Oracle Real Application Clusters in Oracle VM Environments

An Oracle Technical White Paper
October 2009

ORACLE

Oracle Real Application Clusters in Oracle VM Environments

Executive Summary 3
Introduction 4
What is Oracle Real Application Clusters? 5
What is Oracle VM? 6
Why run Oracle RAC on Oracle VM? 7
Business Continuity, High Availability and Scalability 8
Deployment Methods 10
Hardware and Software Requirements 12
Oracle RAC on Oracle VM – Best Practices 16
  Oracle Installation Recommendations 16
  Generic Configuration Recommendations 16
  Storage Configuration 17
  Network Configuration 18
  Sizing the Oracle VM Environment for Oracle RAC 20
Installing Oracle RAC in Oracle VM Environments 21
Summary 22
Appendix A – Guest Configuration File Example 23
Appendix B – Bonding Setup Example 24
Appendix C – References 28
Appendix D – Table of Figures 29
Appendix E – Known Limitations 30

EXECUTIVE SUMMARY

Running today's non-critical business applications in virtualized environments has shown to be efficient and cost saving. More sophisticated or highly available applications on the other hand were most likely incompatible with commonly used software based virtualization solutions.

The availability of Oracle VM overcomes this obstacle. Providing a software based virtualization infrastructure (Oracle VM) and the market leading high availability solution Oracle Real Application Clusters (RAC), Oracle now offers a highly available, grid-ready virtualization solution for your data center, combining all the benefits of a fully virtualized environment.

Oracle VM *and* Oracle Real Application Clusters (RAC) enable the benefits of a virtualized data center infrastructure for highly available applications.

The combination of Oracle VM and Oracle RAC enables a better server consolidation (RAC databases with underutilized CPU resources or peaky CPU utilization can often benefit from consolidation with other workloads using server virtualization) sub-capacity licensing, and rapid provisioning.

It also allows the creation of virtual clusters (product demos, educational settings and sometimes test environments on the same physical machine in a virtual environment). In future, Oracle RAC in Oracle VM will even allow dynamic changes to react to changing service level requirements.

Last but not least, **Oracle VM is the only software based virtualization solution that is fully supported for Oracle Real Application Clusters.**

This paper discusses various deployment scenarios Oracle RAC and provides best practices for an optimized Oracle RAC deployment in Oracle VM environments.

Note: Oracle is constantly in the process of testing additional and advanced Oracle VM features with Oracle RAC. This paper will be updated regularly as new test results are available.

INTRODUCTION

Oracle Real Application Clusters (RAC)[1] is an option to the award-winning Oracle Database Enterprise Edition. Oracle RAC is a cluster database with a shared cache architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches to provide highly scalable and available database solutions for all your business applications. Oracle RAC is a key component of Oracle enterprise grid architecture.

Oracle RAC utilizes **Oracle Clusterware**[2] for the inter-node communication required in clustered database environments. Oracle Clusterware is the technology that transforms a server farm into a cluster. A cluster in general is a group of independent servers that cooperate as a single system. Oracle Clusterware is the intelligence in this system that ensures the required cooperation and as well a key component of Oracle enterprise grid architecture.

In a typical Oracle RAC installation, **Oracle Automatic Storage Management (ASM)**[3] acts as the underlying, clustered volume manager. It provides the database administrator with a simple storage management interface that is consistent across all server and storage platforms. As a vertically integrated file system and volume manager, purpose-built for Oracle database files, ASM provides the performance of raw I/O with the easy management of a file system. Oracle Automatic Storage Management provides the basis for a shared storage pool in Oracle enterprise grid architectures.

Oracle VM[4] is a platform that provides a fully equipped environment for better leveraging the benefits of virtualization technology. Oracle VM enables deployment of operating systems and application software within a supported virtualization environment. Oracle VM completes the Oracle enterprise grid offering by providing an Oracle RAC certified virtualization environment.

---

[1] Oracle Real Application Clusters (RAC) homepage: http://otn.oracle.com/rac
[2] For more information on Oracle Clusterware, visit http://otn.oracle.com/clusterware
[3] For more information on Oracle Automatic Storage Management (ASM), visit http://otn.oracle.com/asm
[4] Oracle VM homepage: http://www.oracle.com/virtualization

WHAT IS ORACLE REAL APPLICATION CLUSTERS?

A RAC Database is a clustered database. A cluster is a group of independent servers that cooperate as a single system. Clusters provide improved fault resilience and modular incremental system growth over single symmetric multi-processor (SMP) systems. In the event of a system failure, clustering ensures highest availability to users. Access to mission critical data is not lost. Redundant hardware components, such as additional nodes, interconnects, and disks, allow the cluster to provide high availbility. Such redundant hardware architectures avoid single points-of-failure and provide execptional fault resilience.

Oracle RAC enables the Oracle Database to run mainstream business applications of all kinds on clusters, including popular packaged products (such as Oracle Ebusiness Suite, Peoplesoft, Siebel, SAP) and in-house developed applications generating OLTP, DSS, or mixed workload.

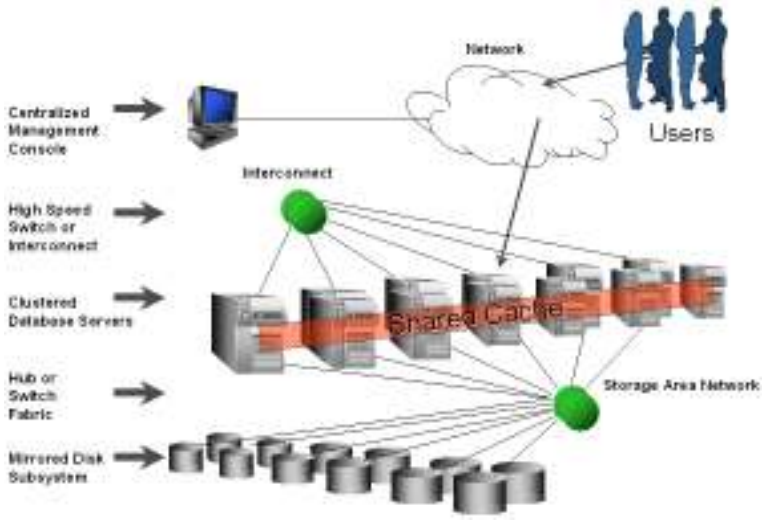

Figure 1: Oracle Real Application Clusters Overview

With Oracle Real Application Clusters (as with any other Oracle database) the Oracle Instance (the processes and memory structures allocated on a server to allow access to the data) is de-coupled from the Oracle Database (the physical structures residing on the storage, which actually hold the data. These structures are commonly known as 'datafiles').

However, a clustered database (using more than one instance) differs from a single instance database in a way that the database can be accessed by multiple instances concurrently. Each instance runs on a separate server in the cluster (formed by Oracle Clusterware).

When additional resources are required, additional nodes and instances can easily be added to the cluster with no downtime. Once a new instance has been started, applications using services can immediately take advantage of it with no changes to the application or application server.

Oracle Real Application Clusters is an extension to the Oracle Database and therefore benefits from the manageability, reliability, and security features built into the Oracle Database.

WHAT IS ORACLE VM?

Oracle VM[5] is a platform that provides a fully equipped environment for better leveraging the benefits of virtualization technology. Oracle VM enables you to deploy operating systems and application software within a supported virtualization environment. The components of Oracle VM are:

- Oracle VM Manager: Provides the user interface, which is a standard ADF (Application Development Framework) web application, to manage Oracle VM Servers[6]. Manages virtual machine lifecycle, including creating virtual machines from installation media or from a virtual machine template, deleting, powering off, uploading, deployment and live migration of virtual machines. Manages resources, including ISO files, virtual machine templates and sharable hard disks.

- Oracle VM Server: A self-contained virtualization environment designed to provide a lightweight, secure, server-based platform for running virtual machines. Oracle VM Server is based upon an updated version of the underlying Xen hypervisor technology, and includes Oracle VM Agent.

- Oracle VM Agent: Installed with Oracle VM Server. It communicates with Oracle VM Manager for management of virtual machines.

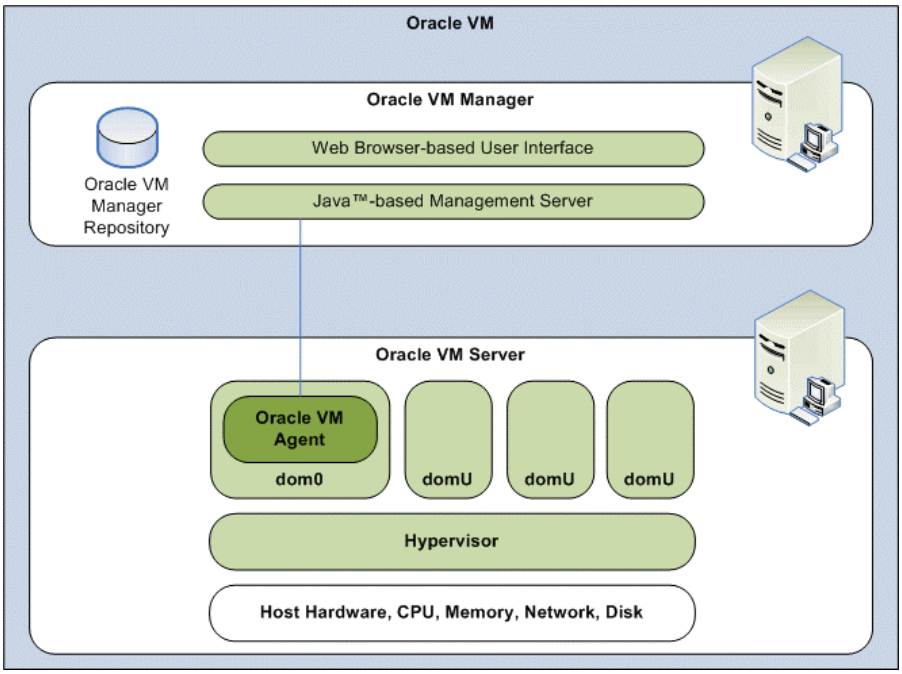

Figure 2: Oracle VM overview

[5] Oracle VM homepage: http://www.oracle.com/virtualization
[6] Oracle VM is also fully integrated into Oracle Enterprise Manager 10.2.0.5: www.oracle.com/technology/software/products/oem/index.html

WHY RUN ORACLE RAC ON ORACLE VM?

There are several reasons why customers may want to run Oracle RAC in an Oracle VM environment. Some of the more common ones are:

1. **Server Consolidation:** Oracle RAC databases with underutilized CPU resources or peaky CPU utilization can often benefit from consolidation with other workloads using server virtualization. A typical use case for this scenario foresees the consolidation of several Oracle RAC databases onto a number of machines with limited resources (e.g. CPU resources), in which case each Oracle VM hosting an Oracle RAC instance can be set up with a fixed and limited number of resources; isolated in order not to interfere with other Oracle instances sharing the same physical hardware.
2. **Sub-capacity licensing:** The current Oracle licensing model requires the Oracle RAC database to be licensed for all CPUs on each server in the cluster. Sometimes customers wish to use only a subset of the CPUs on the server for a particular Oracle RAC database. Oracle VM can be configured in such way that it is recognized as a hard partition. Hard partitions allow customers to only license those CPUs used by the partition instead of licensing all CPUs on the physical server. More information on sub-licensing using hard partitioning can be found in the Oracle partitioning paper[7]. For more information on using hard partitioning with Oracle VM refer to the "Hard Partitioning with Oracle VM" white paper[8].
3. **Create a virtual cluster:** Oracle VM enables the creation of a virtual cluster on a single physical server. This use case is particularly interesting for product demos, educational settings, and test environments. This configuration should never be used to run production Oracle RAC environments. The following are valid deployments for this use case:
    a. Test / development cluster
    b. Demo cluster
    c. Education cluster
4. **Rapid Provisioning:** The provisioning time of a new application consists of the server (physical or virtual) deployment time, and the software install and configuration time. Oracle VM can help reduce the deployment time for both of these components. Oracle VM supports the ability to create templates. These templates can then be used to rapidly provision new (Oracle RAC) systems.

---

[7] Oracle Licensing – Partitioning: www.oracle.com/corporate/pricing/partitioning.pdf
[8] Hard Partitioning with Oracle VM - www.oracle.com/technology/tech/virtualization/pdf/ovm-hardpart.pdf

BUSINESS CONTINUITY, HIGH AVAILABILITY AND SCALABILITY

Business continuity is a key element in today's business and while Oracle RAC is still the ultimate solution when it comes to Oracle Databases, there are more choices to choose from when virtualization technologies are used.

Two different kinds of High Availability (HA) mechanisms can be distinguished in an Oracle virtual environment: Oracle VM HA (external HA) and Oracle Clusterware based, internal HA (used for Oracle RAC) as illustrated in figure 3.

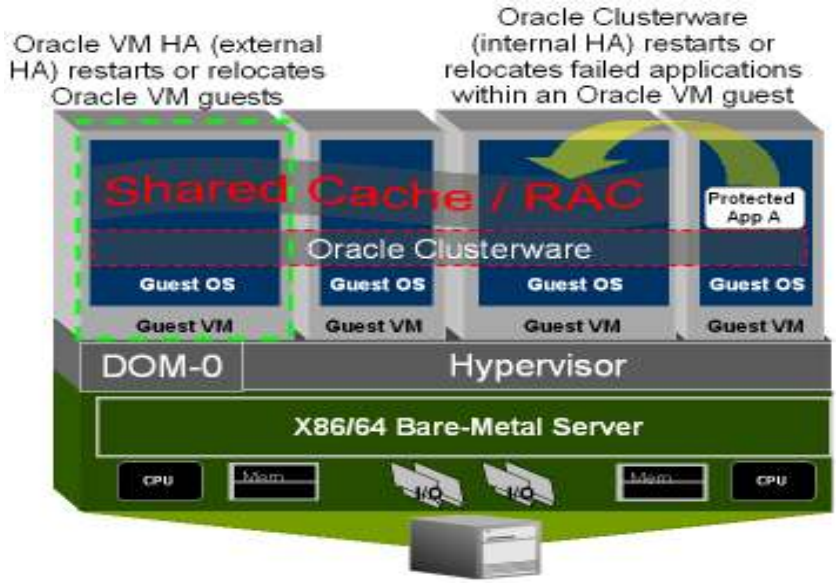

Figure 3: Oracle VM HA and Oracle RAC

In general, Oracle VM HA enables the restart of an Oracle VM guest on either the same physical machine or a different machine, if more than one physical machine is available in the server pool. However, in any case, Oracle VM would operate on the VM guest as a whole, restarting whatever is running within it.

The reason is that the Oracle VM guest is usually unaware of the applications running in the virtualized environment as much as the application is typically unaware of the Oracle VM guest it is running in.

For a fast recovery from a process or application failure within an Oracle VM guest, an internal, ideally cluster based HA solution like Oracle Clusterware (used for Oracle RAC) should be used. Utilizing application specific agents, these solutions will perform corrective actions particular to the failure without the overhead of restarting the whole Oracle VM guest.

While a combination of both such an internal HA and the external HA provided by Oracle VM HA seems to be ideal, it needs to be noted that the Oracle VM HA rule engine is currently limited in its capabilities to consider all requirements for a supported and optimal Oracle RAC (re-) placement.

Using Oracle VM HA for Oracle VM guests hosting an Oracle RAC database is therefore currently not supported.

Possible HA combinations include – currently supported solutions in bold:

1. **Oracle Real Application Clusters High Availability and Scalability**
    a. **Without additional Oracle VM guest HA**
    b. In conjunction with Oracle VM guest HA
2. **Oracle Clusterware (failover cluster) provided High Availability**
    a. Without additional Oracle VM guest HA
    b. In conjunction with Oracle VM guest HA
3. **Standalone Oracle VM guest High Availability (no Oracle RAC)**

Lately, some vendors of software based virtualization software have announced that they will provide a business continuity or better a "continuous availability" or "fault tolerant" solution that would solely be based on their virtualization software.

Those solutions are still under development and might be suitable for lightweight, stateless applications in their first release. However, they are not appropriate for production databases. This means that for production environments the choice is limited to the solutions listed above, highlighted in bold.

For **Oracle RAC production environments, solution 1a** (Oracle RAC HA and scalability without additional Oracle VM guest HA) **must be used**, which still provides the full Oracle RAC HA and scalability benefits.

However, for test systems and small development systems, solution 1b (Oracle RAC HA and scalability in conjunction with Oracle VM guest HA) can be used.

On the other end of the scale, **solution 3** (Standalone Oracle VM guest High Availability) **is fully supported by Oracle**. However, it is limited in its capability to protect the actual application within the virtual environment. It will nevertheless provide optimal protection against physical hardware failures.

Solution 2 (Oracle Clusterware [failover cluster] provided High Availability) might be an alternative when it comes to single instance databases or applications of any kind that need to be protected against hardware and process failures.

For Oracle Clusterware based solutions regardless of Oracle VM, the current support status can be found in **Oracle Metalink Note 790189.1 - Oracle Clusterware and Application Failover Management.** More information can be found on http://otn.oracle.com/clusterware

DEPLOYMENT METHODS

Historically and typically, virtualized environments were **ideal for development or test environments**. Easy to re-install by re-deploying a formerly saved base image or template, once the current environment is 'worn out' by various test runs.

Oracle VM is the ideal infrastructure for Oracle RAC used in virtual development, test, demo, or education clusters, based on only one Oracle VM host.

Oracle VM provides those benefits as part of the virtual cluster creation and the rapid provisioning approach, even when used together with Oracle RAC, as described in the previous section.

Figure 5 shows a typical Oracle RAC deployment for a development or test environment based on Oracle VM. It should be noted that **for virtual test or development clusters** Oracle allows having the 2 Oracle VM Guest Domains, each hosting 1 Oracle RAC database instance, running on only 1 Oracle VM host.

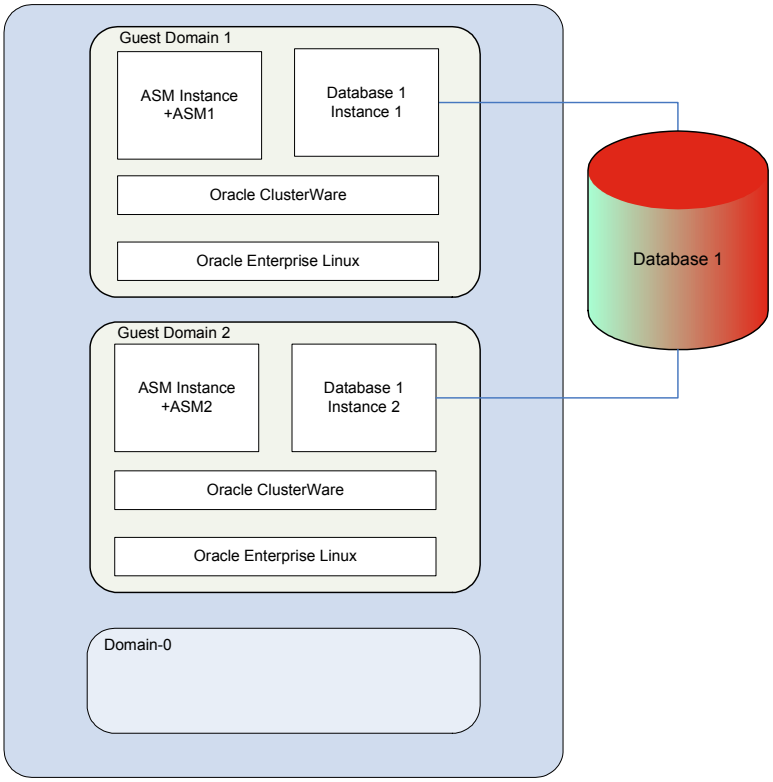

Figure 4: Oracle RAC on Oracle VM: Development Deployment example

Based on extensive tests, Oracle has **certified Oracle RAC on Oracle VM** for **production environments**. This is the first time Oracle supports Oracle RAC in software based, virtual environments and thereby enables these technologies to be used beyond the scope of pure development or test environments.

Oracle VM together with Oracle RAC enable the use of fully virtualized environments as the bases for server consolidation and highly available applications. In order to avoid a single point of failure, a minimum of 2 Oracle VM hosts should be used.

Unlike test and development environments, typical production environments would use more than one Oracle VM host for each Oracle VM Guest Domain (minimum 2), each, again, hosting an Oracle RAC database instance, as illustrated in figure 6. These kinds of configurations are as well most likely used in server consolidation environments.

Having more than one Oracle VM host as the underlying hardware platform to host the Oracle VM Guest domains with the Oracle RAC database instances eliminates the host hardware to be the single point of failure.

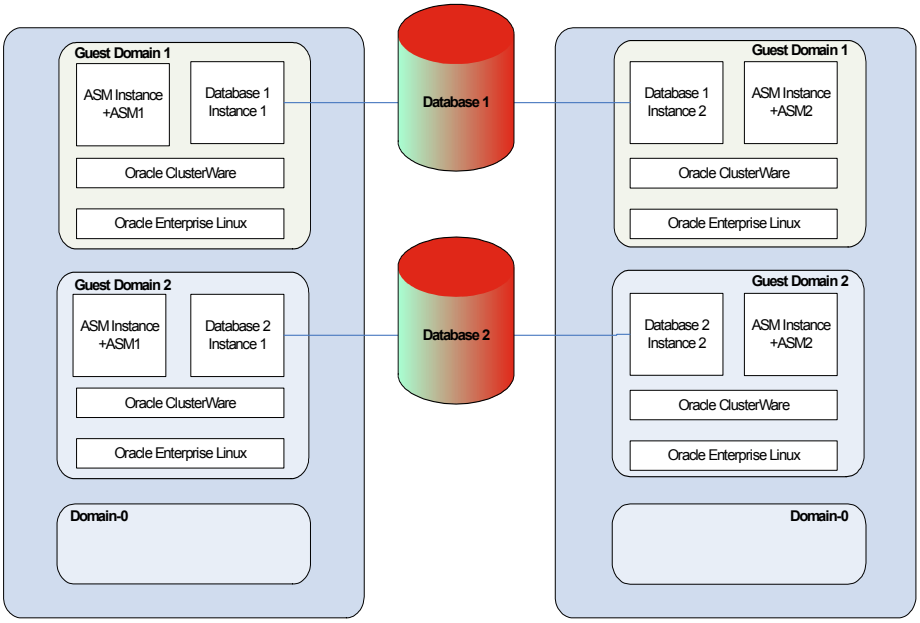

Figure 5: Oracle RAC on Oracle VM: Production Deployment example

HARDWARE AND SOFTWARE REQUIREMENTS

Oracle has currently certified Oracle RAC in Oracle VM environments based on the following hardware and software requirements. Configurations that do not fulfill these requirements are currently not supported.

Note: As a minimum, the hardware and software requirements listed in the Oracle RAC documentation must be met to deploy Oracle RAC on Oracle VM in production environments.[9]

Note: Oracle is constantly in the process of testing additional and advanced Oracle VM features with Oracle RAC. This paper will be updated regularly as new test results are available.

Hardware requirements for Production Environments:

- Minimum of 2 Oracle VM hosts are strongly recommended
    - Each host must provide a minimum of 2 Ethernet NICs for public and private communication (interconnect). 3 Ethernet NICs are required, if network based storage connectivity is used.
    - For the interconnect 1 Gbit Ethernet is required
- Network redundancy requirements will double the number or NICs given above accordingly. (4-6 NICs strongly recommended.)
- Storage redundancy requirements will require a minimum of two HBAs or SCSI controllers.

Hardware requirements for Development Environments:

- For Development or non-production environments on a single Oracle VM host, external storage is recommended, but not required.
- Network connectivity can be established within the Oracle VM host, if no external client connects are required.
- Oracle recommends using a comparable hardware setup for development environments as for production environments.

Oracle RAC Software

- Oracle RAC and Oracle Clusterware 10.2.0.4, 11g Rel. 1, **11g Rel. 2** in 32-bit and 64-bit software versions are currently supported.

[9] http://www.oracle.com/technology/documentation/index.html

General Oracle VM

- Minimum Oracle VM release: 2.1.2
  Older releases are not supported with Oracle RAC or Oracle Clusterware.
- When using Oracle RAC:
  - Dynamic VCPU and memory changes are not supported.
  - Live-migration of an Oracle RAC VM is not supported.

Oracle VM Guest Configuration

- The guest must be a para-virtualized guest
- Oracle Enterprise Linux 5.1 or higher (OEL 5.1 or higher)
- 32-bit and 64-bit Linux is currently supported for the Oracle VM Guest

Oracle VM VCPU configuration for Oracle RAC

Intensive testing has shown that a small amount of over-committing of physical CPUs will not diminish the overall stability of the system or the cluster stack. It is, however, still not recommended to over-commit CPUs. **VCPU allocation** in Oracle RAC / VM environments **must therefore adhere to the following rules**:

- Maintain Oracle VM's default VCPU allocation for dom-0:
  Oracle VM will allocate 1 VCPU for each real CPU or core to dom-0.
- The total amount of VCPUs allocated to guest domains (running Oracle RAC guests), should not exceed two times (2x) the amount of real CPUs / cores in the Oracle VM server.
- The amount of VCPUs allocated to a single guest domain should not exceed the amount of real CPUs / cores in the Oracle VM server.
- CPU pinning is only recommended for hard partitioning. If no hard partitioning is required, CPU pinning should not be used.

VCPU allocation examples

Example 1: A server with 8 cores – 2 CPUs, quad core.

Valid configuration (CPUs are over-committed in accordance to the rules)

- Dom-0 has 8 VCPUs allocated (default allocation)
- Guest domain 1 (running RAC) has 8 VCPUs allocated
- Guest domain 2 (running RAC) has 4 VCPUs allocated
- Guest domain 3 (running RAC) has 2 VCPUs allocated
- Guest domain 4 (running RAC) has 2 VCPUs allocated

Invalid configuration

- Dom-0 has 4 VCPUs allocated (non-default allocation)
    - Violation 1: non-default dom-0 VCPU allocation
- Guest domain 1 (running RAC) has 10 VCPUs allocated
    - Violation 2: allocated VCPUs (10) > real CPUs (8)
- Guest domain 2 (running RAC) has 6 VCPUs allocated
- Guest domain 3 (running RAC) has 8 VCPUs allocated
    - Violation 3: Total allocated VCPUs (24) > 2x real CPUs (16)

Example 2: A server with 24 cores – 4 CPUs, 6-core

Valid configuration (CPUs are not over committed)

- Dom-0 has 24 VCPUs allocated (default allocation)
- Guest domain 1 (running RAC) has 12 VCPUs allocated
- Guest domain 2 (running RAC) has 2 VCPUs allocated
- Guest domain 3 (running RAC) has 4 VCPUs allocated
- Guest domain 4 (running RAC) has 6 VCPUs allocated

Invalid configuration

- Dom-0 has 32 VCPUs allocated (non-default allocation)
    - Violation 1: non-default dom-0 VCPU allocation
- Guest domain 1 (running RAC) has 28 VCPUs allocated
    - Violation 2: allocated VCPUs (28) > real CPUs (24)
- Guest domain 2 (not running RAC) has 8 VCPUs allocated
- Guest domain 3 (running RAC) has 12 VCPUs allocated

Oracle VM supported storage configurations for Oracle RAC

Oracle VM itself allows configuring storage in many different ways. Oracle RAC and Oracle Clusterware are not supported on all possible storage configurations that Oracle VM offers.

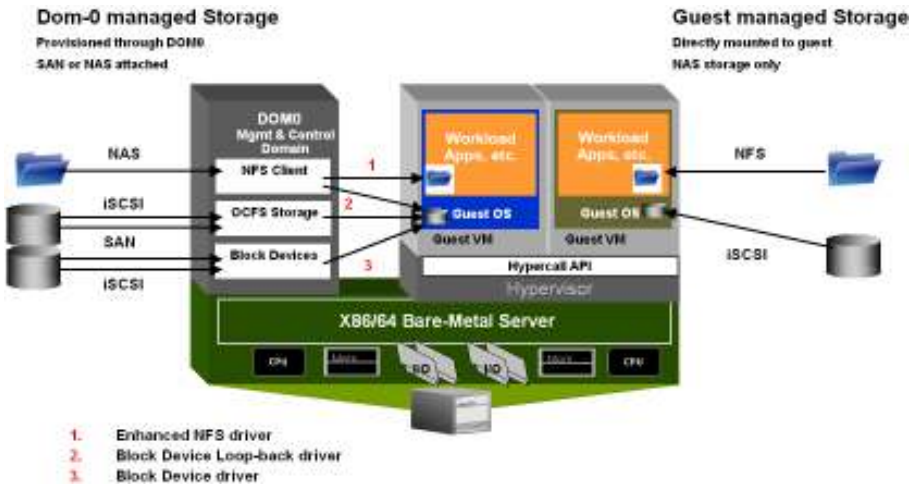

Figure 6: Storage configuration options in Oracle VM

Generally, storage configurations can be divided into two groups:

Dom-0 managed storage

In this method of storage configuration, the storage is made available in dom-0 and then made available in the guest domains using the guest's configuration file.

Guest managed storage

In this method of storage configuration, the storage is made available directly in the guest domain. It is not visible in dom-0.

Storage configuration method	Supported with Oracle RAC
Dom-0 managed storage	
• SAN / block device	Yes
• iSCSI / block device	Yes
• SAN / OCFS	No
• iSCSI / OCFS	No
• NFS	No
Guest managed storage	
• iSCSI	Yes
• NFS	Yes

Table 1: Storage options supported with Oracle RAC

ORACLE RAC ON ORACLE VM – BEST PRACTICES

Based on extensive tests in the course of the Oracle RAC for Oracle VM certification, Oracle has developed some best practices recommendations in order to run Oracle RAC in an optimized virtual environment. Below you will find some of the recommendations as a result of those tests.

Note: Oracle is constantly in the process of testing additional and advanced Oracle VM features with Oracle RAC. This paper will be updated regularly as new test results are available.

Oracle Installation Recommendations

Setting diagwait in Oracle Clusterware

When running Oracle RAC in highly stressed clusters (CPU and memory utilization), Oracle recommends setting the diagwait to 13 seconds. This prevents false evictions and enables a better diagnosis in case of false evictions.

When running Oracle RAC in Oracle VM environments, it is strongly recommended to set diagwait to 13 seconds for all installations, regardless of utilization.

Before changing the diagwait value, please, review Metalink Note 559365.1 (Using Diagwait as a diagnostic to get more information for diagnosing Oracle Clusterware Node evictions).

Generic Configuration Recommendations

Time synchronization

Per default, the Oracle VM Guest time is automatically synchronized with Domain-0. This time synchronization can allow some time drifting in the guest domains running Oracle RAC. Therefore, the following recommendations should be implemented when running Oracle RAC in Oracle VM:

- Configure `xen.independent_wallclock=1`
  in /etc/sysctl.conf in the guest domains only.
  This allows the para-virtualized guests to manage their own system clocks.
- Configure NTPD in dom-0
- Configure NTPD in all the guest domains

Please, see Metalink Note 580296.1 (Clock drift issue between Dom-0 and Dom-U on OVM Server) for more information.

Storage Configuration

Multipathing and Device Persistency in Domain-0

For Oracle RAC in Oracle VM environments, multipathed access to the (SAN) storage is highly recommended. Multipathing should be configured in Domain-0 and not in the guest domains. All supported multipathing tools can be used, e.g. Device Mapper, QLogic multipathing, etc.

Device persistency, if not already configured as part of the multipathing solution mentioned above, should also be set up in Domain-0. If the multipathing software doesn't offer device persistency, the Linux inherent udev configuration should be used. There is no need to set up device persistency in the Guest VMs. The mapping of Oracle VM disks to guest disks is done statically in the guest configuration file (see Appendix A for an example).

Storage Configuration for the Oracle VM Guests hosting the Oracle RAC database

The only configuration that is required to be set up in the Oracle RAC guest domains is the setup of the permissions on the block devices. In addition, when running Oracle RAC 10g Release 2, it is required to configure RAW devices. It is strongly recommended to use the Linux native udev tool to create the RAW devices with the right permissions on the disks in the Oracle VM guest.

Oracle recommends using Oracle Automatic Storage Management (ASM) to manage these RAW devices and as the underlying volume manager for the Oracle RAC database files. Optionally, ASMLib can be used in the guest domains in order to further simplify OS disk administration.

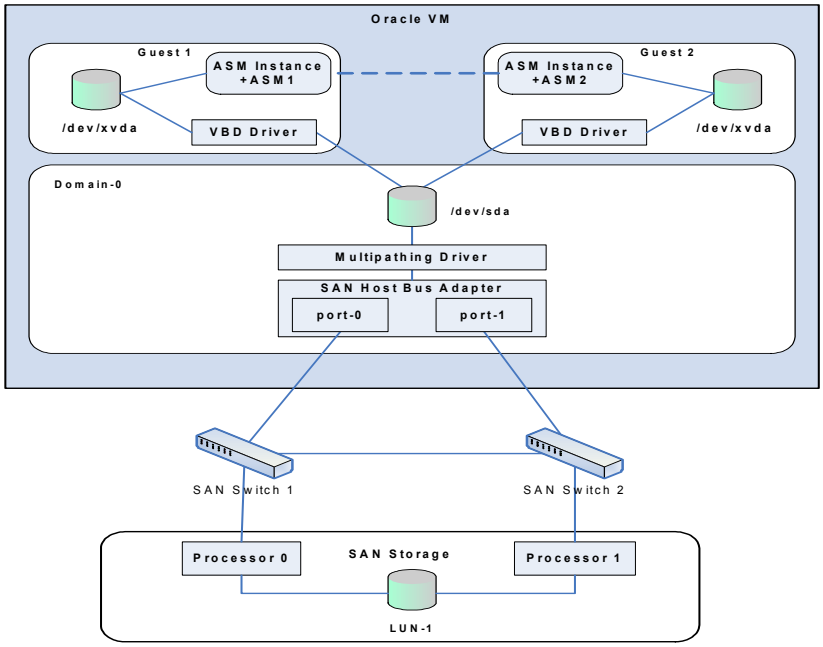

Figure 7: SAN Storage on Oracle VM

Cluster file system requirements

Some applications require running the Oracle RAC database on a cluster file system or relying on some files that need to be available on every node. For these kind of applications the Oracle Cluster File System OCFS2 should be used in the guest domains.

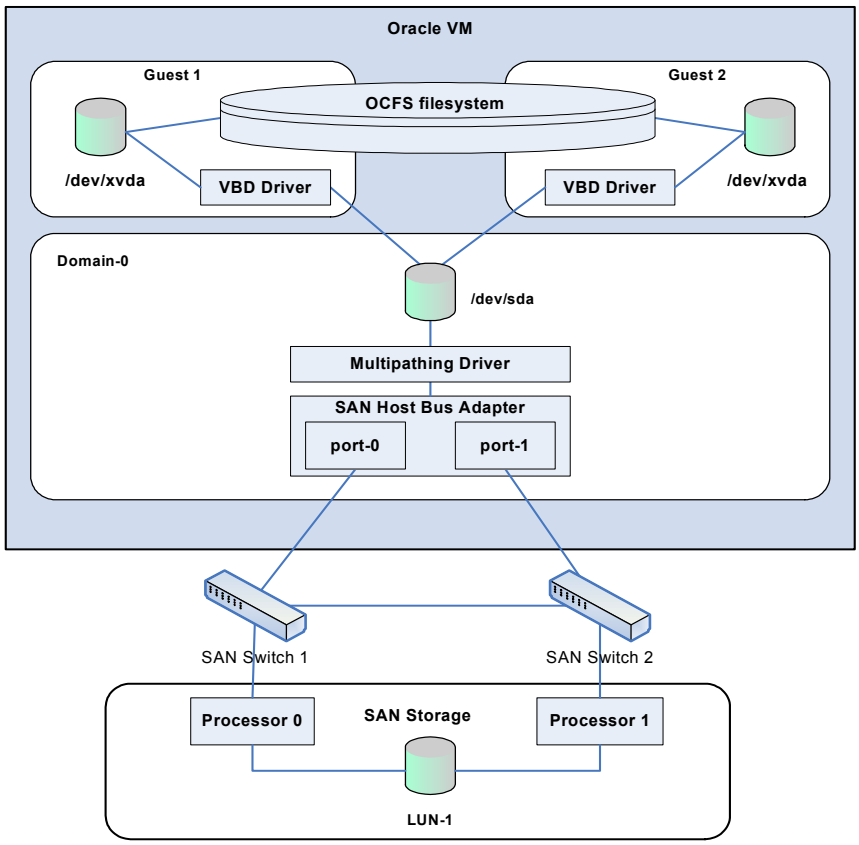

Figure 8: SAN Storage with OCFS on Oracle VM

Network Configuration

Required Networks

The Oracle VM host should have a sufficient number of NICs to provide at least two distinct bridges and virtual NICs to the guest domains. For production systems the number of NICs per Oracle VM host is specified in the "Hardware and Software Requirements" section of this paper. The actual number of NICs depends on the overall configuration.

As for any other Oracle RAC configuration, a separation of networks for the

1. Public network
2. Private network
3. Storage network (if applicable)

is required for production environments. Therefore, the Oracle VM host must ensure that the respective communication would not interfere with each other.

These networks, however, can be shared with multiple Oracle RAC VM Guests in an Oracle VM host. However, network bandwidth and capacity must be considered in this case (see "Sizing the Oracle VM environment for Oracle RAC" for recommendations).

Network Bonding

For production environments it is strongly recommended to have two Network Interface Controllers (NICs) for each network and use Linux bonding to make them highly available. This results in a requirement of a minimum of four NICs (for public and private network) per Oracle VM host running Oracle RAC as illustrated in figure 8, which also shows the additional network interfaces. Appendix B explains setting up bonding devices step-by-step.

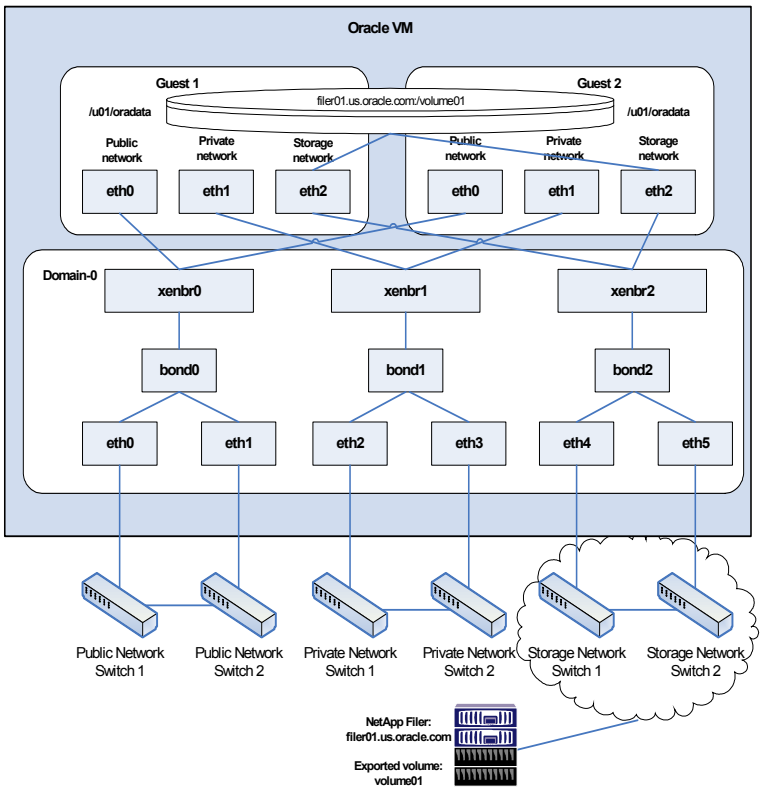

Figure 9: Full network bonding in Oracle VM

Sizing the Oracle VM Environment for Oracle RAC

As a general rule of thumb, you should "size your Oracle VM environment in a similar way as you would size a non-VM environment." Following this rule when sizing an Oracle VM system Oracle RAC should cover most of the optimization. Particular attention should be paid to the following parameters:

CPUs and cores

Follow the requirements as stated in this document for assigning VCPUs to dom-0 and guest domains. It is supported, but not recommended, to over-commit by a factor of 2.

Memory

Memory is not shared between domains in Oracle VM. When sizing memory for your Oracle RAC VM system, make sure your system memory is large enough to accommodate the memory needed for Domain-0, the Linux Operating systems, Oracle Clusterware, and the Oracle RAC databases in each of the guest domains.

Network

For an Oracle RAC in Oracle VM environments the general recommendation is to install a minimum amount of 2 NICs per Oracle VM host machine. For a production environment the recommendation is to have at least 4 network cards installed in the host and use bonding.

Keep in mind that the network bandwidth is shared between the different guest domains installed on one Oracle VM host. For performance reasons it is therefore recommended to use at least 1GBit Ethernet network cards (in production environments, 1 Gbit Ethernet is required).

When configuring a larger number of guest VMs on one host, or when deploying guest VMs that require a high amount of network bandwidth, more than the required minimum amount of network cards should be used. Load balancing should then be configured in Domain-0 to combine the network bandwidth.

Storage

When using SAN storage for your Oracle RAC in Oracle VM deployment, more than one HBA can be used to use a higher bandwidth to the storage in the Oracle VM host. The multipathing drivers usually combine the bandwidth of each path automatically.

When using NAS storage (preferably using iSCSI) it is recommended to use a dedicated network interface to access the NAS storage. The dedicated storage network should be configured in the same way as the public or interconnect network. Again, multiple NICs can be used to eliminate a single NIC as a Single-Point-Of-Failure and appropriate bonding configurations can be used to increase bandwidth.

INSTALLING ORACLE RAC IN ORACLE VM ENVIRONMENTS

The following steps describe the complete installation of Oracle RAC in an Oracle VM environment:

1. Plan your installation carefully. Consider the requirements for your new environment and plan accordingly using the guidelines and best practices in this white paper.
2. Install Oracle VM 2.1.2 on the machine(s) on which you want to perform the Oracle RAC installation. Follow the Oracle VM Server Installation Guide for the details on how to do this.
3. Configure storage, networking and time synchronization in Domain-0 of your Oracle VM host(s) following the guidelines described in this document.
4. Create the para-virtualized guests domains for the Oracle RAC installation using a physical disk partition for the root file system. Oracle VM Manager cannot be used to enable this configuration currently. Follow the instructions in chapter 4 of the Oracle VM Server User's Guide to create the guest domains.
5. Install and configure the guest operating system (Oracle Enterprise Linux 5.1 or higher). Make all necessary configuration changes to the guest operating system as described in the requirements for installing Oracle Clusterware and Oracle RAC. Use udev to configure the raw devices required for installing Oracle RAC 10g Release 2. Optionally, ASMLib can be used in the guest domains in order to further simplify OS disk administration.
6. Install Oracle Clusterware and Oracle RAC 10.2.0.4 as for non-virtualized environments, following the installation guides for these products. Pay attention to Metalink note 414163.1 ("10gR2 RAC Install issues on Oracle EL5 or RHEL5 or SLES10"), which describes some issues and workarounds for installing Oracle 10gR2 on Oracle Enterprise Linux 5.

SUMMARY

Having certified Oracle RAC in Oracle VM based virtual environments and following the recommendations made in this paper; there are no limitations in utilizing these standard technologies in (Oracle) enterprise grid infrastructures like the one illustrated in fig. 9 or even more sophisticated configurations in the future.

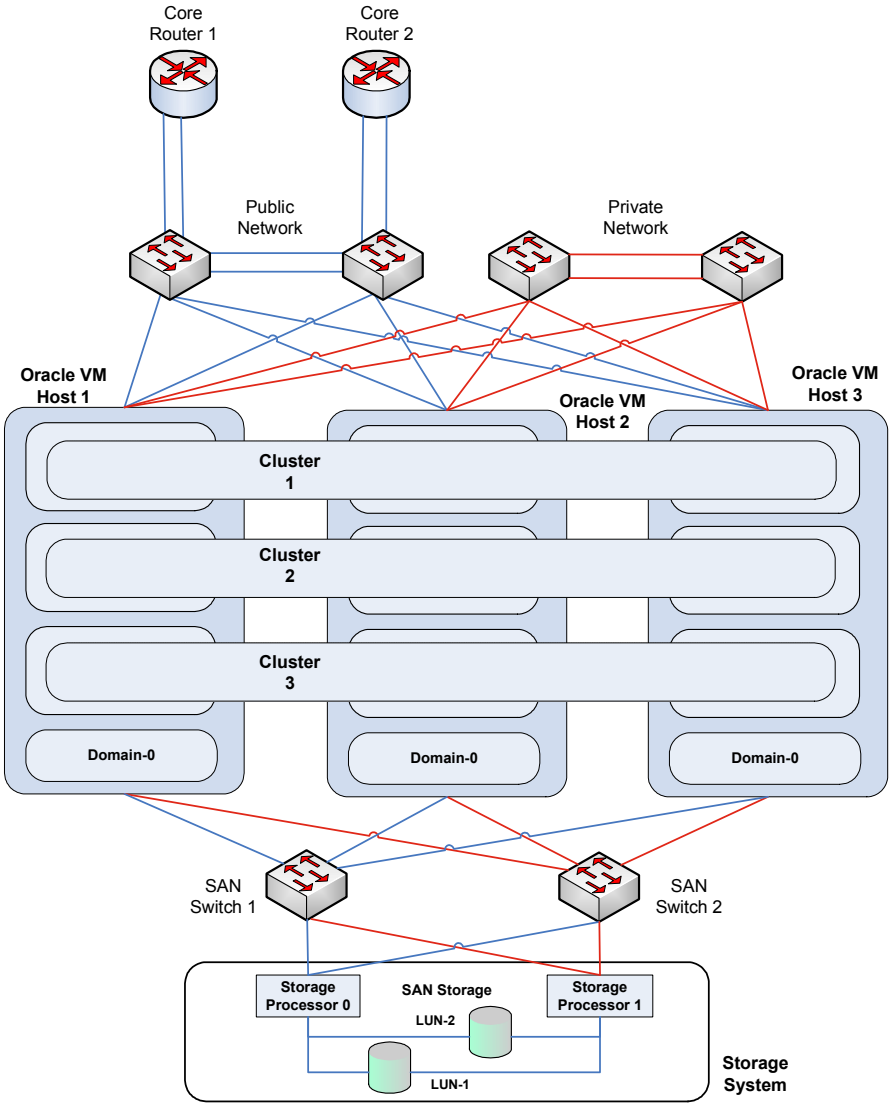

Figure 10: Production Architecture Example

APPENDIX A – GUEST CONFIGURATION FILE EXAMPLE

```
# xen config file example for RAC Guest Domain
name = "vmhost01-g01"
memory = "16384"
disk = [
'phy:/dev/shared/sdsk-a1-l20-spb,xvda,w!',
'phy:/dev/shared/sdsk-a1-l21-spb,xvdb,w!',
'phy:/dev/shared/sdsk-a1-l22-spb,xvdc,w!',
 ]
vif = [
'mac=00:16:3E:00:00:08, bridge=xenbr0',
'mac=00:16:3E:10:A5:96, bridge=xenbr1',
 ]
vfb = ["type=vnc,vncunused=1"]
uuid = "3d6f1de4-626c-e02a-42a1-458c9c17e728"
bootloader="/usr/bin/pygrub"
vcpus=8
on_reboot   = 'restart'
on_crash    = 'restart'
```

APPENDIX B – BONDING SETUP EXAMPLE

Disclaimer: Making changes to the network configuration of an Oracle VM machine to the following extend must be performed using remote or direct console access to the machine. In the course of this configuration, network connectivity to dom-0 will be lost. Direct or remote console access will be required to finish the configuration.

Introduction

The following example shows how to create two bonding devices and two bridges to be used by the guests, based on 4 Ethernet Network Interface Cards (NIC), configured in dom-0. The network devices used in this example are eth0, eth1, eth2 and eth3. This setup is meant to support an Oracle RAC running in VM guest domains. A separate public and private network is required.

The eth0 and eth1 interfaces are dedicated to "public" connectivity. These interfaces will be part of the bond0 bonding device. The bond0 bonding device will be attached to the xenbr0 bridge. The dom-0 IP address will also be configured on bridge xenbr0 in order to ensure connectivity to dom-0.

The eth2 and eth3 interfaces are dedicated to "private" connectivity. These interfaces will be part of the bond1 bonding device. The bond1 bonding device will be attached to the xenbr1 bridge. These interfaces are not used for external connectivity and therefore do not get any other IP addresses assigned.

The bonding mode used in this example is mode 1, active-backup. This is the easiest mode and will work on most switches without any problems. Any other modes might require specific switches and a certain switch configuration.

Step 1: Restoring and disabling the xend Network Configuration

Execute the following commands as root to restore the network configuration changed by xend:

```
# cd /etc/xen/scripts/
# ./network-bridges stop
```

Create a script that only returns "true" in the /etc/xen/scripts directory.
For example: **/etc/xen/scripts/network-bridge-dummy**

```
#!/bin/sh

/bin/true
```

In the /etc/xen/xend-config.sxp change the following line from:

```
(network-script network-bridges)
```

to

```
(network-script network-bridge-dummy)
```

Step 2: Loading the Bonding Modules

Add the following lines to the /etc/modprobe.conf:

```
alias bond0 bonding
alias bond1 bonding
options bonding max_bonds=2
```

Add more alias lines and increase the max_bonds parameter as required.

Step 3: Configuring the network devices and bridges for public connectivity

To configure the eth0 and eth1 interfaces, edit **/etc/sysconfig/network-scripts/ifcfg-eth0** file as follows:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
HWADDR=00:14:22:10:A5:F7
```

For **/etc/sysconfig/network-scripts/ifcfg-eth1:**

```
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
HWADDR=00:14:22:10:A5:F8
```

The HWADDR is the pointer to the network card (MAC address) being used. The SLAVE parameter defines this network card as a slave of a bond-device. The MASTER parameter points to the actual bonding device, which this network interface will be part of. Then configure the bonding device bond0 as follows:

In **/etc/sysconfig/network-scripts/ifcfg-bond0** set:

```
DEVICE=bond0
ONBOOT=yes
BRIDGE=xenbr0
BONDING_OPTS="mode=active-backup miimon=100
downdelay=200 updelay=200 use_carrier=1"
```

Note that the parameter BONDING_OPTS used in the configuration file illustrated above allows for setting the bonding options for each individual bonding interface. The BRIDGE parameter specifies that this bonding interface does not have an IP address configured to it, but will be connected to bridge xenbr0.

In **/etc/sysconfig/network-scripts/ifcfg-xenbr0** set:

```
DEVICE=xenbr0
ONBOOT=yes
STP=off
IPADDR=130.35.166.150
NETMASK=255.255.252.0
```

An IP address is assigned to xenbr0 using the IPADDR and NETMASK in order to ensure connectivity to dom-0. The parameters NETWORK and BROADCAST are deprecated. These will be automatically calculated with ipcalc.

Step 4: Configuring the network devices and bridges for private connectivity

To configure the eth2 and eth3 interfaces, edit **/etc/sysconfig/network-scripts/ifcfg-eth2** as follows:

```
DEVICE=eth2
BOOTPROTO=none
ONBOOT=yes
MASTER=bond1
SLAVE=yes
USERCTL=no
HWADDR=00:04:23:BB:54:66
```

For: **/etc/sysconfig/network-scripts/ifcfg-eth3**

```
DEVICE=eth3
BOOTPROTO=none
ONBOOT=yes
MASTER=bond1
SLAVE=yes
USERCTL=no
HWADDR=00:04:23:BB:54:67
```

Then configure the bonding device bond1 by editing **/etc/sysconfig/network-scripts/ifcfg-bond1** as follows:

```
DEVICE=bond1
ONBOOT=yes
BRIDGE=xenbr1
BONDING_OPTS="mode=active-backup miimon=50 downdelay=200
updelay=200 use_carrier=1"
```

In **/etc/sysconfig/network-scripts/ifcfg-xenbr1** set:

```
DEVICE=xenbr1
ONBOOT=yes
STP=off
```

Step 5: Restart Oracle VM

To activate the changes made to the network configuration, its is recommended to restart the machine, by executing the reboot command in dom-0.

If required: Configuring bridges on interfaces without bonding

For the purpose of completeness and since this guide explains how to use bonding, which requires to disable the xend based network and bridge configuration, in the following it is explained how to set up bridges on a non-bonded interfaces, which otherwise would be performed by xend automatically.

This example assumes a configuration of a bridge configured directly on top of an Ethernet interface, but without the use of bonding. This example uses the network interface eth4 and bridge xenbr3.

Edit **/etc/sysconfig/network-scripts/ifcfg-eth4** as follows:

```
DEVICE=eth4
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
BRIDGE=xenbr3
HWADDR=00:04:23:AA:51:23
```

In **/etc/sysconfig/network-scripts/ifcfg-xenbr3** set:

```
DEVICE=xenbr3
ONBOOT=yes
STP=off
IPADDR=192.168.20.34
NETMASK=255.255.255.0
```

The IPADDR and NETMASK parameters are optional and only required if connectivity to dom-0 needs to be established through this interface.

APPENDIX C – REFERENCES

Oracle VM on oracle.com : http://www.oracle.com/virtualization

Oracle VM on the Wiki : http://wiki.oracle.com/Oracle+VM

Oracle RAC on oracle.com: http://otn.oracle.com/rac

Oracle Clusterware on oracle.com: http://otn.oracle.com/clusterware

Oracle ASM on oracle.com: http://otn.oracle.com/asm

Oracle ASMLib:

http://www.oracle.com/technology/tech/linux/asmlib/index.html

Oracle RAC documentation:

http://download.oracle.com/docs/cd/B19306_01/install.102/b14203/toc.htm

http://www.oracle.com/pls/db111/to_toc?pathname=install.111/b28263/toc.htm
http://www.oracle.com/pls/db111/to_toc?pathname=install.111/b28264/toc.htm

http://www.oracle.com/pls/db112/to_toc?pathname=install.112/e10812/toc.htm
http://www.oracle.com/pls/db112/to_toc?pathname=install.112/e10813/toc.htm

Metalink Notes

Note 414163.1 : 10gR2 RAC Install issues on Oracle EL5 or RHEL5 or SLES10

Note 738269.1 : Enabling network bonding in Oracle VM

Note 735975.1 : Oracle VM: Configuring Quality of Service (QoS) for Guest Virtual Machines

Note 580296.1 : Clock drift issue between Dom-0 and Dom-U on OVM Server

Note 564580.1 : Configuring raw devices (multipath) for Oracle Clusterware 10g Release 2 (10.2.0) on RHEL5/OEL5

Note 465001.1 : Configuring raw devices (singlepath) for Oracle Clusterware 10g Release 2 (10.2.0) on RHEL5/OEL5

Note 790189.1 : Oracle Clusterware and Application Failover Management

APPENDIX D – TABLE OF FIGURES

Figure 1: Oracle Real Application Clusters Overview 5
Figure 2: Oracle VM overview 6
Figure 3: Oracle VM HA and Oracle RAC 8
Figure 4: Oracle RAC on Oracle VM: Development Deployment example 10
Figure 5: Oracle RAC on Oracle VM: Production Deployment example 11
Figure 6: Storage configuration options in Oracle VM 15
Figure 7: SAN Storage on Oracle VM 17
Figure 8: SAN Storage with OCFS on Oracle VM 18
Figure 9: Full network bonding in Oracle VM 19
Figure 10: Production Architecture Example 22

APPENDIX E – KNOWN LIMITATIONS

- Oracle VM Manager 2.1.2 cannot be used to create Guest VMs running Oracle RAC. This is due to the requirement for Oracle RAC to use physical devices as Virtual Block Devices (VBDs). Future versions of Oracle VM Manager will allow creation of guests using physical disks.
- No support for HugePages in Xen kernels
- File backed VBDs are not supported in an Oracle RAC on Oracle VM deployment
- Using P2V to convert from bare metal to virtual images is not supported for RAC guest creation, since P2V produces HVM images

Oracle Real Application Clusters in Oracle VM Environments
October 2009, Version 1.5
Authors: Daniel Dibbets, Markus Michalewicz
Contributing Authors: Wim Coekaerts, Adam Hawley, Rene Kundersma, Barb Lundhild, Michael Timpanaro-Perrotta, Nitin Vengurlekar

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

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