

Configuring a Highly Available Linux Cluster for SAP Services

Clusters of Dell™ PowerEdge™ servers using Oracle9i™ Real Application Clusters (RAC) can provide SAP® software environments with a flexible, scalable, and highly available database platform. The database will continue to run if one of the Oracle9i RAC database nodes fails; however, vital SAP functionality such as the message server and the enqueue server can still be single points of failure. To help protect these services from failure and thus unwanted downtime or even data loss, IT administrators can run them on a Red Hat® Enterprise Linux® OS–based cluster to complement the Oracle9i RAC database cluster and maintain service in a highly available manner.

BY DAVID DETWEILER, ACHIM LERNHARD, FLORENZ KLEY, THORSTEN STAERK, AND WOLFGANG TRENKLE

Related Categories:

Clustering

Database

Dell/EMC storage

High availability (HA)

Linux

Oracle

Red Hat Enterprise Linux

SAP

Visit www.dell.com/powersolutions for the complete category index.

Setting up a highly available SAP system on Linux requires eliminating any possible single point of failure for the database as well as for the various SAP components of the overall system. While the database is made highly available by means of Oracle9i Real Application Clusters (RAC) technology, SAP applications can be made highly available by protecting the SAP central instance—which includes the message server and the enqueue server—from failure. In addition, SAP management tools require a common shared \$ORACLE_HOME directory, which requires the Highly Available Network File System (HA NFS) service exporting the Oracle® executables, the SAP executables, and SAP shared files such as profiles and the sapglobal directory.

On Linux, the node membership for Oracle9i RAC database nodes is managed by the Oracle cluster manager

(oracm), which is designed specifically to manage RAC nodes. Therefore, administrators should implement a second, independent cluster to make the \$ORACLE_HOME directory and the SAP central instance services highly available. This must be performed on a second set of hosts, because each node can be a member of only one cluster. Membership in two independent clusters with potential conflicts on current node status would render the cluster nodes unusable for each of the respective clusters. This second cluster uses the Red Hat Cluster Suite.

Setting up the Red Hat cluster

To set up the Red Hat cluster for the SAP software, administrators should first determine whether the Red Hat Package Manager (RPM™) packages for the Red Hat Cluster Suite are installed (see Figure 1). Depending on

```
[root@ls3220 root]# rpm -q clumanager
clumanager-1.2.16-1

[root@ls3220 root]# rpm -q redhat-config-cluster
redhat-config-cluster-1.0.2-2.0
```

Figure 1. Checking for Red Hat Cluster Suite RPM packages

the availability of updates, the version numbers may differ. Administrators should install the most recent version of these packages. They should then prepare the shared storage and the network connections. Throughout the example scenario used in this article, the server names `ls3219` and `ls3220` are used for the first and second cluster nodes, respectively.

Configuring the network

Both nodes must have two available Ethernet interfaces. One interface is used for cluster communication between the two nodes and should be on a private network. The other is the publicly visible network interface. The private network interfaces in this example are named `dell3219` and `dell3220`, respectively. Depending on the specific requirements of the environment, administrators may want to set up four interfaces—two for each node—using the Linux kernel bonding mechanism. This provides a highly available network connection on each channel and secures the cluster against failure of one single component (network interface card, network cabling, or switch) on the respective communication channel.

Administrators should reserve one public IP address for each node. In this example scenario, these addresses are `10.17.64.25` for node `ls3219` and `10.17.64.26` for node `ls3220`. Administrators should also reserve one private IP address for each node. For `dell3219`, the private address is `172.16.42.34`; for `dell3220`, the private address is `172.16.42.35`.

Additionally, administrators should reserve three IP addresses for the cluster services to be used as virtual IP addresses. They should configure the interfaces (or virtual interfaces) with these addresses, either by using the `redhat-config-network` program or editing the respective interface setup files in `/etc/sysconfig/network-scripts`. Figure 2 shows what the public interface on `ls3219` should look like.

Administrators should set up all the interfaces on the nodes according to the host names and IP addresses. This is the same setup principle that is used in the Oracle9i RAC cluster: one public and one private IP address per node.¹

Configuring the shared storage

After testing the network connections, administrators can set up the shared storage. The cluster software needs two small partitions as quorum devices, which should be configured on separate logical units (LUNs) to maximize independence and minimize possible contention. The partitions must have a minimum size of 10 MB each. However, the usual minimum size for a LUN on a Dell/EMC storage array is 100 MB. The quorum LUNs will be bound later as raw devices.

Additionally, administrators should create one or more LUNs to hold the file systems for the data to be exported via the HA NFS server. They should follow Oracle recommendations regarding the size for `$ORACLE_HOME` and SAP recommendations for the executables (approximately 300 to 400 MB depending on the kernel version), and they should take into account the data that will be stored in the LUNs as well. Once the LUNs have been created on the storage system, administrators can make them available to the nodes.

The next step is to create partitions on the quorum and NFS storage LUNs. For the quorum LUN, one partition is enough. Because administrators will bind these partitions as raw devices, they can set the partition type to “da” (non-file-system data) with the `t` option of `fdisk`. Figure 3 shows what the quorum partitions would look like on the host in the example scenario. Administrators should create and format partitions on the LUNs for the NFS directories.

Next, administrators should create persistent symbolic names for the partitions with `devlabel`. This program makes the partition device names resilient against device name reordering (for example, when the SCSI scan order is different). In the example scenario, the persistent symbolic name `/dev/homedir` is created for the partition used for the NFS export.

Because the quorum disks are raw devices, they must be bound so as to be available to the kernel. When the special symbolic name `/dev/raw/rawn` is used with `devlabel`, the link is created and the partition is bound as a raw device. Note that the identifier changes

```
[root@ls3219 root]# cat /etc/sysconfig/
network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
IPADDR=10.17.64.25
NETMASK=255.255.252.0
ONBOOT=yes
TYPE=Ethernet
GATEWAY=10.17.64.1
```

Figure 2. Example public interface setup file for node `ls3219`

¹ For more information, refer to the “Checking the network connections” section in “Creating Flexible, Highly Available SAP Solutions Leveraging Oracle9i and Linux on Dell Servers and Dell/EMC Storage” by David Detweiler, Achim Lemhard, Florenz Kley, Thorsten Staerk, and Wolfgang Trenkle in *Dell Power Solutions*, November 2005; www.dell.com/downloads/global/power/ps4q05-20050174-SAP.pdf.

```

Disk /dev/sdb: 314 MB, 314572800 bytes
64 heads, 32 sectors/track, 300 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Device Boot Start End Blocks Id System
/dev/sdb1 1 300 307184 da Non-FS data

Disk /dev/sdc: 314 MB, 314572800 bytes
64 heads, 32 sectors/track, 300 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Device Boot Start End Blocks Id System
/dev/sdc1 1 300 307184 da Non-FS data
    
```

Figure 3. Example quorum partitions on host

to “RAW.” Administrators can check the result with the `devlabel status` command (see Figure 4).

Administrators can check whether the raw devices are bound. As shown in Figure 5, the `raw` command displays the major and minor numbers of the bound devices. Administrators can check these numbers against the currently assigned block devices (from `devlabel status`).

Once the `devlabel` settings have been finalized, administrators can copy the `/etc/sysconfig/devlabel` file to the same directory on the other host. Then, they can log in to that host and issue the `devlabel restart` command. Administrators should not try to add raw devices and symbolic names themselves; they should allow `devlabel` sort out the unique IDs collected on the other node to ensure that the same physical device is bound under the same symbolic name.

Configuring a clustered NFS service

Once the network and devices are configured, administrators can activate the Red Hat cluster. Logged in as the root user, administrators can check whether the cluster services are running (see Figure 6). If the output does not show that the cluster services have stopped, administrators should stop them by issuing the `stop` argument to the `init-script`. Then, they can start the `redhat-config-cluster` program, preferably in a Virtual Network Computing (VNC) session.

Administrators should begin by setting up the raw devices for the cluster quorum. In the Cluster Configuration Tool, administrators should go to Cluster > Shared State to display the Shared State dialog box. In this box, administrators should enter the names of the two raw devices: `/dev/raw/raw1` and `/dev/raw/raw2`.

They should then add the two nodes as members of the cluster by clicking the Members tab and going to File > New. In the dialog box, administrators should enter the name of the host (`ls3219` in this

```

[root@ls3220 root]# devlabel status
brw-rw---- root disk /dev/raw/raw1
--[RAW]--> /dev/sdb1
brw-rw---- root disk /dev/raw/raw2
--[RAW]--> /dev/sdc1
brw-rw---- root disk /dev/homedir ->
/dev/sdd1
    
```

Figure 4. Checking the status of partitions

example scenario). They should then repeat this step for the other cluster member (`ls3220`). They can leave “Enable SW Watchdog” checked—this enables the software watchdog timer, which allows a cluster member to reboot itself if the system hangs.

Next, administrators can name the NFS service by clicking the Services tab and then the New button to display the Service dialog box. They should provide a name without spaces or special characters (this simplifies querying the status from the command line). In the example scenario, the HA NFS service name is `RAC_NFS_directories`. Neither a failover domain nor a user script should be assigned. Administrators can click the OK button and go to File > Save to save the changes to the cluster configuration.

In the next step, administrators can add a device to mount, a service IP address, and NFS clients to the service. In the example scenario, `10.17.64.27` is used as the IP address for the HA NFS service. To specify a service IP address, administrators should click the Services tab of the Cluster Configuration Tool. Then they can select the service and click the Add Child button. Next, administrators select “Add Service IP Address” in the popup window and specify an IP address with netmask and broadcast in the next window. Although the netmask and broadcast addresses are optional, best practices recommend setting them.

```

[root@ls3220 root]# raw -qa
/dev/raw/raw1: bound to major 8, minor 17
/dev/raw/raw2: bound to major 8, minor 33
    
```

Figure 5. Checking whether raw devices are bound

```

[root@ls3220 root]# /etc/init.d/clumanager status
clumembd is stopped
cluquorumd is stopped
clulockd is stopped
clusvcmgrd is stopped
    
```

Figure 6. Checking whether cluster services are running

ORACLE CLUSTER FILE SYSTEM

A cluster file system allows all nodes in a cluster to concurrently access a device via the standard file system interface—enabling easy management of applications that need to run across a cluster. Oracle Cluster File System (OCFS) Release 1 was introduced in December 2002 to enable Oracle RAC users to run the clustered database without having to use raw devices. This file system was designed to store database-related files such as data files, control files, redo logs, and archive logs.

OCFS2—the next generation of the Oracle Cluster File System—was introduced in August 2005 for Red Hat Enterprise Linux 4 and Novell® SUSE™ Linux Enterprise Server 9 platforms. This high-performance, general-purpose cluster file system can store not only database-related files on a shared disk, but also Oracle binaries and configuration files (shared Oracle home)—helping to make the management of Oracle RAC easier using OCFS2 than OCFS1. Also, any non-Oracle binaries or non-Oracle configuration files, such as shared SAP directories, can be stored on shared disks.

In addition, OCFS2 provides metadata caching; metadata journaling; cross-node file data consistency; easy administration, including operation as a shared root file system; support for multiple block sizes; support for up to 254 cluster nodes; context-dependent symbolic link (CDSL) support for node-specific local files; asynchronous and direct I/O support for database files for improved database performance; and full integration with Linux kernel 2.6 and later. With the release of OCFS2 on Linux, enterprises can implement Oracle RAC with these capabilities while enhancing the overall environment by not having to use HA NFS Linux volumes for the required shared SAP directories.

To add the device to be mounted by the service, administrators can click the Services tab of the Cluster Configuration Tool, select the service, and click the Add Child button. They can then select “Add Device” and click the OK button. Then they can specify a device special file (in this example, `/dev/homedir`) and a mount point (`/sapmnt/clu_export`). Each device must have a unique device special file and a unique mount point within the cluster and across service boundaries. Administrators can specify a directory from which to mount the device in the Mount Point field. This directory cannot be listed in `/etc/fstab` because it is automatically mounted by the Red Hat Cluster Manager when the service is started.

Administrators should choose a file system type from the FS Type list (`ext3` is used in the example scenario).

Administrators can specify options for the device. If the Options field is left blank, the default mount options (`rw, suid, dev, exec, auto, nouser, and async`) are used.² Administrators can check “Force Unmount” to force any application that has the specified file system mounted to be shut down prior to disabling or relocating the service (when the application is running on the same cluster member that is running the disabled or relocated service). When finished, administrators can click the OK button and go to File > Save to save the changes to the `/etc/cluster.xml` configuration file, and go to File > Quit to exit the Cluster Configuration Tool.

Testing the cluster

Once the cluster is configured, administrators can begin the first cluster test. First, administrators should restart the `redhat-config-cluster` program and go to Cluster > Start local cluster daemons. Once the status display shows that the host has changed from “Unknown” to “Active,” administrators can enable the service by selecting it in the Services window and clicking “Enable.” The service status should change from “Disabled” (red) to “Running” (black). On this node, administrators should now see the mounted device under the configured mount point. If administrators do not see the device, they should check the system log for cluster service error messages.

After a successful test on one node, administrators can copy the `/etc/cluster.xml` file into the same directory on the other node. Then, they can start the cluster services there, either with the init script or with the `redhat-config-cluster` graphical user interface (GUI). Administrators also should test switching the service between the two cluster hosts.

Adding clients to the clustered NFS service

After testing that the cluster runs properly, administrators should extend the configuration of the NFS service to export one or more directories to the clients. Administrators should check that the NFS daemon and the portmapper run on both hosts and are configured to start automatically. They should execute the following commands on both hosts:

```
/sbin/chkconfig --level 345 nfs on
/sbin/chkconfig --level 345 portmap on
```

This enables automatic starting in the runlevels 3, 4, and 5. Administrators can check the result by entering the following command:

```
/sbin/chkconfig -list service
```

² For a description of the available options, administrators should refer to the `mount` man page.

The output should look similar to the following:

```
[root@ls3220 root]# /sbin/chkconfig --list nfs
nfs    0:off  1:off  2:on   3:on   4:on
       5:on   6:off
```

In addition, the output should look similar for services that will be set up later. Administrators should perform these steps on both hosts.

Next, administrators should return to the cluster configuration GUI and click the Services tab. They should select the NFS service and click the collapse/expand indicator, or *twistie*, on the left to display the contents. Administrators should see the service IP address and the service device. They can then select the device and click the Add Child button. A popup window asks for the export directory name. In this example scenario, everything below `/sapmnt/clu_export` is exported, and the following directories are exported with different access permissions:

- `/sapmnt/clu_export/readonly` (ro, async)
- `/sapmnt/clu_export/read_write` (rw, sync)
- `/sapmnt/clu_export/read_write_root` (rw, sync, no_root_squash)

Even when it is not immediately necessary to create an export that is root-writable and preserves the user ID (without reassigning `nfsnobody` to root), best practices recommend configuring the export with these settings—to enable backups and quick file distribution among the hosts.

The allowed client and permissions are attributes of the NFS Export Client object, which is a child of the NFS Export object. Administrators can add the clients again by selecting “NFS Export” and clicking the Add Child button. When finished, the NFS service structure should resemble the structure of the XML file `/etc/cluster.xml` (see Figure 7).

Note: The NFS export is under the control of the Red Hat cluster, and the directories exported there must not appear in the `/etc/exports` file used by the non-clustered NFS daemon.

Client-side mount options

On the client side, the directories are mounted under `/sapmnt/homedir/readonly` following a schema where `/sapmnt/hostname/`

```
Service RAC_NFS_directories

service_ipaddress
ipaddress="172.16.42.60"
netmask="255.255.255.0"
broadcast="172.16.42.255"

device name="/dev/homedir"
mountpoint="/sapmnt/clu_export"
fstype="ext3"
forceunmount="yes"

nfsexport name="/sapmnt/clu_export/readonly"
client name="172.16.42.0/24"
options="ro,async"

nfsexport name="/sapmnt/clu_export/readwrite"
client name="172.16.42.0/24"
options="rw,sync"

nfsexport name="/sapmnt/clu_export/readwrite_root"
client name="172.16.42.0/24"
options="rw,sync,no_root_squash"
```

Figure 7. NFS service directory structure

directory mounts directories exported by *hostname*. All clients mount the exported directories there. The `/etc/fstab` entries for the example scenario are shown in Figure 8.

Adapting the SAP directory structure

Locally, symbolic links point to the NFS-mounted directories. For example, the SAP instance DVBGS00 would expect the directory structure shown in Figure 9 on its server. The directories are located on NFS and can be found in the same location (that is, with the identical pathname) on every host. The `/usr/sap/RAC/SYS` directory links to `/sapmnt/RAC` (see Figure 10), and the `/sapmnt/RAC` directory links to the NFS directories (see Figure 11).

In the example scenario, the NFS directories are organized by system ID (SID) to support more than one SAP system (see Figure 12). The `readonly/` and `readwrite/` incarnations of the `RAC_sapsystem` directory show that the directories used by an SAP system are divided by these attributes, as shown in Figure 13.

```
# HA NFS exports
homedir:/sapmnt/clu_export/readonly /sapmnt/homedir/readonly nfs\ hard,intr,noexec,ro,bg 0 0
homedir:/sapmnt/clu_export/readwrite /sapmnt/homedir/readwrite nfs\ hard,intr,sync,bg 0 0
homedir:/sapmnt/clu_export/readwrite_root /sapmnt/homedir/readwrite_root nfs\ hard,intr,sync,bg 0 0
```

Figure 8. Example `/etc/fstab` entries

```
ls3220:racadm-DVBGS00 > find /usr/sap/RAC/
DVBGS00/* -prune
/usr/sap/RAC/DVBGS00/data
/usr/sap/RAC/DVBGS00/log
/usr/sap/RAC/DVBGS00/sec
/usr/sap/RAC/DVBGS00/work
```

Figure 9. Directory structure for DVBGS00 SAP instance

Configuring SAP central instance services for the cluster

Access to data in the underlying database of an SAP system is synchronized with a special lock system called the SAP enqueue mechanism. This mechanism serializes access and prevents access from being changed for more than one requesting party.

The enqueue server usually runs as a service of the SAP central instance. If clients run in the same SAP instance, they can contact the enqueue server via the UNIX® Interprocess Communication (IPC) mechanism; if they are not part of the central instance, clients contact the enqueue server via the SAP message server. As opposed to all other components of the SAP system on the application layer, the enqueue server holds a state—an in-memory table of granted locks—that cannot be recovered gracefully if the service fails. The message server, which consequently plays an important role in contacting the enqueue server, holds no state; it receives only incoming connection requests and transfers them to the addressee. The message server can be restarted after failure, with no impact other than delayed communications. The enqueue server is a potential single point of failure in an SAP system, isolated from the failover provided at the database layer.

The SAP solution to the enqueue challenge is the stand-alone enqueue server and the enqueue replication mechanism. With these components, the enqueue server runs as a stand-alone program and can be contacted directly by enqueue clients. Additionally, a second enqueue server—called the enqueue replication server—is started; its only task is to maintain a second copy of the enqueue state table (lock table). Communicating regularly with the enqueue server, the enqueue replication server keeps its copy of the enqueue table current. If the enqueue server fails, it can be restarted on the host where the enqueue replication server runs. When the enqueue replication server recognizes that the enqueue server has started up, it will transfer its current lock table before exiting. The newly started enqueue server can now continue without losing valuable enqueue state information. Additionally, OS-level high-availability software makes the enqueue server available via a virtual, clustered IP address, masking the restart from the clients so that they always connect to the same IP address.

Splitting the central instance

To secure the SAP system’s services in a high-availability cluster, administrators must split the traditional central instance into dedicated instances because a large “service block” can be difficult to monitor. Furthermore, this large block makes restarting services difficult, because administrators must also restart parts of the central instance that have not failed.

To run the enqueue server as a master/slave service, the enqueue service and the enqueue replication service should always reside on different hosts. The message server is not bound to a particular host. Because these are the two services that constitute a central instance, the cluster can run only those services,

```
ls3220:racadm-DVBGS00 > ls -l /usr/sap/RAC/SYS/
total 4
drwxr-xr-x  2 racadm  sapsys  4096 Nov 11 12:44 exe
lrwxrwxrwx  1 racadm  sapsys   18 Jan 24 14:48 global -> /sapmnt/RAC/global
lrwxrwxrwx  1 racadm  sapsys   19 Jan 24 14:49 profile -> /sapmnt/RAC/profile
```

Figure 10. Links to /sapmnt/RAC

```
ls3220:racadm-DVBGS00 > ls -l /sapmnt/RAC/
total 0
lrwxrwxrwx  1 racadm  sapsys   49 Jan 24 14:49 exe -> /sapmnt/homedir/readonly/RAC_sapkernel/exe-640-21
lrwxrwxrwx  1 racadm  sapsys   46 Jan 24 14:49 global -> /sapmnt/homedir/readwrite/RAC_sapsystem/global
lrwxrwxrwx  1 racadm  sapsys   46 Jan 24 14:49 profile -> /sapmnt/homedir/readonly/RAC_sapsystem/profile
```

Figure 11. Links to the NFS directories

```
[root@ls3220 root]# ls -l /sapmnt/clu_export/*
/sapmnt/clu_export/readonly:
total 12
drwxr-xr-x  3 root    root    4096 Jan 21 19:05 RAC_oracle
drwxr-xr-x  3 racadm  sapsys 4096 Jan 21 12:19 RAC_sapkernel
drwxr-xr-x  4 racadm  sapsys 4096 Jan 21 17:47 RAC_sapsystem

/sapmnt/clu_export/readwrite:
total 8
drwxr-xr-x  2 orarac  dba    4096 Jan 21 03:03 RAC_oracle
drwxr-xr-x  8 racadm  sapsys 4096 Jan 27 13:50 RAC_sapsystem

/sapmnt/clu_export/readwrite_root:
total 0
```

Figure 12. Directories organized by system ID

and all application servers must be outside the cluster. However, for systems management purposes, the message server can run together with a dialog service, and an application server can reside in the cluster, or close to it. In the example scenario, the enqueue server; the enqueue replication server; the message server; and an application instance with dialog, update, batch, and spool work processes all run as services in the cluster.

In the example scenario, the traditional central instance DVEB-MGS is split into multiple instances as follows (the two numerals at the end of each instance name represent the system number):

- **DVBGS00:** Dialog, update, batch, gateway, and spool work processes
- **DM01:** Dialog service (for local administration) and message server
- **E02:** Enqueue server
- **R02:** Enqueue replication server

Note that the enqueue server and the enqueue replication server must have the same system number (02) separate from the rest of the instances; otherwise, the takeover of the enqueue table will fail.

For each instance, administrators must define an instance profile and a start profile, according

to SAP documentation. However, to put these instances under the control of the high-availability cluster, administrators must provide scripts for the cluster that conform to the UNIX System V init conventions—that is, Bourne shell (bash) scripts that offer a start, stop, and status function.

For example instance profiles and start profiles, see the supplemental online section of this article at www.dell.com/powersolutions.

Switching between different user environments

To start the SAP services, the programs use the environment of the SAP administrative user. Because those environment

parameters differ from instance to instance, a simple way to switch between different environments for the same OS user is desirable. To achieve this switching, administrators can adapt the default environment contained in the `sapenv.sh` and `dbenv.sh` scripts for each instance, and rename the script `sapenv_INSTANCENAME.sh`. Then, they can create a scriptlet—a reusable script element—containing a source statement such as the following:

```
#!/bin/sh
source sapenv_INSTANCENAME.sh
```

```
[root@ls3220 root]# ls -l /sapmnt/clu_export/readonly/RAC_sapsystem/
total 8
drwxrwxr-x  3 racadm  sapsys 4096 Jan 28 03:16 profile

[root@ls3220 root]# ls -l /sapmnt/clu_export/readwrite/RAC_sapsystem/
total 24
drwxrwxr-x  6 racadm  sapsys 4096 Jan 26 18:30 DVBGS00
drwxrwxr-x  2 racadm  sapsys 4096 Jan 28 02:46 global
drwxr-xr-x 11 racadm  sapsys 4096 Nov 11 11:00 trans

[root@ls3220 root]# ls -l /sapmnt/clu_export/readwrite/RAC_sapsystem/DVBGS00/
total 16
drwxrwxr-x  2 racadm  sapsys 4096 Jan 28 06:38 data
drwxrwxr-x  2 racadm  sapsys 4096 Jan 27 11:01 log
drwxrwxr-x  2 racadm  sapsys 4096 Jan 26 18:53 sec
drwxrwxr-x  2 racadm  sapsys 4096 Jan 28 08:36 work
```

Figure 13. Directories used by an SAP system

```
RAC_app_server   DVBGS00  1s3216  10.17.64.22
RAC_message_server  DM01  1s3221  10.17.64.27
RAC_enqueue_server  E02   1s3222   10.17.64.28
R02   no address. Bound to the public IP address
      of the owning member.
```

Figure 14. Virtual IP addresses and cluster services for SAP instances

```
RAC_app_server    /etc/init.d/sapappserver-RAC
RAC_message_server /etc/init.d/sapmsgsrv-RAC
RAC_enqueue_server /etc/init.d/sapenserver-RAC
```

Figure 15. Corresponding init scripts for cluster services

By using the command `source`, administrators can make changes to the environment variables effective for the current shell session. If administrators source this scriptlet, the user has the matching parameters for `INSTANCE_NAME` in the environment. This can also be seen in the `start()` and `stop()` functions of the `initscripts` package, because the scriptlet is sourced before executing the command.

Integrating the SAP instances as a cluster service

Administrators must create virtual IP addresses for all of the SAP instances, except for the enqueue replication server. The enqueue replication server always runs on the host not owning the enqueue server and attaches itself to the enqueue server (as opposed to the enqueue server trying to contact it). It can be bound to the public IP address of the respective cluster member, even if this means that it changes IP address with every service relocation.


The DVBGS, DM, and E instance each require a virtual IP address, so that they are always present under the same address from outside the cluster. Because DVBGS and DM also appear in the instance list (SM51), administrators should adapt the instance name to show the host name belonging to the service IP address, not the currently active cluster member IP address. Administrators can do this by setting `rdisp/myname` to `virtualhostname_SID_SAPSYSTEM`. In this manner, the instance names remain stable after relocation of the service from one cluster member to another.

Administrators should create cluster services for the SAP instances and give each service a virtual IP address as a child. Figure 14 shows this configuration for the example scenario, and Figure 15 shows the corresponding init scripts.

As shown in Figure 14, R02 does not have a service. This configuration is used in the example cluster because a service order cannot be defined, nor can services be set in a relationship. Each service is independently monitored and treated without regard to the other configured services. Because the enqueue server and the enqueue replication server are dependent and must be started and stopped on opposite hosts and in a specific order, administrators must start and stop the enqueue replication server from inside the `sapenserver-SID` script.

Next, administrators can enter the scripts as “user scripts” in the service definitions and configure a check interval, which typically varies from 30 to 60 seconds. Before transferring control of the services to the cluster, administrators should run the scripts manually to test their functionality.

Building a reliable platform for SAP

Oracle9i RAC for SAP on Linux can provide a stable, flexible, and scalable environment, provided administrators follow proper planning and installation procedures. By using the SAP enqueue mechanism with Linux, administrators not only can help protect the database from unplanned downtime, but they also can set up the SAP environment to avoid disruptions to end users. 

David Detweiler is the Dell SAP Alliance Manager in Europe, the Middle East, and Africa (EMEA) and a member of the Dell SAP Competence Center in Walldorf, Germany, which helps ensure that current and future Dell technologies work together with SAP solutions.

Achim Lernhard has worked at the Dell SAP Competence Center in Walldorf, Germany, for three years as part of the SAP LinuxLab. He assisted the Oracle9i RAC on Linux pilot customer from installation to productivity and worked on the hardware certifications.

Florenz Kley is a consultant for SAP Technology Infrastructure. He has worked for five years at the Dell SAP Competence Center in Walldorf, Germany, as part of the SAP LinuxLab. He conducted performance benchmarks to help prove the scalability and performance of Oracle9i RAC for SAP on Linux and helped build the architecture for Dell’s Oracle9i RAC on Linux pilot customer.

Thorsten Staerk is a consultant at the Dell SAP Competence Center in Walldorf, Germany, as part of the SAP LinuxLab. He works extensively on Oracle9i RAC technologies for SAP, researches new SAP technologies and functionality, and certifies Dell platforms for SAP on Linux.

Wolfgang Trenkle is a senior consultant at the Dell SAP Competence Center in Walldorf, Germany, and is also a member of the Dell EMEA Enterprise Solutions Center team in Limerick, Ireland. In addition to serving as a consultant and supporting proof of concepts, Wolfgang provides training materials and tools to Dell’s global SAP community.