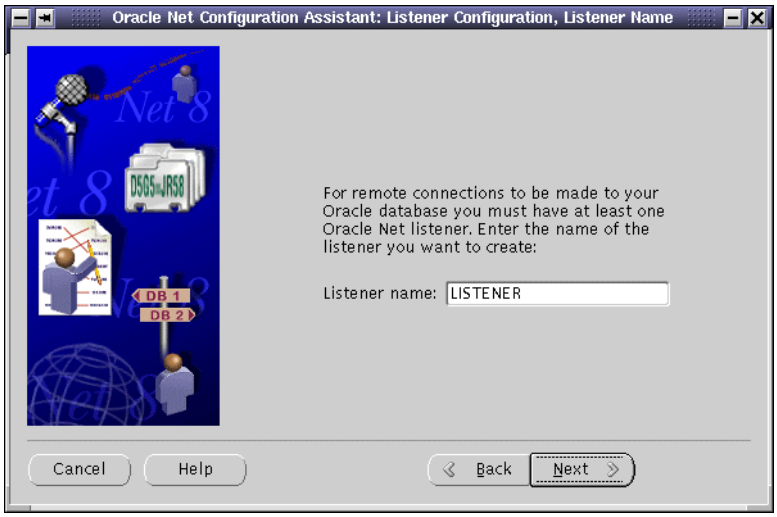

Figure 8-8 Listener name

6. Click **Next** to accept the default, `LISTENER`**,** as the Listener name. The Select Protocols window is displayed (Figure 8-9):

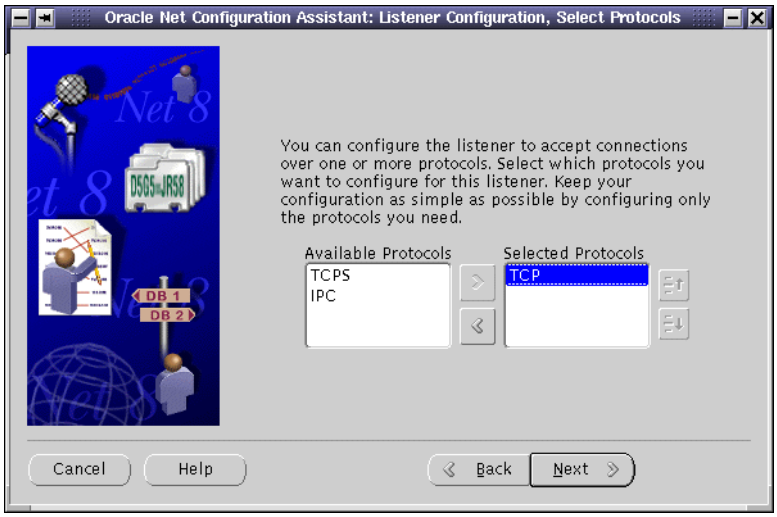

Figure 8-9 Select protocols

7. Verify that **TCP** is in the **Selected Protocols** box. If not, select **TCP** in the Available Protocols box and click **>**. Click **Next** to continue and display the TCP/IP Protocol window (Figure 8-10):

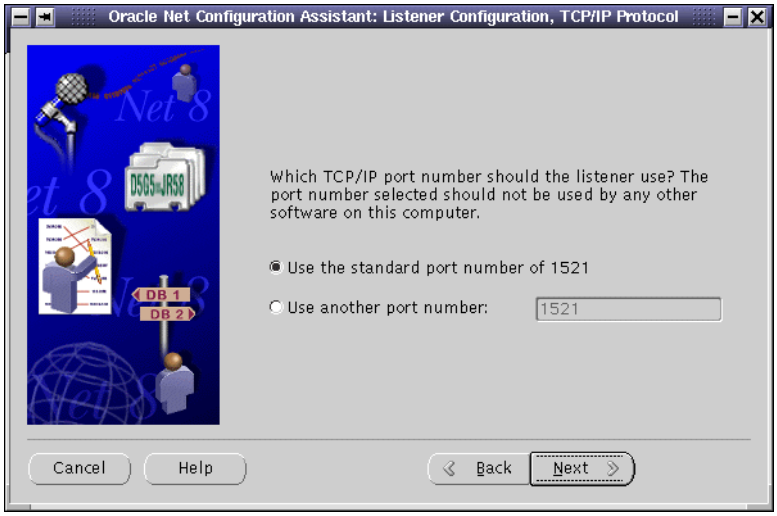

Figure 8-10 Select TCP port

8. Select either **Use the standard port number of 1521** or enter the port number of your choice and click **Next**.
9. The next window asks you whether you want to configure another listener. Select **No** and click **Next**.
10. A window indicating that the listener configuration is complete is displayed. Click **Next** to return to the Welcome window (Figure 8-11):

Figure 8-11 Select Local Net Service Name configuration

11. Select **Local Net Service Name configuration** and click **Next**.The Net Service Name Configuration window is displayed (Figure 8-12):

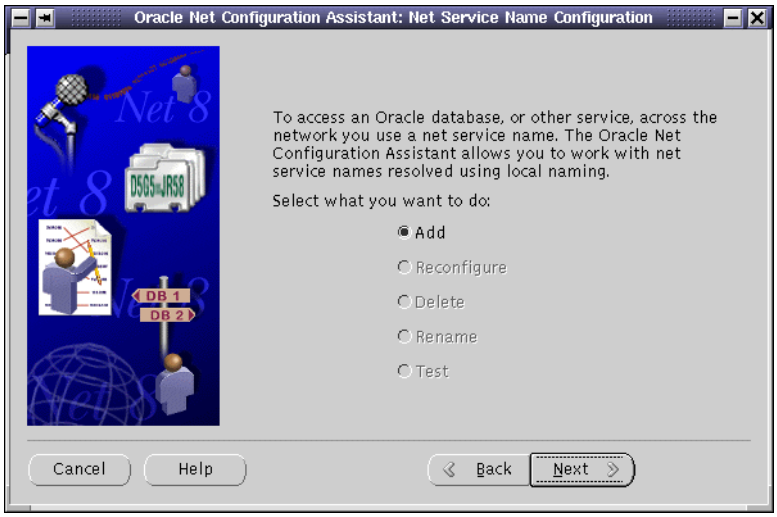

Figure 8-12 Add Net Service Name

12. Select **Add** and click **Next** to display the window that allows you to select which version of Oracle database you wish to access (Figure 8-13):

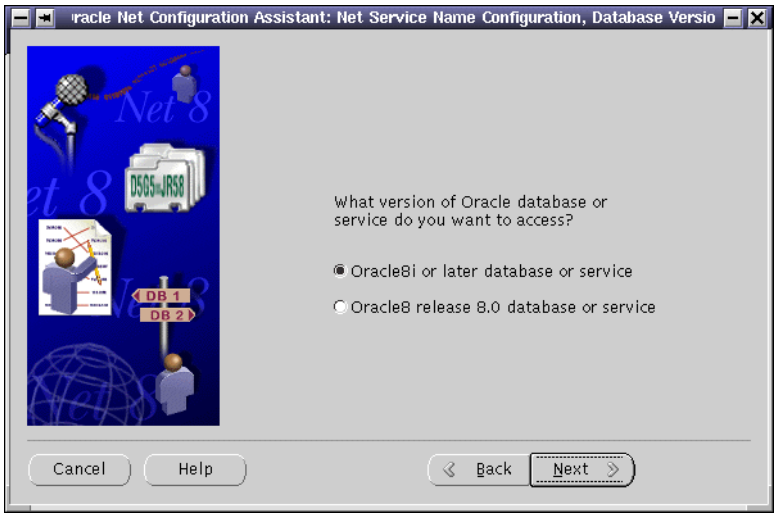

Figure 8-13 Select Oracle database version

13. Select **Oracle8i or later database or service** and click **Next** to display the Service Name window (Figure 8-14):

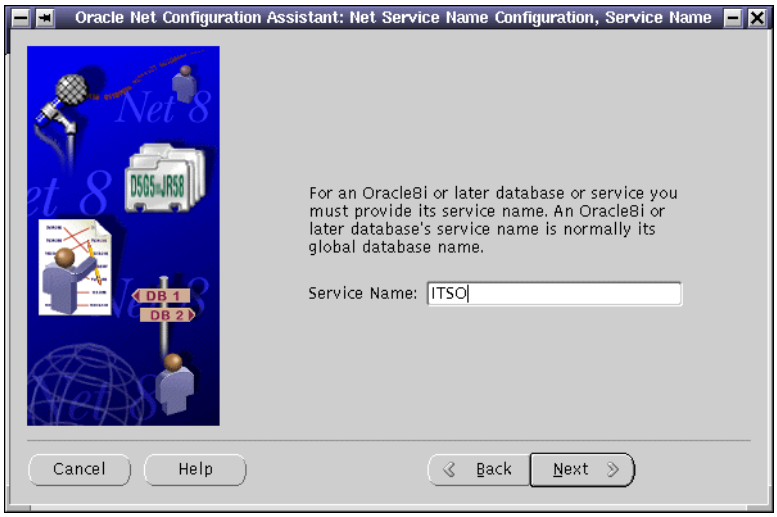

Figure 8-14 Specify database service name

14. Enter a Service Name. We used ITSO as shown. Click **Next** to display the Select Protocol window (Figure 8-15):

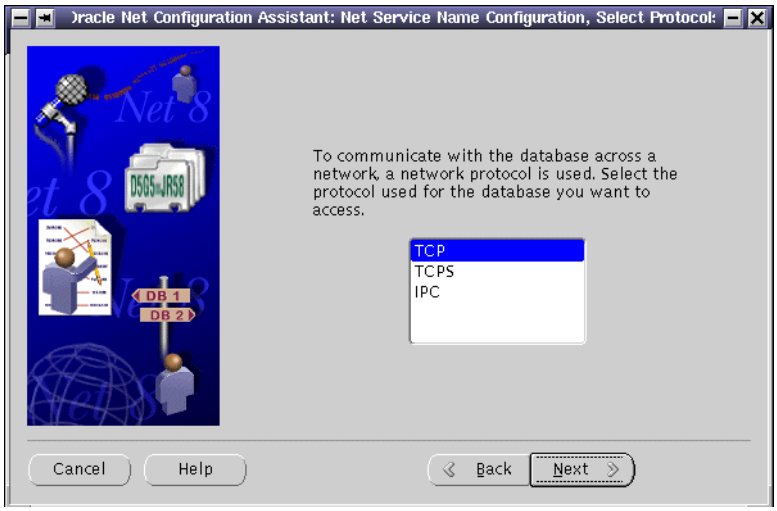

Figure 8-15 Select protocol to use with service

15. Select **TCP** and click **Next**. The next window displayed allows you to enter the host name of the current node (Figure 8-16):

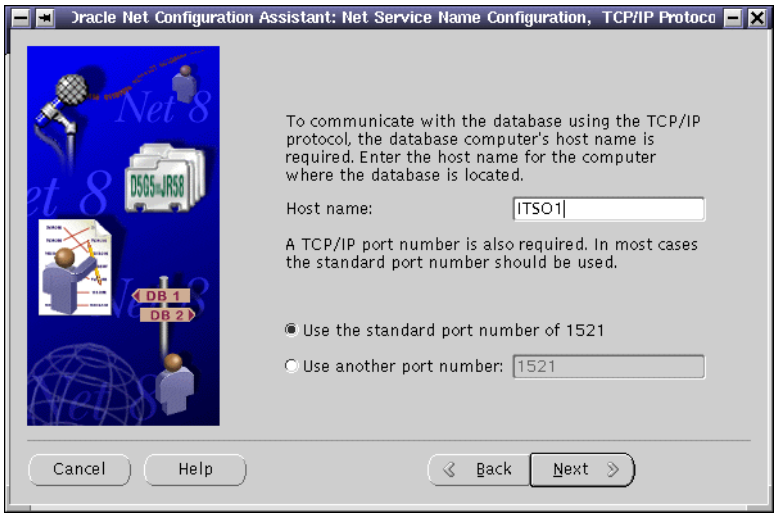

Figure 8-16 Specify the host name and TCP/IP port

16. ,In our case, this is `ITSO1`. Enter the name and select **Use the standard port number of 1521** or your specific port if you did not select the default during the listener configuration. Click **Next** to display the test window (Figure 8-17):

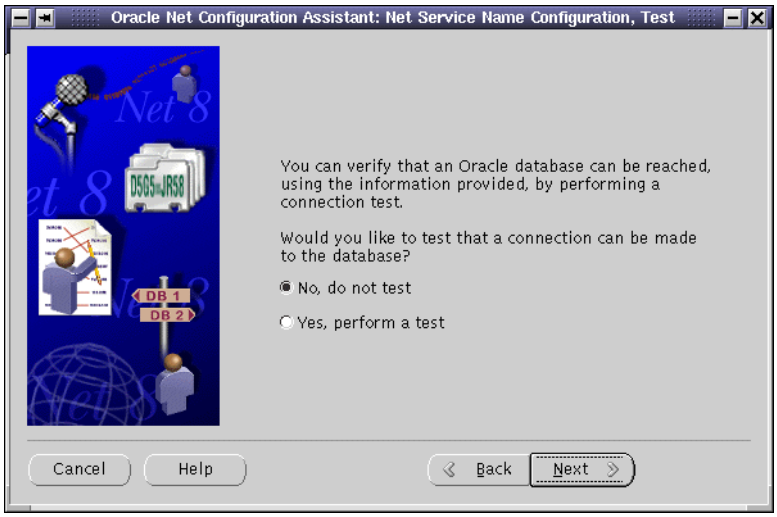

Figure 8-17 Connection test

17. You are asked if you want to test the connection. Select **No, do not test**, because the database has not yet been created. Click **Next** to display the Net service name window (Figure 8-18):

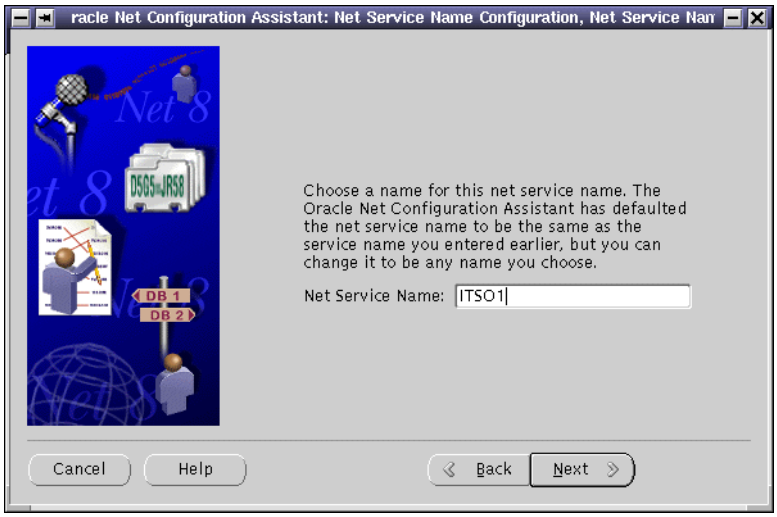

Figure 8-18 Net service name

18. Enter the net service name. We named it `ITSO1`. Click **Next**.
19. The next window asks whether you want to configure another service name. Select **No** and click **Next**.
20. A Configuration Complete window is displayed. Click **Next** to continue.
21. You are now back at the Welcome window. Click **Finish** to exit.

8.6 DBCA configuration file creation

The *Database Configuration Assistant* (DBCA) makes use of a file, named mappingDBCA.cfg, which contains the raw device names for the system datafiles. You should create this file to suit your own tablespaces and associated raw devices and make sure it is owned by the `oracle` user. The file contents for our lab system are shown in Example 8-2:

Example 8-2 Sample mappingDBCA.cfg contents

```
control1=/dev/raw8
control2=/dev/raw9
indx1=/dev/raw10
redo1_1=/dev/raw11
redo1_2=/dev/raw12
redo2_1=/dev/raw13
redo2_2=/dev/raw14
spfile1=/dev/raw3
system1=/dev/raw4
temp1=/dev/raw7
tools1=/dev/raw6
undotbs1=/dev/raw15
undotbs2=/dev/raw16
users1=/dev/raw5
```

The location of this file is passed to DBCA by the variable `DBCA_RAW_CONFIG`. Set this variable to the path of the mappingDBCA.cfg file in .bash_profile for the `oracle` user as shown in Example 8-3:

Example 8-3 /opt/oracle/.bash_profile

```
export DISPLAY=localhost:0.0

export TMPDIR=/tmp

export ORACLE_BASE=/oracle
export ORACLE_HOME=/oracle/product/9.0.1;
export ORACLE_TERM=xterm;
export TNS_ADMIN=/oracle/product/9.0.1/network/admin;
export NLS_LANG=AMERICAN_AMERICA.US7ASCII;
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data;
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/openwin/lib:/usr/local/lib
export THREADS_FLAG=native
export PATH=$ORACLE_HOME/bin:$PATH

export SRVM_SHARED_CONFIG=/dev/raw2

export DBCA_RAW_CONFIG=~/mappingDBCA.cfg
```

To set `DBCA_RAW_CONFIG` for the **dbca** command (which is executed in the next step), source the .bash_profile in the console in which you will run **dbca** as follows:

```
. ~/.bash_profile
```

8.7 Database creation with Database Configuration Assistant

Now we use the Database Configuration Assistant (DBCA) to create the database using the following procedure:

1. As user `oracle`, enter `dbca` to start the Oracle Database Configuration Assistant. The Welcome window shown in Figure 8-19 appears.

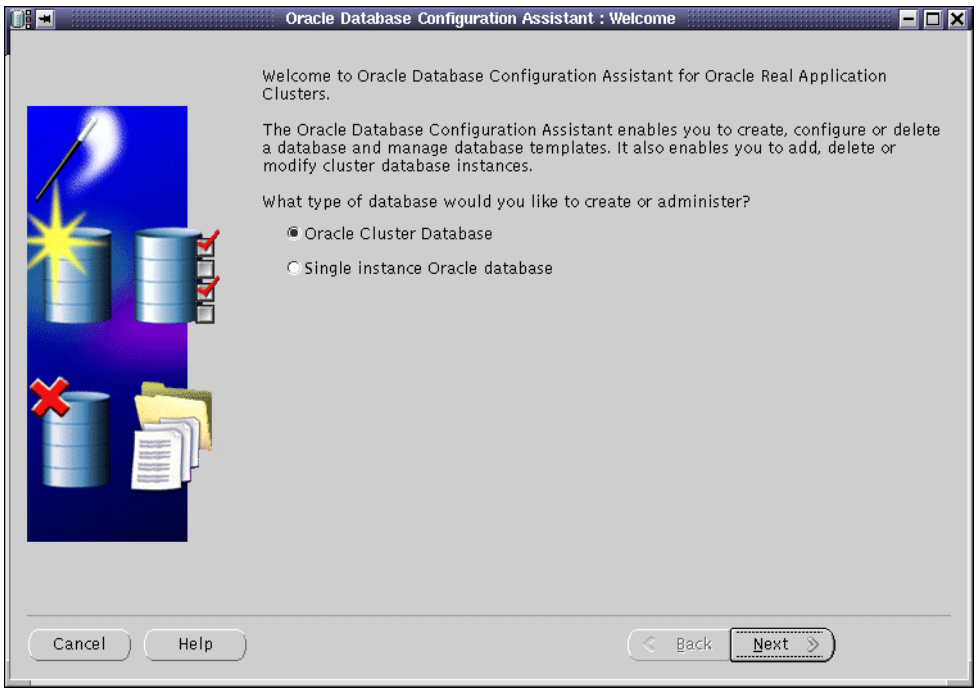

Figure 8-19 DBCA Welcome window

2. Select **Oracle Cluster Database** and click **Next** to display the Operations window (Figure 8-20):

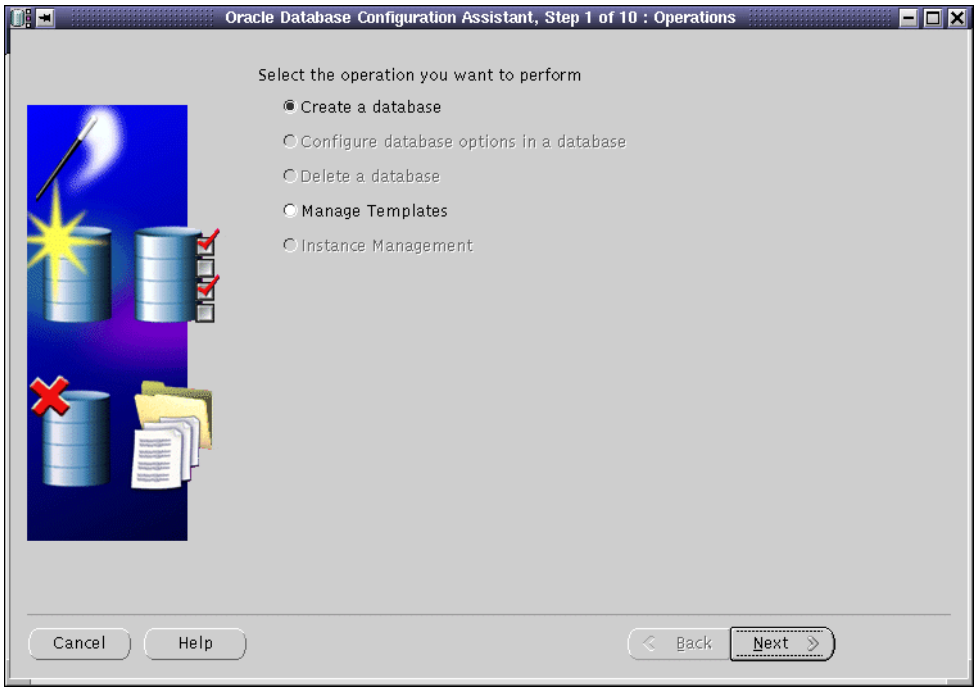

Figure 8-20 Operations window

3. Select **Create a database** and click **Next** to display the Node Selection window (Figure 8-21):

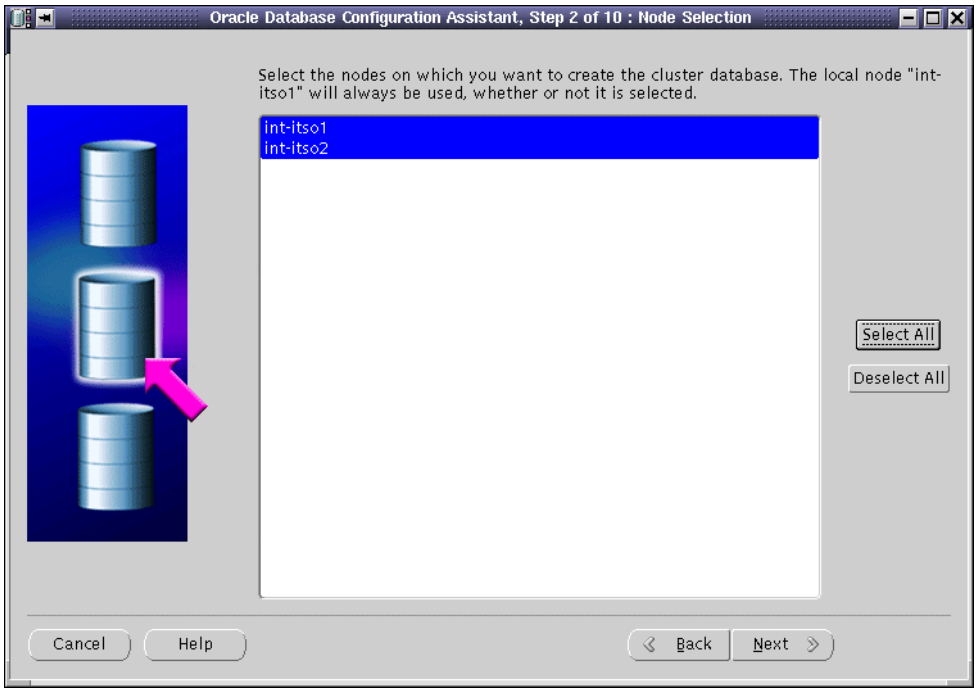

Figure 8-21 Node Selection window

4. Click **Select All** and click **Next** to display the Database Templates window (Figure 8-22):

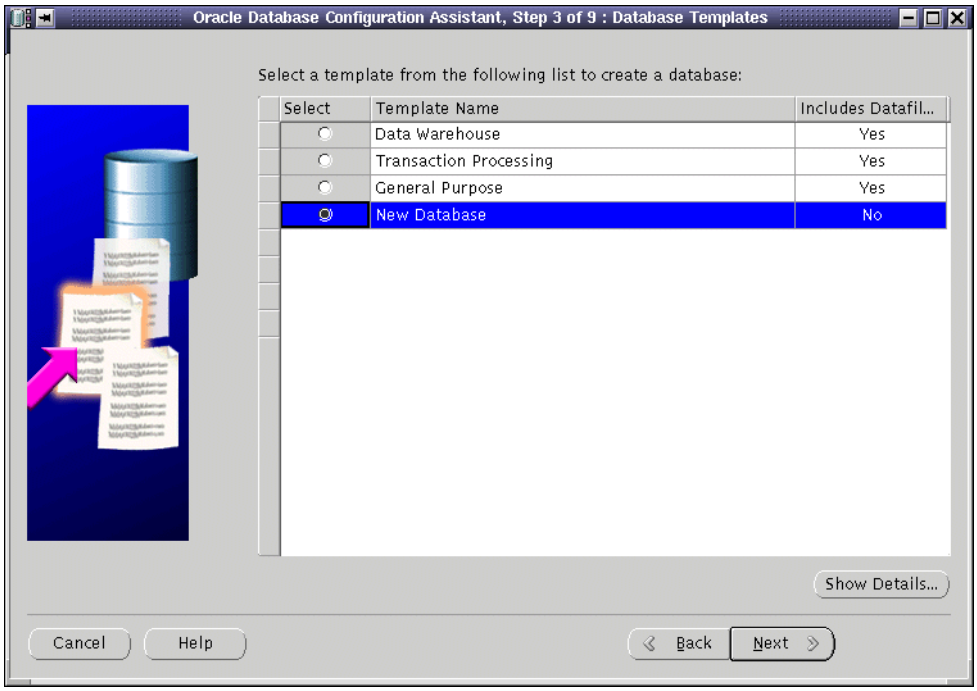

Figure 8-22 Database Templates window

5. Select **New Database** and click **Next** to display the Database Identification window (Figure 8-23):

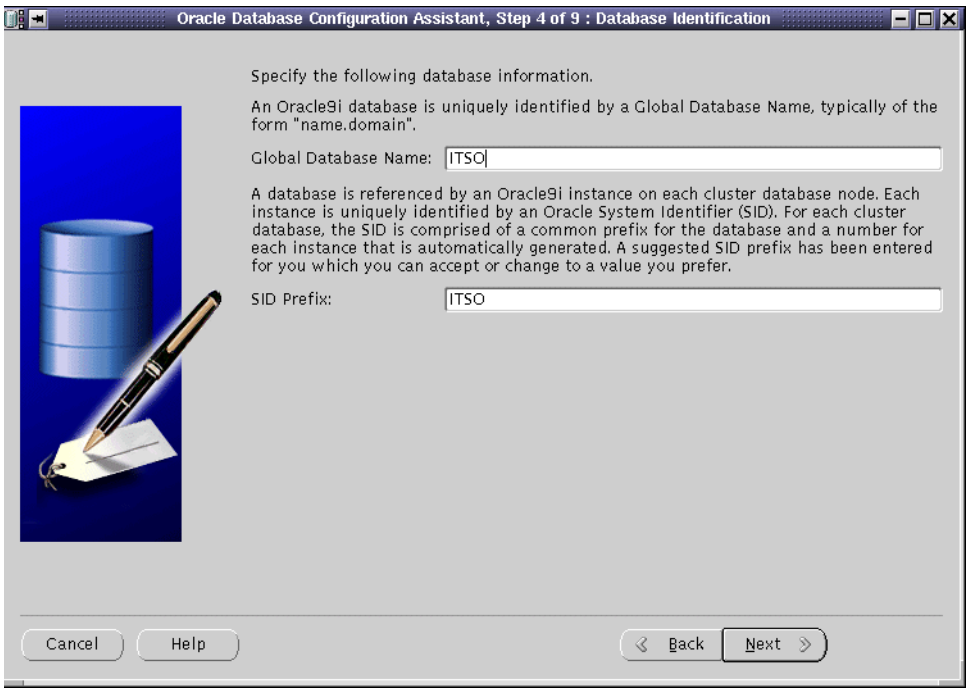

Figure 8-23 Database Identification window

6. Enter the name of your database (we chose `ITSO`) for Global Database Name and verify that the name also appears in SID Prefix. Click **Next** to continue to the Database Options window (Figure 8-24):

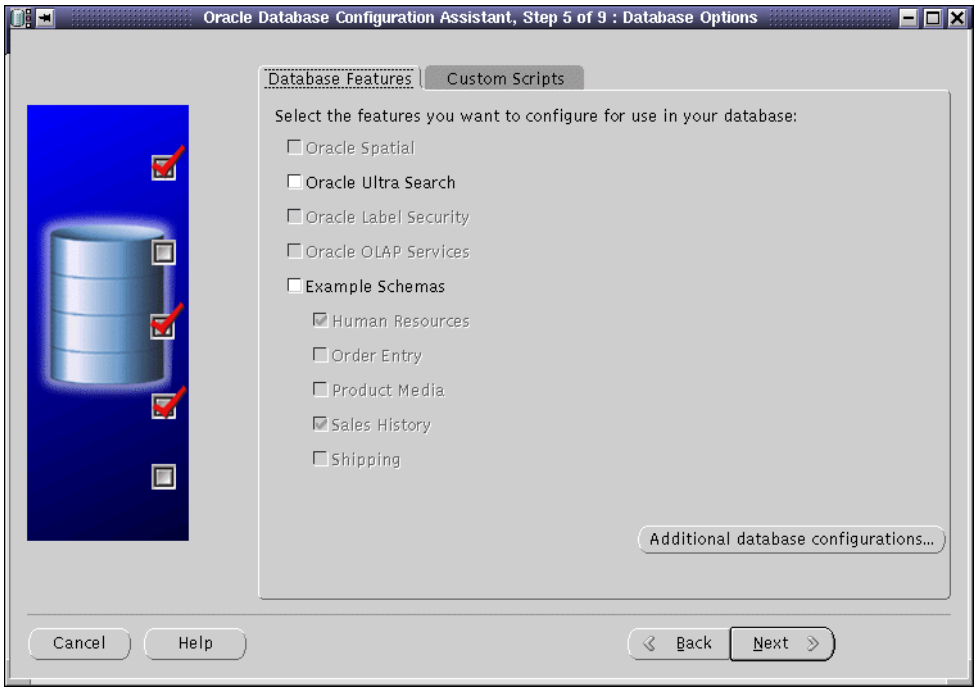

Figure 8-24 Database Options window

7. Deselect everything as shown in Figure 8-24, and click **Additional database configurations...**, to display the pop-up window shown in Figure 8-25:

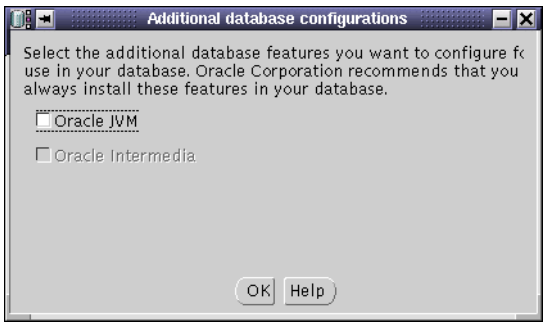

Figure 8-25 Additional database configurations popup window

8. Deselect both items and click **OK**.

   We chose not to install the Java Virtual Machine (JVM) in Oracle because it fails at the database creation. If you need the JVM, check MetaLink to see if there is a patch or solution for this issue.

9. You will be asked if you want to delete the Intermedia associated tablespace, answer **Yes** and the **Database Connection Options** window is displayed (Figure 8-26):

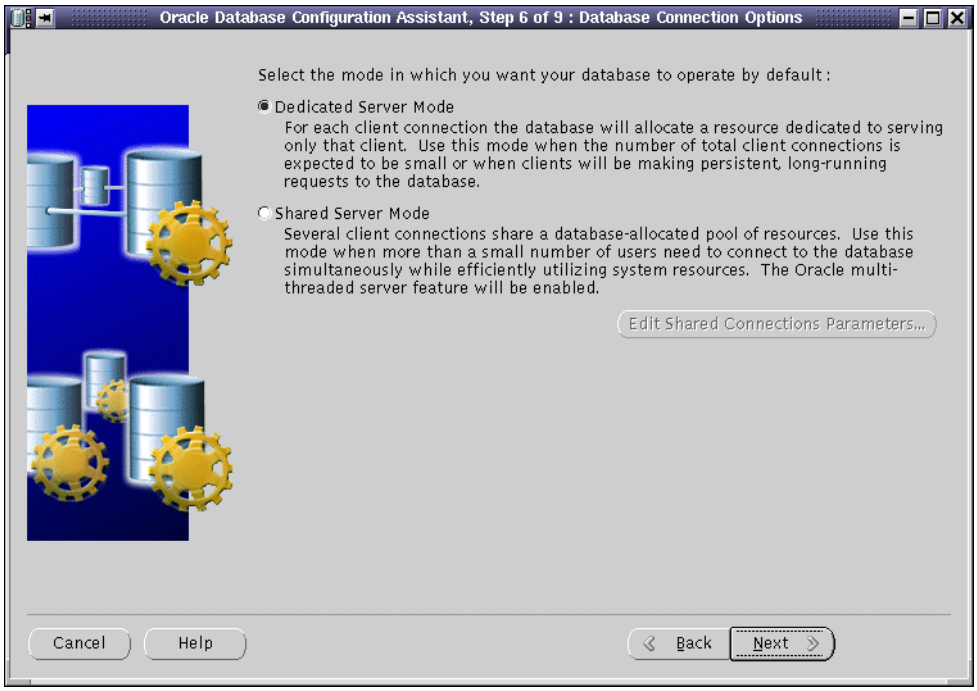

Figure 8-26 Database Connection Options

10. Select **Dedicated Server Mode**.

    Shared Server database creation fails with DBCA. You will be able to convert the Dedicated Server database to a Shared Server database later if you wish.

11. Click **Next** to display the Initialization Parameters window. This window has four tabs. The next four figures show our settings for the Memory tab (Figure 8-27):

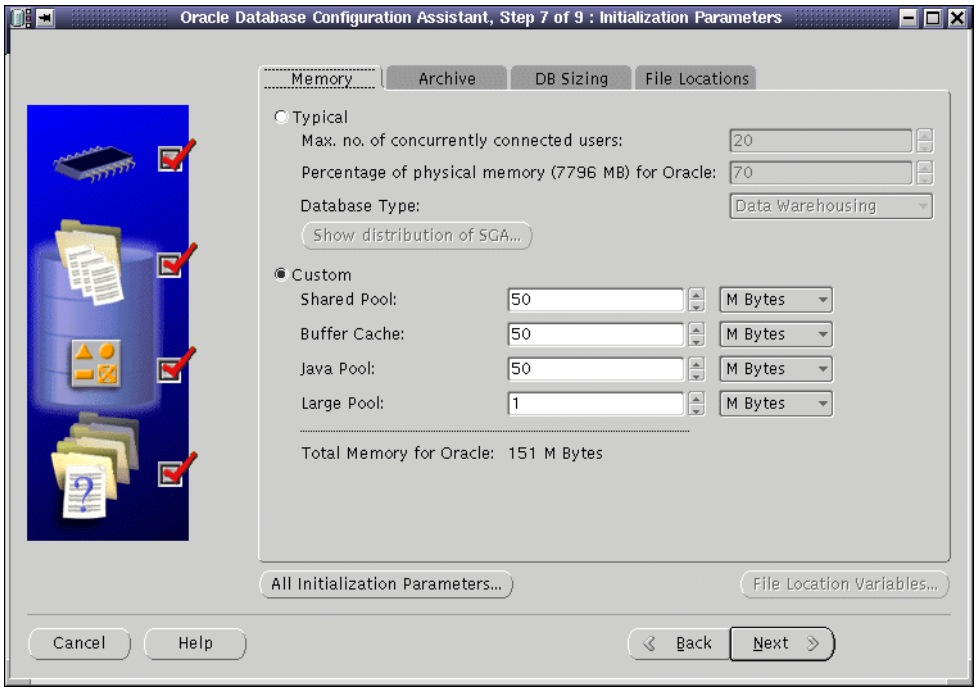

Figure 8-27 Initialization Parameters window: Memory tab

The Archive tab (Figure 8-28):

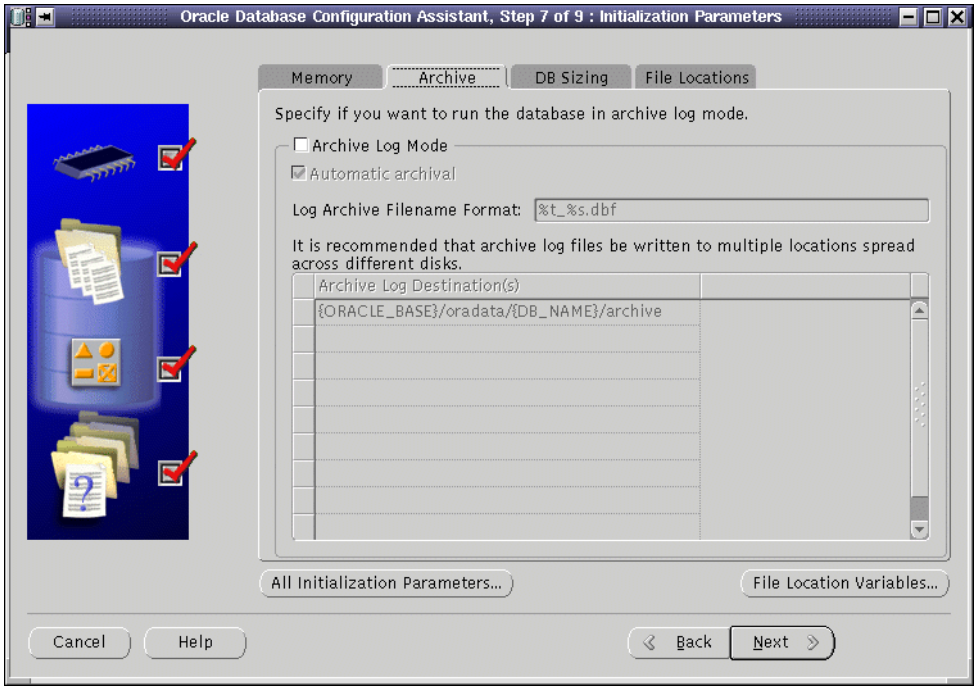

Figure 8-28 Initialization Parameters window: Archive tab

The DB Sizing tab (Figure 8-29):

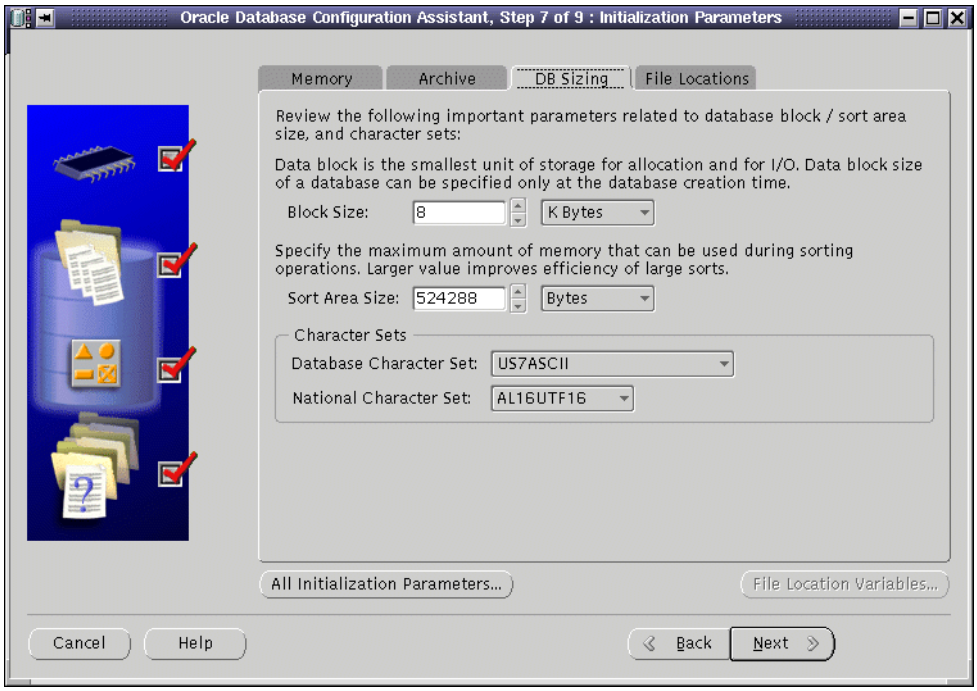

Figure 8-29 Initialization Parameters window: DB Sizing tab

The File Locations tab (Figure 8-30):

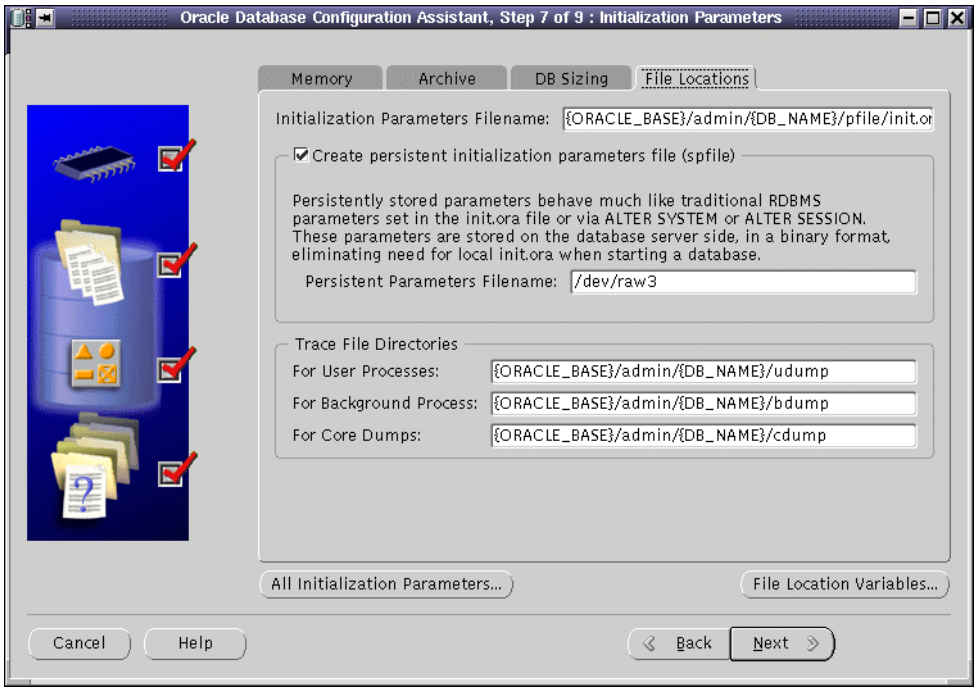

Figure 8-30 Initialization Parameters window: File Locations tab

12. Enter the values for your database initialization parameters and click **Next** to display the Database Storage window (Figure 8-31):

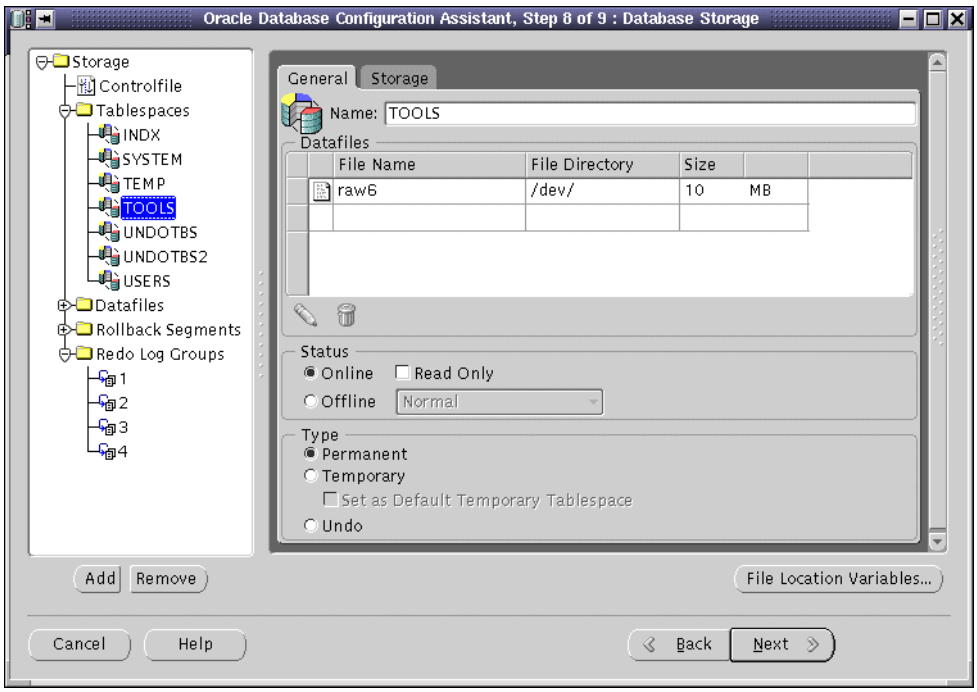

Figure 8-31 Database Storage window

13. Click the items under **Tablespaces**, **Controlfile** and **Redo Log Groups** and verify that the information displayed in the right-hand pane is correct.

The entries under File Name should be the raw devices created earlier. If any entries are incorrect, you have made a mistake in your mappingDBCA.cfg file. If so, click **Cancel**, make the changes to mappingDBCA.cfg as necessary, and restart at step 1 on page 89.

14. Click **Next** to continue and display the Creation Options window (Figure 8-32):

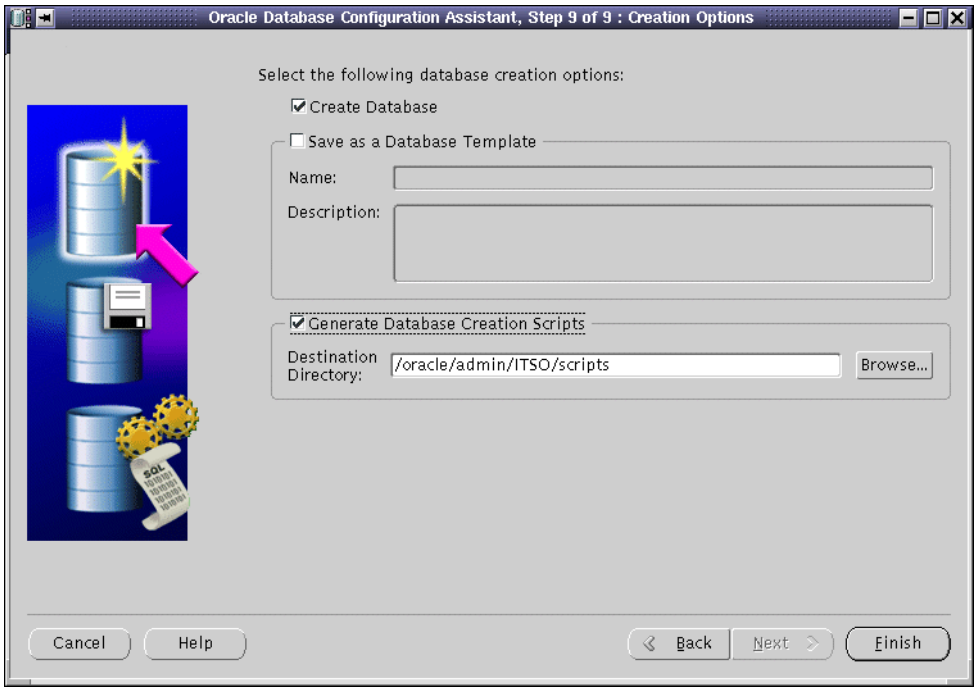

Figure 8-32 Creation Options window

15. Select **Create Database** and **Generate Database Creation Scripts** as shown. Click **Finish** to display the Summary window (Figure 8-33):

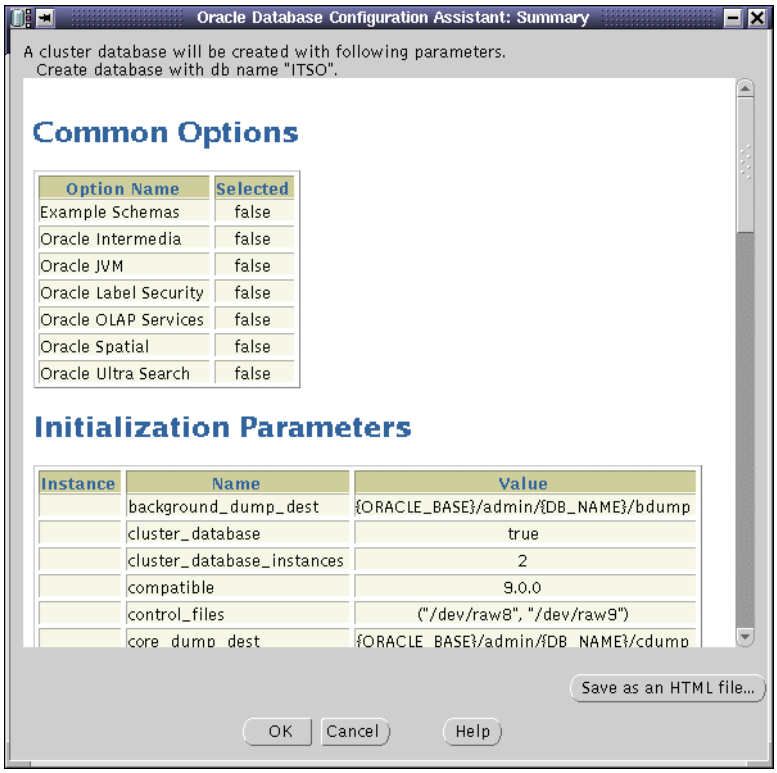

Figure 8-33 Summary window

16. If desired, you may save this information by clicking **Save as HTML file...** before clicking **OK** to continue. At this point, DBCA generates a creation script. When DBCA finishes this step, the pop-up window shown in Figure 8-34 is displayed:

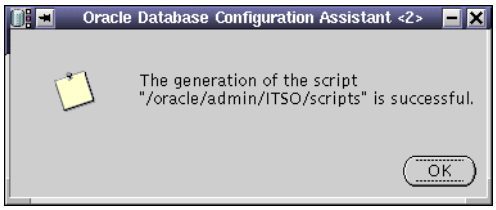

Figure 8-34 Completion of script generation window

17. Click **Ok** in this pop-up window. Do not click Finish again in the Creation Options window (Figure 8-32). Database creation will start and the progress window shown in Figure 8-35 appears:

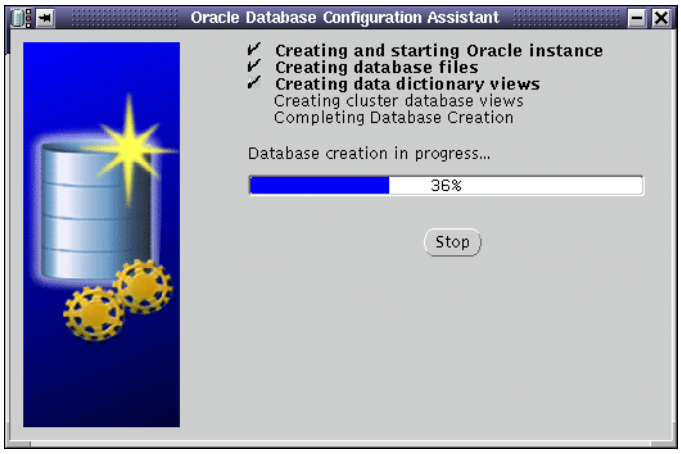

Figure 8-35 Progress window

18. Wait until DBCA completes all of its steps. When complete, the window in Figure 8-36 appears.

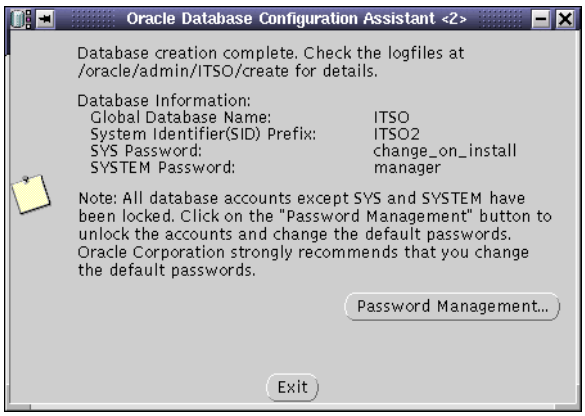

Figure 8-36 Completion window.

19. Click **Exit** to end.

8.8 Oracle Net Services configuration update

Our previous steps have left the listeners monitoring the private interconnect, but we want them to use the production network. To achieve this, after the database has been created using **dbca**, modify the Oracle Net configuration files as follows:

1. On itso1, stop the listener by typing the command:

   ```
   srvctl stop -p ITSO -s lsnr
   ```

2. Modify the files $ORACLE_HOME/network/admin/listener.ora and $ORACLE_HOME/network/admin/tnsnames.ora on both nodes as follows:
   - Change all occurrences `int-itso1` to `itso1`
   - Change all `occurrences int-itso2` to `itso2`
3. Restart the listener by typing:

   ```
   srvctl start -p ITSO -s lsnr
   ```

8.8.1 Oracle user profile update

Now the database has been created, and each node has its own instance. Update each node's .bash_profile for the oracle user with the correct ORACLE_SID value. The instance name for a node is the SID prefix (see step 6 on page 91) concatenated with the thread number (instance number). For example, our first node itso1 will have the instance name ITSO1 because the SID prefix is ITSO (see the highlighted entry in Example 8-4).

Example 8-4 /opt/oracle/.bash_profile

```
export DISPLAY=localhost:0.0

export TMPDIR=/tmp

export ORACLE_BASE=/oracle
export ORACLE_HOME=/oracle/product/9.0.1;
export ORACLE_TERM=xterm;
export TNS_ADMIN=/oracle/product/9.0.1/network/admin;
export NLS_LANG=AMERICAN_AMERICA.US7ASCII;
export ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data;
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/openwin/lib:/usr/local/lib
export THREADS_FLAG=native
export PATH=$ORACLE_HOME/bin:$PATH

export SRVM_SHARED_CONFIG=/dev/raw2

export DBCA_RAW_CONFIG=~/mappingDBCA.cfg

export ORACLE_SID=ITSO1
```

Source the new .bash_profile in your current shell by issuing the command:

```
. ~/.bash_profile
```

9

Configuring RAC for High Availability and Load Balancing

In this chapter, we show you how to configure load balancing and failover on Real Application Clusters, and how to configure the clients that will connect to RAC.

9.1 Fibre Channel high availability

At the time this Redpaper was being written, the multi-path I/O driver had not yet been ported to Linux. A multi-path I/O driver allows you to install two FAStT Fibre Channel Host Adapters in a single server in a failover configuration. If one adapter fails, the other will take over its workload, providing the system with redundancy at the adapter level.

Every other component of the Fibre Channel architecture can be configured for redundancy with Linux. The FAStT500 Fiber Channel controller has a pair of redundant controllers, and redundant mini-hubs and power supplies. The EXP500 disk enclosure can be duplicated and used for RAID arrays, and the Fibre Channel network can utilize redundancy hubs or switches.

Thus, high availability is ensured for the storage.

9.2 Dedicated or Shared Server for the database?

With Oracle8*i* Parallel Server, the load balancing between nodes was only available with a Shared Server (formerly called multithreaded servers) database. With Oracle9*i* Real Application Clusters, load balancing and failover capabilities are also available with a Dedicated Server database.

In *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449, we recommended that the database be converted to Shared Server mode for load balancing. With Oracle9*i* RAC, this recommendation is outdated, since load balancing is available in both Dedicated and Shared Server modes. The decision is yours. We suggest that you start with the Dedicated Server mode, and later, as the number of users and workload increases, there may be a point where it may be beneficial to move to Shared Server mode. This point at which you should switch is unique to your implementation.

The database we created in step 10 on page 93 is a Dedicated Server database. As we said then, DBCA fails if you attempt to create a Shared Server database. You can change a Dedicated Server database to a Shared Server database by setting the database `DISPATCHERS` parameter value.

Specifically, you need to set the `DISPATCHERS` parameter to:

`"(protocol=TCP)(listener=`*`LISTENERS_ITSO`*`)"`

replacing *`LISTENERS_ITSO`* with the name of the entry in the `TNSNAMES.ORA` that represents the listeners on each node in your implementation.

Here are the steps you need to follow:

1. Log in as `oracle` and connect to the database as SYSDBA with these commands:

   ```
   Sqlplus /nolog
   Connect / as sysdba
   ```

2. In the SQLPLUS session, enter this command:

   ```
   Create pfile='/opt/oracle/pfile.ora' from spfile='/dev/raw3'
   ```

   This extracts the existing parameters from the raw device and places them into a text file.

3. Now, to update /dev/raw3 with the updated `DISPATCHERS` parameter, edit /opt/oracle/pfile.ora, and append this line to the file:

   *.dispatchers="(protocol=TCP)(listener=LISTENERS_ITSO)"

   as shown in Example 9-1:

Example 9-1 Updating /opt/oracle/pfile.ora

```
*.background_dump_dest='/oracle/admin/ITSO/bdump'
*.cluster_database_instances=2
*.cluster_database=true,true
*.compatible='9.0.0'
*.control_files='/dev/raw8','/dev/raw9'
*.core_dump_dest='/oracle/admin/ITSO/cdump'
*.db_block_size=8192
*.db_cache_size=52428800
*.db_domain=''
*.db_name='ITSO'
*.fast_start_mttr_target=300
ITSO2.instance_name='ITSO2'
ITSO1.instance_name='ITSO1'
ITSO1.instance_number=1
ITSO2.instance_number=2
*.java_pool_size='52428800'
*.large_pool_size='1048576'
*.open_cursors=300
*.processes=150
*.remote_listener='LISTENERS_ITSO'
*.remote_login_passwordfile='exclusive'
*.resource_manager_plan='SYSTEM_PLAN'
*.shared_pool_size=52428800
```

```
*.sort_area_size=524288
ITSO1.thread=1
ITSO2.thread=2
*.timed_statistics=TRUE
*.undo_management='AUTO'
ITSO1.undo_tablespace='UNDOTBS'
ITSO2.undo_tablespace='UNDOTBS2'
*.user_dump_dest='/oracle/admin/ITSO/udump'
*.dispatchers='(protocol=TCP)(listener=LISTENERS_ITSO)'
```

4. Return to the previous SQLPLUS session, and enter this command:

   Create spfile='/dev/raw3' from pfile='/opt/oracle/pfile.ora'

5. Quit SQLPLUS.

Now, restart the database (See Appendix A, “Useful commands” on page 109, for the commands to stop and start the database).

9.3 Net Service configuration for RAC

You should check a number of items for correct configuration of the Net Service. We discuss them in the sections that follow.

9.3.1 Listener configuration

The listener should be properly configured by **netca** and **dbca** but it is worth checking that the contents of $ORACLE_HOME/network/admin/listener.ora are similar to those in Example 9-2 (this file is for the node itso1):

Example 9-2 listener.ora for itso1

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = itso1)(PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /oracle/product/9.0.1)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (ORACLE_HOME = /oracle/product/9.0.1)
      (SID_NAME = ITSO1)
    )
  )
```

Note: Before making any changes to the listener configuration, stop it on both nodes. When the changes have been made on both nodes, restart it again on both nodes. See Appendix A, “Useful commands” on page 109, for the commands to stop and start the listener.

9.3.2 tnsnames.ora configuration

The file $ORACLE_HOME/network/admin/tnsnames.ora contains the Oracle Net Service configuration for connecting to databases. A sample file is shown in Example 9-3 on page 105.

The file consists of a number of sections. In our two node example, they include:

- `LISTENERS_ITSO`

  This section contains the information about the listeners for all nodes in the cluster. The information is used by the instances to register the listeners at startup.
- `ITSO1` contains the information needed to connect to the instance on the node itso1.
- `ITSO2` contains the information needed to connect to the instance on the node itso2.

  The two previous entries are used mainly for administration tasks, when it is important to connect to a specific instance.
- ITSO contains the information needed to connect to the database, and whether or not load balancing and failover are configured.

The other entries are for other database services or other components, such as Apache.

Load balancing settings

To enable load balancing at connection time, set the parameter `LOAD_BALANCE` to `yes` in TNSNAMES.ORA for the equivalent of the `ITSO` entry in your installation, as shown in Example 9-3.

Note: This parameter is set by default by **`netca`** and **`dbca`**.

Transparent Application Failover settings

To enable Transparent Application Failover (TAF), modify tnsnames.ora to set the parameter `FAILOVER` to `yes`, and insert a `FAILOVER_MODE` section in the `CONNECT_DATA` block for the equivalent of the `ITSO` entry in your installation as shown in Example 9-3.

`FAILOVER_MODE` has four parameters:

- The `TYPE` parameter sets what is done in case of failover (resume an interrupted select query, or keep the session but interrupt any select). In either case, any Insert, Delete, or Update database modification will be undone (rolled back).
- The `METHOD` parameter sets how the failover is performed, by determining whether or not a shadow process is started on the other nodes.
- The `RETRIES` parameter is the maximum number of unsuccessful reconnections to the database that are attempted before an error is returned.
- The `DELAY` parameter is the time between each reconnection attempt.

See the *Oracle9i Net Services Administrator's Guide* for more information.

Example 9-3 Extract from tnsnames.ora

```
LISTENERS_ITSO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso2)(PORT = 1521))
    )
  )

ITSO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso2)(PORT = 1521))
    )
    (FAILOVER = yes)
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVICE_NAME = ITSO)
      (FAILOVER_MODE=
        (TYPE=SELECT)
        (METHOD=BASIC)
        (RETRIES=30)
        (DELAY=1)
      )
    )
  )

ITSO1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ITSO)
      (INSTANCE_NAME = ITSO1)
    )
  )

ITSO2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ITSO)
      (INSTANCE_NAME = ITSO2)
    )
  )
```

9.4 Configuring clients for RAC

To make use of your cluster, the last task to perform is the configuration of the RAC clients.

9.4.1 What is a client for RAC?

Clients for Real Application Clusters differ depending upon the system architecture you have implemented:

- In a 2-tier client-server architecture, the client for RAC is the program on the end-user's workstation. Such workstations must have the Oracle9*i* client installed and a correctly configured tnsnames.ora file.
- In a 3-tier client-server architecture, the application server is the client for RAC. This server must have Oracle9*i* client installed and a correctly configured tnsnames.ora file. The end-user workstation needs to have only a thin client program installed.
- In a Web-based architecture, the Web application server is the client for RAC. This server must have Oracle9*i* client installed and a correctly configured tnsnames.ora file. The end-user workstation needs to have only a Web browser installed.

The Oracle9*i* client software provides libraries and APIs necessary to connect to an Oracle9*i* database.

9.4.2 APIs for client programs

Using OPS 8.1.5, Transparent Application Failover (TAF) was only available to programs that used the Oracle Connection Interface (OCI) database connection. Now, with Oracle9*i*, programs that use Open Database Connectivity (ODBC) or Java Database Connectivity (JDBC) can take advantage of TAF.

For JDBC, the connection must be made using the OCI mode, not THIN mode. Example 9-4 is a sample Java class that connects to a RAC database. Note that, in the connect string "jdbc:oracle:oci:@ITSO", ITSO is the entry in tnsnames.ora that contains the `FAILOVER_MODE` parameters.

Example 9-4 RACdb class that connects to a RAC database

```
import java.io.*;
import java.util.*;
import java.sql.*;

public class RACdb extends Object {

 Connection m_connection=null;

 public RACdb() {
   try{
     DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
     m_connection = DriverManager.getConnection(
                 "jdbc:oracle:oci:@ITSO",
                 "SCOTT",
                 "TIGER");
     m_connection.setAutoCommit(false);
   }catch(SQLException ex){
     System.out.println("Error while Initializing Database Connection " + ex.toString());
   }
 }
}
```

9.4.3 Client Net Service configuration

The client Net Service configuration for an RAC database simply includes the configuration to connect to the database service with TAF and the load balancing feature. See an example of the tnsnames.ora in Example 9-5.

Example 9-5 Client tnsnames.ora

```
ITSO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = itso2)(PORT = 1521))
    )
    (FAILOVER = yes)
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVICE_NAME = ITSO)
      (FAILOVER_MODE=
        (TYPE=SELECT)
        (METHOD=BASIC)
        (RETRIES=30)
        (DELAY=1)
      )
    )
  )
```

A

Useful commands

This appendix provides a brief summary of some of the most useful commands for controlling your Oracle9*i* Real Application Clusters system. The **svrctl** command allows you to manage the database and listeners globally for the entire cluster from the command line. For more information on the srvctl tool, see the Oracle9*i* Real Application Clusters Administration documentation.

Status

- Get the status of the database ITSO (instances and listeners):

```
svrctl status -p ITSO
```

Start

- Start the database ITSO (instances and listeners):

```
svrctl start -p ITSO
```

- Start the listener associated with instance itso1:

```
svrctl start -p ITSO -i itso1 -s lsnr
```

- Start the instance itso1:

```
svrctl start -p ITSO -i itso1 -s inst
```

Stop

- Stop the database ITSO (instances and listeners):

```
svrctl stop -p ITSO
```

- Stop the listener associated with instance itso1:

```
svrctl stop -p ITSO -i itso1 -s lsnr
```

- Stop the instance itso1:

```
svrctl stop -p ITSO -i itso1 -s inst
```

Delete

- Delete the database ITSO:

```
srvctl delete db -p ITSO
```

Configuration

- List the database created on RAC:

  ```
  svrctl config
  ```

- List the configuration of the database ITSO:

  ```
  svrctl config -p ITSO
  ```

Useful sources of information

IBM Redbooks (http://www.redbooks.ibm.com):

- *Oracle Parallel Server and Windows 2000 Advanced Server on IBM Netfinity*, SG24-5449
- *Fibre Array Storage Technology A FAStT Introduction*, SG24-6246

Oracle9i documentation (http://docs.oracle.com/):

- *Real Application Clusters Concepts*
- *Real Application Clusters Administration*
- *Real Application Clusters Installation and Configuration*
- *Real Application Clusters Guard Administration and Reference Guide*
- *Net Services Administrator's Guide*
- *Release Notes, Release 1 (9.0.1) for Linux Intel*
- *Quick Installation Procedure, Release 1 (9.0.1) for Linux Intel*
- *Installation Guide, Release 1 (9.0.1) for UNIX Systems: AIX-Based Systems, Compaq Tru64 UNIX, HP 9000 Series HP-UX, Linux Intel and Sun SPARC Solaris*

SuSE LVM white paper:

http://www.suse.de/en/support/oracle/docs/lvm_whitepaper.pdf

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively

through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.