

Tips for Installing and Configuring Oracle9i Real Application Clusters on Red Hat Linux Advanced Server

An Oracle White Paper
February 2003

Tips for Installing and Configuring Oracle9i Real Application Clusters on Red Hat Linux Advanced Server

Executive overview.....	3
Introduction.....	3
Deployment Principles.....	5
Private networks.....	5
Shared storage.....	5
Operating system services.....	7
Operating system configuration for Red Hat Linux Advanced Server	
2.1.....	7
Public and private network configuration.....	7
Shared storage configuration.....	10
Shared raw devices.....	10
Network attached storage using a Netapp filer.....	12
The Oracle Cluster File System.....	13
Kernel parameter configuration.....	13
Establishing host equivalence.....	14
Oracle RAC installation and configuration.....	15
Cluster Manager installation and configuration.....	16
Using watchdogd with Oracle 9.2 RAC	16
Installing the hangcheck-timer module.....	17
Using the hangcheck-timer module.....	17
Installing and starting the Oracle Cluster Manager.....	17
Oracle 9.2.0.1 with Real Application Cluster installation.....	18
Additional manual configuration.....	21
Related work.....	21
Conclusion.....	22

Tips for Installing and Configuring Oracle9i Real Application Clusters on Red Hat Linux Advanced Server

EXECUTIVE OVERVIEW

There is growing interest in using Oracle9i Real Application Clusters (RAC) to create robust, scalable, and highly available databases running on low cost hardware using the Linux operating system. This is because Oracle9i RAC for Linux allows businesses and organizations to manage data in a scalable, highly available manner using commodity hardware. The combination of Oracle 9i RAC with Linux software offers these groups the opportunity to increase return on investment and reduce total cost of ownership while they continue to enjoy the performance and manageability advantages that Oracle 9i software provides.

This white paper discusses many important installation and configuration steps in the RAC deployment process that are unique to the Linux operating system. It emphasizes operating system-level tools and techniques specifically used with the Red Hat Linux Advanced Server 2.1 distribution, but presents administrative concepts that are useful with all Linux distributions. It also presents specific information about how to deploy Oracle9i RAC on shared raw devices, network attached storage (NAS) with Network Appliance filers, and the Oracle Cluster File System (OCFS). Lastly, it highlights key aspects of the RAC installation and configuration process where incorrect use of Oracle-provided tools commonly cause operating system-specific errors.

INTRODUCTION

The process used to install, configure, and create an Oracle9i Real Application Cluster (or RAC) database on Linux shares much in common with the process used for RAC deployment on any operating system platform. All RAC deployments configure shared storage resources, network resources, and software services that enable RAC node participation, operation, and communication. All RAC deployments also employ the Oracle Universal Installer and Oracle-provided configuration tools to ensure the efficient and correct installation and configuration Oracle9i software.

Despite these commonalties, there are aspects of the deployment process for Oracle9i RAC that are unique for the Linux operating system. Many of these steps involve the installation and configuration of operating system-level

software and services. Others involve the installation and configuration of Oracle-provided software that provides services or functionality incorporated into operating systems other than Linux. As such, the successful completion of these steps depends on specific knowledge about how to modify configuration files used by Linux. It also requires the use of Linux operating system tools. Finally, it relies on the installation and configuration of Oracle-provided software that is specifically needed to deploy Oracle9i RAC for Linux.

This white paper will provide the specific information needed to avoid many potential problems and pitfalls for these Linux-specific steps when deploying Oracle9i RAC. This information will deal with four major parts of the RAC deployment process that have elements or details unique to the Linux operating system:

1. the creation and configuration of the shared public and private networks used by an Oracle9i RAC database on Linux,
2. the use of raw devices, network attached storage (NAS), or the Oracle Cluster File System (OCFS) to share storage with Oracle 9i RAC,
3. the installation and configuration of the Oracle Cluster Manager for Oracle9i RAC on Linux, and
4. the installation and configuration of Oracle9i database software and related services using the Oracle Universal Installer, Network Configuration Assistant, and Database Configuration Assistant.

This document will provide several kinds information about these aspects of Oracle9i RAC deployment on Linux. It will outline specific limitations of the Linux operating system that impact RAC deployment. It will discuss how to modify the configuration files used by the Oracle software and Linux system services listed above. It will show how to use tools provided with Oracle9i RAC and the Red Hat Linux Advanced Server 2.1 distribution to complete the installation and configuration of RAC on Linux. Finally, it will highlight areas in the RAC installation process that must be completed successfully in order to avoid errors.

The remainder of this document will be organized as follows: the next section is devoted to general cluster design principles that govern the success or failure of a Linux RAC deployment in the area of networks, storage, and software services. The third section relates these general principles to the specific techniques and steps used to install and configure the Red Hat Linux Advanced Server 2.1 distribution. The fourth section discusses aspects of Oracle9i RAC installation and configuration that relate to the principles in Section 2 and the configuration in Section 3. The fifth section is a list of related documents and manuals that provide additional background and details about the information and procedures provided here. Finally, the last section provides some brief conclusions.

DEPLOYMENT PRINCIPLES

This section describes general principles that must be considered when deploying Oracle9i RAC on the Linux operating system. As such, these principles apply when RAC is deployed with any Linux distribution. The specific steps and procedures used to implement these principles for Red Hat Linux Advanced Server 2.1 will be discussed in later sections of this document.

Private networks

The one or more private networks used for database communication between nodes are critical components of any RAC deployment on Linux. The communication carried across these networks includes node participation and state information used by the Oracle Cluster Manager. It also includes the Cache Fusion traffic that reduces the amount of disk-based I/O activity generated by a RAC database. A private network may provide a RAC database with access to the shared storage when network attached storage (or NAS, see below) is used. It is therefore imperative that the following be true when RAC database installation, configuration, and/or operation begins:

- the number of private networks between RAC nodes must be established and their respective functions clearly defined,
- these private networks must be configured, tuned, and verified as operational, and
- the names of the hosts attached to these networks must be uniformly published to every RAC node.

The number of private networks that connect nodes is an issue that deserves special attention because of how it affects RAC database installation, configuration, and performance. RAC databases do not strictly require that a private network be present because a public network can handle all network traffic between nodes. A configuration employing a single, public network can degrade database performance, however. High latency or low available bandwidth in a public network may prevent the timely delivery of data required for good RAC database performance. We therefore recommend that at least one high-speed private network (preferably one gigabit or more per second capacity) be used to carry Cluster Manager and Cache Fusion inter-node communication. An additional network of similar capacity is recommended when NAS is used because this separates Cluster Manager and Cache Fusion traffic from the network traffic used to access shared storage. It may also be beneficial to have at least one additional redundant network in place to reduce recovery time when elements of a private network fail.

Shared storage

Shared storage is another critical component of an Oracle9i RAC deployment. The storage shared by a Real Application Cluster contains database datafiles, online redo logs, spfile, and control files, a Global Services Daemon (GSD)

configuration file, and a Cluster Manager (CM) quorum file. The Oracle 9i RAC software distribution and database archived redo logs can also be placed on a shared file system, provided that unique destinations are available for certain log files that are produced while a RAC database operates (not archived redo logs). All of these files are absolutely critical to the operation and management of an Oracle9i RAC database.

Three major approaches exist for providing the shared storage needed by Oracle9i RAC:

1. directly attached raw devices can be used for database datafiles, online redo logs, spfile, and control files, as well as the CM quorum file and GSD configuration file,
2. network attached storage (NAS) can be used in a supported configuration to provide a shared repository for all RAC database files (preferably through a high speed private network),
3. one or more cluster file systems can be used to hold all RAC files on directly attached storage.

Each of these approaches has advantages and disadvantages that can be arranged using the following attributes:

- **Ease of use** – File systems and network-attached storage offer advantages over raw devices because database files can be more easily moved, renamed, copied, and resized.
- **Ease of replication** – Underlying hardware that hosts network-attached and direct attached storage may have underlying mirroring and snapshot capabilities that allow transparent data replication.
- **Greater design flexibility** – Operating systems may impose limitations on the total number of raw devices and the number of possible raw devices per disk drive; file systems and network attached storage do not suffer from these limitations and allow greater flexibility in database design.
- **Improved performance** – The lower number of software layers between Oracle processes and directly attached disk storage allows raw devices and certain clustered file systems (such as the Oracle Cluster File system described below) to offer the highest level of overall database performance.

Each of these attributes should be considered when choosing to implement a shared storage solution for a RAC database.

Once chosen, the shared storage solution must be configured. Direct attached storage may require the installation and configuration of additional hardware (such as a FibreChannel HBA) with accompanying drivers. Network attached storage requires additional network capacity; it is recommended that this capacity be provided by a separate private or storage area network (SAN). Clustered file systems, such as the Oracle Cluster File System (OCFS)

discussed in this document, may require additional drivers and software be installed, configured, and loaded in order to operate.

Operating system services

Oracle9i RAC can require a number of additional software services in order to function correctly. These services generally fall into two major groups:

1. **Access Services** – These services allow RAC instances running on each of the nodes to access the cluster shared storage and include client, server, and driver software such as the software enabling the Network File System (NFS).
2. **Cluster Services** – These services establish the identity and participation of the members of the cluster, allow them to communicate, and provide some information about their operational state.

The software products that provide these services can come from a variety of sources: Oracle software, operating system distributions and add-ons, and third party product software. This document will focus on the Oracle Cluster Manager provided in the Oracle9i RAC distribution for Linux, the NFS client software provided with the Red Hat Linux Advanced Server 2.1 distribution, and the Oracle Cluster File System driver software provided by Oracle.

OPERATING SYSTEM CONFIGURATION FOR RED HAT LINUX ADVANCED SERVER 2.1

This section discusses aspects of the configuration of the Red Hat Linux Advanced Server 2.1 distribution that are important for the successful installation and operation of Oracle9i RAC. It is not intended to be a complete, step-by-step instruction guide for operating system configuration. Rather, it is meant to highlight the Installation steps and concepts that are especially important for RAC nodes. Users are encouraged to supplement this information by reading documentation for the Red Hat Linux Advanced Server 2.1 distribution and Oracle9i RAC. All operating system configuration operations discussed here require super-user access unless explicitly stated otherwise.

Public and private network configuration

Network configuration can be performed in three ways on Red Hat Linux Advanced Server 2.1: manually (not recommended), at installation time (to provide basic hostname and IP information), and using the X Windows-based Network Configuration tool (`/usr/sbin/ncat`) after installation is complete. While some manual configuration steps will be described here, the bulk of the configuration steps will employ the automatic methods used by the Network Configuration Tool.

It is first necessary to allocate network interface cards (NICs) for the public and private networks that will be used by Oracle9i RAC. It is recommended that both public and private network addresses be static (i.e. not dynamically

allocated by a service such as DHCP.) This is because RAC software requires a regular set of host names on the public network to perform administrative tasks such as starting, stopping, and determining the status of RAC instances. Private networks for a RAC cluster do not need to be connected to the Internet and therefore do not require dynamic host name services. It is also useful at this point to stabilize the mapping of cards to networks, in order to avoid sudden shifts in network configuration when node hardware is changed.

Basic network address and hostname information for NICs can be supplied either during installation or using the Network Configuration Utility. The format of the basic information is similar:

- the Network Configuration screen will appear during installation after the choice of Boot Loader is made and before the firewall security level is selected, and will request activate-on-start, fully qualified host name (hostname and domain name), IP address, netmask, gateway, and DNS server information, or
- the Network Configuration window of the Network Configuration Utility appears, and the following information can be modified from various tabs and sub-windows:
 - 1) Click the DNS tab to set information regarding hostname, domain name, and DNS server.
 - 2) Click the Device tab, choose the device to configure, and click the Edit button to set activate-on-start in the Ethernet Device window.
 - 3) Click the Protocol tab in the Ethernet Device window (from the step above), select TCP/IP, and click the Edit button to enter the IP address and netmask information for that NIC.
 - 4) Click Apply and then click Close to exit.

The choice of public and private addresses to assign to each NIC depends on the exact configuration of the network where the RAC database will operate. The local network administrator will generally assign the addresses used by the public network. The addresses for the one or more private networks can be assigned from the following network numbers, provided that the private networks using these numbers will **NOT** be connected to the Internet:

- Class A network 10.0.0.0, netmask 255.0.0.0
- Class B networks 172.16.0.0 to 172.31.0.0, netmask 255.255.0.0.
- Class C networks 192.168.0.0 to 192.168.255.0, netmask 255.255.255.0

Connecting networks using these numbers to the Internet will result in complaints from the ISP providing the Internet connection or the domain name registrar.

Network configuration tools provided with the Red Hat Linux Advanced Server 2.1 distribution solve problems caused when Kudzu (the Red Hat Linux automated configuration tool) manages RAC node configurations with multiple NICs.

Once basic NIC identity has been established, the Network Configuration tool can also be used to perform an additional step that can greatly stabilize the network configuration of a RAC node on the Red Hat Linux Advanced Server 2.1 distribution. The Kudzu Hardware probing tool provides a very useful service but may not handle hardware reconfiguration well when multiple NICs are present; it may randomly re-assign network information to NICs when hardware configuration changes. The association of the MAC address of each card with its network configuration prevents this behavior. To do this, perform these steps:

1. Start the Network Configuration Utility.
2. Click the Device tab on the Network Configuration window.
3. Click to choose the device to configure, click Edit button.
4. In Hardware Device tab, select Use Hardware Address option, click Probe for Address, and click OK.
5. Click Apply to apply the changes and click Close to exit.

A final, critical step in the network configuration process is to publish all public and private host names to all machines in a Real Application Cluster. This can be done manually by adding entries to the `/etc/hosts` file, or by using a network service such as DNS. It is recommended that private network addresses be managed using the former method for small clusters (when not connected to the Internet) in order to avoid the complexity of setting a DNS server for a small network. This also avoids the problem of making a DNS server into a single point of failure for a database cluster. This step can be performed manually or with the Network Configuration Utility using the following techniques:

- start the Network Configuration Utility, click on the Hosts tab, and use the Add, Remove, or Edit buttons to create the necessary host entries, or
- modify the `/etc/hosts` file directly using a text editor such as `vi`.

It is recommended that both partially qualified (hostname only) and fully qualified (hostname and domain name) aliases be inserted into the `/etc/hosts` file for both methods. For example, entries for a four node cluster with a public network and two private networks might look like this:

```
# Public hostnames
172.31.149.14 tiger.fooxyz.com tiger
172.31.149.15 lion.fooxyz.com lion
172.31.149.16 leopard.fooxyz.com leopard
172.31.149.17 cheetah.fooxyz.com cheetah
# Private hostnames
192.168.1.1 tigeri.fooxyz.com tigeri
192.168.1.2 lioni.fooxyz.com lioni
192.168.1.3 leopardi.fooxyz.com leopardi
192.168.1.4 cheetahi.fooxyz.com cheetahi
```

```
# Hostnames used for accessing Netapp filer
10.0.1.11 tigerii.fooxyz.com tigerii
10.0.1.12 lionii.fooxyz.com lionii
10.0.1.13 leopardii.fooxyz.com leopardii
10.0.1.14 cheetahii.fooxyz.com cheetahii
# Public and Private Netapp filer hostnames
172.31.148.226 pluto.fooxyz.com pluto
10.0.1.10 pluto_e3 pluto_private
```

The correct operation of any public or private networks used by a RAC database should be verified prior to database installation by using the ping command.

Shared storage configuration

Three major methods exist for sharing disks with Red Hat Linux Advanced Server 2.1: raw devices, network attached storage, and cluster file systems. Unsuccessful configuration of shared storage can result in multiple database creation errors and the eventual unexpected loss of data if database creation is successful. This section provides installation and configuration tips and examples for three different shared storage configurations: raw devices, network attached storage in the form of NFS mounts used with a NetApp Filer, and the Oracle Cluster File System (OCFS).

Shared raw devices

Shared raw devices are commonly presented under Red Hat Linux Advanced Server 2.1 as a series of SCSI devices by a SCSI controller driver or Fibre Channel HBA driver. As such, shared raw devices have the following characteristics:

1. **Persistent Binding** – Some Fibre Channel adapters use an underlying communication protocol that introduces a race condition in how devices report their identity. This race condition can cause storage devices to be bound to different device names upon system start or reboot because the Linux kernel uses the order of discovery to name devices. This means that a RAC instance set to run on a given cluster node may not start after the node boots up because that instance cannot correctly locate the data files, control files and redo logs it needs. It also means that the RAC instances in a cluster may successfully start or fail in a non-deterministic manner after a cluster-wide reboot because different nodes initiate the device name binding process at slightly different times. Please carefully check manufacturer-provided documentation to determine if this behavior affects any Fibre Channel HBAs in use.
2. **Partition Limits** – The Linux operating system limits the number of SCSI disks that can be connected to a machine running an Oracle RAC instance to 128. More importantly, it also limits the total number of partitions on each disk to a maximum of 15. Of these 15 partitions per disk, only 14 can

The Oracle Cluster File System eliminates many of the limitations that make shared raw devices difficult to manage on Red Hat Linux Advanced Server 2.1.

generally be used because of the rules imposed by the fdisk labels used by the Linux operating system. This causes inflexibility when creating a RAC database on shared raw devices because each partition can only be associated with one raw device, and the number of raw devices is small when the number disks is small. Two ways to avoid partition limits are to either use a cluster file system such as the Oracle Cluster File System or use network-attached storage.

3. **Raw Device Limits** – The method currently used by the Linux operating to recognize raw devices system limits the number of raw devices that the Linux kernel can access to 255. This inevitably limits the maximum size of a RAC database and maximum number of RAC instances that can participate in a database cluster. As with partition limits, limits on the number of raw devices in use can be avoided by employing a cluster file system such the Oracle Cluster File System or network attached storage.

Configuring shared raw devices involves two steps: disk partitioning and raw device binding. The disk partitioning step must be performed once for shared storage and can easily be accomplished either during Red Hat Linux Advanced Server 2.1 installation using fdisk or the Disk Druid, or by using the fdisk utility once operating system installation is complete. The details of this process and the layout of the partitions necessary to create a RAC database are beyond the scope of this document.

Raw device binding must be performed for every RAC node that uses the shared partitions. The recommended procedure for raw device binding is to create a list that associates partitions with raw devices and use a shell script that binds raw devices from this list when a RAC node is re-starts. To do this for Red Hat Linux Advanced Server 2.1, merely edit the `/etc/sysconfig/rawdevices` file on every RAC node. The format for each line in the file is the same as for the raw command itself, minus the raw command name.

Common mistakes made during the deployment of Oracle9i RAC with shared raw devices on Red Hat Linux Advanced Server 2.1 include:

- forgetting to make sure that the kernel on every RAC node has loaded a partition table from the shared disks that includes all the partitions that will be mapped to shared raw devices,
- incorrect or incomplete mapping of raw devices to partitions, and
- creating partitions of incorrect size.

These problems (none of which occur when the Oracle Cluster File System is used) can generally be solved after some careful inspection of node and shared storage configuration. The first problem can quickly be solved by examining the list of partitions found by the kernel with the following command:

```
#cat /proc/partitions
```

Careful inspection of node configuration with the `raw` and `fdisk` utilities can solve many problems caused by mistakes made during storage configuration part of deployment.

If the expected partitions are not present, reboot the affected node. The last two problems can only be solved by taking careful notes regarding the size and number of the partitions mapped to raw devices and carefully comparing those notes with information used during RAC database creation. The actual size of partitions and the state of raw device bindings can be determined using the following commands, respectively:

```
#fdisk -l  
  
and  
  
#raw -qa
```

Network attached storage using a Netapp filer

This section will present the operating system and cluster configuration recommendations needed when Oracle9i RAC is used with a Network Appliance (Netapp) filer. The configuration of the filer itself is assumed to be beyond the scope of this document. Please consult the appropriate documentation provided by Network Appliance for details about filer configuration. It is also assumed that the reader is generally familiar with how to use the Network File System (NFS) on Linux.

The general procedure for using a Netapp filer with Oracle9i RAC includes the following steps:

1. Create, configure, and export the necessary volume(s) on the Netapp filer that will be used by this cluster.
2. Create a mount point for each exported volume on all of the cluster nodes, such as `/mnt/netapp`.
3. Mount the shared storage volume(s) using the the following options (based on tests carried out when RAC is run with a Netapp filer):

```
rw,bg,hard,intr,rsize=32768,wsiz=32768, tcp, noac, vers=3
```

The `noac` and `tcp` mount options are required for Oracle9i RAC.

4. Once mounted, try touching files or creating directories to make sure that the volume is readable and writable, and that files and directories on the shared volume have the correct owner and permissions.
5. Add a line to the `/etc/fstab` file of each RAC node for each Netapp volume with the options above to ensure that the correct mount will be recreated every time that each RAC node reboots.

It is useful to test the completed configuration by rebooting one or more RAC nodes to verify that the shared volume(s) will be re-mounted after a reboot.

Use of these recommended options guarantees the highest level of availability, reliability, and performance for Oracle9i RAC deployments that use Netapp filer volumes. Use of other settings may expose weaknesses in the implementation of the NFS client on the Linux operating system.

The Oracle Cluster File System

The Oracle Cluster File System was developed by Oracle to simplify the management of RAC database data files. It does this by overcoming Raw Device Limits and Partition Limits without incurring the performance degradation (compared to raw devices) that can occur when using Network Attached Storage. Therefore, it is primarily intended for use with Oracle database data files and online redo logs rather than for use as a general-purpose file system (at present).

Installation of the Oracle Cluster File System requires a private network configured with all the recommendations suggested by this document and consists of these general steps:

1. Obtain the necessary OCFS software from Oracle at the following URL:
`http://otn.oracle.com/tech/linux/content.html`
2. Install the OCFS module in the appropriate `/lib/modules` tree in use by the operating system kernel.
3. Copy the provided OCFS utilities to the `/usr/sbin` directory.
4. Create the OCFS configuration using the provided `ocfstool` utility.
5. Create a script that will automatically run at system start to load the OCFS module and mount any file systems using the following commands:

```
#/usr/sbin/load_ocfs  
  
#/sbin/mount -a -t ocfs
```

6. Use `fdisk` to create the appropriate partitions that will be formatted with the Oracle Cluster File System.
7. Use the `ocfstool` utility to format the partitions created in the previous step.
8. Create mount points for all OCFS file systems that will be used.
9. Create a series of lines in the `/etc/fstab` file for each formatted OCFS file system similar to the following example:

```
<partition device name> <mount point name> ocfs uid=1001,gid=100
```

The numbers following the `uid` and `gid` options correspond to the user id of the `oracle` user and the group id of the `dba` group; verify that these values are correct for any RAC deployment where OCFS will be used.

As with network attached storage, it is useful to reboot one or more RAC nodes at this point to verify that the correct number of OCFS volumes re-mounted.

Kernel parameter configuration

An important step in the operating system configuration process for Red Hat Linux Advanced Server 2.1 ensures that the correct amounts of shared memory,

Use of the correct kernel parameters helps to avoid problems during the installation and configuration of all Oracle products, as well as ensuring maximum performance for production RAC deployments.

semaphore, and swap resources are allocated for Oracle9i RAC to run. This is a four-step process that must be performed for each RAC node:

1. Create a shell script called `rhas_ossetup.sh` in `/etc/init.d` that will make the allocate resources using the `/proc` file system interface (these values are examples that may require adjustment for some hardware configurations):

```
#!/bin/sh
echo "65536 " > /proc/sys/fs/file-max
echo "2147483648" > /proc/sys/kernel/shmmax
echo "4096" > /proc/sys/kernel/shmmni
echo "2097152" > /proc/sys/kernel/shmall
echo 1024 65000 > /proc/sys/net/ipv4/ip_local_port_range
ulimit -u 16384
echo "100 32000 100 100" > /proc/sys/kernel/sem
ulimit -n 65536
```

2. Create the soft links in both `/etc/rc5.d` and `/etc/rc3.d` that will cause this script to be run at the proper time on node startup with the following commands:

```
# cd /etc/rc3.d
# ln -s ../init.d/rhas_ossetup.sh S77rhas_ossetup

# cd ../rc5.d
# ln -s ../init.d/rhas_ossetup.sh S77rhas_ossetup
```

3. Reboot the machine, make sure the kernel parameters have been changed to the required values.

```
# echo /proc/sys/kernel/shmmax
# echo /proc/sys/fs/file-max
# echo /proc/sys/kernel/shmmax
# echo /proc/sys/kernel/shmmni
# echo /proc/sys/kernel/shmall
# echo /proc/sys/net/ipv4/ip_local_port_range
# ulimit
# echo /proc/sys/kernel/sem
```

4. Use the following command to determine the amount of swap space available for each RAC node and create additional swap space if it does not meet the requirement specified by the RAC Installation Guide:

```
#!/sbin/swapon -s
```

Establishing host equivalence

Finally, the Oracle Universal Installer can install Oracle9i RAC software in one of two ways:

Proper storage allocation and use of host equivalence prevents failures that occur during the final, post re-linking stages of RAC installation. This is when the Oracle Universal Installer attempts to copy files from the RAC node where it is running to the other nodes in the cluster – if local ORACLE_HOME directories exist on each node.

1. create a single, shared ORACLE_HOME directory tree on network attached storage or a cluster file system (though not OCFS at present, see below) to be used by all nodes of the cluster, or
2. create a separate, individual ORACLE_HOME directory on local storage for each and every node within the cluster.

The choice of which approach to use depends on the shared storage strategy used for the cluster and the amounts of available storage on local and shared disks. It is important to note that while the Oracle Clustered File System can currently be used to hold RAC database datafiles and redo logs, it cannot be used to hold a single, shared ORACLE_HOME directory tree for a RAC installation on Linux. This installation method will be supported in future versions of OCFS for Linux.

The successful deployment of Oracle9i RAC requires that a `oracle` user process on one RAC node be able to run commands via `rsh` and copy files with `rcp` on all other nodes in the cluster without providing a password. This is necessary to copy files in local ORACLE_HOME directories (where such directories are used), modify database instance and network configurations, and to start and stop database instances. There are several ways to allow this, and, some are more secure than others. The approach presented here is moderately secure and favors simplicity over the need to fully close potential security holes:

1. On each cluster node, use the following commands to verify that the `rsh` and `rcp` operating system services are running:

```
#chkconfig --list rlogin
```

```
#chkconfig --list rsh
```

2. Run the following commands if either or both of these services are not active:

```
#chkconfig rlogin on
```

```
#chkconfig rsh on
```

3. Add lines to the `/etc/hosts.equiv` or `/home/oracle/.rhosts` files to establish host equivalence for the `oracle` user. These lines use the following format:

```
<node private hostname> oracle
```

4. Verify the correctness of the configuration by logging into one or more nodes as the `oracle` user and attempting remote logins using `rsh` and remote copy operations using `rcp`.

ORACLE RAC INSTALLATION AND CONFIGURATION

This section will discuss general aspects of the installation process for Oracle9i RAC software. As with the preceding section on operating system

configuration, it will not provide exhaustively complete procedures; its goal is to highlight steps that may cause problems in the installation process.

Cluster Manager installation and configuration

The use of the watchdog daemon is deprecated in 9.2 versions of Oracle9i RAC in favor of a kernel module called the hangcheck-timer. The watchdog daemon must still be run with the 9.2.0.1 version of RAC, but it must be carefully configured. The watchdog daemon must not be run with the 9.2.0.2 version of Oracle9i RAC.

A key feature of the Oracle9i Cluster Manager for Linux is an associated agent that monitors system health and resets a RAC node when that node hangs. For the 9.0.1 and 9.2.0.1 versions of the Cluster Manager, this agent was called watchdogd. A kernel module described below provides higher cluster reliability and availability for the Oracle 9.2.0.2 Cluster Manager. The use of the user-space watchdog daemon with 9.2 versions of the Cluster Manager for Linux and beyond is deprecated.

In place of the watchdog daemon, a new Linux kernel module called the **hangcheck-timer** has been created to perform a similar task. It employs a kernel-based timer to periodically verify that the system task scheduler is functioning correctly and resets the node immediately when a system hang has been discovered. It has two parameters:

1. hangcheck-tick – the period of time between checks of system health by the module (recommended setting: 30 seconds), and
2. hangcheck-margin – the maximum hang delay that the module will tolerate before resetting the node during the time between ticks (recommended setting: 180 seconds).

As the hangcheck-timer is an orthogonal solution to watchdogd for helping to address problems with node availability and reliability, there is no reason that it can not be used with any 9.2 versions of Oracle9i RAC, provided that the correct Cluster Manager settings are used.

Using watchdogd with Oracle 9.2 RAC

It is not necessary to run watchdogd with the 9.2.0.2 version of Oracle RAC and any references to it should be removed or commented out in any scripts or configuration files that start the Cluster Manager (such as `ocmstart.sh` and `ocmargs.ora`). It is also not necessary to load the softdog kernel module that was used by the 9.2.0.1 watchdogd, unless it is used by other services on the RAC node.

The 9.2.0.1 version of the Cluster Manager still requires that watchdogd be run, however. Correct configuration of the 9.2.0.1 version of watchdogd requires that the softdog kernel module (or some similar module providing service to the `/dev/watchdog` device) be loaded and configured with a parameter called the `soft_margin`. The following watchdogd and Cluster Manager parameters will permit coexistence of the hangcheck-timer module and watchdogd for Oracle 9.2.0.1 RAC deployments:

```
watchdogd: -d /dev/null -l 0 -m <softdog soft_margin setting>
```

```
oracm: /a:0
```

These settings should be inserted into the `ocmargs.ora` file or whatever script or configuration file is used to start the Oracle Cluster Manager on all the RAC nodes.

Installing the hangcheck-timer module

Instructions for installing the hangcheck-timer module are included with the Oracle9i RAC 9.2.0.2 patch set. Steps dealing specifically with the hangcheck-timer kernel module in these instructions may be used to setup the hangcheck-timer with both the 9.2.0.1 and 9.2.0.2 versions of Oracle RAC. Other steps that require the alteration or removal of Cluster Manager configuration parameters are specific to Oracle 9.2.0.2 and must be ignored for 9.2.0.1 versions of RAC unless they are discussed in the next subsection.

Using the hangcheck-timer module

The use of the hangcheck-timer module requires coordination between the settings of hangcheck-tick and hangcheck-margin and the MissCount parameter of the Cluster Manager. This is done in order to ensure that the Cluster Manager does not declare an instance to be dead, force cluster reconfiguration, and start RAC node recovery until a hung node is guaranteed to be reset. The MissCount parameter for the Cluster Manager (in the `cmcfg.ora` file) must be set using the following formula:

$$\text{MissCount} > \text{hangcheck-tick} + \text{hangcheck-margin}.$$

The default value of the MissCount parameter is a minimum of 210 seconds using the hangcheck-timer parameters recommended in this document.

Installing and starting the Oracle Cluster Manager

The Oracle Cluster Manager is installed using the Oracle Universal Installer and requires that the public and private hostnames for each node in the cluster be entered on two separate screens during the installation process. Careful attention must be paid during Cluster Manager installation to make sure that the lists of public and private node names exactly correspond (e.g. the first public node that is entered must be the first private node entered and vice versa.) Failure to do so will cause errors in subsequent steps in the RAC software installation process.

At this point, it is necessary to create several directories used by Oracle software before the Cluster Manager in order to be sure that they will be properly replicated across the RAC deployment. The Cluster Manager requires the use of a log directory that is not properly created or replicated across the Real Application Cluster during its installation by the Oracle Universal Installer. Likewise, several log directories used by other Oracle tools and services will not be replicated properly to other nodes in the cluster during the installation of RAC database software if they are not created before installation begins.

The Oracle Universal Installer requests public and private network node names for all nodes in a Real Application Cluster. There must be an exact correspondence between the public and private node name information provided.

The successful start of the Oracle Cluster Manager and installation other Oracle tools requires the manual creation of several directories to hold log files and other information between runs of the Oracle Universal Installer.

The following procedure will create eight log directories for use by the Cluster Manager, the SQL*Net listener, the Oracle Intelligent Agent, and the RAC database. The following command will create the Cluster Manager log directory:

```
$ mkdir -p $ORACLE_HOME/oracm/log
```

These two commands will create the necessary directories for the SQL*Net listener:

```
$ mkdir -p $ORACLE_HOME/network/log
```

```
$ mkdir -p $ORACLE_HOME/network/trace
```

The directories created by these commands will be used by database instances:

```
$ mkdir -p $ORACLE_HOME/rdbms/log
```

```
$ mkdir -p $ORACLE_HOME/rdbms/audit
```

Finally, these commands create directories used by the Oracle Intelligent Agent:

```
$ mkdir -p $ORACLE_HOME/network/agent/log
```

```
$ mkdir -p $ORACLE_HOME/network/agent/reco
```

It is now possible to start the Oracle Cluster Manager and proceed with RAC database installation.

Oracle 9.2.0.1 with Real Application Cluster installation

The installation of Oracle 9.2.0.1 includes the following series of general steps with the associated hints and caveats:

1. The Oracle Universal Installer solicits information concerning the Oracle products to be installed, the locations of components, and the identity of privileged operating system groups. It installs software into the designated ORACLE_HOME location and produces a `root.sh` shell script that must be run on every node.
 - The `root.sh` generated by the Oracle Universal Installer must be run exactly according to instructions provided during RAC installation. The script must be run on all nodes of the cluster. If the script is not present on all nodes of the cluster, a host equivalence problem exists within the cluster. Exit the installation, check host equivalence, and begin again.
 - The behavior of the Oracle Universal Installer indicates whether it is attempting to install single instance Oracle9i database software or RAC software. Normally, the *Cluster Node Selection* screen appears after clicking *Next* on the OUI *Welcome* screen when RAC software is being installed. The *File Locations* screen will appear instead when the Oracle Cluster Manager is not functioning properly. The installation must be aborted by clicking *Exit* and responding *Yes* in the popup window when

It may be necessary to manually run configuration tools to assist in network, cluster, and database configuration if the sequence started by the Oracle Universal Installer should fail. Careful attention should also be paid to where and how the `root.sh` script generated by the Oracle Universal Installer is run.

Careful observation of the behavior of many Oracle-provided installation and configuration tools can prevent errors in RAC deployment. This especially includes problems that occur when the Oracle Cluster Manager is not running properly; tools such as the Oracle Universal Installer and Network Configuration Assistant generally have a *Node Selection* screen that appears when the Cluster Manager is operating correctly.

the latter case occurs. Verify that the Cluster Manager is running once the installation is aborted with the following command:

```
# ps ax | grep oracm
```

Restart the Cluster Manager or repair its configuration based on whether or not any Cluster Manager processes are found. Log information for the Cluster Manager can be found in the `ORACLE_HOME/oracm/log` directory.

The Net Configuration Assistant and other Oracle-provided tools exhibit similar behavior. The method for resolving the problem is identical for all tools.

- Use of a single, shared `ORACLE_HOME` directory by an entire cluster requires that additional configuration steps be taken during the pause provided to run the `root.sh` script. These steps are necessary because each RAC database instance in the cluster expects to have sole access to several configuration and log directories. These directories include:

```
$ORACLE_HOME/network/admin
```

```
$ORACLE_HOME/network/agent
```

```
$ORACLE_HOME/network/log
```

```
$ORACLE_HOME/rdbms/dbs
```

One approach to dealing with this problem is to make separate copies of these directory trees and carefully use of soft links to ensure that a common global directory name points to different local directories on each node. This can be done in three steps using the following commands:

1. For two nodes called `node1` and `node2`, create the necessary local directories and delete the original common directory with the following commands:

```
cp -R $ORACLE_HOME/network/admin \
$ORACLE_HOME/network/admin_node1
```

```
cp -R $ORACLE_HOME/network/admin \
$ORACLE_HOME/network/admin_node2
```

```
rm -rf $ORACLE_HOME/network/admin
```

2. Create identically named symbolic links on the local disks of each RAC node that point to the newly created directories from the previous step. The following command is used on `node1`:

```
ln -s $ORACLE_HOME/network/admin_node1 \
/var/opt/oracle/links/netadmin
```

Likewise, a similar command is used for `node2`:

Careful use of symbolic links solves special problems associated with installing RAC with a single, shared `ORACLE_HOME` directory that arise from the fact that each RAC instance expects to be the sole user of certain common directories.

```
ln -s $ORACLE_HOME/network/admin_node2 \  
/var/opt/oracle/links/netadmin
```

3. Create a pointer in the place of the deleted global directory that points to the link created in the previous step:

```
ln -s /var/opt/oracle/links/netadmin \  
$ORACLE_HOME/network/admin
```

Although the Oracle Cluster File System does not currently support single, shared ORACLE_HOME directory RAC deployments, it will include features that make this procedure unnecessary. These features will allow the Oracle Universal Installer to create groups of directories on a shared file system that will be unique to each node but have an identical absolute path.

2. The *Configuration Tools* window appears and the Cluster Configuration Assistant, the Net Configuration Assistant, and the Agent Configuration Assistant run.

- After the Cluster Configuration Assistant is done, go to a terminal window, log in to the RAC node as the `oracle` user and execute the following command to check on the status of the Oracle Global Services Daemon (gsd):

```
$ gsdctl stat
```

The following message will appear if the gsd is running properly:

```
GSD is running on the local node
```

If the Cluster Configuration Assistant is unable to configure and start up the Oracle gsd on the cluster, stop and cancel all remaining configuration tools and exit from OUI. The following command can manually configure the gsd from one node:

```
$ srvconfig -init
```

and started by running this command from all the nodes:

```
$gsdctl start
```

The remaining configuration tools can then be run manually. The Net Configuration Assistant can be started with the following command:

```
$ $ORACLE_HOME/bin/netca
```

Next, the Intelligent Agent can be started by performing the following three steps:

1. Verify that the `jobout`, `log`, and `reco` subdirectories are present in the `ORACLE_HOME/network/agent` directory. If they are not, create them.

The Oracle Global Services Daemon must be running for network configuration, agent configuration, and database creation to proceed properly. Manually verify that the gsd is operating before other configuration and creation steps are executed.

2. If this is the first attempt to start the Intelligent Agent and the ORACLE_HOME directory used during installation is not used by another database that is registered in an Oracle Enterprise Manager (OEM) repository, run the following commands:

```
$ cd $ORACLE_HOME/network/agent
$ rm -f *.q services.ora db snmp.ver
```

If one or more databases using the same ORACLE_HOME directory have been registered with an OEM repository, delete the databases from any repositories where they are registered, and then run the two commands listed above.

3. Run the following command:

```
$ agentctl start
```

3. Once Oracle Net and Agent configuration are complete, a RAC database may be created using the Database Creation Assistant.

- The Database Creation Assistant should be started with additional flags when a RAC database is created on a shared file system using either network-attached storage or the Oracle Cluster File System. These flags are:

```
$ORACLE_HOME/bin/dbca -datafileDestination \  
<absolute path of shared data file destination>
```

This will ensure that the Database Configuration Assistant uses a shared file system for data files.

Additional manual configuration

The following post installation steps can be performed once a database has been created to verify the success of the installation and to help automate its operation:

1. Login as the oracle user and use the srvctl command to verify that all database instances are running:

```
$srvctl status database -d <database name>
```

2. Connect to a remote instance using SQL*Plus to confirm connectivity. Be careful to set the ORACLE_SID environment variable properly.
3. Create scripts to automatically start the database at boot time on all nodes.

RELATED WORK

Useful additional information can be found in several Oracle guides, Oracle white papers, and partner white papers. The [Oracle9i Installation Guide Release 2 \(9.0.2.1.0\) for UNIX Systems: AIX-Based Systems, Compaq Tru64 Unix, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris](#) provides detailed

information regarding the installation and configuration of RAC on Linux. The [Oracle9i Administrator's Guide Release 2 \(9.0.2.1.0\) for UNIX Systems: AIX-Based Systems, Compaq Tru64 Unix, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris](#) discusses methods for tuning database performance on Linux. The Patch Set Notes for the [Oracle9i Release 2 Database Server Patch Set 1 with Cluster Manager Patch for Linux-32](#) discusses the installation and configuration of the hangcheck-timer module, and the corresponding configuration changes that must be made to the Oracle Cluster Manager. An Oracle white paper, [Tips and Techniques: Install and Configure Oracle9i on Red Hat Linux Advanced Server](#), provides information about kernel settings and virtual memory considerations that apply to all Oracle databases running on Red Hat Linux Advanced Server 2.1, including RAC databases. An additional Oracle white paper, [Linux Virtual Memory in Red Hat Advanced Server 2.1 and Oracle's Memory Usage Characteristics](#) discusses technical innovations made to the Linux kernels used by the Advanced Server 2.1 distributions, and how Oracle databases consume memory. A white paper prepared in conjunction with Network Appliance, [Oracle9i RAC with Network Appliance Filer on Red Hat Linux Advanced Server 2.1](#) presents additional details regarding the configuration of a Netapp filer for use with Oracle RAC. All of these documents can be found at the Oracle Technical Network website.

CONCLUSION

This paper described the installation and configuration of Oracle9i Real Application Clusters (RAC) on Red Hat Linux Advanced Server 2.1. The goal is to provide information about operating system-specific aspects of the deployment process for Oracle9i RAC on Linux in order to prevent problems caused by incorrect installation and configuration. This was achieved by discussing concepts and procedures related to the allocation of public and private network resources, the use of different shared storage strategies, the installation of Oracle Cluster Manager software, and the use of tools like the Oracle Universal Installer on the Linux operating system.

**Tips for Installing and Configuring Oracle9i Real Application Clusters on Red Hat Linux Advanced Server
February 2003**

Author: Ted Haining and Yuanjiang Ou

Contributing Authors: Marcos Matsunaga and John So

**Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.**

**Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com**

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

**Copyright © 2002 Oracle Corporation
All rights reserved.**