

An Oracle Technical White Paper
December 2010

Best Practices and Guidelines for Deploying the Oracle VM Blade Cluster Reference Configuration

Introduction	1
Overview of the Oracle VM Blade Cluster Reference Configuration ..	2
Reference Configuration Components.....	3
Overview of Oracle VM Concepts and Components	3
Oracle VM Templates	4
Key Deployment Concepts	4
Server Pool Planning	7
High Availability Planning.....	7
Recommendations for Deploying Highly Available Server Pools....	8
Pool Capacity, Performance, and Scalability	9
Recommendations for Server Pool Capacity, Performance, and Scalability Planning.....	12
Oracle VM Installation and Configuration	12
Configuring Sun x86 Systems with Oracle VM Pre-installed	13
Oracle VM Server Installation	13
Oracle VM Manager Installation.....	15
Network Configuration and Best Practices	15
Storage Configuration and Best Practices	16
Oracle VM Blade Cluster Reference Configuration Example	17
Conclusion	20
References.....	20
Appendix A. Network Configuration Checklist	21
Appendix B. Storage Configuration Checklist.....	25

Introduction

The Oracle VM blade cluster reference configuration addresses one of the key challenges in deploying a virtualization infrastructure. IT organizations often spend multiple weeks to plan, architect, and deploy a multi-vendor solution. Deployment teams must assemble and integrate a range of hardware and software components from different vendors (e.g. servers, storage, network, virtualization software, and operating systems). The process is not only time-consuming, but also error-prone, making it hard to get the best value from the investment.

The Oracle VM blade cluster reference configuration offers a much simpler approach that speeds deployment and reduces risk. It is a single-vendor solution for the entire hardware and software stack and can be deployed in hours rather than weeks. The reference configuration has gone through Oracle Validated Configuration testing, resulting in a pre-tested, validated configuration that can significantly reduce testing time as well as the time-consuming effort of determining a stable configuration.

This paper provides recommendations and best practices for optimizing virtualization infrastructures when deploying the Oracle VM blade cluster reference configuration. It covers deployment of software, hardware, storage, and network components and is intended to serve as a practical guide to help IT organizations get up and running quickly while maximizing the potential benefits of Oracle VM.

Overview of the Oracle VM Blade Cluster Reference Configuration

The Oracle VM blade cluster reference configuration addresses every layer of the virtualization stack with Oracle hardware and software components (Figure 1). The solution makes use of pre-configured virtual machines that contain optimized configurations for running applications on Sun Blade 6000 Modular Systems from Oracle with Oracle's Sun ZFS Storage Appliances. Tests were conducted by Oracle to validate the reference configuration using the Oracle Validated Configuration test suite. In addition, the components have also been tested with Oracle VM by the respective product teams.

Oracle VM Templates are used with this solution to simplify and accelerate deployment of the reference configuration software stack. The Templates include pre-installed versions of Oracle Linux and Oracle Solaris as well as Oracle VM Server and Oracle VM Manager. Best practices for optimizing the environment are built into the Templates, which define configurations for all components of the software stack.

To deploy the Oracle VM blade cluster reference configuration, customers simply order the recommended hardware components and then download the Oracle VM Templates for the reference configuration. The Oracle VM Templates contain ready-to-run software stacks, so the entire virtualized infrastructure can be up and running in hours as opposed to weeks.

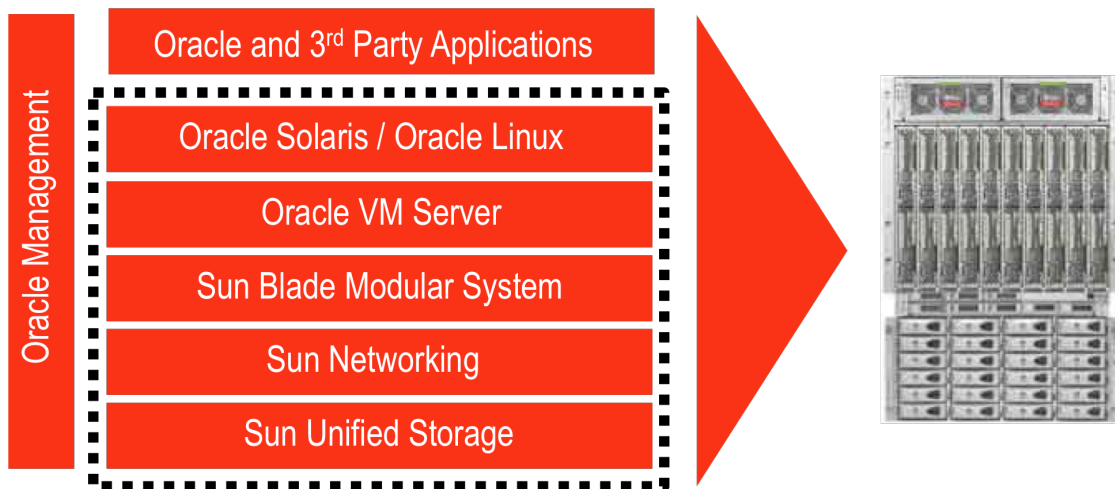


Figure 1. The Oracle VM blade cluster reference configuration provides a complete hardware and software stack that can accelerate deployment and reduce the risk of errors.

Reference Configuration Components

Table 1 includes a list of the supported components in the reference configuration along with relevant configuration details.

TABLE 1. COMPONENTS OF THE ORACLE VM BLADE CLUSTER REFERENCE CONFIGURATION		
STACK LAYER	RECOMMENDED ORACLE PRODUCT	CONFIGURATION DESCRIPTION
Operating System	Oracle Solaris and Oracle Linux	<ul style="list-style-type: none"> Oracle Solaris 10 Oracle Linux 5
Virtualization	Oracle VM Server for x86	<ul style="list-style-type: none"> Oracle VM Server 2.2.1 runs on each compute blade Oracle VM Manager 2.2 running as a VM, HA enabled Oracle VM Templates offer pre-installed and pre-configured software images
Server hardware	Sun Blade 6000 Chassis Sun Blade X6275 M2 Server Module Sun Blade X6270 M2 Server Module	<ul style="list-style-type: none"> 10 single- or dual-node server modules per chassis Start with 2 nodes and scale up to 32 nodes in one Oracle VM server pool Scale to multiple server pools
Networking	Sun Blade 6000 Ethernet Switched NEM. 24p 10GbE	<ul style="list-style-type: none"> 10 10GbE downlinks: 1 link to each blade server via Fabric Expansion Module (FEM) 14 10GbE uplinks: 2 SFP+, 3 QSFP (Quad Small Factor Ports) (equiv to 12 10GbE ports)
Storage	Oracle's Sun ZFS Storage Appliances <ul style="list-style-type: none"> Sun ZFS Storage 7120 Sun ZFS Storage 7320 Sun ZFS Storage 7420 Sun ZFS Storage 7720 	<ul style="list-style-type: none"> Available in different configurations to meet a variety of needs for capacity, price, and performance Use NFS over high-speed 10Gb Ethernet interfaces for Oracle VM Servers to access shared storage

Overview of Oracle VM Concepts and Components

To fully understand the rationale behind the described best practices and their implications, it is important to review some of the key concepts and components of the Oracle VM solution. For x86 servers, Oracle VM consists of Oracle VM Server for x86 and Oracle VM Manager.

- **Oracle VM Server** for x86 installs directly on server hardware with x86 processors (Intel or AMD) that support PAE (Physical Address Extension) and does not require a host operating system. It requires HVM (Hardware Virtual Machine) support (Intel VT or AMD-V) on the underlying hardware platform in order to run HVM guests. The Oracle VM Agent for server management is installed with Oracle VM Server.
- **Oracle VM Manager** is a Java-technology-based management server. It uses an Oracle database as its management repository, which can be installed either on the management server or a separate server. Oracle Database Express Edition (XE), Standard Edition (SE), Enterprise Edition (EE), and

Real Application Clusters (RAC) are supported as the management repository. Oracle VM Manager manages the virtual machines running on Oracle VM Server through the Oracle VM Agent.

Although not discussed in this reference configuration, Oracle VM Servers can alternatively be managed using the Oracle Enterprise Manager Grid Control Oracle VM Management Pack, which is licensed separately from Oracle VM. The Oracle VM Management Pack provides the same management functionality as Oracle VM Manager, but is integrated into Oracle Enterprise Manager Grid Control. This enables management and monitoring of Oracle VM environments alongside all other Oracle and third-party products managed by Enterprise Manager. This integrated management approach offers deeper insights into the health, performance, and configuration of the hardware and software stack.

Oracle VM Templates

Oracle VM provides the ability to rapidly and easily deploy a pre-built, pre-configured, pre-patched guest virtual machine (or multiple machines depending on the application). The guest VM can contain a complete Oracle software solution along with the operating system and related software infrastructure. These guest VMs, called Oracle VM Templates, are available from Oracle's E-Delivery website and are ready to download and run. Already configured for production use, Oracle VM Templates can save users days or weeks learning to install and configure a sophisticated product such as Siebel CRM or Oracle Enterprise Manager Grid Control. Instead, users can simply download and start the VM to begin using it right away.

Within these Templates, Oracle software is laid-out in the same manner as if the software had been installed and patched manually. The exact same directories and Oracle "homes" are used, and the package and patch inventories are completely standard and up-to-date so that no changes to operations procedures are required to maintain the instances over time. Accordingly, Oracle VM Templates can also be fully customized post-install and then re-saved as "golden image" Templates in Oracle VM. Such Templates can serve as a user's enterprise deployment standard to minimize risks and variation across multiple instance deployments.

An up-to-date list of available Templates can be found on Oracle VM Templates website at <http://www.oracle.com/technetwork/server-storage/vm/templates-101937.html>.

Key Deployment Concepts

From a deployment perspective, multiple Oracle VM Servers are grouped into server pools as shown in Figure 2. Every server in a given pool has access to shared storage, which can be NFS, SAN (Fibre Channel) or iSCSI storage. This allows VMs associated with the pool to start and run on any physical server within the pool. Typically this is the server that has the most resources available, or a server that closely matches the resource requirements of the VM. Given the uniform access to shared storage mounted under the /OVS directory in which all resources will reside, VMs may also be securely Live Migrated or automatically started or restarted across any servers in the pool. In this reference configuration, Sun ZFS Storage Appliance offers a simple high-speed 10Gb interface for Oracle VM Servers to share storage via NFS.

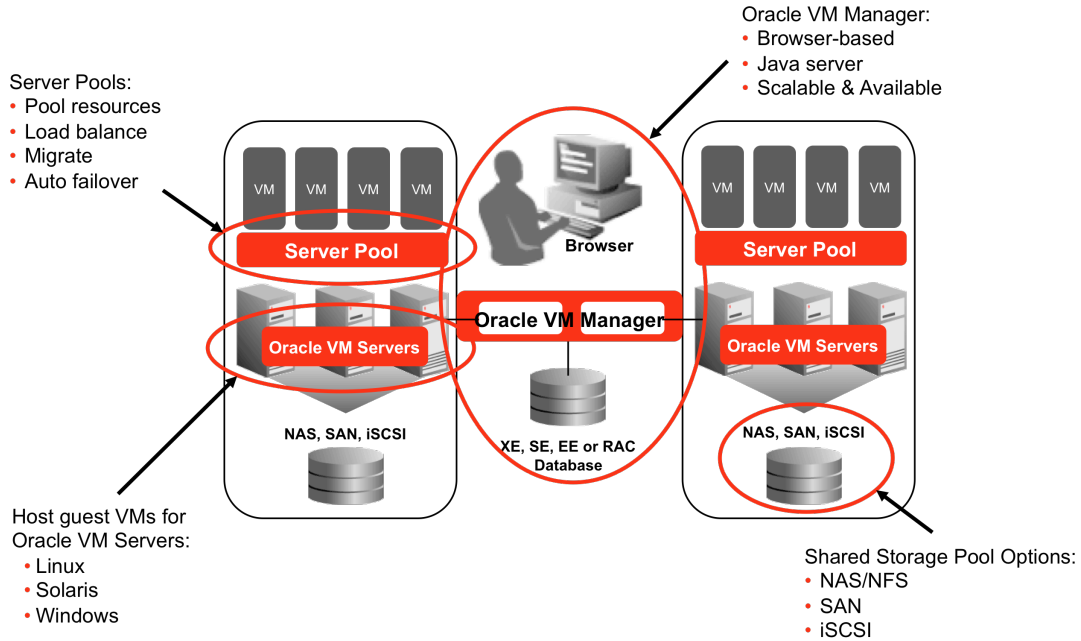


Figure 2: Oracle VM Deployment

VMs are associated with a given server in the pool. This association is made dynamically at power-on and is based on load balancing algorithms or on a user-defined list of named servers called the Preferred Server list. The Preferred Server list identifies which servers are to be used as a host for specific VMs. When VMs are powered-off and not running, they are not associated with any particular physical server. Powered-off VMs simply reside in the shared pool storage.

As a result of this architecture, VMs can easily start-up, power-off, migrate, and/or restart without being blocked by the failure of any individual server or, by the failure of multiple servers as long as there are adequate resources in the pool to support the requirements for all VMs to run concurrently.

As shown in Figure 3, there are different roles of Oracle VM servers in the pool. These roles, which are defined below, are implemented by the agent running on the Oracle VM server. Multiple roles can be combined in a single server, but there is only one Server Pool Master in the server pool at any given time.

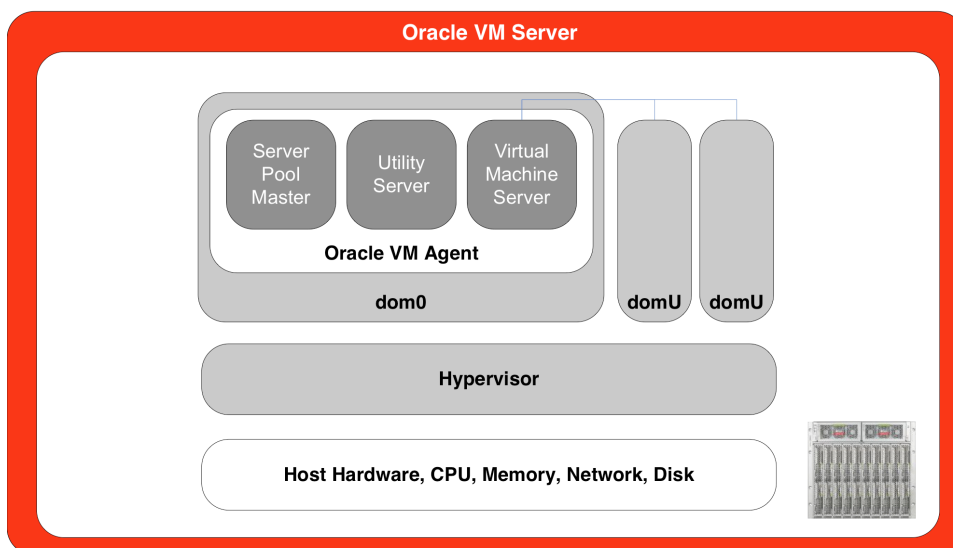


Figure 3: Oracle VM Server Architecture

Server Pool Master

In each server pool, there is exactly one Server Pool Master at any given time. It coordinates a number of the activities of the pool, particularly when that action requires coordination across multiple servers. This includes such key activities as coordinating Secure Live Migration actions as well as Guest VM HA (high availability) auto-restart, and power-on actions amongst others.

All of the Oracle VM agents in the pool communicate directly with the Server Pool Master, which, in turn, communicates with the Oracle VM Manager. This architecture provides a number of benefits including high management scalability in large environments as well as higher availability at the Oracle VM Manager instance level because functionality is distributed and isolated to minimize the impact of any single failure. For example, a management server outage does not prevent the ability of the pool master(s) to complete Secure Live Migration tasks or to automatically fail over/restart failed VMs. Similarly, a pool master outage on one pool does not affect the operation of another pool.

By default, the first server added to a pool is assigned the role of the Server Pool Master. With Oracle VM 2.2, the Server Pool Master can perform auto-failover when server pool is HA enabled and a virtual IP address is set for the server pool.

Utility Server

In each server pool, there can be one or multiple servers designated with the Utility Server role. Utility Servers are responsible for performing resource-intensive copy or transfer activities in the pool. For example, cloning activities, creating VMs from Oracle VM Templates, or saving existing VMs as templates all involve either copying an existing VM image or moving an image from one directory to another. Depending on the size of the files involved, this activity can be resource-intensive. It may be desirable to off-load this activity from servers that are hosting production VMs so as to minimize or

eliminate any service level impact. Multiple servers in a pool can be designated as Utility Servers to provide better load balancing and availability as described later in this paper.

Virtual Machine Server

Virtual Machine Servers are simply servers that host VMs. At least one server in a pool must be a Virtual Machine Server. Depending on total server capacity requirements, it may or may not be separate from the Server Pool Master and/or the Utility Server(s) in the pool as described later in this document.

Server Pool Planning

There are a large number of considerations when planning the virtual infrastructure and one size does not fit all. This section provides some considerations and guidelines to help develop a plan that is well suited to an organization's unique requirements.

It may be helpful to think of the server pool as if it were one big server with an aggregate amount of CPU, memory, storage, and network bandwidth. As such, planning for deploying VMs into a pool is much like planning for a server consolidation. It involves deciding how much aggregate capacity is needed to support normal and peak workloads as well as what types of workloads are appropriate to share the pool or server. Workload profiles should be considered in addition to how predictable or unpredictable the workloads may be.

There are also some significant similarities between server pool planning and physical server planning with regards to node (physical server) size versus overall pool size. For example, in some cases it is better to have relatively fewer but larger servers in a pool. In other cases, a greater number of relatively small servers or blades is a better fit. Both deployments may provide the same aggregate CPU, memory, storage, and bandwidth but the implications of the deployment in a pool can be different.

For the current Oracle VM 2.2 release, each server pool must have its own shared storage resources that can be accessed by Oracle VM servers within the same pool. A separate server pool must have its own separate shared storage.

High Availability Planning

Oracle VM provides the following features to provide maximum up time for VMs running in server pools:

- **Guest VM HA** — Auto-restart on server or VM failure.
- **Secure Live Migration** — Move VMs off of servers that are undergoing planned maintenance.
- **Automatic pool load balancing on VM start-up** — At VM power-on, an algorithm dynamically assigns the host server for the VM in order to load balance, but also to avoid a down server blocking VM start.

Figure 4 highlights how these availability features are typically deployed. The features should be used to collectively maximize up time of guest VMs running in a pool. The details of these features and capabilities are the subject of another white paper titled, “[Oracle VM – Creating & Maintaining a Highly Available Environment for Guest VMs](http://www.oracle.com/us/technologies/026999.pdf),” which is available at <http://www.oracle.com/us/technologies/026999.pdf>.

The following section summarizes best practices and considerations for planning server pools.

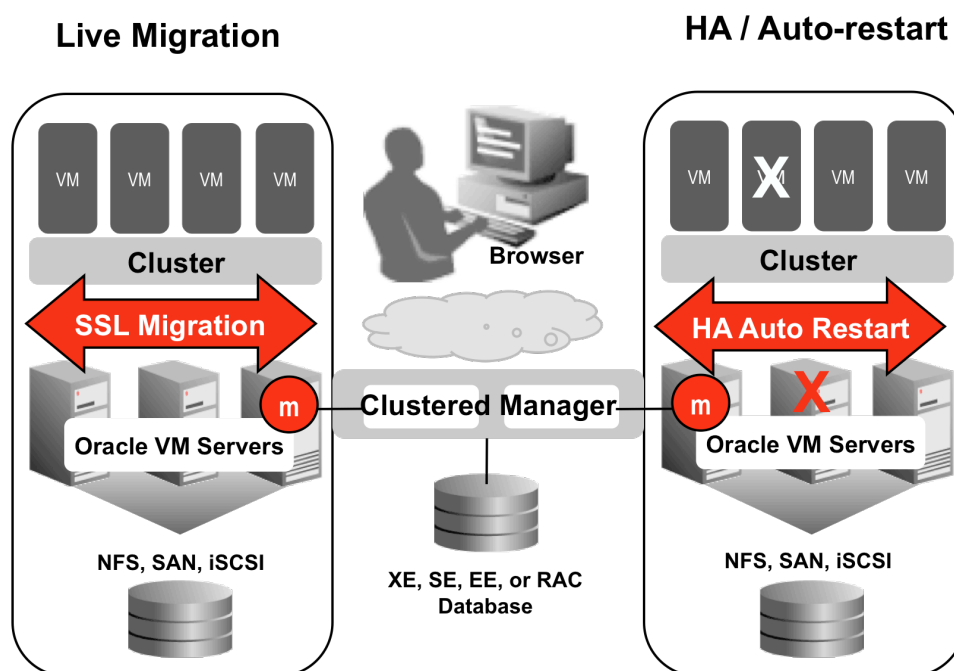


Figure 4. High availability helps minimize downtime due to planned or unplanned events

Recommendations for Deploying Highly Available Server Pools

The following steps are recommended to achieve high availability:

- Enable the “High Availability (HA)” option at both the server pool level and individual VM level to ensure VMs are automatically restarted after an unplanned failure. If the Server Pool Master fails in a High Availability set up of Oracle VM 2.2 environment, another Oracle VM Server is automatically selected from the server pool to act as the Server Pool Master.
- Plan to use the Secure Live Migration feature to migrate VMs in support of planned events like server maintenance to prevent any service outage.
- Plan for enough excess capacity in aggregate across the pool to support running all VMs to an appropriate service level even when one or more servers in the pool are out of service. Up to 32 nodes are supported within a server pool.
- Plan for multiple server pools when there is a need for more than 32 physical servers.

- Enable HA and set up virtual IP (VIP) for the server pool to take advantage of the Server Pool Master auto-failover to minimize risk and thus maximize availability of the management services.
- Configure two or more dedicated and load-balancing Utility Servers to provide continuity of service in the event of a utility server outage. There can be multiple Utility Servers in a pool and Utility Servers are automatically load balancing. If one Utility Server fails, the remaining Utility Server(s) can continue without any disruption of utility services (VM cloning, VM import, Create VM from Template, etc.).

Pool Capacity, Performance, and Scalability

Capacity planning for a server pool is similar to capacity planning for a physical server. However, the following additional considerations are also important when planning capacity for a server pool:

- **Plan extra capacity to support Guest HA/ auto-restart.** There should be sufficient capacity to support hosting additional VMs on relatively fewer machines in the event that one or more of the servers fails and its VMs end up being restarted on the remaining, healthy servers, if only temporarily.
- **Plan for extra capacity to support Live VM Migration during planned events.** When performing maintenance on a server (or servers) in the pool, Live Migration allows administrators to migrate VM(s) to another server in the pool without interrupting service. To take advantage of this capability, there should be enough excess capacity in aggregate across the pool so that a server can be taken offline (after migrating its VMs) without inappropriately impacting service levels.

Determining How Many Servers or VMs a Pool Should Contain

The number of servers or VMs that are ideal for a pool depends on a number of factors that can vary greatly between datacenters and deployments. There is no one correct answer to this question, but there are several factors that should influence such decisions. Some considerations are described below.

Storage topologies, performance, and implementation

Oracle VM pools require that all servers in a pool have shared access to the same storage so that VMs can be moved around easily. This means that server pools must use shared storage such as NFS, OCFS2 (Oracle's Cluster File System), or SAN (iSCSI or FC) storage. In the Oracle VM blade cluster reference configuration, Sun ZFS Storage Appliances are used to provide NFS shared storage. Both the physical make-up of the storage devices and the scalability of the file systems used will dictate how many servers are practical for a given shared-storage pool without adversely affecting I/O performance.

When evaluating how many servers can share a given storage topology, the following questions should be considered:

- How much I/O will each server generate and can the throughput and latency needs be accommodated through the designated NIC or HBA ports?
- How much I/O can the storage device or devices support?
- What are the HA needs (bonding or multipath)?
- Are there any application requirements for directly accessed storage?

Of course, the answers to these storage questions depend on the I/O environment. Is the application I/O intensive? What is the average size of an I/O request?

Determining a realistic number of storage nodes in a storage cluster requires considering similar questions:

- How much I/O will each storage cluster node support?
- How much I/O will each server generate to/from the clustered file system?
- How many servers will be accessing the clustered file system?

Workload profile

Is the workload flat and stable or variable and peaky? The ideal VM has relatively low utilization and is very flat and stable with minimal peaks. These types of VMs can often be very tightly consolidated, with a large number of VMs per server. Since they are very predictable they require little excess capacity or headroom to be available to accommodate unexpected peaks.

The next best scenario is when consolidating multiple VMs that may have peaks, but the peaks are very predictable in both timing and magnitude. For instance, some VMs may contain applications that always peak at the end of the week or end of the month. If those VMs could be consolidated with other VMs that peak at exactly the opposite time – workloads that peak at the beginning of the week or month– they can potentially be packed fairly tightly to maximize the number of VMs per server and per pool.

The worst case scenario is when the VMs are highly variable in load and in timing. In this situation, it is likely that a comparatively large amount of extra headroom will be needed on the servers. Thus fewer VMs will be able to be accommodated per server and per pool.

Service level support strategy

Sometimes service level objectives may dictate that there is enough planned excess capacity to support normal service levels even if all the VMs peak at the same time. This is the most conservative option, but also the most expensive since it requires extra hardware that may not be utilized much of the time. Another alternative is to plan for the average load and accept any performance hit based on resource contention if there is too much of a peak. This certainly reduces hardware expense, but may not provide acceptable service levels if the workloads are too unpredictable or are mission-critical. As a

result, many datacenters plan their capacity to support some percentage of the aggregate peak load. For example, they may plan for 40-60% of the peak above average. This is often a good compromise between meeting service levels and having reasonable hardware utilization. However, it clearly depends on how critical service levels are for the given workloads.

Dynamic server pools

Understanding that the server pool is dynamic, keep in mind that VMs are not associated with any one physical server until they are powered-on and placed on a server. Typically they are placed on the server that currently has the most memory free. Based on HA events or Live Migrations, VMs can be moved around within the pool. Thus a given VM can end-up sharing a server with any other VM in the pool unless policies implemented to restrict which servers can host which VMs. This means that the capacity plan really needs to be at the pool level, not the individual server level. So it may be best to consider keeping highly volatile VMs in their own pool or restricting them to a subset of the pool where a relatively large amount of excess capacity can be maintained for handling unpredictable peaks.

Conversely, highly predictable VMs should be restricted to a separate pool where the resources can be very tightly planned for high utilization without the need for much excess capacity.

Server affinity policies

Consider the impact of VM placement. Oracle VM allows administrators to implement policies on an individual VM basis to restrict which servers are allowed to host that particular VM. Preferred Server lists are used to implement these policies. They contain a list of explicitly named servers that are allowed to host specific guest VMs within a server pool.

Preferred Server policies are typically implemented to assure that two components of the same application stack are not hosted on the same server in order to maximize availability. These policies are respected for all actions within the pool, including Live Migrations, Power-On, and HA / Auto-Restart. One consideration in using Preferred Server policies is that it can have the effect of increasing the number of servers required in the server pool to support HA and Live Migration.

Additionally, for any server in the pool, the Virtualization Server role can be removed to prevent that server from hosting any VMs, although this is typically only done to create dedicated Utility Servers

To give a simplistic example, when deploying a VM application with two clustered components, it is a good idea to create a Preferred Server policy for each VM. Such a policy assures that both of the nodes would never be on the same physical server at the same time. Yet it is still important to make sure there is enough capacity to fail-over and to generally support peak loads. It is also important to have the option to Live Migrate either component (VM) to another server to support server maintenance without any application down time. In this simplistic case, there would need a minimum of four nodes since each of the two VMs must have two servers available. The first server would act as the primary server for running the first VM and another server would be required for migrating or re-starting the first VM in the event of a failure. The second VM would similarly require its own two servers for primary and migration/re-start. Without the restriction that both VMs never operate on the same

server at the same time, the two VMs could be hosted on just two servers and still have HA and migration support.

Recommendations for Server Pool Capacity, Performance, and Scalability Planning

The following guidelines are recommended for designing and sizing the server pool:

- Plan for excess resource capacity at the pool level to support advanced features such as Live Migration and guest VM HA.
- When determining the number of nodes in a server pool, consider storage topologies and their characteristics as well as network requirements, workload characteristics, and HA needs.
- Consider that using Preferred Server VM placement policies may increase the number of nodes required in a server pool.
- Plan excess capacity according to business requirements for meeting peak loads versus only a proportion of the peak load.
- Memory capacity is the most critical resource. I/O capacity and then CPU capacity are the second and third priorities respectively. CPU and I/O should be balanced given that I/O activity is often CPU intensive.
- The amount of memory required for all running VMs must never exceed the amount physically available on the server(s) in the pool. However, the total amount of memory required by all VMs assigned to the pool (running + powered off) may exceed the total physical memory since powered off VMs do not consume memory.
- CPU over commitment is supported (e.g. more virtual CPUs configured than physically present). The over commitment ratio is dependent upon workload requirements (see the *Workload Profile* section). However, the recommendation is to keep the ratio of virtual CPU to physical CPU at 2:1 or less.
- Determining the best physical server node size for a pool is a combination of factors depending on workload characteristics as well as the operational and budgetary issues of the datacenter.
- Use identically configured server nodes throughout the pool to support consistent performance and feature set regardless of individual server failure(s). In order to support the best agility and flexibility, it is also recommended that every server node in the pool be identical in capacity and configuration. This approach ensures that no matter where live migration or (re-) start occurs, the same performance and features are uniformly available.

Oracle VM Installation and Configuration

The sections below discuss the installation and configuration of Oracle VM Server and Oracle VM Manager. If the Sun x86 systems are shipped with Oracle VM pre-installed, they can be easily configured for a new virtual infrastructure or can be easily added to an existing Oracle VM environment.

Configuring Sun x86 Systems with Oracle VM Pre-installed

Select Sun x86 systems offer Oracle VM Server software as pre-install option with three virtual machines:

- Oracle VM Manager
- Oracle Linux
- Oracle Solaris

If an Oracle VM Manager is installed as part of the existing virtualization infrastructure, the new Oracle VM server can be registered to the existing Oracle VM Manager for creating a new server pool or be added to an existing server pool.

However, if there's no Oracle VM Manager in the environment, on-screen prompts will guide the user to start the Oracle VM Manager instance from the pre-installed image when the new system boots initially. For further details, please refer to the chapter, "Configuring the Preinstalled Oracle VM Software," of the server installation guide of the respective server platform (e.g. [Sun Blade X6270 M2 Server Module Installation Guide](#)).

Oracle VM Server Installation

If the systems do not come with Oracle VM pre-installed, installation of Oracle VM Server and Oracle VM Manager is a straightforward process. Oracle VM Server, Oracle VM Manager, and Oracle VM Templates can be downloaded from the [Oracle® E-Delivery](#) Web site at <http://edelivery.oracle.com/oraclevm>.

It could take a few minutes to complete the installation depending on the hardware configuration and installation methods. To get started, there must be a minimum of one server with a static IP address to install Oracle VM Server. Then Oracle VM Manager can run as a guest VM created from the Oracle VM Template for Oracle VM Manager, which also requires a static IP address for the Oracle VM Manager virtual machine.

It's also required to check if the server has the Hardware Virtual Machine (HVM) support. For example, Sun Blade X6270 M2 and X6275 M2 Server Modules are HVM capable. HVM capability can be identified from Intel or AMD web site if the CPU models are given. Usually there are system BIOS settings to enable the hardware virtual machine (HVM) feature. If Oracle VM 2.2 server is installed, `xm info` command can be used to verify if HVM is enabled. For example, "HVM" is present as the attribute of `virt_caps`.

```
# xm info
release : 2.6.18-128.2.1.4.27.el5xen
virt_caps : hvm
xen_major : 3
xen_minor : 4
xen_extra : .0
xen_caps : xen-3.0-x86_64 xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-x86_32p hvm-3.0-x86_64
```

Oracle VM Server installation is like a regular Linux OS installation. Oracle VM server software is installed onto on local disk (/boot, swap, /, and /OVS). By default, Oracle VM Server installer creates four partitions, for /boot, /, swap, and /OVS, with 3GB allocated for root /.

When asked to provide network configuration, make sure that a fully qualified domain name is given for the Oracle VM server (e.g. vmserver1.example.com). The real host name (vmserver1 for example) must not be associated with the loopback address (127.0.0.1). So the /etc/hosts file would look like:

```
127.0.0.1    localhost.localdomain localhost
192.168.100.100 vmserver1.example.com vmserver1
::1 localhost6.localdomain6 localhost6
```

Make sure that Oracle VM servers are listed in the entries of the DNS server; If DNS is not used, make sure the correct setting in /etc/hosts for all the servers in the pool. If DNS is used for all servers, but DNS was not specified during the server installation, please update /etc/resolv.conf file and add the domain name in it as

```
search example.com
domain example.com
nameserver your-dns-ip-address
```

Please note that all the servers in the same pool must have the consistent name resolution, either by DNS or by file (/etc/hosts). Mixed name services for the Oracle VM servers in the same server pool should not be used. For example, some have DNS, while others use /etc/hosts to resolve host names.

Once the server installation is completed, the installer should run up2date to connect to ULN ([Oracle Unbreakable Linux Network](#)) to update the Oracle VM server and its Oracle VM agent.

If there's just one server in the server pool, no further action is needed. However, if there is a need to create a server pool with multiple servers, the local storage repository must be removed since external shared storage will be used:

Find out if there's any local storage repository:

```
# /opt/ovs-agent-2.3/utlis/repos.py -l
```

Remove the local storage repository:

```
# /opt/ovs-agent-2.3/utlis/repos.py -d uuid
```

Then the external shared storage repository offered by Sun ZFS Storage Appliance (NFS) should be created. From the designated server pool master, run the commands:

```
# /opt/ovs-agent-2.3/utlis/repos.py -n nfserver:/your-directory
```

Set the cluster root:

```
# /opt/ovs-agent-2.3/utlis/repos.py -r uuid
```

The Oracle VM Documentation provides additional details.

- [Preparing External Storage and Storage Repositories](#)
- [Managing Storage Repositories](#)

Oracle VM Manager Installation

A quick and easy way to install Oracle VM Manager is to deploy [Oracle VM Template for Oracle VM Manager](#) as a virtual machine. The Oracle VM Manager VM requires a static IP address. The pre-built Oracle VM template for Oracle VM Manager can be downloaded from the [Oracle E-Delivery](#) Web site. The VM running Oracle VM Manager also requires a static IP address. Before running the deployment script of Oracle VM Template for Oracle VM Manager, be sure to read the README. Once the template is deployed, simply use a browser to access the Oracle VM Manager:

```
SSL Enabled:
https://oracle-vm-manage-host-name-or-ip-address:4443/OVS
SSL not Enabled:
http://oracle-vm-manage-host-name-or-ip-address:8888/OVS
```

If the Oracle VM Manager will be installed into an existing Oracle Linux environment, just download the Oracle VM Manager ISO image, and follow the standard installation.

```
# mkdir /OVMCD
# mount -o loop,ro OracleVM-Manager-2.2.0.iso /OVMCD
# cd /OVMCD
# sh runInstaller.sh
```

After the installation is completed, remember to download and install the [tightvnc-java rpm](#) onto the host running Oracle VM Manager to enable access to Oracle VM Manager from any Java enabled web browser.

Network Configuration and Best Practices

Oracle VM provides flexible networking configuration that can be used as-is or modified to meet customers' business requirements.

The management domain (Dom0) of the Oracle VM server has direct access to the physical devices, it exports a subset of the devices in the systems to guest domains (DomU). A virtual device driver (also known as front-end driver) appears to the guest operating system as a real device. The network configuration is the same way and will look like a regular host with MAC address, IP address, etc. It can receive I/O requests from its kernel, but since it does not have direct access to the hardware, it must pass those requests to the back-end driver running in Dom0. The back-end driver, in turn, receiving the I/O requests, validates them for safety and isolation. Then it proxies them to the real device. When the I/O operation completes, the back-end driver notifies the front-end driver that the operation was successful and ready to continue. The front-end driver then reports the I/O completion to the guest OS kernel.

Oracle Linux and Oracle Solaris come with paravirtualized I/O drivers for improved network throughput and higher disk I/O. Moreover, Oracle VM Windows PV Drivers, signed by Microsoft for the Windows Logo Program, are available to improve Windows guest I/O throughput.

In the Oracle VM blade cluster reference configuration, the Sun Blade 6000 Modular System is equipped with a redundant pair of switched NEM (Network Express Module) 24p interconnects. Each

is dedicated to one head of the Sun ZFS Storage Appliance cluster. This solution leverages the bandwidth of 10GbE while ensuring no single points of failure. It also offers 12 unused 10GbE uplinks per NEM for seamless integration into the datacenter.

The Oracle VM server network configuration and datacenter topology should provide enterprise grade redundancy and failover. This is typically achieved by providing dual paths and switches using bonding on virtual machine servers. These settings can be configured in the reference configuration.

Additional details about Oracle VM network configurations can be found on the third-party document called, “[The Underground Oracle VM Manual](#).” This document is available at <http://itnewscast.com/underground-oracle-vm-manual>.

Storage Configuration and Best Practices

Sun ZFS Storage Appliances support multiple protocols such as NFS, Common Internet File System (CIFS), Internet Small Computer System Interface (iSCSI), InfiniBand (IB), and Fibre Channel (FC). Invented by Sun, NFS is one of the most successful examples of open network file sharing. It is designed to be both vendor neutral and operating system-type neutral. It integrates file access, file locking, and mount protocols into a single, unified protocol to ease traversal through a firewall and to improve security. NFS offers the utmost simplicity in attaching storage in Oracle VM virtualization environments over Ethernet. NFS on Sun ZFS Storage Appliance scales to many concurrent I/O threads due to the innovative architecture design of the appliance. This high I/O throughput enables more VM stacks to perform I/O without sacrificing service levels.

Architects can also choose FC or iSCSI protocols for storing and accessing both Oracle VM and user data on the Sun ZFS Storage Appliance. While additional SAN infrastructure is required to use FC interface, NFS and iSCSI can run on the existing LAN infrastructure over 1GbE or 10GbE links. If choosing FC or iSCSI, it is recommended to use OCFS2 clustered file system over the LUNs.

Sun ZFS Storage Appliance offers many choices in terms of RAID layout to address capacity, protection, and performance. Mirrored or triple mirrored protection is the recommended RAID layout for Oracle VM storage repository and user data. However, depending on the capacity requirement and the service level agreement, RAID Z2 (double parity RAID) can also be deployed for storage repositories.

The high-performance storage capabilities in this configuration results from using solid state drives (SSDs) in some models of the Sun ZFS Storage Appliance. SSDs enable rapid write capabilities for fast data placement in the storage pool. They optimize I/O rates by providing a fast buffer for reads and/or writes. Both Sun ZFS Storage 7420 and Sun ZFS Storage 7320 models have read-optimized and write-optimized SSDs, which enable excellent response time and throughput for demanding virtualized environments. This especially boosts VM cache performance because of the low latency and higher performance possible with SSD or flash memory technology. These platforms also have clustering capabilities to provide high availability for the storage.

To address the synchronous write performance of Oracle VM, it is strongly recommended to have two or more write-optimized flash devices per storage pool. To fully utilize the hybrid storage pool model with shorter read response time, it is recommended to use two or more read-optimized flash devices.

One project per Oracle VM server pool with dedicated storage repository is recommended. The shared Oracle VM storage repository (/OVS) could be a file system or a LUN that is seen from the Oracle VM servers in the server pool.

For storing the structured user data such as databases, which are accessed from the virtual machine, it is recommended to match the share record size with the application block size for optimal performance (for example 8kB for OLTP databases). The unstructured data and binary files can be stored on the shares with 128kB.

Oracle VM Blade Cluster Reference Configuration Example

This section provides an Oracle VM deployment scenario using Sun Blade 6000 and Sun ZFS Storage Appliance. It's a typical enterprise deployment model using high-end x86 servers and storage.

The reference configuration illustrated in Figure 5 offers high performance, availability, reliability, and manageability and is specifically intended for database virtual machines. While these requirements can be met in numerous ways, this reference architecture implements an optimal approach. The following sections detail the server components, virtual machine allocation, storage configuration, and project allocation with the Sun ZFS Storage 7420 Cluster storage and network configuration.

Either blade-based or rack-mounted Sun x64 servers can be used to run the Oracle VM infrastructure in a large, high availability configuration. As an example, a Sun Blade 6000 server (with 10 Sun Blade X6270 M2 or 5 X6275 M2 server modules) can be configured to run the Oracle VM software. Each server is attached to a clustered Sun ZFS Storage Appliance that is accessed using NFS, or via FC/iSCSI LUN. This approach enables additional features such as live migration, high availability, distributed resources scheduling to be used for the VMs in the server pool. Sets of servers that are identified as part of a single server pool are recommended to be configured in a similar way so that the service level agreements can be met during failovers or migration.

Each Sun Blade server module must be fitted with a Sun Intel 32599 Dual 10GbE PCIe 2.0 Fabric Expansion Module (FEM). The FEM supplies each server with two shared 10 GbE ports. The two 10 GbE interfaces are used by VMs to access NFS services on the Sun ZFS Storage 7420. Two 10 GbE interfaces supplies plenty of bandwidth and cable aggregation for virtual machine data. For NFS and iSCSI access, 10 GbE ports are also used.

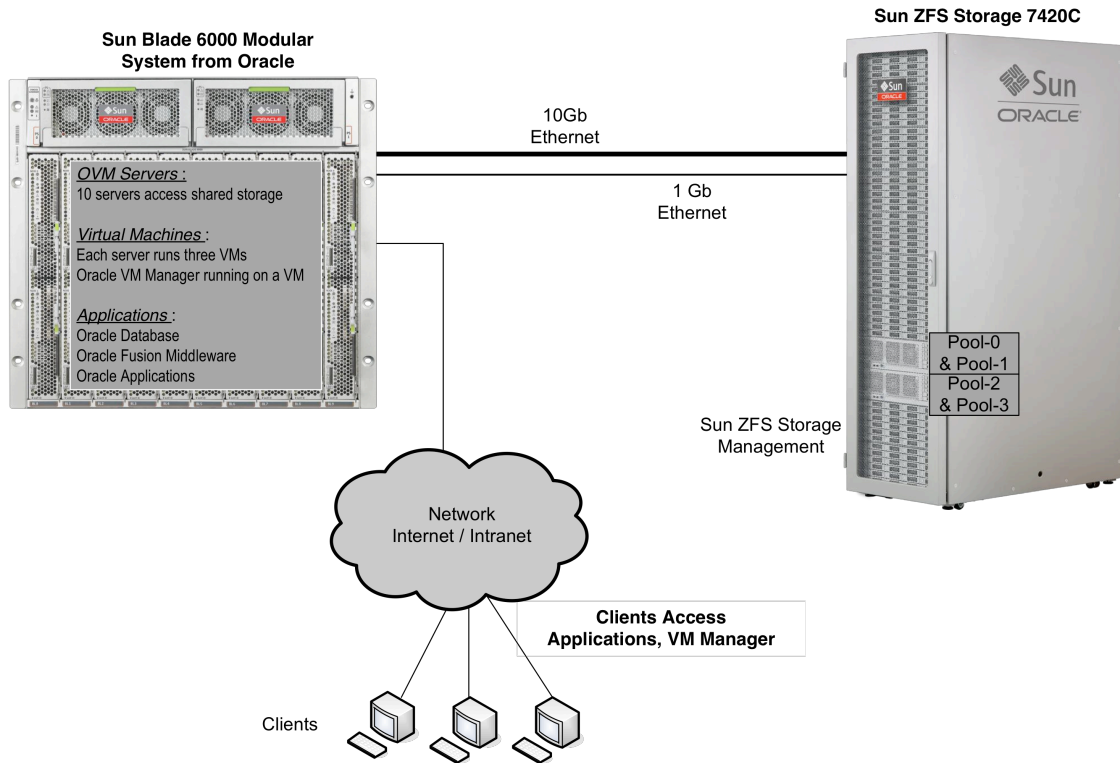


Figure 5. Oracle VM Blade Cluster Reference Configuration

For each virtual machine, the number of virtual CPUs and memory is dependent on the kind of applications that are deployed. In this example architecture, a minimum of 4GB and two virtual CPUs are configured for each of the middleware and application virtual machines. For the virtual machine that hosts the database, 16GB of memory and four virtual CPUs are allocated. Note that the number of virtual CPUs can be tuned in a live system.

Choosing the number of virtual machines per VM server as well as the type and number of network links are also critical considerations for a successful deployment. For example, for certain I/O loads over a single 1GbE link, the network could be saturated first before either the server or the storage. In such case, adding more links or moving to one or more 10GbE links may provide a better performance. If the CPUs are pegged running multiple virtual machines on the same host, either add additional servers to the server pool and live migrate the virtual machines to balance the Pool, or move the Oracle VM infrastructure from the midrange server to an enterprise class server.

For the high availability and optimal performance, Sun ZFS Storage 7420 clustered system is used as the storage platform. Table 2 shows the recommended configuration of four distinct storage pools and the purpose of each pool. Further details about the recommended pools are provided below.

TABLE 2. STORAGE CONFIGURATION FOR HIGH PERFORMANCE WITH ORACLE VM				
POOL NAME	RAID CONFIGURATION	SUN ZFS STORAGE 7420 CLUSTER HEAD	SYSTEM / VM	PURPOSE
pool-0	RAID-Z2	HEAD-1	Dom0	For storing the Oracle VM storage repository. This pool is accessed from domain 0 and contains the OS images from which the virtual machines are launched.
pool-1	MIRRORED	HEAD-1	Virtual machines that run the database	For storing database files and database binaries. One or more projects are created to cater to each database instance.
pool-2	MIRRORED or RAID-Z2	HEAD-2	Virtual machines that run the Oracle Fusion Middleware	For storing the middleware components, including binaries and configuration files.
pool-3	MIRRORED or RAID-Z2	HEAD-2	Virtual machines that run the Oracle Applications	For storing the application layer components – including the binaries, configuration files.

Each storage head is set to be active for 2 pools. This enables both load-sharing and high availability for all storage pools. In the event of a failure in HEAD-1, HEAD-2 will take over ownership of pool-0 and pool-1 as well as continuing to serve the clients.

- **Pool-0** — One Oracle VM project is created in pool-0 to store Oracle VM storage repositories. This is accessed from the dom0 of the Oracle VM server.
- **Pool-1** — One project is created to share the ORACLE_HOME across the various database instances. Additionally, one project per database is created. Compared to the rest of the pools, a greater number of disks are to be allocated for this pool. This is because the random reads (db sequential reads) benefit from more disks during scenarios when there is a read miss from the cache.
- **Pool-2** — One or more projects are created for storing the various middleware components. Projects might be dedicated to items such as Web server, SOA binaries, Tlogs, admin binaries and so on.
- **Pool-3** — One or more projects are created for storing the various application binaries and configurations for the Oracle E-Business Suite, Oracle Siebel, Oracle Peoplesoft and so on.

The shares (file systems) created under these projects are mounted from the various virtual machines. For security purposes, access to certain projects and file systems can be restricted to specific clients (virtual machines).

In this example architecture, all 27 virtual machines are active and the various applications access the Sun ZFS Storage 7420 Cluster system via redundant data paths. The high availability architecture provides a high performance infrastructure for demanding enterprise virtualization needs.

Conclusion

The Oracle VM blade cluster reference configuration offers the means to greatly simplify deployment and management of a virtualized infrastructure while also reducing risk. By following the guidelines outlined in this paper, IT organizations can take full advantage of the hardware and software components in the reference configuration, thereby realizing greater business benefits from their virtualized infrastructure.

References

For more information, visit the Web resources listed in Table 3 and the white papers listed in Table 4.

WEB RESOURCE DESCRIPTION	WEB RESOURCE URL
Oracle Solaris	http://oracle.com/solaris
Oracle Linux	http://oracle.com/linux
Oracle VM Templates	http://www.oracle.com/technetwork/server-storage/vm/templates-101937.html
Oracle Virtualization	http://oracle.com/virtualization
Sun Blade Systems	http://www.oracle.com/goto/blades
Sun ZFS Storage Appliances	http://www.oracle.com/us/products/servers-storage/storage/unified-storage/
Sun Networking	http://www.oracle.com/us/products/servers-storage/networking/
Oracle Enterprise Manager	http://www.oracle.com/us/products/enterprise-manager/
Oracle Enterprise Manager Ops Center 11g	http://www.oracle.com/us/products/enterprise-manager/opscenter/
Oracle Premier Support for Systems	http://www.oracle.com/us/support/systems/premier/

WHITE PAPER TITLE	WEB URL
Architecting Oracle VM solution with Sun ZFS Storage Appliances and Sun Servers	http://www.oracle.com/technetwork/articles/systems-hardware-architecture/vm-solution-using-zfs-storage-174070.pdf
Sun Blade 6000 Modular Systems from Oracle	http://www.oracle.com/technetwork/articles/systems-hardware-architecture/sb6000-architecture-163892.pdf
Creating and Using Oracle VM Templates: The Fastest Way to Deploy Any Enterprise Software	http://www.oracle.com/us/technologies/027001.pdf
Oracle VM - Creating & Maintaining a Highly Available Environment for Guest VMs	http://www.oracle.com/us/technologies/026999.pdf

Appendix A. Network Configuration Checklist

The steps below can be used to verify that the network is properly configured.

Verify that `/etc/hosts` is formatted correctly

As mentioned in the Oracle VM server installation section, the real host name (vmserver1 for example) must not be associated with the loopback address (127.0.0.1)

The following is incorrect will cause a variety of serious problems when trying to add the new node to a cluster:

```
127.0.0.1 server1 server1.mydomain.com localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
```

The correct format is:

```
127.0.0.1 localhost.localdomain localhost
10.1.2.3 server1.mydomain.com server1
::1 localhost6.localdomain6 localhost6
```

Ping the hostname to verify that it returns the assigned IP and not loopback address

This is an additional check to verify that `/etc/hosts` is set up correctly.

```
ping -c 1 vmserver1
PING vmserver1.mydomain.com (10.1.2.3) 56(84) bytes of data.
64 bytes from vmserver1.mydomain.com (10.1.2.3): icmp_seq=1 ttl=64 time=0.023 ms
```

If it responds with 127.0.0.1, fix `/etc/hosts`.

Test network connectivity by pinging the default gateway or a known host

Find the default gateway and verify that it responds to a ping:

```
# ip route show | grep default
default via 10.1.2.254 dev xenbr0
```

The network is active if there is a response similar to the following:

```
# ping -c 1 10.1.2.254
PING 10.1.2.254 (10.1.2.254) 56(84) bytes of data.
64 bytes from 10.1.2.254: icmp_seq=1 ttl=255 time=0.328 ms
```

If there is no response, try another known host on the same network as the new VM server. If there is still no response, check that the public interface is cabled correctly by physical verification. To blink the link light on a given interface, run:

```
# ethtool -p <the interface name, eth0, eth1, eth2, etc>
```

If the cabling is correct, check the network settings in `/etc/sysconfig/network`. Verify that the `HOSTNAME` and `GATEWAY` variables are set correctly. Also check

```
/etc/sysconfig/network-scripts that the IPADDR and NETMASK are set correctly in the ifcfg-eth* files.
```

If changes are made, do the following to apply them. Restart the OS networking and do step 3 again:

```
# service network restart
```

If there is still no response from the network and the default gateway is on the same subnet, try to reach it at layer 2 with the arping command. First get the default gateway device:

```
# ip route show | grep default
default via 10.1.2.254 dev xenbr0
```

Then use the `-I` parameter to specify the device and issue the command:

```
# arping -I xenbr0 -c 1 10.1.2.254
ARPING 10.148.36.1 from 10.1.2.3 xenbr0
Unicast reply from 10.1.2.3 [00:00:0C:07:AC:70] 0.794ms
Sent 1 probes (1 broadcast(s))
Received 1 response(s)
```

If the default gateway or a known host still cannot be reached, ask a network administrator to check the cabling and switch port configuration for the new node.

Additional network checks:

Run: `ethtool <the interface name, eth0... etc>` to insure that the interface is in full duplex, at the correct line speed and has an active link:

```
# ethtool eth0
Settings for eth0:
Supported ports: [ TP MII ]
Supported link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
Advertised auto-negotiation: No
Speed: 100Mb/s
Duplex: Full
Port: MII
PHYAD: 32
Transceiver: internal
Auto-negotiation: off
Supports Wake-on: pumbg
Wake-on: d
Current message level: 0x00000007 (7)
Link detected: yes
```

Verify that kernel routing is correct and a default gateway is in place with:

```
# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.1.2.0 0.0.0.0 255.255.255.0 U 0 0 0 xenbr0
10.1.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
0.0.0.0 10.1.2.254 0.0.0.0 UG 0 0 0 xenbr0
```


Test and verify bonded interface failover before deploying to production

Verify that two operational dns servers are listed in /etc/resolv.conf

A correctly formatted /etc/resolv.conf looks as follows:

```
search mydomain.com
nameserver 10.1.3.90
nameserver 10.1.4.90
```

The search directive is optional. Additional information about the formatting of this file can be obtained from: `man resolv.conf`

Ping these servers to make sure they are network reachable, then do a test query on each one to verify that the dns service is operational. For the test, pick an internal hostname that is known to be in the local network's DNS:

```
# host vmserver1.mydomain.com 10.1.3.90
Using domain server:
Name: 10.1.3.90
Address: 10.1.3.90#53
Aliases:

vmserver1.mydomain.com has address 10.1.2.3
```

Verify forward and reverse dns lookups on all Oracle VM pool hostnames and IPs

DNS is required for proper operation of the Oracle VM product. Verify the node hostname with both forward and reverse lookups using the existing /etc/resolv.conf server settings:

```
# host server1.mydomain.com
server1.mydomain.com has address 10.1.2.3

# host 10.1.2.3
3.2.1.10.in-addr.arpa domain name pointer server1.mydomain.com.
```

Do this for ALL nodes in the Oracle VM pool that is being built. If any hostnames don't resolve in either direction, please contact the DNS administrator and insure that they are added and verified.

Confirm that /etc/ntp.conf contains at least 2 operational ntp servers

In order to conform to best practices, a reliable pair of ntp servers must be reachable from each node of an Oracle VM cluster. If the default Internet servers are not reachable or desired, delete them and add the following to /etc/ntp.conf where the server parameter is a reachable ntp host:

```
server mytimeserver1.mydomain.com
server mytimeserver2.mydomain.com
```

Ping each ntp server to make sure it is reachable.

Confirm that the system time and date are correct and the ntpd service is active

The date, time and time zone must match the other servers in the pool:

```
# date
Tue Sep 28 19:17:44 MDT 2010
```

If the time zone is not correct, run the system-config-date utility and set the time zone to conform to the proper locale and the other nodes in the Oracle VM cluster.

Now do the following command sequence to initially set the server time/date to the ntp primary server and activate the ntpd service:

```
service ntpd stop
ntpdate mytimeserver1.mydomain.com
chkconfig ntpd on
service ntpd start
```

At this point, the ntpd service can be confirmed to be operating with the following command:

```
# ntpq -p
remote refid st t when poll reach delay offset jitter
=====
*mytimeserver1.m 144.25.255.140 3 u 557 1024 377 0.427 0.192 0.024
+mytimeserver2.m 144.25.255.140 3 u 374 1024 377 0.350 0.276 0.007
LOCAL(0) .LOCL. 101 35 64 377 0.000 0.000 0.001
```

It is also useful to monitor /var/log/messages for ntpd events to verify that the daemon maintains reliable contact with the specified ntp servers and has synchronized the OS to them:

```
Sep 28 20:51:13 server1 ntpd[2584]: synchronized to 10.1.3.100, stratum 3
Sep 28 20:57:41 server1 ntpd[2584]: synchronized to 10.1.4.100, stratum 3
```

Verify network heartbeat peers and paths

When creating the server pool with HA enabled, communications must be possible between all nodes on tcp port 7777. This should be tested from all nodes to all nodes with the following command:

```
# nc -zv <another node ip> 7777
root@server1:~ # nc -zv server0 7777
Connection to server0 7777 port [tcp/cbt] succeeded!
```

Repeat this command on all members of the cluster to all members of the cluster. If it fails, check the firewall rules on each node and verify that there are no blocking rules for this port on the cluster switch or router.

Appendix B. Storage Configuration Checklist

Clean the node in preparation for cluster service with cleanup.py

If the new node is to be used in a pool with shared storage, it is essential to remove the local disk repository that was created during the installation process. If an old node is being recycled or given a new ip address, a cleanup process is also required to re-initialize and make it ready for addition to a new cluster. Both of these goals are achieved by running the cleanup.py script:

```
# /opt/ovs-agent-2.3/utils/cleanup.py
This is a cleanup script for ovs-agent.
It will try to do the following:
*) stop o2cb heartbeat
*) offline o2cb
*) remove o2cb configuration file
*) umount ovs-agent storage repositories
*) cleanup ovs-agent local database
Would you like to continue? [y/N] y
```

This will remove all repository mounts and place the node into an initialized, single node state. The cleanup.py utility does not affect the contents of any remote repositories that were previously mounted. It simply clears all repository information from the node and unmounts any existing repositories.

Caution: Shutdown all VMs on existing nodes before running cleanup.py.

Confirm that the root repository storage is available on the new node

Shared storage availability is essential for creating an Oracle VM server pool and master server. Setting up access to shared storage for nfs or ocfs2 is covered in the External Storage section of the Oracle VM 2.2 product documentation. Please review that documentation before proceeding with this step.

NFS Repository

For new or existing NFS based storage repositories offered by Sun ZFS Storage Appliance, run the following command to verify that an NFS mount is available from the NFS server:

```
# showmount -e <nfs server>
```

The desired, exported mount must be visible in the listing and show access from the network that the new Oracle VM server resides on. If so, no other preparatory steps are needed and it is alright to proceed with adding the repository per the NFS section of the product documentation.



Best Practices and Guidelines for Deploying
the Oracle VM Blade Cluster Reference
Configuration
December 2010

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1010

Hardware and Software, Engineered to Work Together