ORACLE VM

An Oracle White Paper
March 2014

RUNNING ORACLE REAL APPLICATION CLUSTERS ON ORACLE VM SERVER FOR SPARC

ORACLE

Introduction 1
Running Oracle RAC on Oracle VM Server for SPARC..................... 2
Oracle VM Server for SPARC Overview 3
Oracle RAC Nodes 4
Deployment Options 5
Software Requirements 6
Oracle VM Server for SPARC Requirements................................. 6
Oracle Database... 6
Hardware Configuration.. 6
Server Configuration... 8
Configuration Guidelines 8
Configuration Details 9
Software Installation and Configuration 11
Oracle Solaris Installation 11
Oracle VM Server for SPARC Installation.................................... 11
Control Domain Configuration.. 12
Guest Domains... 13
Guest Domain Configuration 14
Network Configuration 16
Network Layout 16
Public Network.. 16
Private Network 17
Network Interface Summary.. 18
Network Interfaces and Host Names 19
Private Network Configuration 21
Storage Configuration.. 22

Shared and Virtual Disks ... 22
Adding Shared Disks to Guest Domains ... 23
Oracle Installation ... 24
Configuring IPMP for the cluster interconnect ... 24
Additional Information ... 25
NTP ... 25
CPU Dynamic Reconfiguration ... 26
Memory Dynamic Reconfiguration ... 26
Performance Considerations ... 27
Jumbo Frames ... 28
Summary ... 29
Appendices ... 30
Logical Domains Manager Configuration Example ... 30
Configuration of the First Control Domain (rac01) ... 30
Configuration of the First Guest Domain (ldom1) ... 33
Configuration of the Second Control Domain (rac02) ... 36
Configuration of the Second Guest Domain (ldom2) ... 39
References ... 42

Introduction

This article discusses running Oracle® Real Application Clusters (RAC) on Oracle servers configured with Oracle VM Server for SPARC® (previously called Sun Logical Domains). Oracle VM Server for SPARC virtualization technology allows the creation of multiple virtual systems on a single physical system, and enables fine-grained assignment of CPU and memory resources to an Oracle RAC workload. When deployed on Oracle SPARC servers, with up to 3,072 CPU threads per system, this solution provides a powerful platform for both development and production environments. In development environments, multiple Oracle RAC nodes can be deployed on the same physical server to reduce hardware costs, while production environments can either share physical servers or place each Oracle node on a separate physical server for increased availability.

This paper addresses the following topics:

- “Running Oracle Real Application Clusters on Oracle VM Server for SPARC” provides an overview of Oracle RAC and Oracle VM Server for SPARC, and discusses the various deployment options for this solution.

- “Software Requirements” lists the software and firmware requirements, and “Hardware Configuration” describes the hardware used for an example configuration.

- “Software Installation and Configuration,” “Network Configuration,” and “Storage Configuration” describe the steps necessary to install and configure an example configuration.

- “Additional Information” provides supplemental information, including performance considerations and known issues.

For a detailed example on RAC installation, please refer to the white paper: Installing Oracle RAC 11gR2 on the Oracle Solaris 11 OS by Using Oracle VM Server for SPARC.

Running Oracle RAC on Oracle VM Server for SPARC

Oracle Real Application Clusters (RAC) 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 the Oracle enterprise grid architecture.

Oracle RAC utilizes Oracle Clusterware 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 is a key component of the Oracle enterprise grid architecture as well.

In a typical Oracle RAC installation, Oracle Automatic Storage Management (ASM) acts as the underlying, clustered volume manager. ASM 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 ASM provides the basis for a shared storage pool in Oracle enterprise grid architectures.

Oracle VM Server for SPARC allows the creation of multiple virtual systems on a single physical system.

Oracle VM Server for SPARC is a virtualization and partitioning solution supported on a broad range of SPARC servers such as SPARC T-Series and SPARC M-Series. The SPARC M-Series servers can be optionally divided into physical domains ("PDoms"), each of which can host independent Oracle VM Server for SPARC environments. Oracle VM Server for SPARC makes it possible to create multiple virtual systems on a single physical system or physical domain. Each virtual system is called a logical domain and runs its own copy of the Oracle Solaris operating system. Oracle's SPARC servers, configured with up to 3,072 virtual CPUs and 32 TB of physical memory, are powerful systems which can easily be configured with logical domains to consolidate and virtualize multiple physical servers onto a single platform.

Oracle Database and Oracle Real Application Clusters (Oracle RAC) are the leading database applications, and as such are frequently used on servers. With virtualization and consolidation, Oracle Database and Oracle RAC can now run on selected certified virtual environments, such as Oracle VM Server for SPARC, and use virtual devices. In addition, multiple Oracle Database servers or Oracle RAC nodes can be located on the same physical platform, introducing a new way to deploy databases.

Oracle VM Server for SPARC Overview

In Oracle VM Server for SPARC, logical domains have roles that define their characteristics. The roles of a domain depend on how the domain is configured, which resources it owns and what services it provides. A single domain can have one or multiple roles. An Oracle RAC node can be located on a domain with any role, as long as that domain provides the appropriate devices and resources required and supported by the Oracle environment.

A logical domain can take one or more of the following roles:

- **I/O Domain** — An I/O domain is a domain that has direct access to some physical I/O resources, such as physical disks or physical network interfaces. An I/O domain can own a PCIe bus, or it can own a PCIe slot or an on-board PCIe device by using the direct I/O (DIO) or single-root I/O virtualization (SR-IOV) features. The operating system running in an I/O domain uses regular (non-virtualized) drivers of the operating system to access physical I/O devices. Running the Oracle Solaris operating system in an I/O domain is very similar to running Oracle Solaris on a non-virtualized system. Similarly, running and configuring Oracle RAC in an I/O domain is no different than running and configuring Oracle RAC on any other Oracle SPARC server. By having direct access to physical I/O devices, an I/O domain provides optimal I/O performance. However, the number of I/O domains that can be created on a single platform is limited by the I/O resources available on that platform, based on the number of PCIe buses, PCIe devices, and SR-IOV virtual functions on the server. As a result, only a limited number of I/O domains can be created. An I/O domain is usually also a service domain, described below, which lets it make its physical I/O devices available to other domains as virtual devices.
- **Root Domain** — A root domain is a domain that has a PCIe bus (also known as a PCIe root complex) assigned to it. A root domain is also an I/O domain, as it owns and has direct access to physical I/O devices. The number of root domains that can be created on a single platform is limited by the number of PCIe bus available on that platform. For example, a maximum of four root domains can be created on the SPARC T4-4 server, and a maximum of 16 on a T5-8 server. As a root domain is also an I/O domain, running Oracle RAC in a root domain is similar to running Oracle RAC in an I/O domain.
- **Control Domain —** The control domain is the first domain that boots up when the server is powered on, and is the domain that runs the Logical Domains Manager, which is used to configure and manage all domains on the platform. The control domain has the name “primary” and is also a root complex I/O domain, since it requires physical I/O to boot up. Therefore running Oracle RAC in the control domain is similar to running Oracle RAC in an I/O domain. A control domain is also

A domain can take one or more of the following roles:

- I/O Domain
- Root Domain
- Control Domain
- Service Domain
- Guest Domain

Running Oracle RAC in an I/O domain can result in better performance, but only a limited number of I/O domains can be created per server. It is not generally recommended to run Oracle RAC in a control domain because of security concerns. It is not generally recommended to run Oracle RAC in a service domain, for performance reasons and to avoid the possibility of negatively affecting virtual device services. Oracle RAC can run in a guest domain and use virtual devices.

usually a service domain, described below, in order to leverage its physical I/O resources by making them available to other domains. However, for management and security concerns, it is not generally recommended to run Oracle RAC in the control domain.

The control domain has a privileged connection with the hypervisor, which allows it to control other domains. Control domain security should be carefully protected. If control domain security is compromised, a malicious user gaining privileged access on the control domain could take control of all other domains on the platform. Any application running in the control domain should be properly configured to prevent such security risks.

- **Service Domain** — A service domain is a domain that provides virtual disk and virtual network services to other domains. A service domain is also usually an I/O domain in order to have access to physical devices underlying the virtual I/O devices it provides. Consequently, running Oracle RAC in a service domain is similar to running Oracle RAC in an I/O domain. However, for performance reasons and to avoid negatively affecting virtual device services, running Oracle RAC in a service domain is not recommended.

A service domain consumes some CPU, memory and I/O resources to provide virtual I/O services to other domains. Therefore, a service domain must be configured with sufficient resources to handle the workload due to services provided to other domains and for any workload generated by applications (such as Oracle RAC) running on the domain itself.

Moreover, Oracle RAC can sometimes reboot the system it is running on. Reboot of a service domain will not bring down or reset guest domains using that service domain. Instead, I/O requests from these guest domains are suspended while the service domain is down, and automatically resume once the service domain is up and running again. If an alternate, redundant service domain has been configured then guest virtual I/O can proceed using the other service domain. When running Oracle RAC in a service domain, keep in mind that Oracle RAC might reboot that service domain and that this will temporarily block I/O requests of guest domains using that service domain. For that reason, it is not recommended to run Oracle RAC in a service domain unless its client guest domains can access their I/O from a redundant path on a different service domain.

- **Guest Domain** — A guest domain is a domain which is not an I/O domain but which is a consumer of virtual device services provided by one or more service domains. A guest domain does not have physical I/O devices, and only has virtual I/O devices such as virtual disks and virtual network interfaces. This is the usual way to provide I/O to applications running in domains. Oracle RAC can run in a guest domain and use virtual devices, and this configuration is the focus of this document.

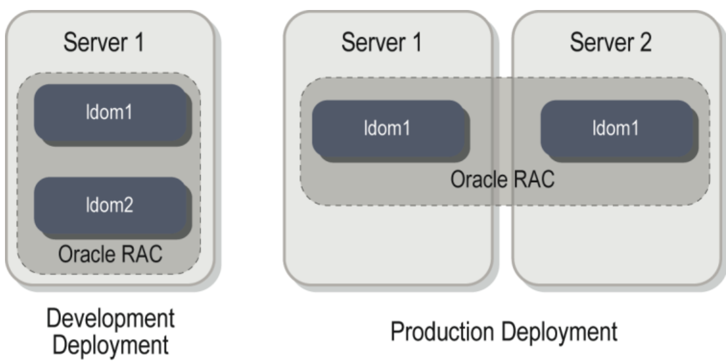
Oracle RAC Nodes

Oracle RAC can be run on any domain regardless of its roles, although it is not recommended to run Oracle RAC in the control domain or in a service domain for the reasons noted above. The main choice is to run Oracle RAC on either a guest domain or an I/O domain. An I/O domain provides optimal I/O performance because it has direct access to physical I/O devices, but the number of I/O domains on a platform is limited. On the other hand, a guest domain is more flexible and supports more dynamic operations, and can even be live migrated to another server, because its resources and devices are virtualized.

A configuration should not mix Oracle RAC nodes running in I/O domains with Oracle RAC nodes running in guest domains. Although such a configuration might be technically possible, it can create a complex setup that is difficult to manage, and may be error prone. Either all Oracle RAC nodes in an Oracle RAC cluster should be running in I/O domains, or all should be running in guest domains.

Deployment Options

With server virtualization, it is possible to run multiple virtual machines and operating systems on a single physical system. This makes it possible to host multiple nodes of the same cluster on a single physical system. Two main variants for deploying Oracle RAC with logical domains are considered, as shown in Figure 1:



Oracle RAC nodes can be placed in logical domains on the same physical server (development variant) to reduce costs, or can be placed on logical domains on separate physical server (production variant) for better availability.

Figure 1: Deployment options

- **Development** — All Oracle RAC nodes are located on domains on the same physical server. This is convenient for development or evaluation because it reduces the amount of hardware required (only one physical server). However, this configuration is not intended for a production environment because the physical server is a single point of failure: if the entire server goes down then all nodes will be down and the entire Oracle RAC cluster will be unavailable.
- **Production** — Oracle RAC nodes are placed on separate physical servers. This is recommended for a production environment, because in this configuration nodes are on different physical servers and there is no single point of failure. Other applications can reside in other domains on the same server without interference.

Of course, both deployment options can be mixed. One variation is to have different Oracle RAC nodes on different physical domains (“PDoms”) of the same M5-32 or M6-32 server, since there can be hardware redundancy across different physical domains. Just keep in mind that nodes located on the same physical server can represent a single point of failure.

Software Requirements

This section discusses the software requirements for deploying Oracle RAC on Oracle VM Server for SPARC, including Logical Domains Manager software, required operating system version and patches, firmware, and Oracle RAC software.

Oracle VM Server for SPARC Requirements

Running Oracle RAC with Oracle VM Server for SPARC requires the use of Logical Domains Manager version 2.2 or later. At this writing, the current version is 3.1, which provides features for enhanced performance, availability and manageability. The operating system in all domains should be at least Oracle Solaris 10 10/11 (Update 10) or Oracle Solaris 11.

Servers should have firmware corresponding to the Oracle VM Server for SPARC version in use. For example, the Oracle VM Server for SPARC 3.1 Release Notes section "System Requirements" shows the firmware version required for each supported platform. The documentation can be found at http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html

Oracle Database

Running Oracle RAC on Oracle VM Server for SPARC requires the use of Oracle 10*g* R2 (Oracle 10.2.0.4) or later with the same patch set on all nodes. Refer to the Oracle documentation to check if a particular version is supported, and for any additional requirements. Information can be found on the Oracle Support Web site:

- https://support.oracle.com
- http://www.oracle.com/technetwork/database/clustering/tech-generic-unix-new-166583.html

Hardware Configuration

Any server supporting Oracle VM Server for SPARC can be used with Oracle RAC. Refer to the latest *Oracle VM Server for SPARC Release Notes* documentation for a complete list of systems supporting Oracle VM Server for SPARC. Some additional hardware, such as external storage arrays or network switches, may be needed depending on the selected configuration and deployment option.

As an example, this document studies the deployment of a two-node Oracle RAC cluster where each Oracle RAC node is located on an Oracle VM Server for SPARC guest domain on a different physical server (production variant). Figure 2 shows the hardware configuration and cabling.

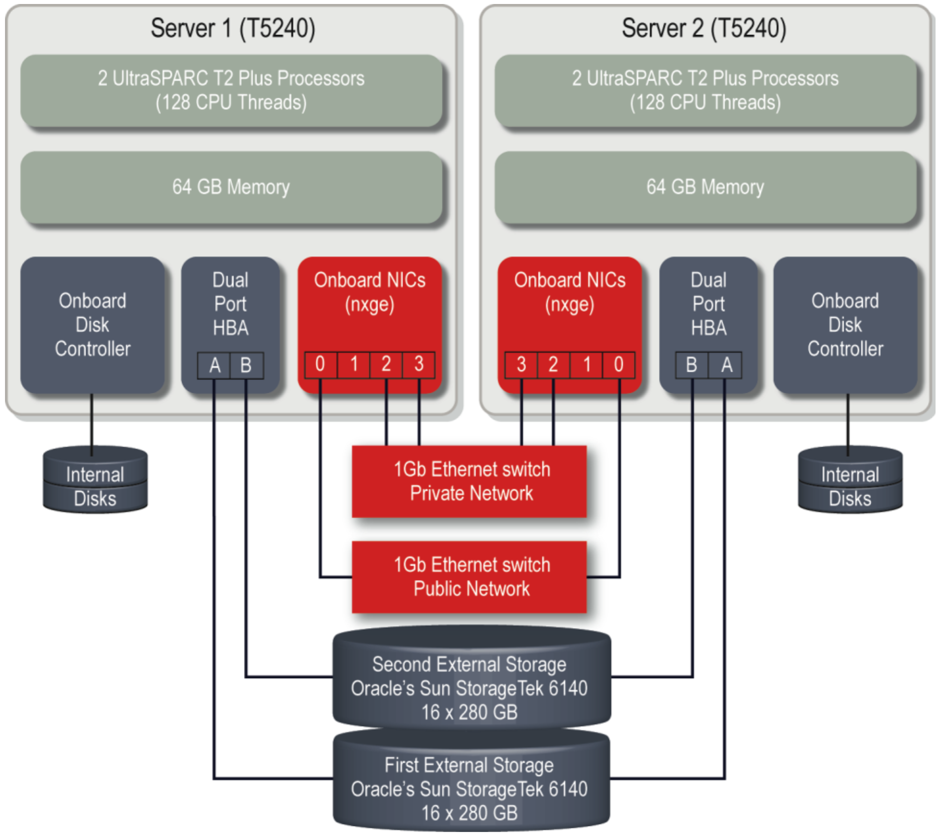


Figure 2: Hardware configuration

This example uses the following hardware. Details will vary for other hardware platforms:

- Two Oracle Sun SPARC Enterprise T5240 servers, each with 128 CPU threads and 64 GB of memory. This is a fairly old server with much less scale and performance than current SPARC servers, but is compatible with current Oracle Solaris and Oracle VM Server for SPARC versions, and remains a good example of a test system when performance is not critical. The Sun SPARC Enterprise T5240 server contains two UltraSPARC T2 Plus processors. Each UltraSPARC T2 Plus processor is made of 8 CPU cores, and each CPU core has 8 CPU threads, providing a total of 2x8x8 = 128 CPU threads per system. The Oracle VM Server for SPARC software virtualizes each CPU thread as a virtual CPU, making 128 virtual CPUs available for creating domains.
- One internal disk is used on each server as the system disk of the control domain. The Sun SPARC Enterprise T5240 server can have up to sixteen internal disks.
- Two Oracle Sun StorageTek 6140 storage arrays, each with sixteen 280 GB Fibre Channel (FC) disk drives. Each storage array is connected to both servers, and each server should be connected to the same array controller (A or B). The first storage array provides one local disk to each guest domain. The second storage array is used to provide five shared disks (voting disks, OCR disk and ASM disk) to the Oracle RAC nodes. Each volume is owned by one and only one controller; therefore, each

server using this volume must be connected to the same owning controller (for example, Server A to port 1 on Controller A, and Server B to port 2 on Controller A).

- Two 4 Gb dual-port FC-AL host bus adapters (HBA), one for each server. These HBAs are used to connect the two storage arrays to both servers. Each HBA has two ports. On each server, the first port is connected to the first storage array, and the second port to the second storage array.
- Two 1 Gb Ethernet switches. One switch is used to interconnect the two servers and create a private network to be used by Oracle RAC. The second switch is also connected to the two servers, but it provides the public network and could be connected to a LAN or a WAN.
- The on-board network interfaces present on each server are used for network communications. The Sun SPARC Enterprise T5240 server has four on-board 1 Gb Ethernet network interfaces. The first interface (port 0) is used for the public network and connected to the corresponding switch. Two other interfaces (port 2 and 3) are used to provide redundant access to the private network and they are connected to the other switch. One interface (port 1) is not used.

Note that this configuration provides a minimum hardware redundancy. It provides a dual connection to the private network to meet the minimum requirement of Oracle RAC. Hardware redundancy can be improved by:

- Using additional HBAs to provide redundant access paths to the storage arrays.
- Using hardware or software RAID solutions to duplicate on-disk data.
- Using the available on-board network interface (port 1) to provide redundant access paths to the public network.
- Using additional network switches to create a redundant private network and have redundant access to the public network.
- Using additional network interface adapters to provide PCI bus-level redundancy to access private and public networks.

Please see References section for more information how to setup network and storage redundancy.

Server Configuration

This section includes general configuration guidelines that are recommended when setting up Oracle RAC on logical domains, and the specific configuration details used for this example configuration.

Configuration Guidelines

The following configuration guidelines are recommended when setting up Oracle RAC on logical domains:

- It is not recommended to run Oracle RAC in the control domain, unless the system is configured with a single domain (which is the control domain), or unless the security and availability considerations are fully understood.

- It is not recommended to run Oracle RAC in a service domain, unless its client guest domains have redundant access to I/O devices through an additional service domain, or have loose availability requirements.
- In an Oracle RAC configuration, Oracle RAC nodes should either all be running in I/O domains, or all be running in guest domains. An Oracle RAC configuration should not mix Oracle RAC nodes running in I/O domains with Oracle RAC nodes running in guest domains.
- Each domain running Oracle RAC should have a minimum of 8 virtual CPUs (1 CPU core) and 4 GB of memory. This is a small resource allocation for illustrative purposes.
- CPU cores should be allocated on a core boundary for performance reasons. This can be done by allocating CPU threads in increments of 8, or by using whole-core allocation to assign CPU resources.
- Each service domain providing virtual devices (virtual disk and virtual network) to an Oracle RAC guest domain should have a minimum of 16 virtual CPUs (2 CPU cores) allocated on core boundaries, and 4 GB of memory.
- When running Oracle RAC in guest domains, the private network redundancy should be configured using IP multipathing (IPMP) in those guest domains.
- Virtual disks shared by Oracle RAC guest domains should be backed by full physical SCSI disks or SCSI LUNs in order to provide SCSI command and multihost disk control functionality. A shared virtual disk should not be backed by a file or a volume, and it should not be a single-slice disk.

Configuration Details

In the example presented in this document, each server is configured with two domains: the control domain and one guest domain (Figure 3). The control domain is also the primary domain, and therefore also an I/O domain. Furthermore, the control domain is used as a service domain, providing virtual device services (virtual disk and virtual network) to the guest domain. For simplicity, the name “control domain” is used to designate all roles of the control, root, I/O, and service domain.

Each domain is allocated the following set of resources, listed in Table 2:

TABLE 2. RESOURCES ALLOCATED TO EACH DOMAIN.

DOMAIN	CPU	MEMORY	DEVICES
Control Domain	32 virtual CPUs (4 CPU cores)	16 GB	All PCI buses (internal disks, on-board NICs, HBA connected to the two storage arrays)
Guest Domain	48 virtual CPUs (6 CPU cores)	24 GB	Virtual disks and virtual network interfaces provided by the control domain

This configuration does not use all resources of the system. There are still 48 virtual CPUs (6 CPU cores) and 24 GB of memory available. These resources can be used to create additional domains or to reconfigure the existing domains if more resources are required, for example to handle a more important workload. The `ldm list-devices` command can be used to display available resources. A valuable feature of Oracle VM Server for SPARC is that resources can be dynamically allocated as needed.

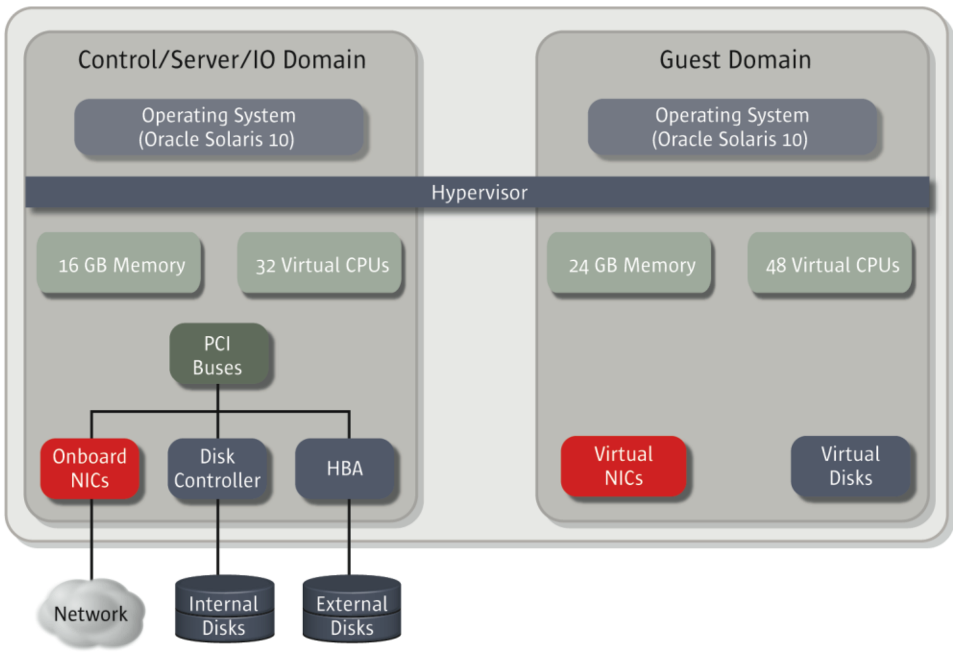


Figure 3: Resource assignment within a physical server

To act as a service domain, the control domain will have the following virtual device services:

- One virtual disk service (`primary-vds0`) used to export physical disks as virtual disks to guest domains.
- One virtual console concentrator service (`primary-vcc0`) used to provide access to the virtual consoles of the guest domains. This service will use the port range 5000-5100 for virtual console access.
- One virtual switch service (`primary-vsw0`) associated with the primary network interface (`nxge0`). This virtual switch will be part of the public network required by Oracle RAC. This interface is directly connected to the virtual switch and will allow the control domain to have a network connection with the guest domain and with the external network (through `nxge0` which is associated with `primary-vsw0`).
- Two virtual switch services (`primary-vsw1` and `primary-vsw2`) associated with physical network interfaces `nxge2` and `nxge3`. These virtual switches will be part of the private network required by Oracle RAC.

The complete configuration of the public and private networks is described in “Network configuration”.

Software Installation and Configuration

The following tasks are part of server software installation:

- Oracle Solaris installation (see below)
- Oracle VM Server for SPARC installation (see below)
- Control domain configuration
- Guest domain configuration

Additional configuration tasks, including network and storage related tasks, are covered in later sections of this document. Oracle installation occurs as a separate step, after the network and storage devices are configured.

Oracle Solaris Installation

Oracle SPARC systems come pre-installed with Oracle Solaris OS and Oracle Solaris 11 comes with Oracle VM Server for SPARC pre-installed. Ensure that the appropriate releases of Oracle Solaris OS and Oracle VM Server for SPARC are installed for your server, and check that required patches are present.

You may also choose to re-install the entire system so that it conforms to your installation policy and matches your requirements. In this situation, the first step is to install Oracle Solaris OS and the appropriate patches. Refer to the Oracle Solaris OS installation documentation for more information on how to install the Oracle Solaris OS on servers with SPARC processors. In our example, the operating system is installed on the first internal disk of the server.

After Oracle Solaris OS is installed, the system can be configured and enabled to use logical domains.

Oracle VM Server for SPARC Installation

Refer to the *Oracle VM Server for SPARC Administration Guide* for a complete procedure on how to install logical domains. Basically, the following actions are performed on each physical server. This example assumes that the Oracle Solaris OS and the required patches are already installed on each server. Note that Oracle VM Server for SPARC software is included by default with the Oracle Solaris 11 OS, so the following installation steps can be skipped. Instead, if the pre-installed version of Oracle VM Server for SPARC is not up to date, use the instructions at http://docs.oracle.com/cd/E38405_01/html/E38406/upgradingtoldoms3.1.html#LDSAGupdates11task For Oracle Solaris 10:

1. Ensure that the system firmware matches the Oracle VM Server for SPARC release that is planned for installation. Refer to the *Oracle VM Server for SPARC Release Notes* to find the appropriate firmware version and to the *Oracle VM Server for SPARC Administration Guide* for instructions to upgrade the system firmware.

2. Download Oracle VM Server for SPARC software from the Oracle web site.

3. Extract the archive.
4. Install the `SUNWldm.v` package

```
# pkgadd -d OVM_Server_SPARC-2_2/Product SUNWldm.v
```

5. Ensure that the Logical Domains Manager (`ldmd`) and the Virtual Network Terminal Server (`vntsd`) services are enabled.

```
# svcadm enable ldmd
# svcadm enable vntsd
```

Control Domain Configuration

Configure a control domain on each physical server.

After the Oracle VM Server for SPARC software has been installed, the current system has to be reconfigured to become the control domain (also known as the primary domain). To do so, the following actions are performed on each physical server:

1. Add a virtual disk server (`vds`) and a virtual console concentrator (vcc). .

```
# ldm add-vds primary-vds0 primary
# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

2 Determine the primary network interface of the system. This is usually (but not necessarily) the first configured interface, for example `nxge0` or `igb0`. If the control domain is running Solaris 11, use the "vanity name" for the network adapter driver, which is usually `net0`. Adjust the commands used in the following steps for the actual device names on your server. Review the hardware and Solaris documentation for your server to select appropriate network devices and drivers.

2. Add a virtual switch associated with the primary network interface. This virtual switch will be used for the public network.

```
# ldm add-vsw net-dev=nxge0 primary-vsw0 primary
```

Add a second and third virtual switch, associated with the physical network interfaces selected for the Oracle RAC private network. Both switches must have option `linkprop=phys-state` set in order to implement link-based IPMP private network interfaces redundancy.

```
# ldm add-vsw net-dev=nxge2 linkprop=phys-state primary-vsw1 primary
# ldm add-vsw net-dev=nxge3 linkprop=phys-state primary-vsw2 primary
```

If running Solaris 10 in the control domain, change the primary interface to be the first virtual switch interface. The following command configures the control domain to plumb and use the interface `vsw0` instead of `nxge0`.

```
# mv /etc/hostname.nxge0 /etc/hostname.vsw0
```

3. Configure the control domain (`primary`) with 32 virtual CPUs (4 CPU cores) and 16 GB of memory. Note that the syntax `ldm set-vcpu 32 primary` can be used instead of `set-core` for backwards compatibility, but whole core allocation is preferred. The `ldm set-crypto` command is used on machines preceding the T4 to assign a hardware cryptographic accelerator to a domain, with one crypto accelerator per core. This command is not needed on a T4 server or later.

```
# ldm set-core 4 primary
# ldm set-crypto 4 primary   # 1 per core, only for servers prior to T4
# ldm start-reconf primary
# ldm set-mem 16g primary
```

4. Save the configuration and reboot the system.

```
# ldm add-spconfig initial
# init 6
```

After the system reboots, Oracle VM Server for SPARC will be enabled and the system will now be configured with one domain: the control domain (primary domain). Additional domains can then be created and configured from the control domain.

Guest Domains

After the control domain has been configured, you can create the guest domains that will be used as Oracle RAC nodes. One guest domain is created on each physical server. The first guest domain is created on the first server with the name `ldom1`, the second guest domain is created on the second server with the name `ldom2`.

Each guest domain (`ldom1` and `ldom2`) is initially created with the following resources:

- 48 virtual CPUs (6 cores).
- 6 cryptographic accelerators (for pre-T4 systems). Oracle RAC can leverage SPARC hardware cryptographic acceleration.
- 16 GB of memory
- One virtual network interface (`vnet0`) connected to the virtual switch `primary-vsw0`. This virtual network interface will provide access to the public network.

- Two virtual network interfaces (`vnet1` and `vnet2`) connected to the virtual switches `primary-vsw1` and `primary-vsw2`. This virtual network interfaces will provide access to the private network.
- One virtual disk (which appears as `c0d0` in the guest domain), which is a LUN from the first storage array. The domain `ldom1` uses the LUN1 of the storage array (`c3t0d1`), and `ldom2` uses the LUN2 of the same storage array (`c3t0d2`). This virtual disk will be used as the system disk of the guest domain. and hosts the operating system.

To simplify the description of the configuration, guest domains are initially created with only one disk (`c0d0`), which is used as the system disk for the operating system. Additional disks required for the Oracle RAC configuration are added later. The initial configuration of guest domains is shown in Figure 4.

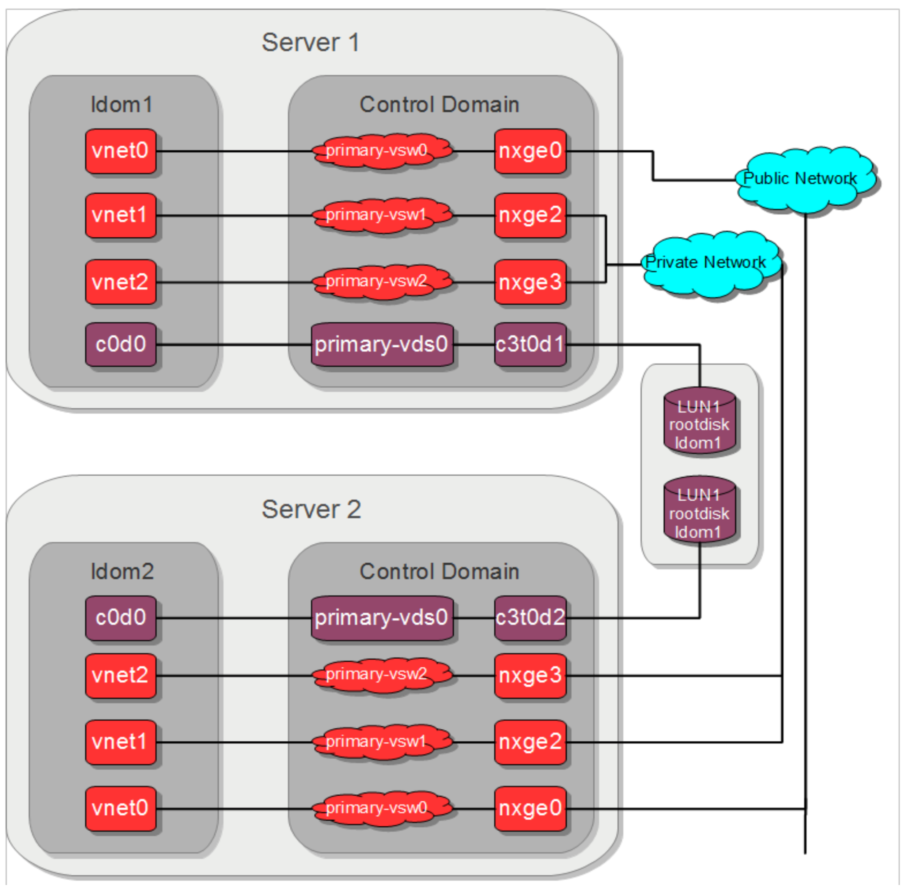


Figure 4: Guest domain initial configuration

Guest Domain Configuration

Each guest domain can be created with the following commands:

1. Create guest domain `ldom1`, from the control domain on server 1. In this example we again illustrate whole-core assignment, allocating 6 cores instead of 48 virtual CPU threads. Again, omit the `set-crypto` command if the server is a T4, since it is unneeded on that server platform:

```
# ldm create ldom1
# ldm set-core 6 ldom1
# ldm set-memory 24G ldom1
# ldm set-crypto 6 ldom1
# ldm add-vnet vnet0 primary-vsw0 ldom1
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom1
# ldm add-vnet linkprop=phys-state vnet2 primary-vsw2 ldom1
# ldm add-vdsdev /dev/rdsk/c3t0d1s2 ldom1@primary-vds0
# ldm add-vdisk ldom1 ldom1@primary-vds0 ldom1
```

2. Create guest domain `ldom2` from the control domain on server 2:

```
# ldm create ldom2
# ldm set-core 6 ldom2
# ldm set-memory 24G ldom2
# ldm set-crypto 6 ldom2
# ldm add-vnet vnet0 primary-vsw0 ldom2
# ldm add-vnet linkprop=phys-state vnet1 primary-vsw1 ldom2
# ldm add-vnet linkprop=phys-state vnet2 primary-vsw2 ldom2
# ldm add-vdsdev /dev/rdsk/c3t0d2s2 ldom2@primary-vds0
# ldm add-vdisk ldom2 ldom2@primary-vds0 ldom2
```

3. After domains have been created, they can be bound and started with the following commands. To bind and start ldom1, execute the following commands from the control domain on server 1:

```
# ldm bind ldom1
# ldm start ldom1
```

4. To bind and start ldom2, execute the following commands from the control domain on server2:

```
# ldm bind ldom2
# ldm start ldom2
```

The console of a domain can be accessed when the domain is bound. To do so, the console port associated with the domain is retrieved with the command “`ldm ls`”. Then the console can be accessed using the `telnet` command on the appropriate console port. For example, if the output from the `ldm ls` command indicates that the console port associated with a domain is 5000, then the command “`telnet localhost 5000`” can be used to access the console of that domain from the control domain.

After each guest domain is started, the appropriate Oracle Solaris OS and patches must be installed in those domains. The installation can be done over the network, from a DVD, or by using a DVD ISO image. Refer to the *Oracle VM Server for SPARC Administration Guide* for more information.

Once the installation of all guest domains is complete, you can continue setting up the systems so that they can be used with Oracle RAC. The additional setup includes configuration of a public and a private network (see "Network configuration" below), and the addition of shared storage (see the section on "Storage configuration").

Network Configuration

This section describes the networking configuration used in this example and includes network interfaces layout, IP address assignment and host names. It also describes the IPMP configuration.

Network Layout

The configuration requires two networks:

- A public network which is available for general use and which connects the domains. This network can also be interconnected with any other network such as a LAN or a WAN.
- A private network used by Oracle RAC (for example for the heartbeat and cache fusion). This network interconnects the two Oracle nodes and should not be connected to any other network. In particular, the private network should not be connected to the public network and it should not be used by any application other than Oracle RAC.

Public Network

The public network connects the primary network interfaces of the control domains (`vsw0`) and of the guest domains (`vnet0`), as shown in Figure 5. Each interface is connected to the first virtual switch (`primary-vsw0`) of the control domain. The virtual switch is also connected to the external public network using the `nxge0` interface. This public network can also be connected to a LAN or WAN.

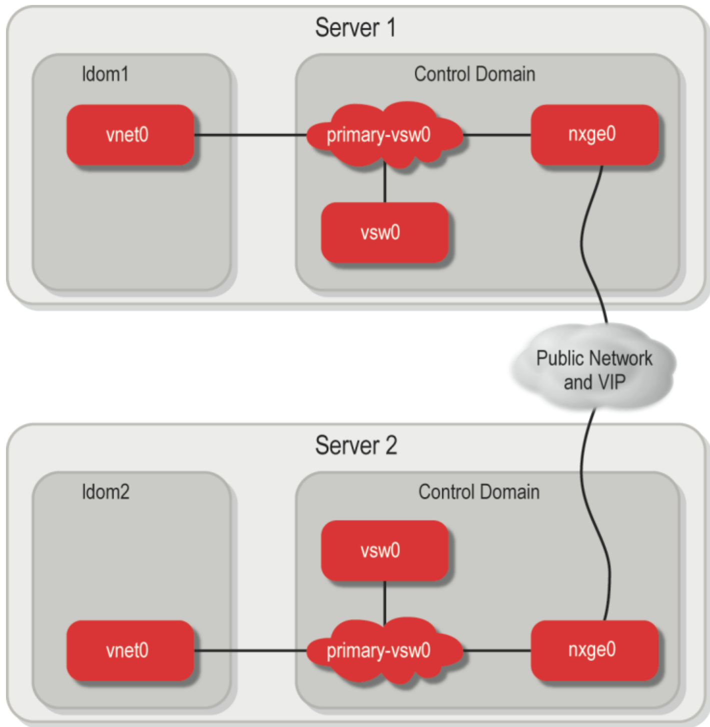


Figure 5: Public network configuration

Private Network

The private network connects the secondary network interfaces of the guest domains (`vnet1` and `vnet2`), as shown in Figure 6. Each interface (`vnet1` and `vnet2`) is connected to the virtual switch (`primary-vsw1` and `primary-vsw2`). The two servers are physically connected using interfaces `nxge2` and `nxge3` on each system to the physical network switch. Link-based IPMP is used on top of `vnet1` and `vnet2` to provide redundancy. There is no need to create virtual network interfaces in control domains to connect to the private network.

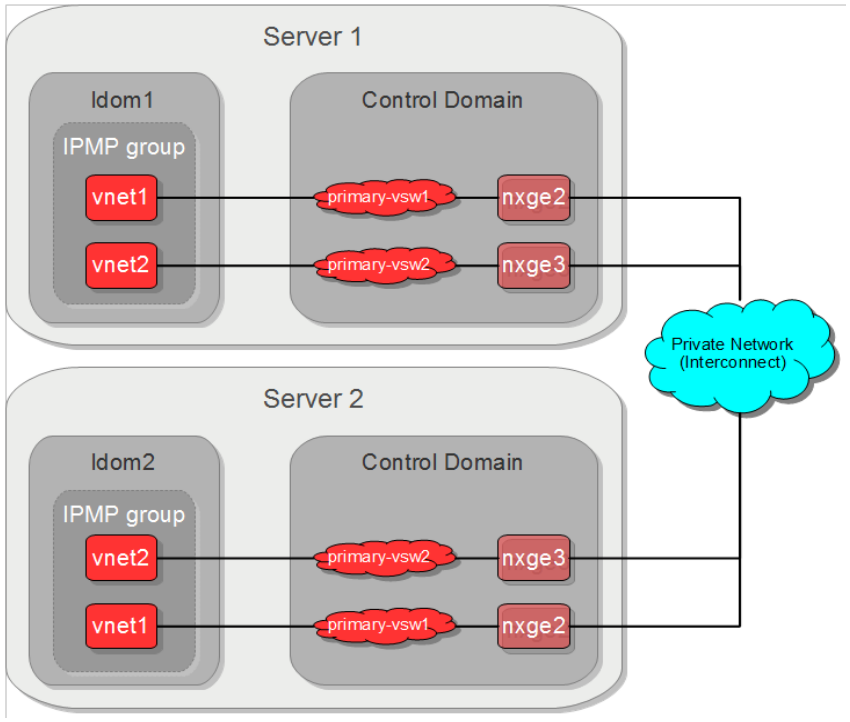


Figure 6: Private network configuration

Network Interface Summary

Figure 7 summarizes the interconnection of the network interfaces of the control and guest domains within a single server.

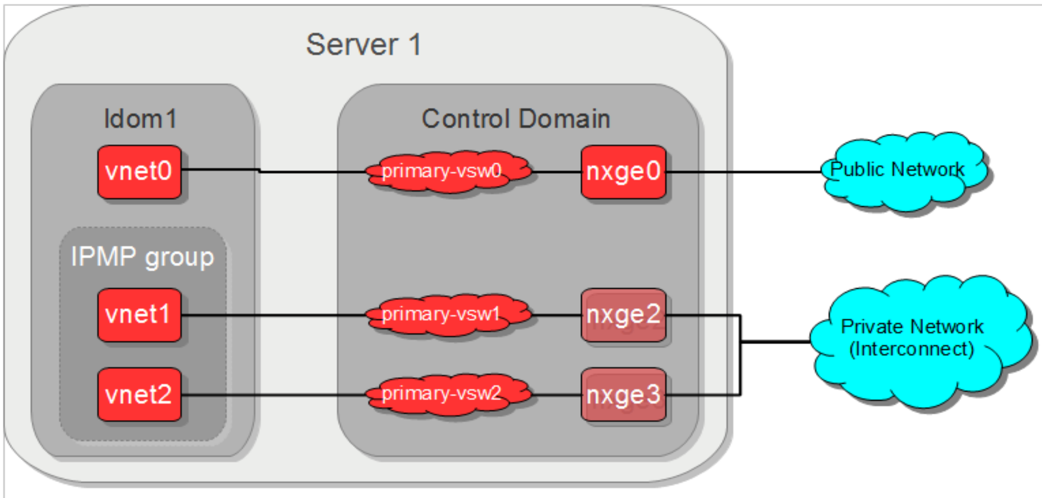


Figure 7: Networking within a physical server

Network Interfaces and Host Names

The configuration requires multiple host names and IP addresses. Table 3 summarizes the different host names and their association with network interfaces and systems:

TABLE 3. HOST NAME AND IP ADDRESS ASSIGNMENTS.

SERVER	DOMAIN	INTERFACE	HOST NAME	DESCRIPTION
		CONTROL DOMAIN — SERVER #1		
1	control	vsw0	rac01	Public network, vsw interface (primary)
		GUEST DOMAIN — SERVER #1		
1	ldom1	vnet0	rac-node1	Node 1, public network (primary)
1	ldom1	vnet1	node1-priv	Node 1, private network (IPMP group primary)
1	ldom1	vnet2	Not assigned	Node 1, private network (IPMP group standby)
1	ldom1	none	rac-nodevp1	Node 1, public network, virtual IP
		CONTROL DOMAIN — SERVER #2		
2	control	vsw0	rac02	Public network, vsw interface (primary)
		GUEST DOMAIN — SERVER #2		
2	ldom2	vnet0	rac-node2	Node 2, public network (primary)

2	ldom2	vnet1	node2-priv	Node 2, private network (IPMP group primary)
2	ldom2	vnet2	Not assigned	Node 2, private network (IPMP group standby)
2	ldom2	none	rac-nodevp2	Node 2, public network, virtual IP

These host names and the associated IP addresses must be defined in the naming service used by the domains, for example with DNS, NIS or `/etc/hosts`. The definition must be consistent across all domains. In addition, VIP and SCAN (for 11gR2) names must be available in DNS.

In this example configuration, the `/etc/hosts` file is defined as follows on all domains:

```
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1       localhost        loghost

# Public Network

10.1.9.101      rac01            # vsw0 - control domain server-1
10.1.9.102      rac02            # vsw0 - control domain server-2
10.1.9.111      rac-node1        # vnet0 - guest domain, server-1
10.1.9.112      rac-node2        # vnet0 - guest domain, server-2
10.1.9.121      rac-nodevp1      # vip of rac-node1
10.1.9.122      rac-nodevp2      # vip of rac-node2

# Private Network

192.168.10.111  node1-priv       # vnet1,2 - guest domain, server-1
192.168.10.112  node2-priv       # vnet1,2 - guest domain, server-2
```

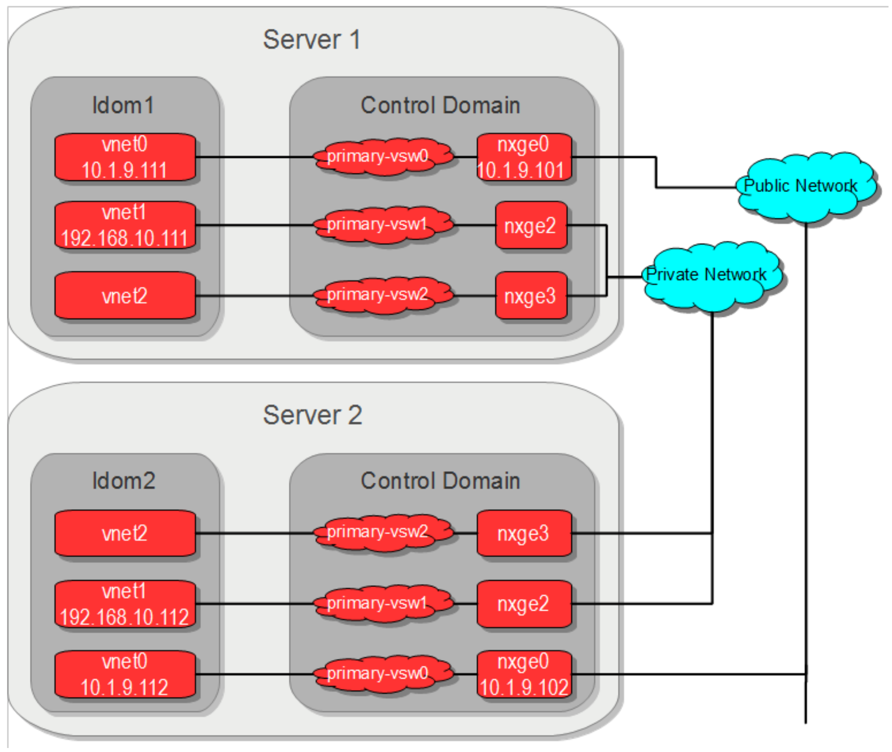


Figure 8: IP address assignment

Figure 8 shows the assignment of IP addresses for each network interface.

- The `nxge0` interface has no IP address because it is associated with the first virtual switch (`primary-vsw0`) and the system is using the interface of that virtual switch (`vsw0`). (This distinction is not needed in Solaris 11 control domains)
- The second and third virtual switches (`primary-vsw1` and `primary-vsw2`) are associated with physical network interfaces `nxge2` and `nxge3`.
- The `vnet1` and `vnet2` interfaces belong to network 192.168.10.0/24 and are used for the Oracle RAC private network. Link-based IPMP is configured on top of this interfaces. This network is not available to users.
- The `vnet0` and `vsw0` interfaces belong to network 10.1.9.0/24 and are used for public networking and virtual IP (VIP). This network is available to users.

Private Network Configuration

Configuration of the private network involves specifying the network interface name and configuring link-based IPMP in active/standby node.

The following setup is performed in Solaris 10 to configure IPMP in active/standby mode for interfaces `vnet1` and `vnet2`:

1. In guest domain `ldom1`:

```
# cat /etc/hostname.vnet1
node1-priv group ipmp1
# cat /etc/hostname.vnet2
group ipmp1 standby
```

2. In guest domain `ldom2`:

```
# cat /etc/hostname.vnet1
node2-priv group ipmp2
# cat /etc/hostname.vnet2
group ipmp2 standby
```

IPMP is configured differently in Oracle Solaris 11, which no longer relies on editing configuration files. Instead each domain would use commands like the following:

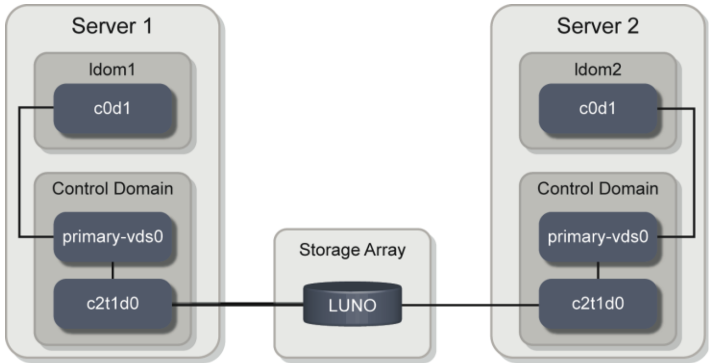
```
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i vnet1 -i vnet2 ipmp0
```

Storage Configuration

Guest domains were initially created with only one virtual disk (`c0d0`) used as the system disk for the operating system. In order to run Oracle RAC, additional disks must be assigned to both guest domains. These additional disks need to be visible from both guest domains.

Shared and Virtual Disks

Figure 9 shows the virtualization of a shared LUN, exported from external storage, as a virtual disk to the guest domain. The storage array is connected to the control domain, which exports the LUN as a virtual disk to the guest domain. When adding virtual disk drives to the guest domains it is recommended that LUNs be exported in the same order on both systems so that virtual disks will have the same device names on both nodes.



LUN0 is multihosted and visible with the same name (`c2t1d0`) on both control domains.

Figure 9: Shared storage is accessible to multiple domains using the same device name

Adding Shared Disks to Guest Domains

In this configuration, five LUNs from the second storage array are used as shared disks. Because the hardware configuration is the same on both servers, the LUNs from the second storage array appear with the same device names (`c2t1d0` to `c2t1d4`) on both control domains.

Note: LUNs may (and often will) appear with different names on different servers. LUN names in control domains are not required to be identical. LUN names within guest domains must be identical. This can easily be achieved by importing LUNs in the same order across all domains.

To export a LUN as virtual disk, execute the following commands. In the following example, `c2t1d0s2` is the device name for the OCR LUN.

1. To add the LUN to `ldom1`, execute the following commands from the control domain of server 1:

```
# ldm add-vdsdev /dev/rdsk/c2t1d0s2 ocr1@primary-vds0
# ldm add-vdisk ocr1 ocr1@primary-vds0 ldom1
```

2. To add the LUN to `ldom2`, execute the following commands from the control domain of server 2:

```
# ldm add-vdsdev /dev/rdsk/c2t1d0s2 ocr2@primary-vds0
# ldm add-vdisk ocr2 ocr2@primary-vds0 ldom2
```

The new disk will be immediately added to the guest domain. Once the disk is visible, use the `format(1m)` command from any guest domain to partition the new disk and create the required partition for OCR. Check from all guest domains that the correct partitioning is visible.

The same steps must be repeated for each shared disk (voting disks and ASM disk). Figure 10 shows the final configuration after all shared disks have been added to guest domains.

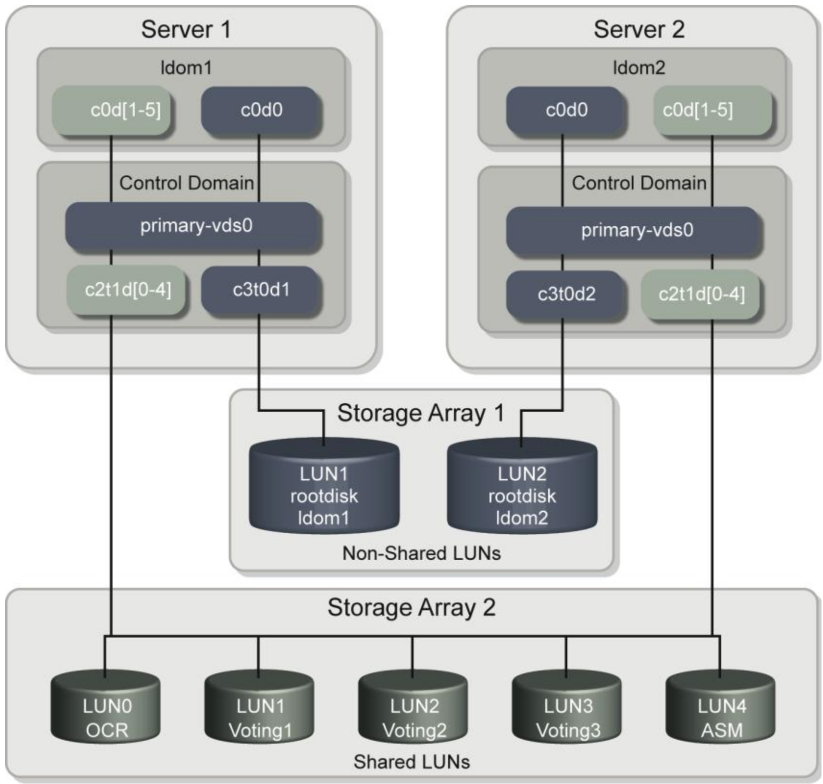


Figure 10: Storage cabling

Oracle Installation

The installation of the Oracle RAC and Oracle database software is similar to a standard Oracle installation. Use network 10.1.9.0/24 for public and VIP networks, and 192.168.10.0/24 for the private network. Start with the installation of Oracle Clusterware and then apply the latest patch set. Continue with the installation of Oracle database, and apply the same patch set this time to the Oracle database.

Configuring IPMP for the cluster interconnect

Consult with Doc ID 1069584.1 on http://support.oracle.com to configure IPMP for the cluster interconnect. For Oracle 11gR2 it is sufficient to select `vnet1` as the private interface when doing Oracle installation and execute the following post-installation steps:

```
# $GRID_HOME/bin/oifcfg getif
vnet0  10.6.241.0  global  public
vnet1  192.168.1.0  global  cluster_interconnect
# $GRID_HOME/bin/oifcfg setif -global vnet2/192.168.1.0:cluster_interconnect
# $GRID_HOME/bin/oifcfg getif
vnet0  10.6.241.0  global  public
vnet1  192.168.1.0  global  cluster_interconnect
vnet2  192.168.1.0  global  cluster_interconnect
```

Additional Information

This section contains additional information relevant to Oracle RAC deployments on Oracle VM Server for SPARC, including:

- NTP
- CPU dynamic reconfiguration
- Memory dynamic reconfiguration
- Performance considerations
- Jumbo Frames

NTP

The system clock of the different domains should be synchronized. This can be done by using the network time synchronization protocol (NTP) across all domains.

In this example the control domain on the first server (`rac01`) is used as a time source, and configured as an NTP server:

```
# grep -v ^# /etc/inet/ntp.conf
server 127.127.1.0 prefer
broadcast 224.0.1.1 ttl 4
enable auth monitor
driftfile /var/ntp/ntp.drift
statsdir /var/ntp/ntpstats/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable
keys /etc/inet/ntp.keys
trustedkey 0
requestkey 0
controlkey 0
# touch /var/ntp/ntp.drift
# svcadm enable ntp
```

Other domains (`rac02`, `rac-node1` and `rac-node2`) are configured as NTP clients:

```
# grep -v ^# /etc/inet/ntp.conf
server rac01 prefer
slewalways yes
disable pll
# svcadm enable ntp
```

CPU Dynamic Reconfiguration

Oracle VM Server for SPARC supports CPU dynamic reconfiguration. CPUs can be dynamically added or removed from any active domain (including the control domain), using the `ldm add-vcpu` or `ldm rm-vcpu` commands from the control domain. CPUs can be allocated in units of whole cores by using the `ldm add-core` or `ldm rm-core` commands. These commands can be used with a domain running Oracle RAC.

When reducing the number of CPUs in a domain, ensure that enough CPUs are still available to efficiently handle the workload of the domain (an Oracle RAC guest domain should always have at least 8 virtual CPUs). The `ldm list` command can be used to display CPU utilization from the control domain. The `vmstat, mpstat` and `prstat` commands can be used within a domain to show CPU utilization and latency due to waiting for CPU resources, just as they can be used in Solaris when not running in a virtual environment. Changes in the number of CPUs are recognized immediately, without requiring a reboot.

If a domain is configured with hard partitioning then you can only add or remove entire CPU cores from the domain. Also the number of CPU cores assigned to a domain can be limited by the number of CPU cores specified for the domain by using the `ldm set max-cores` command.

If several domains are deployed on the same physical server then a CPU resource manager policy can be set to automate CPU dynamic reconfiguration in response to changing workload conditions. Policies can be set to give CPUs to domains that need them and remove excess CPUs from domains that don't. The following example will add or remove CPUs to and from ldom1 in the range of a minimum of 24 and a maximum of 64, depending on the load and time of day. If CPU utilization in the domain drops below 25%, CPUs are removed till the lower limit of 24 CPUs is reached or CPU utilization exceeds 25%. If CPU utilization exceeds 75%, CPUs are added till the upper limit of 64 CPUs is reached or utilization drops below 75%:

```
# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-upper=75 vcpu-min=24
vcpu-max=64 attack=8 decay=8 priority=1 name=high-usage ldom1
```

Memory Dynamic Reconfiguration

Oracle VM Server for SPARC supports memory dynamic reconfiguration. Memory can be dynamically added or removed from any active domain, using the `ldm add-mem`, `ldm rm-mem` or `ldm set-mem` commands. Adding or removing memory does not require a Solaris restart. When reducing the amount of memory assigned to a guest domain, consider the amount of memory assigned to each Oracle database instance running with this domain. Don't go below this amount or your system will experience swapping and have reduced performance.

- In order to reduce the amount of memory, you may need to first reduce the amount of memory assigned to Oracle database instances running on the domain. You may also need to restart these instances depending on the type of memory management used.

- After adding memory to a domain, you may need to adjust the memory setting of database instances running on that domain in order to exploit the additional memory. You may also need to restart these instances depending on the type of memory management used.

Performance Considerations

With the configuration shown in this example, control domains are responsible for I/O and network flow processing. Tests have shown that the CPU's system load on control domains is around 15% for a storage I/O load of 300 MB/sec on each control domain and a network load around 25 MB/sec.

Avoid swap activity on the guest domains because any swap activity on the guest domains will generate additional I/O requests, which must be handled by the control, service, and domains. Avoiding swapping is a best practice in a non-virtual environment, but has additional importance with a virtual machine.

Do not share CPU cores between domains, especially between two domains running the Oracle database. If CPU threads from the same CPU core are assigned to different domains, then this can reduce the efficiency of these CPU threads due to contention for the level 1 cache on each core. This is best done by using the whole-core constraint described previously, but can also be done by uniformly allocating CPU threads in increments of 8 (the number of threads on a core).

It can also be helpful to prevent domains from spanning multiple sockets on larger, multi-socket servers like the M5-32 or T5-8, since that can increase memory latency within a domain due to Non Uniform Memory Access (NUMA) effects. This can be explicitly controlled by using named CPU resources, as described in the section "Assigning Physical Resources to Domains" in the *Oracle VM for SPARC 3.1 Administration Guide,* but is automatically optimized by the Oracle VM Server for SPARC logical domains manager, which optimizes for NUMA latency when a domain is bound.

Jumbo Frames

Starting with Oracle VM Server for SPARC 2.2, virtual switches and virtual network devices can be configured for jumbo frames, and you can set the maximum transmission unit (MTU) up to 16,000 bytes. For more information about configuring jumbo frames for virtual switches and virtual network devices, see the *Oracle VM for SPARC Administration Guide*.

In order to configure Jumbo Frames, all NICs and network switches must support them. The following is an example configuration with `nxge2` and `nxge3` interfaces configured to support Jumbo Frames. It is better to configure jumbo frames while configuring the primary domain and before creating the guest domain. This will save your time by reducing the number of system restarts.

1. Determine the device path of the network interfaces that you want to configure with Jumbo Frames - `nxge2` amd `nxge3`:

```
# cat /etc/path_to_inst | grep nxge
"/pci@500/pci@0/pci@c/network@0" 0 "nxge"
"/pci@500/pci@0/pci@c/network@0,1" 1 "nxge"
"/pci@500/pci@0/pci@c/network@0,2" 2 "nxge"
"/pci@500/pci@0/pci@c/network@0,3" 3 "nxge"
```

2. Edit the driver configuration file (`nxge.conf`) to enable Jumbo Frames on the specified devices:

```
# cd /platform/sun4v/kernel/drv
# vi nxge.conf
name = "pciex108e,abcd" parent = "/pci@500/pci@0/pci@c" unit-address = "0,2"
accept_jumbo = 1;
name = "pciex108e,abcd" parent = "/pci@500/pci@0/pci@c" unit-address = "0,3"
accept_jumbo = 1;
# init 6
```

3. Modify virtual switches to support MTU size 9000:

```
# dladm show-link
vsw0          type: non-vlan  mtu: 1500      device: vsw0
vsw1          type: non-vlan  mtu: 1500      device: vsw1
vsw2          type: non-vlan  mtu: 1500      device: vsw2
nxge0         type: non-vlan  mtu: 1500      device: nxge0
nxge1         type: non-vlan  mtu: 1500      device: nxge1
nxge2         type: non-vlan  mtu: 9194      device: nxge2
nxge3         type: non-vlan  mtu: 9194      device: nxge3
```

```
# ldm set-vsw mtu=9000 primary-vsw2
# ldm set-vsw mtu=9000 primary-vsw3
# init 6
```

4. `vnet2` and `vnet3` in guest domains will automatically receive MTU size 9000.

Summary

Oracle RAC can be installed on servers configured with Oracle VM Server for SPARC, a virtualization technology that allows the creation of multiple virtual systems on a single physical system. Multiple Oracle RAC nodes can be configured on logical domains on the same physical server, for a lower-cost development option. Alternatively, Oracle RAC nodes can be placed on logical domains on separate physical servers to provide better availability for production deployments.

This paper presents an example configuration of the production variant, with Oracle RAC nodes placed in logical domains on two separate physical servers. Step-by-step instructions describe the complete configuration process, including logical domains creation and configuration, networking setup, and storage configuration. Configuration guidelines and software requirements are also included, to help administrators plan for their deployments of Oracle RAC on Oracle VM Server for SPARC.

Appendices

Logical Domains Manager Configuration Example

This section shows the Oracle VM Server for SPARC configuration described as an example in this document. These are only examples; actual configurations may be different.

Configuration of the First Control Domain (rac01)

```
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
primary          active     -n-cv-  SP      32    16G      2.9%  2d 16h 18m

SOFTSTATE
Solaris running

UUID
    701ddc0f-a616-e982-bba8-a9bbfbec2c04

MAC
    00:14:4f:aa:c4:b8

HOSTID
    0x84aac4b8

CONTROL
    failure-policy=ignore
    extended-mapin-space=off

    cpu-arch=native


DEPENDENCY
    master=

CORE
    CID    CPUSET
    0      (0, 1, 2, 3, 4, 5, 6, 7)
    1      (8, 9, 10, 11, 12, 13, 14, 15)
    2      (16, 17, 18, 19, 20, 21, 22, 23)
    5      (40, 41, 42, 43, 44, 45, 46, 47)

VCPU
    VID    PID    CID    UTIL STRAND
    0      0      0      3.9%   100%
    1      1      0      2.9%   100%
    2      2      0      2.7%   100%
    3      3      0      0.6%   100%
```

```
    4      4      0      1.3%   100%
    5      5      0      4.7%   100%
    6      6      0      1.2%   100%
    7      7      0      2.2%   100%
    8      8      1      1.4%   100%
    9      9      1      2.0%   100%
    10     10     1      5.7%   100%
    11     11     1      0.2%   100%
    12     12     1      0.1%   100%
    13     13     1      4.7%   100%
    14     14     1      9.9%   100%
    15     15     1      0.2%   100%
    16     16     2      0.4%   100%
    17     17     2      1.8%   100%
    18     18     2      2.8%   100%
    19     19     2      4.2%   100%
    20     20     2      0.7%   100%
    21     21     2      0.2%   100%
    22     22     2      0.9%   100%
    23     23     2      4.7%   100%
    24     40     5      7.8%   100%
    25     41     5      0.1%   100%
    26     42     5      0.2%   100%
    27     43     5      1.8%   100%
    28     44     5      7.7%   100%
    29     45     5      0.2%   100%
    30     46     5      0.2%   100%
    31     47     5       18%   100%

MAU
    ID     CPUSET
    0      (0, 1, 2, 3, 4, 5, 6, 7)
    1      (8, 9, 10, 11, 12, 13, 14, 15)
    2      (16, 17, 18, 19, 20, 21, 22, 23)
    5      (40, 41, 42, 43, 44, 45, 46, 47)

MEMORY
    RA                PA                SIZE
    0x10000000        0x4a0000000       256M
    0x402000000       0x12000000        2G
    0x890000000       0x130000000       14080M


CONSTRAINT

    cpu=whole-core

    max-cores=unlimited
```

```
    threading=max-throughput


VARIABLES
    keyboard-layout=US-English

IO
    DEVICE           PSEUDONYM        OPTIONS
    pci@400          pci_0
    pci@500          pci_1
    pci@400/pci@0/pci@d MB/PCIE0
    pci@400/pci@0/pci@c MB/PCIE1
    pci@400/pci@0/pci@1 MB/HBA
    pci@500/pci@0/pci@d MB/PCIE4
    pci@500/pci@0/pci@9 MB/PCIE5
    pci@500/pci@0/pci@c MB/NET0

VCC
    NAME             PORT-RANGE
    primary-vcc0     5000-5100

VSW
    NAME              MAC               NET-DEV   ID   DEVICE     LINKPROP   DEFAULT-VLAN-ID PVID
VID                   MTU   MODE
    primary-vsw0      00:14:4f:fb:d0:c0 nxge0     0    switch@0              1               1
1500
    primary-vsw1      00:14:4f:fa:99:8b nxge2     1    switch@1   phys-state 1               1
9000
    primary-vsw2      00:14:4f:f9:87:e8 nxge3     2    switch@2   phys-state 1               1
9000

VDS
    NAME             VOLUME         OPTIONS          MPGROUP        DEVICE
    primary-vds0     ldom1                                          /dev/rdsk/c3t0d1s2
                     ocr1                                           /dev/rdsk/c2t1d0s2
                     voting11                                       /dev/rdsk/c2t1d1s2
                     voting21                                       /dev/rdsk/c2t1d2s2
                     voting31                                       /dev/rdsk/c2t1d3s2
                     ASM1                                           /dev/rdsk/c2t1d4s2

VCONS
     NAME            SERVICE                     PORT
     SP
```

Configuration of the First Guest Domain (ldom1)

```
NAME           STATE    FLAGS    CONS    VCPU  MEMORY   UTIL   UPTIME
ldom1          active   -n----   5000    48    24G       15%   3h 58m

SOFTSTATE
Solaris running

UUID
    f3e1006b-16d4-e266-8447-8eccc1518345

MAC
    00:14:4f:fa:f7:c8

HOSTID
    0x84faf7c8

CONTROL
    failure-policy=ignore
    extended-mapin-space=off

    cpu-arch=native


DEPENDENCY
    master=

CORE
    CID    CPUSET
    3      (24, 25, 26, 27, 28, 29, 30, 31)
    4      (32, 33, 34, 35, 36, 37, 38, 39)
    6      (48, 49, 50, 51, 52, 53, 54, 55)
    9      (72, 73, 74, 75, 76, 77, 78, 79)
    10     (80, 81, 82, 83, 84, 85, 86, 87)
    11     (88, 89, 90, 91, 92, 93, 94, 95)

VCPU
    VID    PID    CID    UTIL STRAND
    0      24     3       51%   100%
    1      25     3       32%   100%
    2      26     3       22%   100%
    3      27     3       28%   100%
    4      28     3       33%   100%
    5      29     3       37%   100%
    6      30     3       19%   100%
    7      31     3       29%   100%
    8      32     4       22%   100%
    9      33     4       28%   100%
    10     34     4       29%   100%
```

```
    11     35     4       24%   100%
    12     36     4       16%   100%
    13     37     4       16%   100%
    14     38     4       15%   100%
    15     39     4       15%   100%
    16     48     6       15%   100%
    17     49     6       33%   100%
    18     50     6       31%   100%
    19     51     6       23%   100%
    20     52     6       21%   100%
    21     53     6       23%   100%
    22     54     6       29%   100%
    23     55     6       27%   100%
    24     72     9      4.3%   100%
    25     73     9      1.7%   100%
    26     74     9      5.4%   100%
    27     75     9      3.7%   100%
    28     76     9      2.3%   100%
    29     77     9      3.8%   100%
    30     78     9      2.9%   100%
    31     79     9      2.8%   100%
    32     80     10     4.5%   100%
    33     81     10     1.9%   100%
    34     82     10     1.8%   100%
    35     83     10     3.7%   100%
    36     84     10     3.4%   100%
    37     85     10     3.5%   100%
    38     86     10     3.7%   100%
    39     87     10     3.0%   100%
    40     88     11     2.0%   100%
    41     89     11     5.3%   100%
    42     90     11     5.4%   100%
    43     91     11     6.8%   100%
    44     92     11     3.2%   100%
    45     93     11     2.8%   100%
    46     94     11     1.6%   100%
    47     95     11     4.9%   100%

MAU
    ID     CPUSET
    3      (24, 25, 26, 27, 28, 29, 30, 31)
    4      (32, 33, 34, 35, 36, 37, 38, 39)
    6      (48, 49, 50, 51, 52, 53, 54, 55)
    9      (72, 73, 74, 75, 76, 77, 78, 79)
    10     (80, 81, 82, 83, 84, 85, 86, 87)
    11     (88, 89, 90, 91, 92, 93, 94, 95)

MEMORY
    RA                PA                SIZE
```

```
    0x2000000        0x92000000       2528M
    0x400000000      0x4b0000000      13856M
    0x800000000      0x820000000      8G


CONSTRAINT

    cpu=whole-core

    max-cores=unlimited

    threading=max-throughput


VARIABLES
    boot-device=/virtual-devices@100/channel-devices@200/disk@0:a vnet0
    keyboard-layout=US-English

NETWORK
    NAME             SERVICE                     ID   DEVICE     MAC               MODE   PVID VID
MTU   LINKPROP
    vnet0            primary-vsw0@primary        0    network@0  00:14:4f:f8:51:5f        1
1500
    vnet1            primary-vsw1@primary        1    network@1  00:14:4f:f8:b4:fb        1
9000  phys-state
    vnet2            primary-vsw2@primary        2    network@2  00:14:4f:fa:13:f0        1
9000  phys-state

DISK
    NAME            VOLUME                     TOUT  DEVICE    SERVER
    primary-vds0    ldom1@primary-vds0               disk@0    primary
                    ocr1@primary-vds0                disk@1    primary
                    voting11@primary-vds0            disk@2    primary
                    voting21@primary-vds0            disk@3    primary
                    voting31@primary-vds0            disk@4    primary
                    ASM1@primary-vds0                disk@5    primary

VCONS
     NAME           SERVICE                     PORT
     ldom1          primary-vcc0@primary        5000
```

Configuration of the Second Control Domain (rac02)

```
NAME             STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
primary          active     -n-cv-  SP      32    16G      2.8%  5d 106h 22m

SOFTSTATE
Solaris running

UUID
    701ddc0f-a616-e982-bba8-a9ccfbec2c04

MAC
    00:14:4f:bb:c4:b8

HOSTID
    0x84bbc4b8

CONTROL
    failure-policy=ignore
    extended-mapin-space=off

    cpu-arch=native

DEPENDENCY
    master=

CORE
    CID    CPUSET
    0      (0, 1, 2, 3, 4, 5, 6, 7)
    1      (8, 9, 10, 11, 12, 13, 14, 15)
    2      (16, 17, 18, 19, 20, 21, 22, 23)
    5      (40, 41, 42, 43, 44, 45, 46, 47)

VCPU
    VID    PID    CID    UTIL STRAND
    0      0      0      3.9%   100%
    1      1      0      2.9%   100%
    2      2      0      2.7%   100%
    3      3      0      0.6%   100%
    4      4      0      1.3%   100%
    5      5      0      4.7%   100%
    6      6      0      1.2%   100%
    7      7      0      2.2%   100%
    8      8      1      1.4%   100%
    9      9      1      2.0%   100%
    10     10     1      5.7%   100%
    11     11     1      0.2%   100%
    12     12     1      0.1%   100%
    13     13     1      4.7%   100%
```

```
    14     14     1      9.9%   100%
    15     15     1      0.2%   100%
    16     16     2      0.4%   100%
    17     17     2      1.8%   100%
    18     18     2      2.8%   100%
    19     19     2      4.2%   100%
    20     20     2      0.7%   100%
    21     21     2      0.2%   100%
    22     22     2      0.9%   100%
    23     23     2      4.7%   100%
    24     40     5      7.8%   100%
    25     41     5      0.1%   100%
    26     42     5      0.2%   100%
    27     43     5      1.8%   100%
    28     44     5      7.7%   100%
    29     45     5      0.2%   100%
    30     46     5      0.2%   100%
    31     47     5       18%   100%

MAU
    ID     CPUSET
    0      (0, 1, 2, 3, 4, 5, 6, 7)
    1      (8, 9, 10, 11, 12, 13, 14, 15)
    2      (16, 17, 18, 19, 20, 21, 22, 23)
    5      (40, 41, 42, 43, 44, 45, 46, 47)

MEMORY
    RA                PA                SIZE
    0x10000000        0x4a0000000       256M
    0x402000000       0x12000000        2G
    0x890000000       0x130000000       14080M


CONSTRAINT

    cpu=whole-core

    max-cores=unlimited

    threading=max-throughput


VARIABLES
    keyboard-layout=US-English

IO
    DEVICE           PSEUDONYM        OPTIONS
    pci@400          pci_0
    pci@500          pci_1
```

```
    pci@400/pci@0/pci@d MB/PCIE0
    pci@400/pci@0/pci@c MB/PCIE1
    pci@400/pci@0/pci@1 MB/HBA
    pci@500/pci@0/pci@d MB/PCIE4
    pci@500/pci@0/pci@9 MB/PCIE5
    pci@500/pci@0/pci@c MB/NET0

VCC
    NAME            PORT-RANGE
    primary-vcc0    5000-5100

VSW
    NAME             MAC               NET-DEV   ID   DEVICE     LINKPROP   DEFAULT-VLAN-ID PVID
VID                  MTU   MODE
    primary-vsw0     00:14:4f:fa:d0:c0 nxge0     0    switch@0              1               1
1500
    primary-vsw1     00:14:4f:fb:88:8b nxge2     1    switch@1   phys-state 1               1
9000
    primary-vsw2     00:14:4f:fb:86:e8 nxge3     2    switch@2   phys-state 1               1
9000

VDS
    NAME            VOLUME         OPTIONS           MPGROUP        DEVICE
    primary-vds0    ldom2                                           /dev/rdsk/c3t0d2s2
                    ocr2                                            /dev/rdsk/c2t1d0s2
                    voting12                                        /dev/rdsk/c2t1d1s2
                    voting22                                        /dev/rdsk/c2t1d2s2
                    voting32                                        /dev/rdsk/c2t1d3s2
                    ASM2                                            /dev/rdsk/c2t1d4s2

VCONS
     NAME           SERVICE                     PORT
     SP
```

Configuration of the Second Guest Domain (ldom2)

```
NAME          STATE    FLAGS    CONS    VCPU  MEMORY   UTIL   UPTIME
ldom2         active   -n----   5000    48    24G       19%   12h 38m

SOFTSTATE
Solaris running

UUID
    f3e1006b-16d4-e266-8447-8ebbb2619456

MAC
    00:14:4f:fb:43:cc

HOSTID
    0x84fb43cc

CONTROL
    failure-policy=ignore
    extended-mapin-space=off

    cpu-arch=native

DEPENDENCY
    master=

CORE
    CID    CPUSET
    3      (24, 25, 26, 27, 28, 29, 30, 31)
    4      (32, 33, 34, 35, 36, 37, 38, 39)
    6      (48, 49, 50, 51, 52, 53, 54, 55)
    9      (72, 73, 74, 75, 76, 77, 78, 79)
    10     (80, 81, 82, 83, 84, 85, 86, 87)
    11     (88, 89, 90, 91, 92, 93, 94, 95)

VCPU
    VID    PID    CID    UTIL STRAND
    0      24     3       51%   100%
    1      25     3       32%   100%
    2      26     3       22%   100%
    3      27     3       28%   100%
    4      28     3       33%   100%
    5      29     3       37%   100%
    6      30     3       19%   100%
    7      31     3       29%   100%
    8      32     4       22%   100%
    9      33     4       28%   100%
    10     34     4       29%   100%
    11     35     4       24%   100%
```

```
    12     36     4       16%   100%
    13     37     4       16%   100%
    14     38     4       15%   100%
    15     39     4       15%   100%
    16     48     6       15%   100%
    17     49     6       33%   100%
    18     50     6       31%   100%
    19     51     6       23%   100%
    20     52     6       21%   100%
    21     53     6       23%   100%
    22     54     6       29%   100%
    23     55     6       27%   100%
    24     72     9      4.3%   100%
    25     73     9      1.7%   100%
    26     74     9      5.4%   100%
    27     75     9      3.7%   100%
    28     76     9      2.3%   100%
    29     77     9      3.8%   100%
    30     78     9      2.9%   100%
    31     79     9      2.8%   100%
    32     80     10     4.5%   100%
    33     81     10     1.9%   100%
    34     82     10     1.8%   100%
    35     83     10     3.7%   100%
    36     84     10     3.4%   100%
    37     85     10     3.5%   100%
    38     86     10     3.7%   100%
    39     87     10     3.0%   100%
    40     88     11     2.0%   100%
    41     89     11     5.3%   100%
    42     90     11     5.4%   100%
    43     91     11     6.8%   100%
    44     92     11     3.2%   100%
    45     93     11     2.8%   100%
    46     94     11     1.6%   100%
    47     95     11     4.9%   100%

MAU
    ID     CPUSET
    3      (24, 25, 26, 27, 28, 29, 30, 31)
    4      (32, 33, 34, 35, 36, 37, 38, 39)
    6      (48, 49, 50, 51, 52, 53, 54, 55)
    9      (72, 73, 74, 75, 76, 77, 78, 79)
    10     (80, 81, 82, 83, 84, 85, 86, 87)
    11     (88, 89, 90, 91, 92, 93, 94, 95)

MEMORY
    RA               PA               SIZE
    0x2000000        0x92000000       2528M
```

```
    0x400000000      0x4b0000000      13856M
    0x800000000      0x820000000      8G


CONSTRAINT

    cpu=whole-core

    max-cores=unlimited

    threading=max-throughput


VARIABLES
    boot-device=/virtual-devices@100/channel-devices@200/disk@0:a vnet0
    keyboard-layout=US-English

NETWORK
    NAME             SERVICE                     ID   DEVICE     MAC               MODE   PVID VID
MTU   LINKPROP
    vnet0            primary-vsw0@primary        0    network@0  00:14:4f:f9:02:a7        1
1500
    vnet1            primary-vsw1@primary        1    network@1  00:14:4f:f8:c4:fb        1
9000  phys-state
    vnet2            primary-vsw2@primary        2    network@2  00:14:4f:fa:23:f0        1
9000  phys-state

DISK
    NAME            VOLUME                     TOUT  DEVICE    SERVER
    primary-vds0    ldom2@primary-vds0               disk@0    primary
                    ocr2@primary-vds0                disk@1    primary
                    voting12@primary-vds0            disk@2    primary
                    voting22@primary-vds0            disk@3    primary
                    voting32@primary-vds0            disk@4    primary
                    ASM2@primary-vds0                disk@5    primary

VCONS
     NAME           SERVICE                     PORT
     ldom2          primary-vcc0@primary        5000
```

References

Relevant Oracle publications mentioned throughout the article are listed in the table below.

TABLE 5. REFERENCES FOR MORE INFORMATION.

REFERENCES	URL
Oracle Virtualization	http://oracle.com/virtualization
Oracle VM Server for SPARC Technical White Paper	http://www.oracle.com/technetwork/server-storage/vm/overview/index.html
Oracle Real Application Clusters (RAC)	http://www.oracle.com/technetwork/database/options/clustering/overview/index.html
Oracle Clusterware	http://www.oracle.com/technetwork/database/clusterware/overview/index-090666.html
Oracle Automatic Storage Management	http://www.oracle.com/technetwork/database/database-technologies/cloud-storage/index.html
Best Practices for Data Reliability with Oracle VM Server for SPARC	http://www.oracle.com/technetwork/articles/systems-hardware-architecture/vmsrvrsparc-reliability-163931.pdf
Best Practices for Network Availability with Oracle VM Server for SPARC	http://www.oracle.com/technetwork/articles/systems-hardware-architecture/vmsrvrsparc-availability-163930.pdf

ORACLE®

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

Running Oracle Real Application Clusters on Oracle VM Server for SPARC
March 2014, version 1.2
Author: Alexandre Chartre, Roman Ivanov, John Mchugh and Jeff Savit

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

Hardware and Software, Engineered to Work Together