Red Hat Enterprise Linux 5

Configuration Example - Oracle HA on Cluster Suite

Configuring Oracle for High Availability (HA) on Red Hat Cluster Suite

Red Hat Enterprise Linux 5 Configuration Example - Oracle HA on Cluster Suite
Configuring Oracle for High Availability (HA) on Red Hat Cluster Suite
Edition 1

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701

This manual provides a step-by-step installation of Oracle for High Availability (HA) using the Red Hat Advanced Platform product, Cluster Suite. This manual presents both the simple RDBMS Enterprise Edition failover scenario, as well at Oracle RDBMS Real Applications Clusters (RAC) on Shared GFS filesystems. A sample 2-node cluster is provided for both installation types, and incorporates best practices that are both common and specific to the chosen method of Cluster Suite HA.

Introduction

1. About This Guide

This manual provides a step-by-step installation of Oracle for High Availability (HA) using the Red Hat Advanced Platform product, Cluster Suite. This manual presents both the simple RDBMS Enterprise Edition failover scenario, as well at Oracle RDBMS Real Applications Clusters (RAC) on Shared GFS file systems. A sample 2-node cluster is provided for both installation types, and incorporates best practices that are both common and specific to the chosen method of Cluster Suite HA.

2. Audience

This book is intended to be used by system administrators managing systems running the Linux operating system. It requires familiarity with Red Hat Enterprise Linux 5, Red Hat Cluster Suite, GFS file systems, Oracle Enterprise Edition HA for Cluster Suite, and Oracle Real Aplication Clusters for Cluster Suite.

3. Related Documentation

This manual is intended to be a standalone Install Guide, so it should not be necessary to seek out other manuals, unless further information is required to research configuration customization or advanced topics. Notes and Tips throughout the document provide some insight into why certain decisions were made for this guide. Much of that rationale is found in these reference documents, which provide further information on administering Red Hat Cluster Suite:

- *Red Hat Cluster Suite Overview* — Provides a high level overview of the Red Hat Cluster Suite.
- *Logical Volume Manager Administration* — Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- *Global File System: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS (Red Hat Global File System).
- *Using Device-Mapper Multipath* — Provides information about using the Device-Mapper Multipath feature of Red Hat Enterprise Linux 5.
- *Red Hat Cluster Suite Release Notes* — Provides information about the current release of Red Hat Cluster Suite.

Red Hat Cluster Suite documentation and other Red Hat documents are available in HTML, PDF, and RPM versions on the Red Hat Enterprise Linux Documentation CD and online at *http://www.redhat.com/docs/*.

4. Feedback

If you spot a typo, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (*http://bugzilla.redhat.com/bugzilla/*) against the component **`Documentation-cluster`**.

Be sure to mention the manual's identifier:

```
Bugzilla component: TBD
```

```
Book identifier: TBD(EN)-5 (2010-07-23T15:20)
```

By mentioning this manual's identifier, we know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

5. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

5.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F1`** to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

[1] https://fedorahosted.org/liberation-fonts/

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

5.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

5.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

Overview

This manual provides a step-by-step installation of Oracle for High Availability (HA) using the Red Hat Advanced Platform product, Cluster Suite. This manual provides installation instructions for the following two scenarios:

- Simple RDBMS Enterprise Edition failover
- Oracle RDMBS Real Applications Cluster (RAC) on shared GFS file systems

A sample two-node cluster is provided for both installation types. Each installation incorporates best practices that are both common and specific to the chosen method of Red Hat Cluster Suite HA.

The remainder of this chapter describes the components of the sample installation configurations and provides general overviews of the configuration issues an Oracle HA installation must address. It is organized as follows:

Note

Installing Oracle for use with Red Hat Cluster Suite HA is complex and requires close collaboration across the entire IT organization, including development when RAC is deployed. HA computing is a single platform that must span these departments successfully, in order to achieve the intended reliability. The quality of this collaboration cannot be under-estimated.

1.1. Oracle Enterprise Edition HA Components

The first installation scenario this document describes requires Oracle Enterprise Edition HA for Red Hat Cluster Suite. The second installation scenario this document describes requires the Real Application Clusters (RAC) option of Oracle Enterprise edition. The following sections summarize these components and their certification requirements.

1.1.1. Oracle Enterprise Edition HA for Red Hat Cluster Suite

Oracle has supported a simple, exclusive failover, since Oracle7. Customers familiar with HP’s Serviceguard will recognize this Red Hat Cluster Suite HA configuration.

In this configuration, there are two servers that are licensed to run Oracle Enterprise Edition, but only one server may access the database at any given time. Oracle refers to this as single-instance, non-shared operation. Red Hat Cluster Suite ensures isomorphic, or mutually exclusive operation of these two servers. If both servers access the database simultaneously, corruption may result. Red Hat Cluster Suite is responsible for ensuring this does not happen. The Enterprise Edition HA failover case will assume the file system is ext3, but others are supported.

There are no specific certification requirements for combinations of Red Hat Cluster Red Hat Cluster, RHEL file systems and Oracle Enterprise Edition HA. Oracle supports any *certified, non-local* file system that is supported by Red Hat Cluster Suite. For more information on Oracle HA on Red Hat Cluster Suite, see the kbase article “Red Hat Support for Oracle Enterprise Edition and Cold Failover Cluster Suite configurations”: http://kbase.redhat.com/faq/docs/DOC-21631.

1.1.2. Oracle Real Application Clusters for Red Hat Cluster Suite and GFS

Oracle Enterprise Edition has a separately priced option called Real Application Clusters (RAC), and this does provide for shared access, or multi-instance, shared operation. Red Hat Cluster Suite Oracle RAC is certified only for use with GFS shared volumes.

Although Oracle RAC supports more than eight nodes, most customer deployments are typically four to eight nodes. The mechanics of a multi-node RAC installation can be demonstrated with the same two-node cluster that can be used for Enterprise Edition HA. This provides an equivalent configuration for comparison and to determine which option is best for your requirements.

Oracle RAC has very specific certification requirements that include a *minimal* update release level of RHEL and a patchset specific version of the Oracle RDBMS RAC kernel. Certified configurations of Oracle RAC with GFS can be found in the Oracle Support document ID 329530.1.

In the RAC configuration described in this document, there are two servers licensed to run Oracle Enterprise Edition simultaneously. This is referred to as *shared disk architecture*. The database files, online redo logs, and control files for the database must be accessible to each node in the cluster. Red Hat Cluster Suite and Oracle Clusterware work together to ensure the health of the cluster is optimal.

1.2. Sample Two-Node Cluster

The sample two-node cluster that will be used for this configuration is a simple cluster that can be configured for either of the two install types. Tips and Notes will be provided to help with the process of customizing the install to a particular set of business requirements.

Figure 1.1, “Sample Two-Node Oracle Cluster” shows a generalized overview of the configuration this installation yields. In this configuration, there are two nodes, each with a fencing agent and each connected to shared storage. A quorum disk has been configured. There is also an application tier network that accesses the nodes.

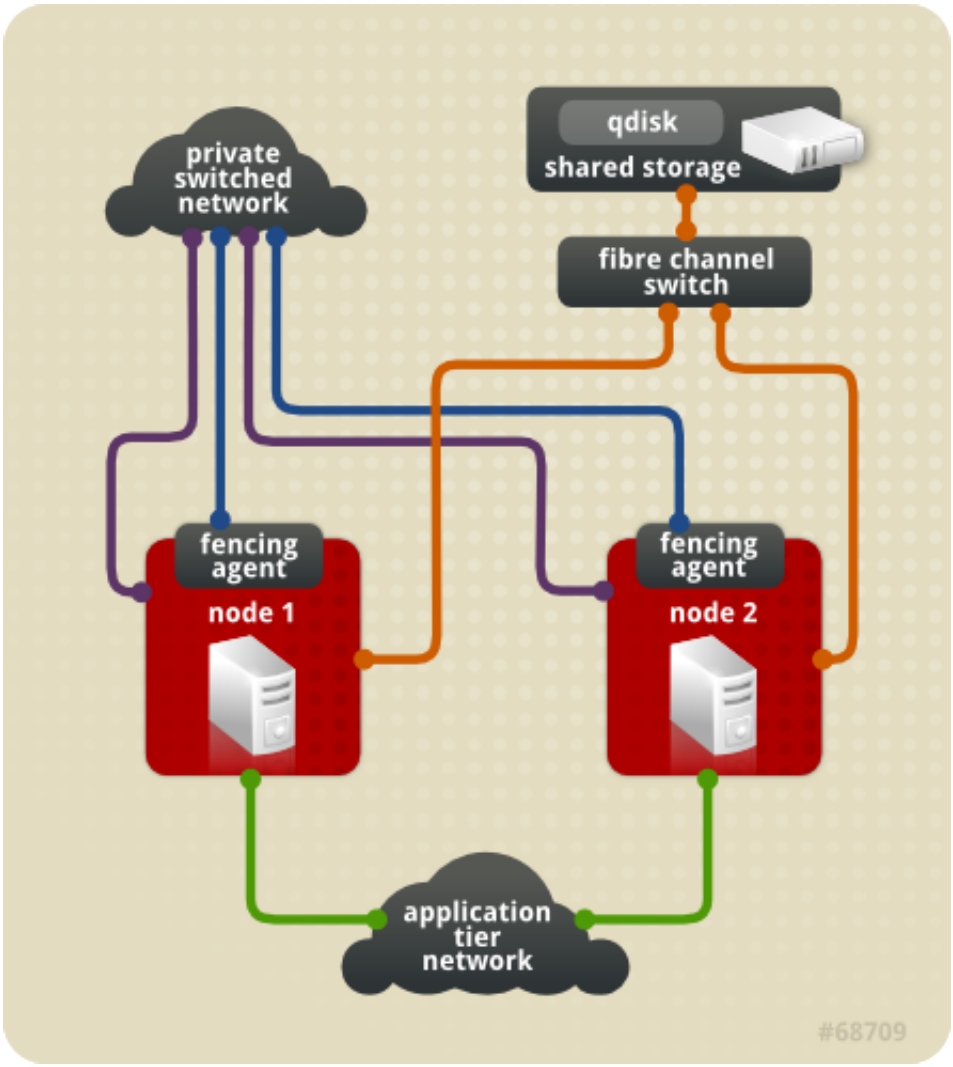

Figure 1.1. Sample Two-Node Oracle Cluster

Figure 1.2, "Cluster Node Connections" shows a generalized summary of the connections for each node in the configuration. Each node is connected to a public network, to a private network, and to shared storage. In addition, each node is configured with a fencing device that is also connected to the private network.

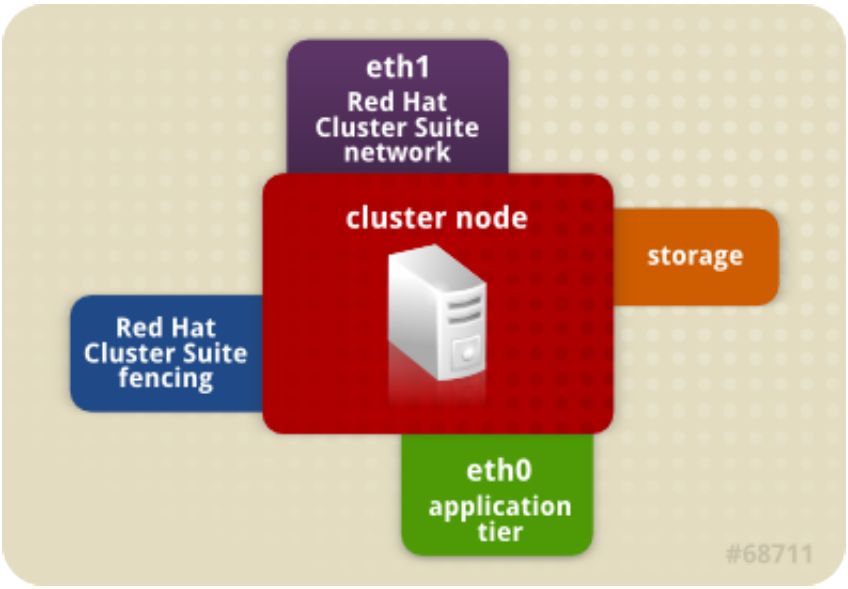

Figure 1.2. Cluster Node Connections

Asymmetrical RAC topologies

RAC clusters are often configured to be symmetrical; the type of workload presented to the nodes is similar. In this topology, the servers are also of the same relative computing strength. Typically, the servers are over-configured by 50% in order for the failover node to handle the work of both nodes. However, this assumes that the business requirements for degraded operation are identical to normal operation, which is not always the case.

An alternative configuration is to build asymmetric topologies. In our simple two-node case, one node might be used to update the database (ETL - Extract, Transform and Load), and the other node may be used to run queries. Some nodes could be significantly larger in order to be dedicated to just one form of processing (e.g., Parallel Queries). Oracle RAC is not universally transparent to SQL workloads; awareness of when and where writes occur (SQL inserts, updates and deletes) can dramatically improve scalability, even in the two-node case.

Asymmetrical RAC topologies do have implications for failover, as a set of nodes might be tuned for queries and now must handle other work on behalf of the failed node. This topology is more common with higher node counts.

1.3. Storage Considerations

A database does only a couple of things: it reads a lot of data and it writes a lot of data. It produces and consumes I/O, and with few exceptions most of those I/O operations are in the form of small, random reads and writes. A well-tuned application (they do exist) will access most of the data in the most efficient way possible. This means extensive use of indexes and that translates into random IOPS, or I/Os per second.

Disk drives are physical media and are at the mercy of the law of physics. A disk drive (or spindle) must deliver as many IOPS as possible to make it a good candidate for database use. This usually means a high RPM, and support for SCSI. Modern SAS drives (Serial Access SCSI) replaced the SCSI bus with a cheaper, serial bus. Modern SATA (Serial ATA) replaced the ribbon cable in your PCI with a much cheaper cable. SAS drives tend to be higher RPM, support something called tagged queuing and usually have the best IOPS/spindle. However, disk drive technology changes often, so

insist on the highest IOPS/spindle/$; regardless of the technology. It is not possible to buy too many spindles.

The storage layer must absolutely preserve the persistency of the data, so the data is still there when the lights go out. Be more aware of what hardware actually fails in a typical no single point of failure configuration. Drives fail, the grid fails, power supplies fail, in that order. Most other components outlast the lifetime of the deployed cluster.

RAM disks versus Flash RAM cards

Drive technology has not kept up with CPU and memory technology, and much of this has to do with basic physics. A recent trend is the use of Flash technology in a disk form factor (Solid State Drives or SSD). The other trend is the use of large Flash RAM cards (connected by 8-16 lanes of PCI-e) to operate as a coherent, write cache, either in the storage array or somewhere between you and the physical disks. Both Flash cards and SSDs are very fast, but must be just as persistent. Since Red Hat Cluster Suite Oracle HA requires shared storage (in either case), the storage vendor tends to have both options. Either can work well for a given workload, but it is always the workload adaptability that will determine the success of these technologies (or any disk technology).

RAID technology

There seem to be more RAID options than ever before. A simple thing to remember for databases is that, on average, a 144GB 15K drive is the same speed as a 36GB 15K, so if you factor for IOPS throughput, you don’t need to worry about space.

RAID5 is often used as a speed/space compromise, but is very slow especially for random writes, which databases do a lot. Sometimes the RAID controllers can hide this effect, but not well, and not forever. Another common algorithm uses one or more parity drives (most notably Netapp and HP EVA), and this option is a much better alternative to RAID5.

For database performance, the gold standard is RAID10 (a stripe of mirrored drives), which can tolerate the loss of 50% of the spindles and keep running at full performance. It might seem like a “waste” of space, but you are purchasing IOPS/spindle/$; the size of the drive is not relevant to database performance.

Various RAID options can create extra I/O in order to maintain the persistency, so the actual numbers of IOPS available to the database (payload IOPS), tends to be less than the spindle count, as is a function of the selected RAID algorithm.

Shared or non-shared file systems tend to be blocks-based file systems that are constructed on a set of physical or logical LUNs, as in the case of Red Hat’s Logical Volume Manager (LVM), or the clustered equivalent for the shared GFS install, CLVMD. An example of a files-based file system would be the NFS file system. This guide assumes that LUNs are presented for formatting into the appropriate filesystem type for either Enterprise Edition HA, or RAC.

IOPS Math

There are 3 main factors in calculating IOPS or I/O's Per Second:

- Rotational Speed – AKA spindle speed (RPM)
- Average Latency – Time for sector being accessed to be under a r/w head
- Average Seek – Time it takes for hard drive's r/w head to position itself over the track to be read or written.

IOPS is calculated as 1/(Avg. Latency + Avg. Seek)(ms)

Total IOPS = IOPS * Total number of drives

For example, let's say we want to find the total IOPS in our storage subsystem and we have the following storage:

```
4 X 1TB 10kRPM SAS (RAID 0)
Avg. Latency = 3ms
Avg. Seek = 4.45ms
```

1(.003 + .0045)= 133 IOPS

Total IOPS = 4 * 133 IOPS = 532 IOPS

1.4. Storage Topology and DM-Multipath

Blocks-based clusters still tend to be deployed on Fiber Channel (FCP), but other technologies, such as iSCSI and FCoE (FCP transport captured into, typically, 10GbE Ethernet) also exist and are supported Oracle technologies. Device-Mapper multipath is a RHEL service that provides multiple pathways to the shared storage array. The array is usually accessed through an FCP switch that contains partitioned zones to regulate the use of the array. It is recommended that the all the FCP ports from the servers be isolated into a zone with the access ports (or HCAs) on the storage array. Although many customers use large storage arrays and then sub-divide their use amongst many consumers in the datacenter, we would not recommend this for Oracle clusters. Production Oracle databases tend to place great demands on storage arrays, and sharing them will only complicate the deployment. And complication *always* means a reduction in reliability.

In the iSCSI case, bonded NICs can be used in lieu of DM-multipath, but should be a separate set of NICs that will be used for the Red Hat Cluster Suite heartbeat service, or Oracle RAC Clusterware services.

1.5. Fencing Topology

Fencing is one of the most important features of Red Hat Cluster Suite HA. Fencing is ultimately responsible for guaranteeing the integrity of the cluster and preventing wholesale corruption of an Oracle database.

Red Hat Cluster Suite HA supports a variety of fencing technologies, but we recommend the use of fencing agents that manage the power to the server nodes. This is performed using either the server's Baseboard Management Controller (BMC), or with external Power Distribution Units (PDUs). Most IPMI-based BMCs, as well as a few popular PDUs are supported.

In this sample cluster, the nodes are HP Proliants, and the IPMI-based BMC is call iLO (Integrated Lights-Out Management).

For the certified Oracle RAC configurations, Oracle Clusterware does not support power-managed fencing and must be configured so as to not interfere with Red Hat Cluster Suite HA fencing agents.

1.6. Network Topology Overview

Up to three sets of network pathways are needed for an Oracle HA cluster to operate properly:

- The application tier, on behalf of users, must have access to the database nodes.
- The Red Hat Cluster Suite node monitoring services need access to all nodes to determine the state of the cluster.
- In the RAC case, Oracle Clusterware needs a high-speed pathway to implement Oracle Cache Fusion (or Global Cache Services – GCS).

Oracle RAC was certified with GCS running over bonded pairs of Ethernet interfaces, as was the Red Hat Cluster Suite heartbeat service to ensure coordinated cluster activity. It is optional to use bonded links for the Red Hat Cluster Suite heartbeat (or Oracle Clusterware GCS), but is highly recommended.

Note

The Oracle Cache Fusion links can run over standard UDP Ethernet (GbE links, at least), but can also run over a proprietary RDMA Infiniband network, and is fully supported by Oracle on RHEL, starting with RDBMS version 10.2.0.3. The Oracle GCS protocol functions like many other conventional cache protocols and must broadcast to all nodes when requesting a cache block. On a hardware bus, this is called Snooping. With GCS, this tends to generate geometric UDP broadcast traffic when node count exceeds eight. Most applications become the bottleneck long before, but GCS snoop traffic eventually limits the scalability of all RAC clusters. The IB/rdma feature (AKA Reliable Datagram Sockets – RDS/skgxp), mitigates this geometric growth, and makes larger node counts more practical. This effect is rarely seen in the majority of clusters, which tends to less than eight nodes.

The heartbeat networks for a cluster should be isolated into a dedicated, private VLAN that also filters for UDP broadcasting, in the RAC cluster case. All of these service networks should be a private and physically isolated as is possible. These fabrics should not be considered public, production networks. Make this clear when requesting network provisioning from NetOPS, or give them a copy of this document.

Hardware Installation and Configuration

A Cluster is a complex arrangement of bits and pieces that, once combined with the software configuration, produces a highly available platform for mission critical Oracle databases. The hardware configuration requires some knowledge of the application, or at a minimum, its expectation of performance. The goal is always to produce a reliable Red Hat Cluster Suite HA platform, but rarely at the expense of performance or scalability. Oracle uses the terms MAA or Maximum Availability Architecture, but whatever the term, optimizing a platform for availability, scalability and reliability often feels like juggling chainsaws.

2.1. Server Node

Most servers that are configured to run Oracle must provide a large amount of memory and processing power, and our sample cluster is no exception. Each node is an HP Proliant DL585, with 32GB of RAM, and multi-core processors.

The server comes standard with HP's Integrated Light Out processor management that will be used as the Red Hat Cluster Suite fencing mechanism. It also has two built-in GbE NICs. This configuration also includes an additional dual-ported GbE NIC used by Red Hat Cluster Suite and Oracle Clusterware (in the RAC install).

The local storage requirements on each server are minimal and any basic configuration will have more than adequate disk space. It is recommended that you configure the local array for reliable speed, not space (i.e., not RAID5). Oracle can produce a trace log load, especially Clusterware, which may impact cluster recovery performance.

2.2. Storage Topology

Storage layout is very workload dependent, and some rudimentary knowledge of the workload is necessary. Historically, database storage is provisioned by space, not speed. In the rare case where performance is considered, topology bandwidth (MB/sec) is used as the metric. This is the wrong performance metric for databases. All but the largest data warehouses require 1000s of IOPs to perform well. IOPS only come from high numbers of spindles that are provisioned underneath the file system.

The easiest way to configure an array for both performance and reliability is to use a RAID set size of 8-12 (depending on the RAID algorithm). Many RAID sets can be combined to produce a single large volume. It is recommended that you then use this volume and strip the LUNs off this high IOP volume to create the specific number of sized LUNS. This is often called the "block of cheese" model, where every strip independent of size has full access to the IOP capacity of this large, single volume. This is the easiest way to produce high performance LUN for a database.

Acquire as many 15K spindles as is practical or affordable. Resist the temptation to use large, low RPM drives (i.e., SATA). Resist the temptation to use drive technology (including controllers and arrays) that don't support tagged queuing (i.e., most SATA). Tagged queuing is critical to sustained high IOP rates. In the SATA world, it is called NCQ (Native Command Queuing). In the FCP/SAS world, it is called Tagged Queuing. It is usually implemented at the shelf level; insist on it.

Contrary to some detailed studies, in general a 15K 72GB drive has better performance than a 10K 300GB drive. Outer track optimizations cannot be relied upon over the lifecycle of the application, nor can they be relied upon with many storage array allocation algorithms. If you could ensure that *only*

the outer tracks were used, then larger capacity drives *should* seek less. It is difficult to buy small, high RPM drives, but they will always have the best IOP price/performance ratio.

Software, or host-based RAID is less reliable than array-based RAID, especially during reconstruction, and load balancing. Host-based RAID operations compete for resources and could compromise throughput on the database server.

Many storage topologies include FCP switch infrastructure and this can be used to isolate the I/O traffic to the array. We recommend that the storage array HCAs and the four ports of the two HBAs all be placed in one zone. For more information on HBA configuration see *Section 3.2.2, "Multipath Configuration"*.

We do not recommend the *multi-purposing* of a storage array. Many customers buy very large arrays and place multiple Oracle databases (including dev and test) all on one array. This is ill advised, and the write-back cache policies in the array (which *will* become the bottleneck) are difficult to tune. Relative to the cost of Oracle and the critical nature of most Oracle databases to their respective enterprises, the storage is free; dedicate the storage, if possible. Oracle workloads are voracious, and unpredictable consumers of arrays.

2.2.1. Storage Allocation

Red Hat Cluster Suite requires a single, 64MB LUN for quorum disk support. It is recommended that the qdisk feature be used for Oracle Cold Failover.

RAC/GFS Requirement

The qdisk feature is mandatory for RAC/GFS clusters.

RAC/GFS clusters require Oracle Clusterware to be installed, and they require five 384MB LUNS (two for registry, three for quorum). It is recommended that three Clusterware voting (quorum) disks be configured, but a single, externally (array) redundant Clusterware vote disk is fully supported.

In either the HA or RAC/GFS install, the LUNs will be used to create file systems. Oracle supports AIO and DIO for both EXT3 and GFS; this provides raw device performance. In our configuration, the performance of any given LUN is the same; the *size* of the LUN does not affect performance. However, the size of the LUN *may* affect filesystem performance if large numbers of files are placed in many directories. Most Oracle databases use a relatively low number of *datafiles* in a file system, but this is at the discretion of the DBA and is determined by the ongoing operational requirements of the database. Tablespaces consist of datafiles, and contain base tables and indexes. Tables are indexes are usually in separate tablespaces (if you are lucky) and the datafiles are usually created to be as large as possible. In some cases, tablespaces and datafiles are intentionally created small, with **`AUTOEXETEND`** disabled. This generates alerts that cause DBAs to be notified of dynamic growth requests in the database. No two shops have the same policy towards **`AUTOEXTEND`**.

Redo Logs, UNDO tablespaces and Redo Archive logs often get their own file system. Redo log file systems normally have write latency sensitivity, and can be impacted by an Archive log switch (**`ARCHIVELOG`** is usually enabled for production databases).

Tip

During a log switch, the previously closed log is copied to the archive destination, and is usually not throttled. This can impact transaction commit response times. One of the simplest ways to mitigate this effect is to place the Archive Log destination on DIO-enabled NFS mount, and the network connection be forced to 100TX. This is the easiest way to throttle archive log copies. Customers often use NFS as an archive log destination, so this can be as simple as a NIC re-configuration request.

A LUN (and subsequent file system) should be allocated for **`ORACLE_HOME`**. This file system should not contain any database files. This LUN must only hold the product home, and spare capacity for trace files. It could be as small as 8GB.

RAC/GFS Requirement

For RAC/GFS, Oracle Clusterware Home (**`ORA_CRS_HOME`**) cannot be located on a clustered GFS mount point.

Virtualized Storage

Like virtualized anything else, Oracle and virtualization tend to make very strange bedfellows. Oracle database applications are voracious consumers of hardware resources and rarely share well with other applications, and often not well even with the host OS. Oracle is a fully portable OS that is completed implemented in user space. It is best to dedicate the hardware to Oracle, and this goes for the storage array too. EMC invented “virtualized” storage years ago with the concept of busting up a single, big disk into four pieces, or Hypers. These Hypers combine in a way that create a Meta LUN. This looks like a highly efficient utilization of storage, but misses the point -- A 15K drive busted up into four pieces, does not serve four times the IOPS. If you run several instances of Oracle on a virtualized server and several copies of Oracle databases on a virtualized storage array, your life will be much harder (and very likely shorter).

2.3. Network Topology

There are only two main network pathways used by the cluster: the frontside, or *public*, and the backside, or *private*, cluster interconnect network.

Clients or application servers mostly use the public network in order to connect to the database. When a node fails, existing transactions, sessions and connections disappear and this can create an interruption in service to these connections. The decision to deploy Cold Failover or RAC/GFS depends on how fast connections and transactions must restart. Cold Failover does not preserve any state, but can still restart very quickly, without having to reconstruct, re-connect, and re-synchronize with the application. RAC provides the ability to preserve much more context about sessions and transactions. If configured properly (including the application tier), this can dramatically reduce the downtime, but it increases both cost and complexity.

The most difficult situation is with existing connections that have opened a TCP/IP socket to the database. When the database node fails, the client socket needs to be notified as soon as possible. Most JDBC drivers now use out-of-band signaling to avoid the dreaded hung socket. Connection pools within application servers must be configured correctly, so failover delay is minimized.

The backside network is a private, dedicated network that should be configured as a four-port VLAN, if a non-private switch is used.

Most customers buy dual-ported NICs, which are not as reliable as two single-ported NICs. However, bonding ports across different drivers is also not recommended (bonding a TG3 port and an e1000 port, for instance). If possible use twp outboard single-ported NICs. Servers with the same out-board ports as the built-in ports (all e1000 ports, for instance), can safely cross-bond.

Connecting the ports to two different switches may also not work in some cases, so creating a fully redundant bonding NIC pathway is harder than it should be. Since the goal of the back-side network is for heartbeat, if the NIC fails but the server is up the server is still fenced. Statistically, the cluster might fence a little more often, but that's about it.

2.4. RAC/GFS Considerations

- Oracle Clusterware implements Virtual IP routing so that target IP addresses of the failed node can be quickly taken over by the surviving node. This means new connections see little or no delay.
- In the GFS/RAC cluster, Oracle uses the back-side network to implement Oracle Global Cache Fusion (GCS) and database blocks can be moved between nodes over this link. This can place extra load on this link, and for certain workloads, a second dedicated backside network might be required.
- Bonded links using LACP (Link Aggregation Control Protocol) for higher capacity, GCS links, using multiple GbE links are supported, but not extensively tested. Customers may also run the simple, two-NIC bond in load-balance, but the recommendation is to use this for failover, especially in the two-node case.
- Oracle GCS can also be implemented over Infiniband using the Reliable Data Sockets (RDS) protocol. This provides an extremely low latency, memory-to-memory connection. This strategy is more often required in high node-count clusters, which implement data warehouses. In these larger clusters, the inter-node traffic (and GCS coherency protocol) easily exhausts the capacity of conventional GbE/udp links.
- Oracle RAC has other strategies to preserve existing sessions and transactions from the failed node (Oracle Transparent Session and Application Migration/Failover). Most customers do not implement these features. However, they are available, and near non-stop failover is possible with RAC. These features are not available in the Cold Failover configuration, so the client tier must be configured accordingly.
- Oracle RAC is quite expensive, but can provide that last 5% of uptime that might make the extra cost worth every nickel. A simple two-node Red Hat Cluster Suite Oracle Failover cluster only requires one Enterprise Edition license. The two-node RAC/GFS cluster requires two Enterprise Edition licenses and a separately priced license for RAC (and partitioning).

2.5. Fencing Configuration

Fencing is a technique used to remove a cluster member from an active cluster, as determined by loss of communication with the cluster. There are two fail-safe mechanisms in a typical Oracle HA configuration: the quorum voting disk service, **`qdisk`**, and the **`cman`** heartbeat mechanism that operates over the private, bonded network. If either node fails to "check-in" within a prescribed time, actions are taken to remove, or *fence* the node from the rest of the active cluster. Fencing is the most important job that a cluster product must do. Inconsistent, or unreliable fencing can result in corruption of the Oracle database -- it must be bulletproof.

Red Hat Cluster Suite provides more fencing technologies than either Veritas Foundation Suite, or Oracle Clusterware. The fencing methods that we recommend for use with Oracle databases, are all power-based, and have been in the Red Hat Cluster Suite for several releases. Mature, power-based fencing methods are, indeed, the foundation of any robust cluster.

Most Tier 1 server vendors provide built-in baseboard management controllers (BMC), but they are called many things (HP iLO, Dell DRAC, Sun ILOM). All BMCs provide network-signaled access to the server's power supply. When Red Hat Cluster Suite must fence a node in the cluster, the fencing process on the node that detected the fault will connect to the other nodes BMC and literally power-off the server node. This is the most discrete form of fencing, and it the mechanism we use. In this case, we use HP iLO, which comes standard on all Proliant 300 and 500 series.

Red Hat Cluster Suite also supports *levels* of fencing for those who find BMC-based fencing insufficient. Among many other available methods (such as FCP switch port disable), Red Hat Cluster Suite also supports signaled power distribution units (PDUs). PDUs are also connected to an Ethernet network, and when engaged for fencing, they cut the power to the server's power supply, much as the BMC does on-board. The need to use multi-levels can be necessary because most, if not all, BMC interfaces are single Ethernet ports. This could be a single point of failure. Most PDUs also only have 1 network interface, but combined, these two methods provide redundant power signaling.

Our example will show iLO, and how it can be combined with an APC switched PDU infrastructure.

Red Hat Cluster Suite is typically configured to access the fencing network over the private bonded fabric, but any network fabric can be subsequently configured if a dedicated (and likely bonded) network is dedicated just to the fencing network. Our example will access the fencing network over the private, bonded network.

Software Installation and Configuration

A Cluster is a complex arrangement of bits and pieces that, once combined with the software configuration, produces a highly available platform for mission critical Oracle databases. We probably can't repeat that often enough, but complexity is public enemy #1. Clusters, by definition, are complex. When clusters are poorly configured, they completely defeat the purpose for which they were originally sought: high availability.

The software components of a cluster combine with a particular set of hardware components to often produce a unique platform that could fail because it was not fully tested in this specific configuration. This is the just the reality of the modern enterprise. Under abnormal operation conditions (when you most want the cluster most to work), it is safe to say that no two clusters are alike in their ability to produce conditions that might cause instability. Do not assume that your unique combination of hardware and software has ever existed, let alone been tested in some mythical, multi-vendor testing lab. Torture it before you put it into production.

The steps outlined in this chapter assume one node at a time, and most of the process simply replicates to each node.

3.1. RHEL Server Base

Some customers install every last package onto an Oracle database server, because that simplifies their process. Some Oracle customers have been known to hand build kernels and delete every non-essential package, with everybody else in between.

For our sanity (and we hope, yours), we install the minimum set of RPM groups that are necessary to run Red Hat Cluster Suite and Oracle Enterprise Edition.

The following shows the kickstart file for:

HP Proliant server, with iLO, Storageworks controller, outboard e1000, and Qlogic 2300 series FCP HBA.

You should take the following into account when considering what sofware components to install.

- This example is an NFS-based install. As always, no two kickstart files are the same.
- Customers often use auto-allocation, which creates a single logical volume to create partitions. It is not necessary to separate the root directories into separate mounts. A 6GB root partition is probably overkill for an Oracle node. In either install configuration, **`ORACLE_HOME`** must be installed on an external LUN. **`ORA_CRS_HOME`** (Oracle Clusterware for RAC/GFS) must be installed on a local partition on each node. The example below is from our RAC/GFS node.
- Only the groups listed below are required. All other packages and groups are included at the customer's discretion.
- SELINUX must be disabled for all releases of Oracle, except 11gR2.
- Firewalls are disabled, and not required (customer discretion).
- Deadline I/O scheduling is generally recommended, but some warehouse workloads might benefit from other algorithms.

```
device scsi cciss
```

```
device scsi qla2300

install
nfs --server=192.168.1.212 --dir=/vol/ed/jneedham/ISO/RHEL5/U3/64
reboot yes
lang en_US.UTF-8
keyboard us
network --device eth0 --bootproto=static --device=eth0 --gateway=192.168.1.1 --ip=192.168.1.1
 --ip=192.168.1.114 --nameserver=139.95.251.1 --netmask=255.255.255.0 --onboot=on
rootpw "oracleha"
authconfig --enableshadow --enablemd5
selinux --disabled
firewall --disabled --port=22:tcp
timezone --utc America/Vancouver
bootloader --location=m br --driveorder=cciss/c0d0 --append="elevator=deadline"

# P A R T I T I O N   S P E C I F I C A T I O N

part swap --fstype swap --ondisk=cciss/c0d0 --usepart=cciss/c0d0p2  --size=16384 --asprimary
part / --fstype ext3 --ondisk=cciss/c0d0 --usepart=cciss/c0d0p3     --size=6144 --asprimary
part /ee --fstype ext3 --ondisk=cciss/c0d0 --usepart=cciss/c0d0p5   --noformat --size 32768

%packages
@development-libs
@x-software-development
@core
@base
@legacy-software-development
@java
@legacy-software-support
@base-x
@development-tools
@cluster-storage
@clustering
sysstat
```

Following installation, we often disable many of the services in the **`/etc/rc3.d`** file. Most of these services are not required when the server is configured for Oracle use.

Tip

ext3 file systems that are created during an install do not have the maximum journal size. For RAC/GFS nodes, where **`ORA_CRS_HOME`** must live on this mount, we recommend that you rebuild the file system with the maximum journal size:

```
$ mke2fs –j –J size=400 /dev/cciss/cod0p5
```

Oracle Clusterware can churn a file system, so larger journals and a local RAID algorithm that favors performance will be beneficial.

3.2. Storage Topology

Once the server has all the required software installed, configuration can commence. Configuring the server node to map the external LUN will require some incremental configuration on the server, some incremental configuration on the array and then back to the server to verify that all LUNs were mapped. Since the storage pathways will be multipathed, all LUNS must be visible down both ports on the HBA before moving onto the multipath configuration.

3.2.1. HBA WWPN Mapping

Fiber Channel HBAs typically have two ports or, for extra redundancy, two single-ported HBAs are deployed. In either case, the World-Wide Port Number (WWPN) for each port must be acquired for both nodes and used to register the LUNS so the storage array will accept the connection request. Try to install the FCP ports into the server before RHEL is installed. This will insure they are configured for use, once the install is complete.

When FCP switch zones are deployed to isolate database traffic to a specific set of FCP array ports on the array, the switch will identify the ports physically, or you can also use the specific WWPN. Most storage administrators know how to do this, but this is what must happen to make sure two *copies* of the LUNS show on each server node.

The storage array typically bundles the LUNS that are reserved for this cluster into an *initiator group*, and this group list must contain all four WWPNs so that all four requesting HBA ports can see the set of LUNS.

On RHEL, the easiest place to look for the HBA WWPNs is in the **/sys** directory, but the switch often has logged the port names as well, so you can look there if you know how the HBAs are connected to the switch.

```
$ cat /sys/class/block/fc_host/host0/port_name
0x210000e08b806ba0

$ cat /sys/class/block/fc_host/host1/port_name
0x210100e08ba06ba0
```

Use the hex values from the **/sys** inquiry. Do not use the WWNN or node name. WWPNs needed to be added to the initiator group on the array, and to the appropriate zone on the switch. Once these steps are complete, reboot the server and you should see two sets of identical LUNS. You cannot proceed to the multipath configuration section until there are two identical sets.

3.2.2. Multipath Configuration

The software feature Device-Mapper Multipath (DM-Multipath) was installed as part of the kickstart and is used to provide pathway redundancy to the LUN. Configuring DM-Multipath must be the next step. Both the Red Hat Cluster Suite quorum disk and the Oracle Clusterware support disks will need to use the resulting DM-Multipath objects. Once DM-Multipath is configured, the block device entries that will be used appear in **/dev/mapper**.

The installation of DM-Multipath creates an **rc** service and a disabled **/etc/multipath.conf** file. The task in this section is to create reasonable aliases for the LUN, and also to define how failure processing is managed. The default configuration in this file is to blacklist everything, so this clause must be modified, removed, or commented out and then multipath must be restarted or refreshed. Be sure the multipah daemon is set to run at reboot. Also, reboot of the server should take place now to ensure that the duplicate sets of LUN are visible.

To create aliases for LUNs, the WWID of the scsi LUN must retrieved and used in the alias clause. The previous method for gathering WWIDs required the execution of the **scsi_id** command on each LUN.

```
$ scsi_id -g -s /block/sdc #External LUN, returns 360a98000567246716841a514137392d65
$ scsi_id -g -s /block/sdd #External LUN, returns 360a98000567246716841a502d34555579
```

The following example of a multipath configuration file shows the Red Hat Cluster Suite quorum disk and, for the RAC/GFS node, the first of three Oracle Clusterware Voting Disks. This excerpt is the stanza that identifies the WWID of the LUNS in the **multipath.conf** file.

```
multipath {
                no_path_retry               fail
                wwid                            360a98000056724671684a514137392d65
                alias                             qdisk
        }
#The following 3 are voting disks that are necessary ONLY for the RAC/GFS configuration!
multipath {
                no_path_retry               fail
                wwid                            360a98000056724671684a502d34555579
                alias                             vote1
        }
multipath {
                no_path_retry               fail
                wwid                            360a98000056724671684a502d34555578
                alias                             vote2
        }
multipath {
                no_path_retry               fail
                wwid                            360a98000056724671684a502d34555577
                alias                             vote3
        }
```

The only two parameters in the mulipath configuration file that must be changed are **path_grouping_policy** (set to **failover**) and **path_checker** (set to **tur**). Historically, the default was to **readsector0**, or **directio**, both of which create an I/O request. For voting disks on highly loaded clusters, this may cause voting “jitter”. The least invasive path checking policy is TUR (Test Unit Ready), and rarely disturbs **qdisk** or **Clusterware** voting. TUR and zone isolation both reduce voting jitter. The voting LUNS could be further isolated into their own zone, but this would require dedicated WWPN pathways; this would likely be more trouble than it is worth.

Some storage vendors will install their HBA driver and also have specific settings for the **multipath.conf** file, including procedures, defined by the **prio_callout** parameter. Check with the vendor.

The following example shows the remaining portion of the **multipath.conf** file.

```
defaults {
        user_friendly_names     yes
        udev_dir                /dev
        polling_interval        10
        selector                "round-robin 0"
        path_grouping_policy    failover
        getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
        prio_callout            /bin/true
        path_checker            tur
        rr_min_io               100
        rr_weight               priorities
        failback                immediate
        no_path_retry           fail
        user_friendly_name      yes
}
```

Now that the **multipath.conf** file is complete, try restarting the **multipath** service.

```
$ service multipathd restart
$ tail -f /var/log/messages #Should see aliases listed
$ chkconfig multipathd on
```

Customers who want to push the envelope to have both performance and reliability might be surprised to find that **multibus** is slower than **failover** in certain situations.

Aside from tweaking for things like **failback** or a faster **polling_interval**, the bulk of the recovery latency is in the cluster take-over at the cluster and Oracle recover layers. If high-speed takeover is a critical requirement, then consider using RAC

RAC/GFS Considerations

Because RAC (and therefore Clusterware) is certified for use with Red Hat Cluster Suite, customers may chose a third configuration option of using either OCFS2 or ASM. This is an unusual configuration, but this permits RAC/asm use, combined with the superior fencing of Red Hat Cluster Suite. This configuration is *not* covered in this manual.

3.2.3. qdisk Configuration

A successful DM-Multipath configuration should produce a set of identifiable inodes in the **/dev/mapper** directory. The **/dev/mapper/qdisk** inode will need to be initialized and enabled as a service This is the one of the first pieces of info you need for the **/etc/cluster.conf** file.

```
$ mkqdisk –l HA585 –c /dev/mapper/qdisk
```

By convention, the label is the same name as the cluster; in this case, HA_585. The section of the **cluster.conf** file looks like the following.

```
<?xml version="1.0"?>
<cluster config_version="1" name="HA585">
  <fence_daemon post_fail_delay="0" post_join_delay="3"/>
        <quorumd interval="7" device="/dev/mapper/qdisk" tko="9" votes="3" log_level="5"/>
```

Tip

You may need to change the maximum journal size for a partition. The following procedure provides an example of changing the maximum journal size of an existing partition to 400MB.

```
tune2fs -l /dev/mapper/vg1-oracle |grep -i "journal inode"
debugfs -R "stat <8>" /dev/mapper/vg1-oracle 2>&1 | awk '/Size: /{print $6}
tune2fs -O ^has_journal /dev/mapper/vg1-oracle
tune2fs -J size=400 /dev/mapper/vg1-oracle
```

Warning

Fencing in two-node clusters is more prone to fence and quorum race conditions than fencing in clusters with three or more nodes. If node 1 can no longer communicate with node 2, then which node is actually the odd man out? Most of these races are resolved by the quorum disk, which is why it is important for the HA case, and mandatory for RAC/GFS.

RAC/GFS Requirement

Red Hat Cluster Suite *must* be implemented with qdisk, or the configuration is unsupported. Red Hat Cluster Suite has to retain quorum to support a single, surviving RAC node. This single-node operation is required for certified combinations of RAC/GFS.

3.3. Network Topology

A cluster's network is either complicated, or *really* complicated. The basic cluster involves several sets of logical network pathways. Some of these share physical interfaces, and some require dedicated physical interfaces and VLANs, depending on the degree of robustness required. This example is based on a topology that Red Hat uses to certify Oracle RAC/GFS, but is also suitable for the HA configuration.

Tip

Cluster networks require several VLANs and multiple address assignments across those VLANs. If bonds are going to span VLANs or switches, then it might be required to use **ARP** to ensure the correct behavior in the event of a link failure.

3.3.1. Public Network

The public network is the pathway used by the application tier to access the database. The failure scenario is the loss of an entire node, so although bonding does provide protection in the event of the public interface failure, this is not as likely. Bonded public interfaces complicate application tier network configuration and failover sequencing. This network is not bonded in our example.

The hostnames of the server nodes are identified by the public address. All other network interfaces are private, but they still may need addresses assigned by network operations.

RAC/GFS Considerations

Oracle Clusterware (CRS) creates it's own set of Virtual IPs (VIP) on the public interface. This mechanism makes it possible for CRS on another node to provide continued access to the failed node's specific public address. Bonded public interfaces, in the presence of CRS VIPs, are not recommended. See Oracle SQL*Net Configuration in both the HA and RAC/GFS Chapters.

3.3.2. Red Hat Cluster Suite Network

The Red Hat Cluster Suite network is used by CMAN to monitor and manage the health of the cluster. This network is critical to the proper functioning of the cluster and is the pathway that is bonded most often.

RAC/GFS Considerations

RAC requires GFS clustered file systems, which utilize the Distributed Lock Manager (DLM) to provide access to GFS. The Oracle Global Cache Services (GCS) is often configured to use this pathway as well. There is a risk of overloading this network, but that is very workload dependent. An advanced administrator may also choose to use Infiniband and Reliable Data Sockets (RDS) to implement GCS.

The network is private, and only ever used by cluster members. The dual-ported e1000 NIC is used for the Red Hat Cluster Suite hearbeat service or Oracle RAC Clusterware services.

The file **`/etc/modprobe.conf`** contains all four interfaces, and the two ports of the e1000 will be bonded together. The options for bond0 set the bond for failover (not load balance), and the sampling interval is 100ms. Once the file **`modprobe.conf`** file is modified, either remove and reload the e1000 kernel module, or the modification will take effect at the next reboot.

```
alias eth0 tg3
alias eth1 tg3
alias eth2 e1000
alias eth3 e1000
alias bond0 bonding
options bond0 mode=1 miimon=100
```

The configuration of the bond requires three network-scripts files: One for bond0, and then the corresponding interface files have to be set as well, as shown in the following example.

```
ifcfg-eth2

# Intel Corporation 82546GB Gigabit Ethernet Controller
DEVICE=eth2
HWADDR=00:04:23:D4:88:BE
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
TYPE=Ethernet
ONBOOT=no

ifcfg-eth3

# Intel Corporation 82546GB Gigabit Ethernet Controller
DEVICE=eth3
HWADDR=00:04:23:D4:88:BF
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
TYPE=Ethernet
ONBOOT=no

ifcfg-bond0

DEVICE=bond0
```

```
IPADDR=192.168.2.162
NETMASK=255.255.255.0
NETWORK=192.168.2.0
BROADCAST=192.168.2.255
BOOTPROTO=none
TYPE=Ethernet
ONBOOT=yes
```

3.3.3. Fencing Network

When Red Hat Cluster Suite has determined that a cluster node must be removed from the active cluster, it will need to fence this node. The methods used in this cluster are both power-managed. The HP iLO BMC has one Ethernet port, which must be configured, and this information must exactly match the fencing clauses in the **/etc/cluster.conf** file. Most IPMI-based interfaces only have one network interface, which may prove to be a single point of failure for the fencing mechanism. A unique feature of Red Hat Cluster Suite is the ability to nest fence domains to provide an alternative fence method, in case the BMC pathway fails. A switched Power Distribution Unit (PDU) can be configured (and it frequently has only one port). We do not recommend the use of FCP port fencing, nor T.10 SCSI reservations fence agent for mission critical database applications. The address and user/password must also be correct in the **/etc/cluster/conf** file.

```
<fencedevices>
  <fencedevice agent="fence_ilo" hostname="192.168.1.7" login="rac" name="jLO7"
 passwd="jeff99"/>
  <fencedevice agent="fence_ilo" hostname="192.168.1.8" login=”rac” name="jLO8"
 passwd="jeff99"/>
</fencedevices>
```

Note

You can test the fencing configuration manually with the **fence_node** command. Test early and often.

3.3.4. Red Hat Cluster Suite services

There are now enough hardware and software pieces in place that the **cluster.conf** file can be completed and parts of the cluster can be initialized. Red Hat Cluster Suite consists of a set of services (**cman**, **qdisk**, **fenced**) that ensure cluster integrity. The values below are from the RAC example, with HA values in comments. The timeouts are good starting points for either configuration and comments give the HA equivalent. More details on the RAC example will be provided in *Chapter 4, RAC/GFS Cluster Configuration*. More details on the HA example will be provided in *Chapter 5, Cold Failover Cluster Configuration*.

```
<cluster config_version="2" name="HA585">
  <fence_daemon post_fail_delay="0" post_join_delay="3" />
    <quorumd interval="7" device="/dev/mapper/qdisk" tko="9" votes="1" log_level="5"/>
    <cman deadnode_timeout="30" expected_nodes="7"/>
    <!-- cman deadnode_timeout="30" expected_votes=”3”/ -->
    <!-- totem token=”31000”-->
```

```
                <multicast addr="225.0.0.12"/>
    <clusternodes>
            <clusternode name="rac7-priv" nodeid="1" votes="1">
                    <multicast addr="225.0.0.12" interface="bond0"/>
                    <fence>
                            <method name="1">
                                    <device name="jLO7"/>
                            </method>
                    </fence>
    </clusternode>
    <clusternode name="rac8-priv" nodeid="2" votes="1">
            <multicast addr="225.0.0.12" interface="bond0"/>
                        <fence>
                                <method name="1">
                                        <device name="jLO8"/>
                                </method>
                        </fence>
     </clusternode>
<fencedevices>
                <fencedevice agent="fence_ilo" hostname="192.168.1.7" login="rac"
name="jLO7"
 passwd="jeff123456"/>
                <fencedevice agent="fence_ilo" hostname="192.168.1.8" login="rac"
name="jLO8"
 passwd="jeff123456"/>
</fencedevices>
```

The cluster node names **rac7-priv** and **rac8-priv** need to be resolved and therefore are included in all nodes' **/etc/hosts** file:

```
192.168.1.7          rac7-priv.example.com          rac7-priv
192.168.1.8          rac8-priv.example.com          rac8-priv
```

Tip

When doing initial testing, set the **init** level to 2 in the **/etc/inittab** file, to aid node testing. If the configuration is broken and the node reboots back into **init 3**, the startup will hang, and this impedes debugging. Open a window and tail the **/var/log/messages** file to track your progress.

The **qdiskd** service is the first service to start and is responsible for parsing the **cluster.conf** file. Any errors will appear in the **/var/log/messages** file and **qdiskd** will exit. If **qdiskd** starts up, then **cman** should be started next.

Assuming no glitches in configuration (consider yourself talented, if the node enters the cluster on first attempt) we can now ensure that the **qdisk** and **cman** services will start on boot:

```
$ sudo chkconfig –level 3 qdiskd on
$ sudo chkconfig –level 3 cman on
```

At this point, we should shut down all services on this node and repeat the steps in this chapter for our second node. You can copy the **multipath.conf** and **cluster.conf** configuration files to the second node to make things easier. Now the configuration process diverges to the point that further configuration is very RAC/GFS or HA specific. For information on configuring a RAC/GFS cluster,

continue with *Chapter 4, RAC/GFS Cluster Configuration*. For information on configuring cold failover HA cluster, continue with *Chapter 5, Cold Failover Cluster Configuration*.

RAC/GFS Cluster Configuration

This chapter provides information on a configuring RAC/GFS cluster. For information on configuring a cold failover HA cluster, see *Chapter 5, Cold Failover Cluster Configuration*.

Preparing a cluster for RAC requires additional package installation and configuration. Deploying Oracle RAC on a certified GFS cluster requires additional software and configuration. The aim of this section is to demonstrate these scenarios.

Oracle RAC is a shared-disk option of Enterprise Edition that requires another Oracle product (Clusterware) to be installed as well. This complicates the Red Hat Cluster Suite install, as there are now 2 independent clustering layers running *simultaneously* on the cluster. Oracle requires that Clusterware (CRS) be installed on top of Red Hat Cluster Suite, and this will be the chapter's focus. The chapter assumes that the user can install CRS (as well as the RDBMS).

All Oracle database files can reside on GFS clustered volumes, except Oracle Clusterware product files (**`ORA_CRS_HOME`**). The Oracle RDBMS product files (**`ORACLE_HOME`**) can be installed on shared GFS volumes, although Context Dependent Pathnames (CDPN) will be required for some **`ORACLE_HOME`** directories.

4.1. Oracle Clusterware

Oracle Clusterware is a stand-alone cluster layer that Oracle provides for use with the RAC option. CRS mimics all the functionality of Red Hat Cluster Suite, but must be tuned so as to not interfere with Red Hat Cluster Suite's ability to manage the cluster (and the GFS clustered file systems).

CRS requires a set of dedicated LUNs (that were allocated and configured for use with Multipath). Starting with 11gR1, the helper LUNS no longer need to be *raw devices*, but can be standard block devices. The inodes in the **`/dev/mapper`** file can now be used directly for the CRS Cluster Registry (OCR) and quorum (VOTE) files.

Oracle CRS installation permits *external redundancy* and *internal redundancy*. The external option assumes the storage array is responsible for their protection. In this installation option, only one copy of OCR and one copy of VOTE are allocated. In the internal redundancy configuration, Oracle creates two OCR files, organized as a simple RAID1 mirror, and generates three quorum VOTE files. The number of VOTE files can be higher, providing it is a prime number of files. Most installations choose three VOTE files, and most installations choose internal redundancy. CRS is certified for use in both internal and external redundancy.

Oracle CSS network services must be configured, and then set with sufficiently high timeouts to insure that only Red Hat Cluster Suite is responsible for heartbeat and fencing. These values *must* be set, or the configuration will not be supported.

CSS Timeout should be set to at least 300 seconds to 500 seconds. CSS Disk Timeout should be set to 500 seconds.

Tip

Oracle cluster nodes are usually set to reboot and automatically re-enter the cluster. If the nodes should remain fenced, then the **`option="off"`** value in the **`fence`** section of the **`cluster.conf`** file can be set to ensure nodes are manually restarted. (The **`option`** value can be set to **`"reboot"`**, **`"on"`**, or **`"off"`**; by default, the value is **`"reboot"`**.)

Tip

The time a node takes to reboot depends on several factors, including BIOS settings. Many servers scan all of memory and then scan PCI buses for boot candidates from NICs or HBAs (of which there should only be one). Disabling these scans and any other steps in the BIOS that take time, will improve recovery performance. The **`grub.conf`** file often continues a built-in 5-second delay for screen hold. Sometimes, every second counts.

4.1.1. Cluster Recovery Time

In RAC/GFS, the road to transaction resumption starts with GFS filesystem recovery, and this is nearly instantaneous once fencing is complete. Oracle RAC must wait for CRS to recover the state of the cluster, and then the RDBMS can start to recover the locks for the failed instance (LMS recovery). Once complete, the redo logs from the failed instance must be processed. One of the surviving nodes must acquire the redo logs of the failed node, and determine which objects need recovery. Oracle activity is partially resumed as soon as RECO (DB recovery process) determines the list of embargoed objects that need recovery. Once roll-forward is complete, all non-embargoed and recovered objects are available. Oracle (and especially RAC) recovery is a complex subject, but its performance tuning can result in reduced downtime. And that could mean $Ms in recovered revenue.

Tip

It is possible to push the CSS Timeout below 300 seconds, if the nodes can boot in 60 seconds or less.

4.2. Network Topology

Clusterware requires a heartbeat network, and an inter-node network for moving database block between nodes (GCS). These are usually the same network, and often the same network as the Red Hat Cluster Suite network.

It is critical that Red Hat Cluster Suite operates heartbeat services over the private, bonded network and not the public network. If the private network fails for a node, then this node must be removed from the cluster. If the public network fails, the application tier cannot access the database on the node, but the CRS VIP service is responsible for the public network.

```
<clusternode name="rac7-priv" nodeid="1" votes="1">
        <multicast addr="225.0.0.12" interface="bond0"/>
        <fence>
                <method name="1">
                        <device name="jLO7"/>
                </method>
        </fence>
 </clusternode>
 <clusternode name="rac8-priv" nodeid="2" votes="1">
        <multicast addr="225.0.0.12" interface="bond0"/>
        <fence>
                <method name="1">
                        <device name="jLO8"/>
                </method>
        </fence>
 </clusternode>
```

While most customers do not bond this interface, it is supported by Oracle.

4.3. GFS Configuration

GFS file systems are certified for use with specific versions of Oracle RAC. For Oracle customers, see the Oracle Support Document 329530.1 for all currently certified combinations.

Clustered GFS requires that the Distributed Lock Manager (DLM) and the Clustered LVM services be configured and started. The DLM, if present will be started by CMAN. The RPM group should have installed all relevant components.

CLVMD only requires 1 change to the **`/etc/lvm/lvm.conf`** file; you must set **`locking_type`** to 3:

```
# Type of locking to use. Defaults to local file-based locking (1).
# Turn locking off by setting to 0 (dangerous: risks metadata corruption
# if LVM2 commands get run concurrently).
# Type 2 uses the external shared library locking_library.
# Type 3 uses built-in clustered locking.
locking_type = 3
```

Tip

The GFS service will not start up if the **`fenced`** service has not started.

Tip

Host-based mirroring (or more importantly host-based RAID) is not recommended for use with RAC (especially mission critical databases). RAC requires a storage array and any storage array worthy of running an Oracle RAC cluster will have superior RAID and RAIDSET management capabilities. Concatenating volumes does not involve RAID management, so that is less bug prone than using multiple layers of RAID.

Warning

GFS volumes can be grown if the file system requires more capacity. The **`gfs_grow`** command is used to expand the file system, once the LUN has been expanded. By keeping the filesystem mapping to single LUNs, it reduces an errors (or bugs) that might arise during **`gfs_grow`** operations. There is no performance difference between using the DDM inode, or subsequent CLVMD created logical volumes, built on these inodes. However, it must be stressed that you should perform a backup of your data before attempting this command as there is a potential to render you data unusable.

4.3.1. GFS File System Creation

For RAC, the file system must be created with arguments that are specific to the locking mechanism (always DLM), and the name of the cluster (HA585, in our case).

```
$ sudo gfs_mkfs -r 512 -j 4 -p lock_dlm -t rac585:gg /dev/mapper/ohome

$ sudo gfs_mkfs-j 4 -p lock_dlm -t rac585:gg /dev/mapper/db
```

Oracle manages data files with transaction redo logs, and with Oracle configuration in AIO/DIO mode, the writes always go to disk. The default journal is usually sufficient. The increased size of resource groups for GFS file systems is recommended for **ORACLE_HOME**, where the **$OH/diag** directory can contain thousands of trace files, spanning tens of GBs.

Note

Oracle Clusterware **HOME** is not supported on GFS clustered volumes at this time. For most installations, this will not be an imposition. There are several advantages (including, async rolling upgrades) to placing **ORA_CRS_HOME** on the node's local file system, and most customers follow this practice.

4.3.2. /etc/fstab Entries

```
/dev/mapper/ohome   /mnt/ohome      gfs     _netdev 0 0
/dev/mapper/db      /mnt/db         gfs     _netdev 0 0
```

The **_netdev** mount option is also useful as it ensures the file systems are unmounted before cluster services shut down.

4.3.3. Context Dependent Pathnames (CDPN)

When **ORACLE_HOME** (**$OH**) is located on a GFS clustered volume, certain directories need to appear the same to each node (including names of files, such as **listener.ora**), but have node-specific contents.

To enable CDPN for $OH/network/admin, perform the following steps.

1. Change to the **OH/network** directory:

   ```
   $ cd $OH/network
   ```

2. Create directories that correspond to the hostnames:

   ```
   $ mkdir rac7
   $ mkdir rac8
   ```

3. Create the admin directory in each directory:

   ```
   $ mkdir rac7/admin
   $ mkdir rac8/admin
   ```

4. Create the CPDN link (from each host).

   ON RAC7, in **$OH/network**:

```
$ ln –s @hostname admin
```

On RAC8, in **$OH/network**:

```
$ ln –s @hostname admin
```

4.4. Oracle Settings and Suggestions

Among the thousands of tuning variables in Oracle, the 2 most important are **SGA_TARGET** and **FILESYSTEMIO_OPTIONS**. Oracle performs more efficient I/O operations of the files on the GFS volumes are opened with DirectIO (DIO) and AsyncIO (AIO). This is accomplished using the **filesystemio_options** parameter:

```
filesystemio_options=setall
```

DirectIO bypasses the page cache for all I/O operations. If DIO is disabled, all datafile I/O will be use the page cache, which effectively double buffers the I/O. Oracle already contains a page cache, called the db block buffer cache. Double buffering increases response time latency for reads, and when the page cache runs the server out of Free memory, system throughput usually drops by 30-50%.

The third most important **init.ora** parameter must be decided upon first: **db_block_size**. The default **db_block_size** for Oracle on Linux is 8K. GFS uses 4K blocks (as does x64 hardware). Although 4K blocks will out-perform 8K blocks in GFS, other factors in the application might mask this effect. Application performance requirements take precedence, and do not change it unless you know what you are doing. It is not recommended that 2K blocks be used on GFS. Most customers leave it 8K.

RAC/GFS was certified using both 4K and 8K blocks, but supports all block size values that the Oracle RDBMS supports.

4.4.1. RHEL Settings and Suggestions

The RDBMS needs non-default values in the **/etc/sysctl.conf** that involve shared memory, semaphores. Clusterware requires the network settings to be altered. These are documented in the Oracle Install Guide or release notes for that particular version.

It is highly recommended that you install the 64-bit (x64) version of RHEL, Red Hat Cluster Suite and Oracle. Although 32-bit (x86) platforms are still fully certified and supported by both Oracle and Red Hat, Oracle performs better when allowed to access more memory.

4.4.2. Huge TLBs

The Oracle SGA (Shared Global Area) contains several memory structures that used to improve the performance of the executing SQL. The largest, and most critical is the db block buffer cache. This cache typically consumes over 80% of the SGA. Several SQL pools used for results and complex parallel operations consume the next largest block.

The advent of x64 systems make it possible to SGA ranges in the 8-1024GB range. For any SGA that is over 16GB, a consistent improvement of 8-15% should be possible; the larger the SGA, the more the improvement. In addition to making it possible for the hardware to do less work when providing

memory to the RDBMS, it also saves user memory by reducing the number of process page table entries (TLBs) that must be stored by each process.

For information on optimizing SGA settings, consult your Oracle user guide.

Cold Failover Cluster Configuration

This chapter provides information on configuration a cold failover HA cluster. For information on configuring a RAC/GFS cluster, see *Chapter 4, RAC/GFS Cluster Configuration*.

Long before RAC (and its progenitor, OPS) was suitable for high availability, customers still needed Oracle databases to be more reliable. The best way to do this was with a (relatively) simple two-node cluster that provided a second server node to take over in the event the primary node crashed. These early clusters still required many of the shared attributes that OPS/RAC databases required, but mandated that only one Oracle instance could be running at once; the storage was shared, but Oracle access was not. The use of this “failover” configuration remains in wide use today.

Note

An Oracle instance is the combination of OS resources (processes and shared memory) that must be initiated on a server. The instance provides coherent and persistent database access, for the connecting users or *clients*. Oracle workloads are extremely resource intensive, so typically there is only one instance/server. Oracle RAC consists of multiple instances (usually on physically distinct servers), all connecting to the *same set* of database files. Server virtualization now makes it possible to have more than one instance/server. However, this is not RAC unless these instances all connect to the same set of database files. The voraciousness of most Oracle workloads makes multiple instance/server configurations difficult to configure and optimize.

The OS clustering layer must insure that Oracle is *never* running on both nodes at the same time. If this occurs, the database will be corrupted. The two nodes must be in constant contact, either through a voting disk, or a heartbeat network, or both. If something goes wrong with the primary node (the node currently running Oracle), then the secondary node must be able to terminate that server, take over the storage, and restart the Oracle database. Termination is also called fencing, and is most frequently accomplished by the secondary node turning off the power to the primary node; this is called *power-managed* fencing. There are a variety of fencing methods, but power-managed fencing is recommended.

Note

The Oracle database is a fully journaled file system, and is capable of recovering all relevant transactions. Oracle calls the journal logs *redo* logs. When Oracle or the server fails unexpectedly, the database has *aborted* and requires *crash recovery*. In the failover case, this recovery usually occurs on the secondary node, but this does affect Oracle recovery. Whatever node starts up Oracle after it has aborted must do recovery. Oracle HA recovery is still just single instance recovery. In RAC, there are multiple instances, each with it’s own set of redo logs. When a RAC node fails, some other RAC node must recover the failed node’s redo logs, while continuing to provide access to the database.

The Oracle database must be installed on a shared storage array and this file system (or these file systems) can only be mounted on the active node. The clustering layer also has agents, or scripts that must be customized to the specific installation of Oracle. Once configured, this software can automatically start the Oracle database and any other relevant services (like Oracle network listeners). The job of any cluster product is to ensure that Oracle is only ever running on one node.

Clusters are designed specifically to handle bizarre, end-case operating conditions, but are at the mercy of the OS components that might fail too. The heartbeat network operates over standard

TCP/IP networks, and is the primary mechanism by which the cluster nodes identify themselves to other members. This ensures the cluster is viable and that Oracle can continue to operate on the primary node. Some failure cases can cause the heartbeat to become erratic or unreliable, so modern clustering products provide a second check-in mechanism, which insures that *quorum* is maintained. Quorum voting causes each cluster member to identify itself by *voting*, in the form of a simple write to a shared vote, or quorum disk. The combination of heartbeat and quorum disk minimizes the risk of *split-brain* cluster states. Split-brain clusters occur when the two nodes think they are both in the *correct* cluster state, so both access the shared storage. Split-brain states create the highest risk of database corruption, so this is the functional core of the cluster.

The two most common examples of Oracle Cold Failover are HP ServiceGuard and Veritas Cluster Server (VCS). Red Hat Cluster Suite's implementation for Oracle closely models these products, so customers familiar with them will be immediately familiar with Red Hat Cluster Suite Oracle HA. Of course, the devil is most definitely in the details.

5.1. Red Hat Cluster Suite HA

Red Hat Cluster Suite contains all the requisite components to implement Oracle HA: heartbeat, quorum disk voting, fencing and a resource harness to relocate the Oracle instance, when necessary. The major differences between how Red Hat Cluster Suite is set up for RAC and how it is set up for single instance involves appropriate timeout settings and the configuration of the resource harness, aptly named **rgmanager**.

5.2. Red Hat Cluster Suite Timers

When Oracle RAC is installed, Red Hat Cluster Suite must interact with Oracle Clusterware, but also is in control of the timeouts and eventually the fencing. When Oracle HA is configured, Red Hat Cluster Suite is also in charge, so the timeouts are very similar.

Tip

It is critical that the Red Hat Cluster Suite heartbeat service operates over the private, bonded network, not the public network. If the private network fails for a node, then this node must be removed from the cluster.

All installations will have subtly different timeout requirements, but start with these recommended settings:

```
<cluster config_version="11" name="dl585">
  <fence_daemon clean_start="1" post_fail_delay="0"post_join_delay="3" />
  <quorumd device="/dev/mapper/qdisk" interval="2" log_level="5" tko="8" votes="1" />
  <cman expected_votes="3" two_node="0" />
  <totem token="33000" />
```

In this example, the quorum disk is the level fencing mechanism with a timeout of 16 seconds; that is, two intervals of 8 seconds. The **tko** parameter stands for Technical Knock Out — a boxing metaphor. The CMAN heartbeat timeouts must be more than two time the **tko** timeouts; we choose 33 seconds (value in ms). This delay gives the quorum daemon adequate time to establish which node is the master during a failure, or if there is a load spike that might delay voting. The **expected_votes** parameter is set to the number of nodes + 1.

5.3. RGManager Configuration

Note

At this point in the process, we have installed Oracle on a shared volume disk (i.e. SAN).

The Resource manager is required only in the HA configuration and is responsible for ensuring that the selected node is capable of supporting an Oracle instance. The manager must ensure that network connectivity (provided by a Virtual IP address) is available, and mount the appropriate shared file systems that contain the Oracle database and the Oracle product installation and finally start the Oracle instance.

RGManager is capable of terminating Oracle services, dismantling the file systems and network so that the other node may safely start the Oracle services. There is a sample script, **`oracledb.sh`**, found in **`/usr/share/cluster`**. The customer must always modify this script so that RGManager can identify the Oracle services that require cluster management. Oracle environment variables, such as **`ORACLE_HOME`** and **`ORACLE_SID`** are critical to this identification. Oracle will likely use several file system mount points, and all mounts that are required to successfully run the database must be made known to RGManager.

RGManager is not enabled to start upon RHEL boot, so it must be enabled for the appropriate run level (typically 3):

```
$ sudo chkconfig –level 3 rgmanager on
```

5.3.1. Customizing oracledb.sh Environment Variables

Here are some notes from the script's preamble:

```
# (1) You can comment out the LOCKFILE declaration below.  This will prevent
# the need for this script to access anything outside of the ORACLE_HOME
# path.
#
# (2) You MUST customize ORACLE_USER, ORACLE_HOME, ORACLE_SID, and
# ORACLE_HOSTNAME to match your installation if not running from within
# rgmanager.
#
# (3) Do NOT place this script in shared storage; place it in ORACLE_USER's
# home directory in non-clustered environments and /usr/share/cluster
# in rgmanager/Red Hat cluster environments.
```

The original script supported Oracle's attempt at an application server (iAS), which has been replaced by a real product, Weblogix. It is also not common to co-locate the connection pool on the database server.

The script also attempts to start up the OEM (Oracle Enterprise Manger) console. The console is also not generally run on server, but runs on some random Windows box, and the OEM *agents* are configured on the server. If you happen to be using an older version of RHEL, the **`oracledb.sh`** script might need to have all of this disabled, including references to these obsolete services in the **`start_db`**, **`stop_db`** and **`get_db_status`** functions in the script.

5.3.1.1. DB_PROCNAMES

There has been some debate over what constitutes a running instance and RGManager uses a list of Oracle background processes (daemons in unix) to test for their existence. Historically, PMON and SMON are the most frequently selected and they are a good set. PMON is the Process Monitor part of the instance, and mostly cleans out latches and locks left by dead processes. It does dead process detection, including the death of critical background process like LGWR (redo log writer), DBWR (database writer), including SMON (system monitor). It is SMON's job to make sure the instance is in a healthy state. If SMON crashes for some reason, then PMON will kill the instance. The presence of both of these is a reasonable test.

5.3.1.2. LSNR_PROCNAME

The SQL*Net Network Listener must be running for clients running on other machines to connect to the instance. The database can be accessed on the server node with the use of a Listener. Some instances that don't need client access (like batch ETL, or extremely security sensitive instances), do not have to run a Listener. Most conventional databases with a mid-tier connection pool need a listener, so RGManager ensures that the Listener is up and running. The lack of a Listener will look like a dead database to the mid-tier.

5.3.2. Network VIP for Oracle Listeners

The Oracle SQL*Net Listener service must listen on a common IP address that must be accessible from either host, and this is accomplished using a Virtual IP, or VIP.

```
# ip addr add 192.168.1.20/24 dev eth0
```

The VIP is managed and relocated by **rgmanager**, and must be in the same submit as the public, or front-side physical network interfaces. The front-side network is the network the Listener uses, and clients will also have access.

```
<rm log_level="7">
<service domain="OracleHA" autostart="1" exclusive="1" name="oracle11g" recovery="relocate">
<oracledb home="/ff/11g/db" name="ed" type="11g" user="jneedham" vhost="192.168.1.60"/>
<ip address="192.168.1.60" />
<fs device...
<rm/>
```

Alternatively, you can use a hostname for the virtual IP:

```
edb home="/ff/11g/db" name="ed" type="11g" user="jneedham" vhost="hacf-vip"/>
<ip address="hacf-vip" />
```

The **vhost** argument must match the IP address clause in the service domain definition, and the **/etc/hosts** file must contain all the required addresses:

```
# Cold Failover VIP
192.168.1.60    hacf-vip
```

```
192.168.1.160   rac5
192.168.2.160   rac5-priv
192.168.2.5     rac5-jlo

192.168.1.161   rac6
192.168.2.161   rac6-priv
192.168.2.6     rac6-jlo
```

5.3.2.1. listener.ora Configuration

The listener is managed by the **rgmanager** package, but the functionality is determined by the Oracle configuration file, **listener.ora**. The bolded **LISTENER** tag in the file is the specific name of this listener instance. This is the default, but this can be changed, and often is, when there is more than 1 SQL*Net Listener service for this database.

```
LISTENER =
  (ADDRESS_LIST=
       (ADDRESS=(PROTOCOL=tcp)(HOST=hacf-vip)(PORT=1521)) #1521 is too common
       (ADDRESS=(PROTOCOL=ipc)(KEY=PNPKEY)))
SID_LIST_LISTENER =
  (SID_LIST =
   (SID_DESC =
    (GLOBAL_DBNAME = ed)      # Needs to match DBNAME in inited.ora
    (ORACLE_HOME = /ff/11g/db)
    (SID_NAME = ed)           # Needs to match the instance's ORACLE_SID
   )
   (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = /ff/11g/db)
    (PROGRAM = extproc)
   )
  )
```

The Listener must listen on the VIP, *not* on the host-specific public interfaces. The connecting clients use an SQL*Net **tnsnames.ora** configuration file that contains an alias that directs them to the virtual IP. The location of the database instance is now transparent to clients.

```
rhcs11g=
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = hacf-vip)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME =ed)   # This is the ORACLE_SID
    )
  )
```

Most JDBC clients do *not* install Oracle client libraries, so must use the **Thin** client driver. (More advanced JDBC connectivity does require an Oracle client install). For JDBC thin, the connection string cannot use the SQL*Net **alias**, but must encode the same information:

```
… getConnection ("jdbc:oracle:thin:@hacf-vip:1521:ed", "scott", "tiger")
```

5.3.3. Files System Entries

Oracle single-instance can run on any RHEL-supported filesystem type. Ths is unlike RAC, where only GFS is specifically certified for use. Most customers use EXT3, but EXT4, GFS and NFS are supported as well.

```
<fs device="/dev/mapper/dbp5" force_unmount="1" fstype="ext3" mountpoint="/ff" name="ora_ff"/
>
<fs device="/dev/mapper/dbp6" force_unmount="1" fstype="ext3" mountpoint="/gg" name="ora_gg"/
>
```

And for NFS mounts:

```
<netfs host="F3040" export="/vol/ed" force_unmount="1" mountpoint="/mnt/ed"
 options="rw,hard,nointr,vers=3,rsize=32768,wsize=32768,actimeo=0,proto=tcp"
 name="ora_nfs_ed"/>
```

Note

NFS must be mounted using only the mount options that are required by Oracle. The most important of these is **`actimeo`** and it should be set to zero to ensure the access times stay current.

Appendix A. Sample cluster.conf File

This appendix provides a sample **`cluster.conf`** file for a two node cold failover configuration with power fencing via an APC power strip.

```
<?xml version="1.0"?>
<cluster config_version="1" name="HA585">
     <fence_daemon clean_start="1" post_fail_delay="0" post_join_delay="3"/>
     <quorumd interval="2" device="/dev/mapper/qdisk" tko="8" votes="1" log_level="7"/>
     <cman expected_votes="3" two_node="0"/>
     <totem token="33000"/>
     <fencedevices>
          <fencedevice agent="fence_apc" ipaddr="192.168.91.59" login="adminl" name="apc"
 passwd="password"/>
     </fencedevices>
     <clusternodes>
          <clusternode name="ora11-priv" nodeid="1" votes="1">
               <fence>
                    <method name="1">
                     <device name="apc" option="off" switch="1" port="2"/>
                     </method>
               </fence>
          </clusternode>
          <clusternode name="ora12-priv" nodeid="2" votes="1">
               <fence>
                    <method name="1">
                         <device name="apc" option="off" switch="1" port="5"/>
                    </method>
               </fence>
          </clusternode>
     </clusternodes>
     <rm log_level="7">
          <service domain="OracleHA" autostart="1" exclusive="1" name="oracle11g"
 recovery="relocate">
               <ip address="10.10.8.200"/>
               <fs device="/dev/mapper/diskdp1" force_unmount="1" fstype="ext3" mountpoint="/
diskd" name="diskd"/>
               <oracledb home="/diskd/ora11gR1/db_1" name="oracledb" type="11g" user="oracle"
 vhost="10.10.8.200"/>
          </service>
     </rm>
</cluster>
```

Appendix B. Revision History

Revision 1.0 **Fri Jul 23 2010** **Steven Levine**

First edition

Index

P

Q

R

S

T

V

W