



An Oracle White Paper  
September 2010

# Deploying Oracle WebLogic (Java EE) applications on Oracle Sun T-series Servers and Sun Storage 7000 Unified Storage

<a href="#">Introduction.....</a>	<a href="#">4</a>
<a href="#">Reference Architecture .....</a>	<a href="#">6</a>
<a href="#">Major Components.....</a>	<a href="#">6</a>
<a href="#">Architecture overview.....</a>	<a href="#">10</a>
<a href="#">Benefits .....</a>	<a href="#">16</a>
<a href="#">Summary.....</a>	<a href="#">20</a>
<a href="#">Variations .....</a>	<a href="#">21</a>
<a href="#">Separate T-series Database Server.....</a>	<a href="#">21</a>
<a href="#">Oracle Real Application Cluster (RAC).....</a>	<a href="#">22</a>
<a href="#">WebLogic Clustering .....</a>	<a href="#">23</a>
<a href="#">Other Reliability Variations.....</a>	<a href="#">23</a>
<a href="#">Setup Guide.....</a>	<a href="#">24</a>
<a href="#">System installation and setup .....</a>	<a href="#">24</a>
<a href="#">Software installation.....</a>	<a href="#">35</a>
<a href="#">References .....</a>	<a href="#">41</a>
<a href="#">Appendix A – Creating and Installing a Zone .....</a>	<a href="#">42</a>
<a href="#">Appendix B – Creating an iSCSI target and ZFS pool.....</a>	<a href="#">44</a>

## Introduction

Java Enterprise Edition (Java EE) applications have become the standard for developing and deploying enterprise web applications. Oracle provides hardware and software that is designed to work together, and gives many benefits to do so:

- **Full platform** – All components from the application server to the database, the hardware they run on, are part of the Oracle platform: no third party software is needed to deploy your Java EE applications.
- **Tested together** – Having the Oracle database and Oracle WebLogic server tested by Oracle engineers on Oracle server hardware and storage hardware ensures a more robust platform than when such testing is done by third parties.
- **Supported together** – The support for all the components: hardware and software including visualization and operating system is from one vendor with a single point of contact.
- **Fully certified** – All of the infrastructure components into which your Java EE application is being deployed have been certified by Oracle engineers.

With these benefits and others in mind this paper documents a reference architecture for deploying Oracle WebLogic Server / Java EE applications onto Oracle Sun T-series servers and storage.

Specifically this document provides an example of consolidating multiple Java EE applications and workloads onto a single Sun SPARC T-series server with Sun Storage. In this architecture the WebLogic 11g application server and Oracle 11gR2 database tiers are combined onto a single server by utilizing Oracle virtualization technologies. Additionally, this paper presents some alternatives to the described architecture to allow the reader to explore other, more complex options such as greater availability via clustering.

## **How this document is organized**

This paper is divided into three sections, the first describes the reference architecture and the second describes possible variations of this architecture. The third section is specifically designed for Java EE application deployers and system administrators to assist in the system configuration and deployment of their Java EE applications.

## **Scope**

This paper is a reference architecture, detailing systems Oracle engineers have built and tested and are typical of Oracle customer deployments. This paper is not, however, a “best practices guide”. Full system tuning and performance recommendations are beyond the scope of this document. References are supplied for more information that may assist the reader in these areas.

## **Audience**

This document is intended to be read by systems administrators, Java EE application developers and deployers and to anyone interested in modern virtualized systems architectures. Additionally it may be of interest to those wanting to know more about the various Oracle software and hardware components and advantages in delivering a consolidated topology from a single vendor.

It assumes the reader is familiar with Java EE concepts, a general understanding of computer systems hardware and the command line, and some familiarity with a UNIX or Linux-like environment.

## Reference Architecture

### Major Components

The reference architecture presented in this paper has many components integrated together from hardware to software, and from applications to workloads. This chapter provides a summary and details about each of these.

The following is a summary of components that was used to test the reference architecture. The architecture is generally applicable to all T-series servers and Storage 7000 Unified Storage Systems:

- Server: Sun SPARC Enterprise T5240 Server
- Storage: Sun Storage 7410 Unified Storage System
- Operating system: Oracle Solaris 10 Operating System
- Virtualization:
  - Oracle VM Server for SPARC v1.3
  - Oracle Solaris Containers
- Database: Oracle Database 11g Release 2
- Java EE Application Server: Oracle WebLogic Server Release 1
- Java: Sun JDK version 6.0
- Applications/Workloads:
  - MedRec sample Java EE application
  - SPECjEnterprise2010 benchmark

**Note: The number of Java EE applications that can be deployed onto a single Sun SPARC T-series server is determined by the resource requirements of the applications and the utilization (number of users of the application). This architecture demonstrates the principles of deploying multiple applications but the sizing of these applications is beyond the scope of this document.**

### Detailed Component Description

#### Server

The Sun SPARC Enterprise T5240 Server is a two socket, eight core per socket server using the UltraSPARC T2 Plus processor with chip multithreading technology. The server has eight hardware threads per core and between the two processors provides a total of 128 hardware threads available for virtualization. The server is equipped with 64 GB memory and can be configured with up to 256 GB

memory and a maximum of 16 internal disk drives. The T5240 features a built-in hypervisor (on chip) for virtualization and is SPARC binary compatible.

### **Storage**

The Sun Storage S7410 Unified Storage System is a high capacity storage device with high throughput features and a simple to use management interface that makes it ideal for storage requirements of consolidated Java EE applications. Major features include:

- Use of flash memory to speed data reads and writes
- Predictive Self-Healing and diagnosis of all system hardware failures
- ZetaByte File System (ZFS) end-to-end data checksums of all data and metadata, protecting data throughout the stack
- RAID-6 (Double Parity) and optional RAID-6 across disk shelves
- Active-active clustering for high availability (7310 and 7410)
- Link aggregations and IP multi-pathing for network failure protection
- I/O multi-pathing between the controller and disk shelves
- Integrated software restart of all storage system software services

### Operating System

The Oracle Solaris 10 Operating System is an enterprise-grade operating system, and is used for all Sun T-series and M-Series servers, as well as being available on x86-based servers from Oracle. Oracle Solaris Containers (see below) and Oracle Solaris ZFS are used in this architecture to securely and conveniently host and deploy Java EE applications. Some of the major features include:

- 64-bit operating environment
- High performance
- Support for large memory and high CPU count systems
- Excellent scalability for richly threaded environments
- Advanced filesystem
- Predictive, self-healing features
- Built-in virtualization
- Extensive instrumentation and diagnostic capabilities to assist performance and availability

### Virtualization

The virtualization technologies of Oracle VM Server for SPARC and Oracle Solaris Containers are included as part of the Oracle T-series servers and Solaris operating system respectively. They both allow the rapid deployment of virtual environments, resource controls and efficient use of shared resources, but also offer distinct differences allowing them to be used in a wide variety of situations.

**Oracle VM Server for SPARC** (formerly known as Sun Logical Domains) provides full hardware based hypervisor virtualization capabilities for SPARC, allowing up to 128 virtual machines to be created. OVM server for SPARC enables reliable consolidation and deployment of applications by providing secure isolation between running instances of the Solaris operating system.

**Oracle Solaris Containers** (also known as zones) are an included feature of the Oracle Solaris 10 operating system (on both SPARC & x86 platforms) and allows kernel-level separation of applications running in a single Oracle Solaris 10 instance. Oracle Solaris Containers are rapid to deploy, offer low overhead and are used in this reference architecture to separate instances of Oracle WebLogic Server within a single virtual machine which are both running the SPECjEnterprise2010 benchmark application.

### Database

TPC-C benchmarks provide a good illustration of the performance benefits of running Oracle database on Oracle hardware . For details see <http://www.oracle.com/us/corporate/press/042052>

The Oracle Database 11g Release 2 Database is a full featured enterprise class relational database which this reference architecture uses as the persistence tier of the Java EE applications.

### Java EE Application Server

The Oracle WebLogic Server 11g Release 1 is a fully compliant Java EE 5.0 application server which is feature rich and holds benchmark world records for Java EE performance. Oracle WebLogic Server 11g takes full advantage of the 64-bit addressable memory and also the large number of hardware threads available in Sun SPARC T-series servers such as the Sun SPARC T5240 used in this architecture.

### Java Standard Edition 6.0

Java Standard Edition 6 (Java SE 6) is the current shipping edition of the Java SE platform runtime and code development kit. Java SE 6 provides the platform on which the Oracle WebLogic Server 11g executes. For more information on Java SE 6 refer to the comprehensive resources available on the Oracle technical network at <http://www.oracle.com/technetwork/java/index.html>.

### Java EE Applications

This architecture uses two applications to demonstrate various deployment methodologies:

**Note: the name Java Enterprise Edition (Java EE) encompasses the current Java EE 5.0 version and the previous version which was formerly Java 2 Enterprise Edition (J2EE)**

**Avitek Medical Records (or MedRec)** is designed as an educational tool for Java EE developers; it showcases the use of Java EE components, and illustrates design patterns for component interaction and client development, and is included in the WebLogic peripheral downloads as an example architecture that highlights the features of the WebLogic Server. The application demonstrates a framework for patients, doctors, and administrators to manage patient data using a variety of different clients. The mock patient data includes:

- Patient profile information—A patient's name, address, social security number, and login information.
- Patient medical records—Details about a patient's visit with a physician such as the patient's vital signs and symptoms as well as the physician's diagnosis and prescriptions.

**SPECjEnterprise2010** is a benchmark workload which involves many aspects of a system and was created by SPEC as a standard Java EE benchmark. In this reference architecture, the SPECjEnterprise2010 application is not used as a benchmark but as a good example of a complex and resource intensive application typical of enterprise businesses that run and rely on Java EE and WebLogic for their core business functions.

## Architecture overview

This section describes the physical and virtual architecture of the system and the software deployment.

### Physical architecture

The physical architecture of the system is simple, the SPARC T-series server and the Sun Storage 7000 Unified Storage System are attached to the network switch using the four on-board 1 Gigabit ethernet interfaces that are a standard part of the server and storage systems. Other connection options are available such as fibre channel and these can be used instead of, or in conjunction with the ethernet connections. This architecture uses and assumes the simple network connection shown Figure 1.

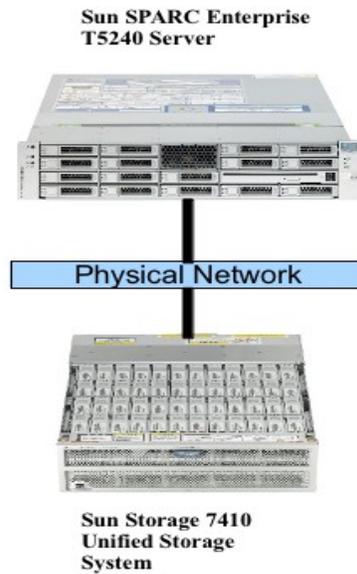


Figure 10. Connections.

## Virtual Environment

**Tip for larger and more complex deployments, the naming scheme becomes important so as to enable easy identification of virtual machine/container and it's associated function.**

Using Oracle VM Server for SPARC, the SPARC Enterprise T5240 is logically divided into four guest domains and one administration domain as shown below. The domain names highlight the purpose of the domain. The first part of the name refers to “logical domain” or “ld” for short. The second part of the domain name refers to application server (“as” for short) or database server (“db” for short). In the ld03-as02 domain, there 2 Oracle Solaris containers (a.k.a zones) which are used to support instances of the WebLogic Server. The reason for the use of both logical domains and Solaris zones will be discussed in a later section.

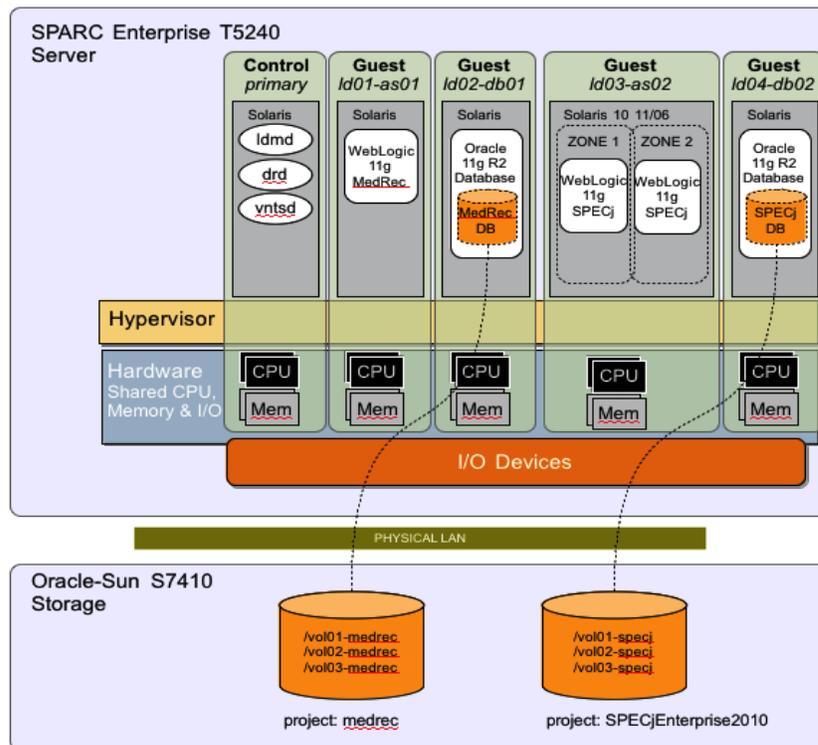


Figure 1: Virtual Architecture

**For a full introduction to Oracle VM server for SPARC see the “Beginners Guide to Ldoms”**

**Hardware threads are a feature of the Oracle Sun T-series servers and can be thought of as a virtual cpu (vcpu)**

**In this reference architecture the “primary” domain serves as the I/O domain and the Service Domain.**

The domains created by Oracle VM Server for SPARC have dedicated access to their CPU and memory (not shared) and have certain “roles”. These determine the hardware components available directly to them, and the functions and features each domain can deliver and subscribe to. The following is an overview of the various roles, which are NOT custom to the sample applications, but rather exist as the way OVM Server for SPARC separates out the server’s resources:

- **Control Domain** – This default domain is created when the operating system is installed onto the T-series hardware. The Control Domain communicates with the hypervisor to make changes to configuration, and is the domain where access to the logical domains command line (ldm) is accessed. There can be only one of these domains per-system, and is referred to as “primary”.
- **I/O Domain** - This domain has physical access to I/O devices such as network and disk. By default the control domain “owns” all physical I/O devices but then share these devices out to other domains in various ways depending upon the hardware platform. Multiple I/O domains can be created, although only one was used in the tested architecture.
- **Service Domain** – This domain provides virtualized services based upon physical devices such as virtual disk services and virtual network switches. These devices are essentially published to other domains which can connect to them with the service domain acting as a “proxy”.

**It is also possible to create services that do not have a hardware component, such as in the case of an internal virtual switch service to route between two domains without an external interface. This is called a “routed-mode” virtual switch.**

- **Guest Domain(s)** – These domains have no access to physical devices. In Figure 1, ld01-as01, ld02-db01, ld02-as02, ld04-db02 are guest domains. They subscribe to virtual services such as vdisk server and virtual switches with virtual devices such as vdisk and vnet.

## Resource Allocations

**Note:** A feature of this architecture is that resource assignment is extremely flexible, vcpu, memory and virtual network adapters can be easily re-assigned to the busiest guest domains and vcpus can be allocated dynamically, perhaps switched during times of peak load for an application.

The diagram below shows the initial resource allocation of the SPARC server. Note the primary domain requires adequate cpu and memory resources as the act of enforcing the various services and I/O falls to the Control Domain.

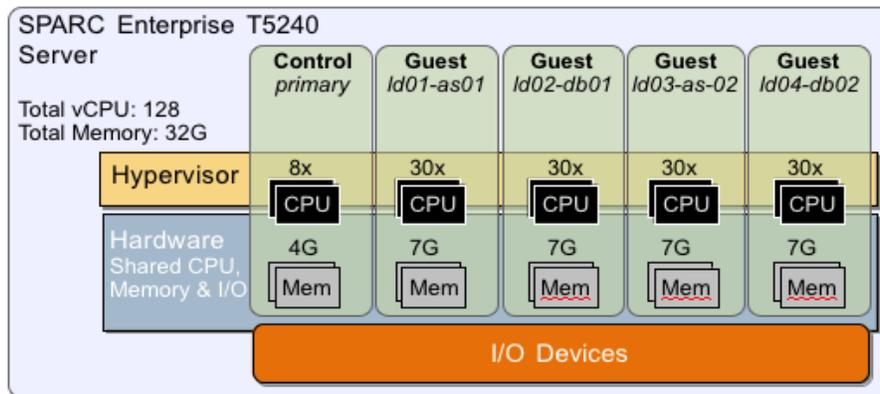


Figure 2: resource allocations

**Note:** Resources such as memory and VCPU are not shared between domains (often referred to as over-subscription in other virtual environments). In Oracle VM for SPARC resources are assigned to domains exclusively, and this assignment persists across reboots.

From the primary domain and logged in as the root user the Virtual Machines (LDoms) and resource allocations can be listed.

**Note:** Oracle VM server for SPARC was formerly known as Sun Logical Domain Manager and some terminology is still embedded in the technology, thus an LDom or domain is a virtual machine.

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv-	SP	8	4G	5.8%	22d 10h 31m
ld01-db01	active	-n----	5000	30	7G	0.2%	14d 3h 21m
ld02-as01	active	-n----	5001	30	7G	0.0%	14d 7h 42m
ld03-db02	active	-n----	5002	30	7G	0.1%	11d 22h 31m
ld04-as02	active	-n----	5003	30	7G	0.0%	22d 2h 10m

### Networking Architecture

The architecture uses an Oracle VM feature that allows for “routed mode” virtual switches (vsw) to be created for all network traffic between application server and database. These virtual switches require no physical network adapter nor cabling and can be created dynamically, therefore routing their network traffic completely inside of the server itself, eliminating the need for external networking ports. See the “Setup Guide” section for details.

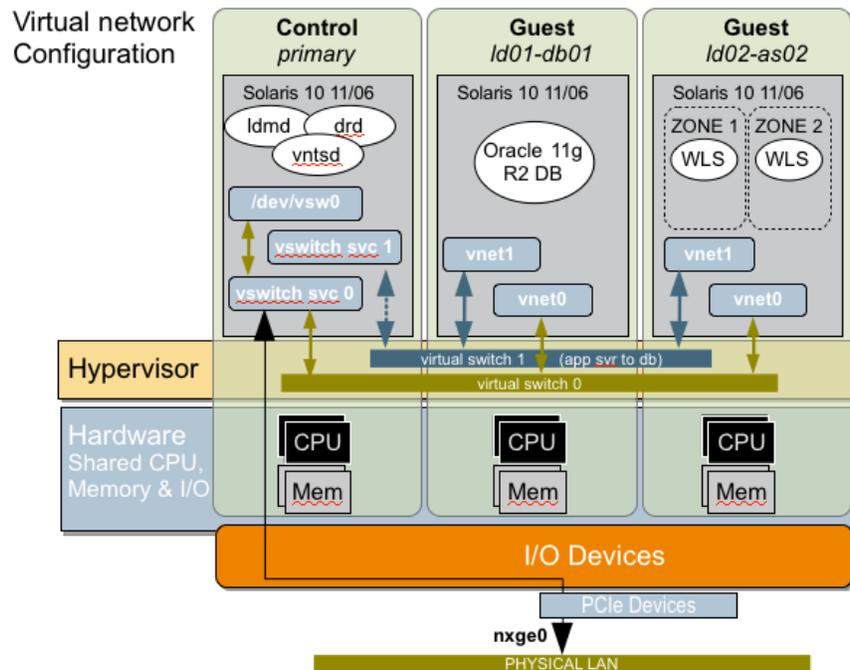


Figure 3: virtual network configuration

For external network access, virtual switches are created using a physical device e.g. the gigabit ethernet nxge0 interface and then the virtual networks are created using the virtual switch and plumbed in the normal fashion. Because no Solaris domain has direct access to the hardware, this virtualization layer needs to be created, even if the specified Solaris domain has sole access to the networking hardware.

#### Networks established

- Primary domain - uses the nxge adapter to allow users to connect and administer the server.
- Application Server Virtual Machines – one routed mode (internal network connection) vsw to the database and one nxge hosted vsw to the user network to allow for incoming requests.
- Database Server Virtual Machines – one routed mode vsw to the application server, one nxge hosted vsw to the storage and one nxge hosted vsw for administrator access.

**Note that in this architecture using routed mode switches means that the network ports for the database need not even be available on the company intranet, this enhances the security of the database and corporate data. Previously this level of database isolation would require expensive switches and complicated VLAN configurations.**

## Virtual Machine Images

**Note: See Appendix B for an example of how to create the virtual machine images on the Sun Storage S7410.**

To establish the architecture a template virtual machine or golden LDom image with a vanilla Solaris installation is established on a ZFS zvol on the Sun Storage S7410. The advantage of this approach is that the ZFS “snapshot” and “clone” features can be utilized to instantly create the Solaris installations on the four guest virtual machines. The need to install the operating system more than once is eliminated. This approach can also be used to update the operating system and software installation of the virtual machines in the future. Whilst the internal disks of the server could be used to host the root filesystems the use of the Sun Storage S7410 to host the virtual operating system installs offers two significant advantages. Firstly the Sun S7410 provides a large amount of storage, ensuring that the “/” filesystem is unlikely to “fill up” in any of the guest virtual machines and secondly the images are highly available due to the use of the reliability features of the Sun Storage S7410 i.e. zfs raid-z is used for these root volumes.

## Software Architecture

Each application (MedRec and SPECjEnterprise2010) is installed into separate sets of WebLogic servers and Oracle database server instances.

The Oracle database and the Oracle WebLogic server that host each of the user applications are installed into separate virtual machines. This keeps the applications, the data and the user base for each server and for each function isolated from each other and from other applications running on the same physical hardware.

In many Java EE applications there will be a need to run more than one instance of the WebLogic application server. In this architecture it is assumed that SPECjEnterprise2010 application will require more than one instance of the WebLogic server. Given the low resource overhead and the ease of use characteristics of Oracle Solaris Containers (zones) it is both efficient and convenient to separate these multiple application servers instance by installing them into different zones. This architecture takes advantage of the clone feature of Solaris 10 and zone 2 is created by “cloning” zone 1 which has an installed WebLogic instance and deployed SPECjEnterprise application.

Networking and communication between application server virtual machines and database virtual machines is provided via OVM for SPARC virtual switches. Use of these virtual switches mean that the application server to database traffic is kept on the backplane of the Sun SPARC T5240 server and never goes over a network cable. This is both efficient and secure.

## Benefits

The architecture presented offers a number of significant advantages to users wishing to deploy Java EE applications, especially when compared to deploying each Java EE application and database on a separate server. These include:

- Flexible resource allocation for Java EE applications
- Security
- Scalability
- Reliability and availability of Java EE application data
- Power and space savings
- Benefits from a single vendor solution
- Certification, testing and inclusion (i.e. cost savings)

## Flexible Resource Allocation

**Note: With Oracle VM server for SPARC V1.3, a restart of the server is required for changing memory; however it is anticipated that in future releases memory will be dynamically allocated in a similar fashion to cpu resources.**

The Oracle VM Server for SPARC allows reassignment of compute resources (vcpu's) dynamically e.g. to move a virtual cpu (VCPU) from the first database instance to the second use the ldm command

```
#ldm remove-vcpu 1 ld01-db01
#ldm add-vcpu 1 ld03-db02
```

In a similar fashion memory can be moved between virtual machines, new networks can be created or destroyed and or storage links established. The key benefit is that because the virtual machine boundaries are flexible, deployers and system administrators can re-allocate resources at any time or for any reason. For example, if there is a cpu-intensive batch application that needs to run every night, the vcpus can be re-assigned to the domain in which that application is deployed.

**Note: Oracle VM Server for SPARC also allows CPU resources to be reassigned based on a policy, for instance extra cpu resources could be allocated to the database virtual machines in the evenings for “reporting or batch work” or other database administration tasks.**

## Security

This reference architecture is a secure platform for deployment of Java EE applications. This is due to both the underlying security of the Oracle Solaris 10 operating system and the implementation of the Oracle VM Server for SPARC in conjunction with Oracle Solaris 10.

### **Solaris 10 security**

Security is a fundamental part of the design of the Oracle Solaris 10 Operating System, as such the installation defaults are such that attack surface is minimized, in addition to offering robust security features in the operating system itself. Important security features that are part of the architecture or can be applied to the architecture include:

- Auditing and File Integrity – allows identification of access and changes in the file information on the system.
- User and Rights Management – allows the minimum privileges to be assigned to users and systems administrators hence minimizes the use of the super user privilege.
- Network Service Protection – reduces attack footprint of Solaris systems by setting sensible and restricted network services defaults.
- The Solaris Security Toolkit is available and provides security for devices critical to the management of your server, including the control domain in the Logical Domains Manager. Other available functions are:- Hardening, Minimizing, Authorization, Auditing and compliance.

### **Oracle VM Server for SPARC Security**

The Oracle VM server for SPARC hypervisor implements secure separation for each guest virtual machine and this means that:

- Each guest virtual machine is unaware of other guest virtual machines.
- Guest virtual machines can't access the cpus or memory from other virtual machines and there is no programming interface to allow this.
- Guest virtual machines can't see nor access traffic on virtual switches of other guest machines, unless configured to do so from the control domain.
- Guest virtual machines can't see virtual disks outside of those which they have been specifically assigned.
- Guest virtual machines do not have an interface to access the control domain.

### **Other Security Advantages**

Access to the console of the guest virtual machines is only via the control domain and the control domain itself is secured by Solaris 10, this limits again limits the attack surface of the guest virtual machines.

## Availability/Reliability

This architecture provides a high level of Java EE data and application availability due to the overall quality of the hardware and software components. Some features that add to this level of availability are listed below:

**Note:** See the section titled “variations” for the many other features of the Oracle products that could be used to further increase application and data availability.

- The database is stored on the Sun Storage S7410 Unified Storage System which uses ZFS and enhanced error detection and self-healing. This minimizes errors and database downtime as a result of disk failure.
- All of the database volumes configured in the Sun Storage S7410 for the database are utilizing ZFS raid-z which is capable of dealing with 1 or 2 simultaneous device failures. The WebLogic Server transaction logs are similarly provisioned and robust.
- The virtual machine boot-devices are provisioned using ZFS and raid-z so the operating system image is protected from a single disk failure.

## Tuning Resources Available

The architecture features two Java EE applications running on Oracle WebLogic Server and the performance and throughput of the Oracle WebLogic Server running on Sun SPARC T-series hardware is demonstrated by benchmark results for the Industry Standard J2EE and Java EE 5.0 SPECjAppServer2004 and SPECjEnterprise2010 benchmarks respectively. Refer to <http://www.spec.org/jAppServer2004> or <http://www.spec.org/jEnterprise2010> for more details and results and importantly resources. For example the Oracle T-Series result at <http://www.spec.org/jAppServer2004/results/res2010q2/jAppServer2004-20100518-00142.html> contains a wealth of tuning and configuration information e.g. the Java tunings listed below are in the “full disclosure report” of the benchmark. These tunings along with the the Solaris, network, storage, WebLogic 11g and other system tunings form a valuable resource that can be drawn from for Java EE applications running in WebLogic 11g on Sun SPARC T-series servers.

```
JVM Options:
-server -Xms3300m -Xmx3300m -Xmn1024m -Xss128k -XX:+AggressiveOpts -XX:+UseParallelGC
-XX:ParallelGCThreads=32 -XX:PermSize=128m
-XX:LargePageSizeInBytes=4m -XX:+UseParallelOldGC
-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCTimeStamp
-XX:-UseAdaptiveSizePolicy -XX:+PrintAdaptiveSizePolicy
-XX:MaxTenuringThreshold=15 -XX:InitialSurvivorRatio=10
-XX:SurvivorRatio=10 -XX:TargetSurvivorRatio=90 -XX:ReservedCodeCacheSize=64m
-XX:CICompilerCount=4
-Dweblogic.management.discover=false -Dweblogic.SocketReaders=4
-Dweblogic.StuckThreadMaxTime=900
-Dweblogic.diagnostics.debug.DebugLogger.DISABLED=true
-Doracle.jdbc.defaultRowPrefetch=300
...
```

## Power and Space Savings

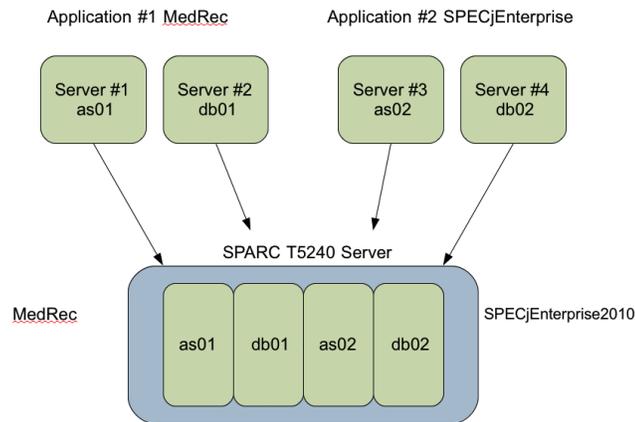


Figure 4: consolidate multiple servers onto one

**Note:** Note if more than two Java EE applications and servers are consolidated onto on Sun SPARC T-series server then there is a greater saving in floor space, power and cooling.

This architecture demonstrates how multiple physical servers can be replaced by one physical server. The advantages of this consolidated architecture include:

- Improved resource utilization. It is very common for independent servers run at very low utilization e.g. 20% or less of capacity. Consolidating the servers as indicated will typically lead to much better resource utilization and much less idle hardware.
- Floor or rack space savings. In this example, assuming each server consolidated is just 1 RU and the Oracle Sun SPARC T5240 server is 2RU then the saving is 2RU of space.
- Savings on power and cooling. Consider that to provide a level of redundancy each separate server will typically be configured with a minimum of 2 internal drives, 2 fans and 2 power supplies i.e. 8 total fans, 8 total internal disks and 8 total power supplies. Moving to the consolidated architecture means that this same level of redundancy can be achieved with just 2 internal disks, 2 fans and 2 power supplies.

### **Certification and Inclusion**

All of the components of this architecture for deploying Java EE applications are from one vendor and all of the virtualization facilities of Oracle VM Server for SPARC or Oracle Solaris containers are included with the hardware and operating system. No 3<sup>rd</sup> party virtualization software or tools are needed and no 3<sup>rd</sup> party support is required.

Oracle WebLogic Server 11g Release 1 and Oracle database 11g R2 are fully certified for Oracle Solaris 10 and fully supported by Oracle.

All of these components are being assembled and tested together by Oracle as part of Oracle's integrated stack testing approach.

### **Summary**

The Oracle Sun SPARC Enterprise T-series servers are available in a wide range of configurations and the Sun Storage 7000 Series Unified Storage systems offer terabytes of easy to manage and reliable disk storage. Java EE application servers such as Oracle WebLogic 11g and the Java applications they support are highly multi-threaded which means they are well suited to make good use of the threads and memory available on these servers. The in-built virtualization features and associated flexibility of the platform offer an attractive consolidation platform for deploying Java EE applications using Oracle WebLogic 11g and Oracle Database. Enhanced security and resource separation features included in both the hardware and the software offer ways through which data center managers, system and database administrators can consolidate multiple applications on to one platform, while maintaining manageability and performance targets.

## Variations

The reference architecture described above demonstrates many benefits and the principles of consolidating Java EE / WebLogic server applications onto the Oracle Solaris 10 Operating System and Oracle SPARC T-series servers. This section highlights some possible variations of this reference architecture, the variations use the same basic building blocks but will highlight further possibilities.

### Separate T-series Database Server

The database virtual machines are created on a separate physical Sun SPARC Enterprise T-series server and the databases are installed onto these virtual machines. This demonstrates physical separation of the architectural tiers while using the same principles of virtualization of the reference architecture. This variation is likely appeal to Java EE deployers and system administrators who are uneasy about having the database and the application server on the same physical tier.

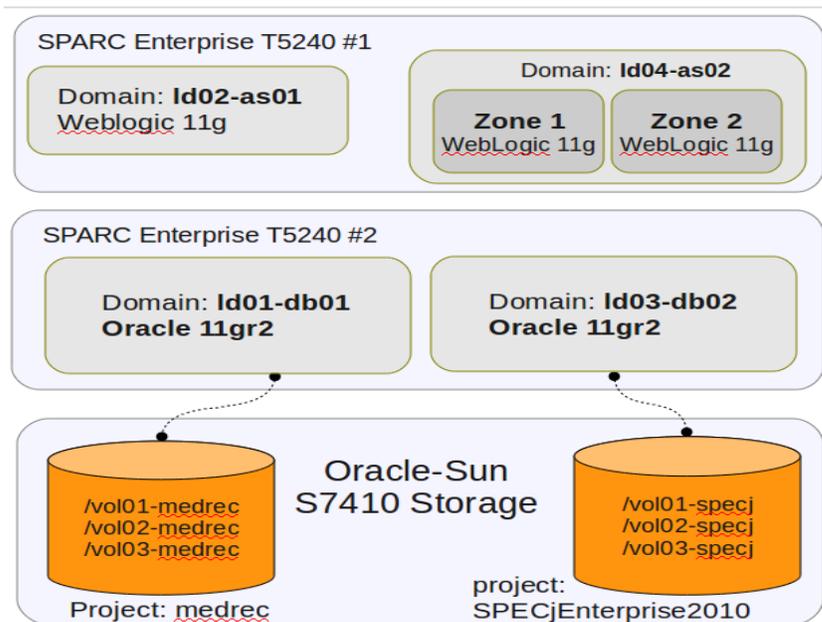


Figure 5: adding a separate server for the Oracle Database

## Oracle Real Application Cluster (RAC)

A further variation is to (again) add a second Sun SPARC T-series server but to also add RAC clustering for the database. On each node a virtual machine running Oracle RAC is configured as well as one or more virtual machines (and possibly containers) running WebLogic 11g. The diagram below describes the configuration and how the virtual machines could be established across these 2 machines which now include RAC clustering. Advantages of this variation are :

- It is highly resilient and, can be configured so that even the failure of a complete Sun SPARC T5240 server will not cause loss of service for the Java EE application.
- It is one of the standard configurations that is being tested by Oracle engineers as part of the Oracle's integrated testing philosophy.

- Provides further scaling benefits.

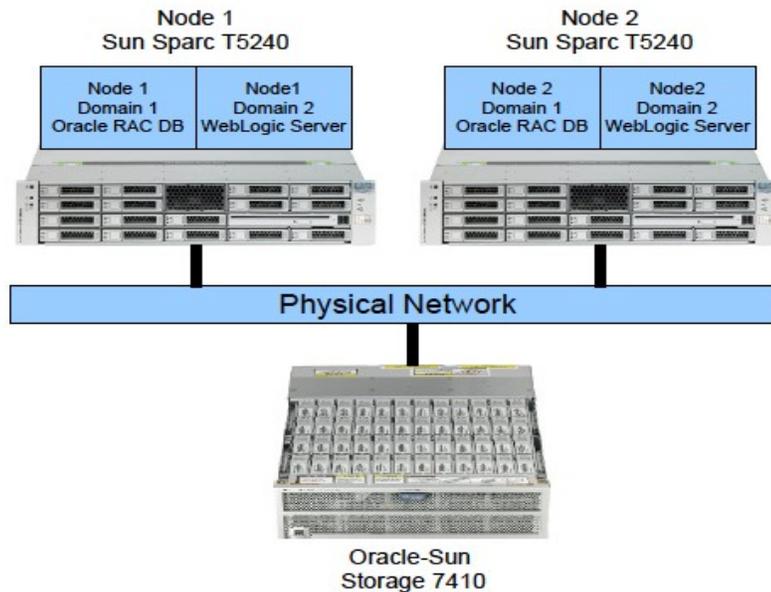


Figure 6: adding Oracle Real Application Clusters

## WebLogic Clustering

Application availability could be further enhanced by utilizing WebLogic Cluster. A WebLogic cluster is a collection of WebLogic Server instances that work together to provide a reliable, scalable environment for Java EE applications. WebLogic Server clusters increase reliability by supporting failover; WebLogic automatically switches requests and processing to a redundant server upon the failure or abnormal termination of the currently-active server. A WebLogic cluster always contains one Administration Server and this could run in a dedicated zone or virtual machine or potentially run on a separate management machine providing services for zone 1, zone 2 as well as as01-as02 i.e. providing services for all of the configured WebLogic instances no matter if they were deployed on separate physical servers like the 2 node configuration above or the single server featured in the detailed reference architecture.

## Other Reliability Variations

There are a number of other features of Oracle Solaris 10 and of the Oracle SPARC T-series virtualization environment, that are available to further increase the availability of the deployed Java EE applications. For example:

- On the Sun Storage S7410, unlimited read-only and read-write snapshots, with snapshot schedules could be configured to constantly and quickly back up data.

- The addition of redundant network links from the Sun SPARC T5240 server and the use of Solaris link aggregations to manage these redundant links would increase network availability.

## Setup Guide

### System installation and setup

The following section provides instructions on how to setup the virtualization environment and deploy the Java EE applications into the virtualization environment.

#### Establishing the virtual environment

The following is a summary of the major steps required to establish the virtual environment. It is assumed that the systems already have the base Oracle Solaris operating environment installed:

1. Install the latest Oracle VM for SPARC release
2. Allocate system resources for the primary (control) domain
3. Create a template Oracle VM for SPARC virtual machine
4. Clone the template virtual machine to create guests
5. Allocate system resources to the virtual machines
6. Create the Oracle Solaris Containers in the appropriate virtual machine.
7. Configure virtual network connections
8. Configure the filesystems on the Oracle Enterprise 7410 storage.

Detailed instructions for each of these steps follow. After these steps are completed the physical systems are established along with the virtual machines and containers, and are ready for the installation of the WebLogic server and of the Oracle database and deployment of the MedRec and SPECjEnterprise2010 applications.

#### Install latest Oracle VM Server for SPARC

**Note:** This step will only be required for Sun T-series servers which do not already have the latest LDOM manager installed.

LDom Manager 1.3 needs to be installed in Oracle Solaris 10u8 operating system to setup the virtualization environment. See: <http://www.sun.com/servers/coolthreads/ldoms/get.jsp>

Download LDoms\_Manager-1\_3.zip file. unzip it and run the install-ldm script from the unzip folder

```
# cd $DIR/LDoms_Manager-1_3
```

```
# Install/install-ldm
```

Verify the installation with executing following command.

```
# /opt/SUNWldm/bin/ldm list
```

Create the default services:

Create the virtual disk server(vds)

```
# ldm add-vds primary-vds0 primary
```

Create the virtual console concentrator server(vcc)

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

Create the virtual switch server(vsw)

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

### Allocate systems resources to the primary (or control) domain

#### Creating the control domain with 8 vcpu's and 4 GB RAM.

Note - If you have any cryptographic devices in the control domain, you cannot dynamically reconfigure CPUs. So if you are not using cryptographic devices, set-mau to 0. A possible use for a cryptographic unit is to speed up domain migrations. Each core in the T5240 server has a dedicated cryptographic processor incorporated in the hardware of the processor itself. The crypto cores can be used to dramatically increase performance of calculating SSL transactions, but have the limitation that any cores that share the crypto processor cannot be dynamically moved out of the domain without a reboot. It is up to the system architect to decide whether the enhanced crypto performance is worth the tradeoff in less flexibility of resource movement.

Assign one cryptographic resource to the control domain, primary. This leaves the remainder of the cryptographic resources available to a guest domain.

```
# ldm set-mau 1 primary
```

Assign 8 virtual CPUs and 4GB memory to the control domain, primary. This leaves the remainder of the virtual CPUs available to a guest domain.

```
# ldm set-vcpu 8 primary  
# ldm set-memory 4G primary
```

Make the modified configuration permanent using list-spconfig option

```
# ldm list-spconfig
factory-default [current]
```

Add a logical domain machine configuration to the system controller (SC).

```
# ldm add-spconfig initial
# ldm list-spconfig
factory-default [current]
initial [next]
```

Reboot the server to come up with the initial configuration.

```
# shutdown -i6 -g0 -y
```

#### Configure the virtual switch on the primary domain.

By default, networking between the control/service domain and other domains in the system is disabled. To enable this, the virtual switch device (vsw0) should be configured as a network device. The virtual switch can either replace the underlying physical device (nxge0 in this example) as the primary interface or be configured as an additional network interface in the domain.

Plumb the virtual switch (vsw0) on the primary domain

```
# ifconfig vsw0 plumb
```

Bring down the primary interface. Note as this step brings down the interface the user should be logged into the console of the server.

```
# ifconfig nxge0 down unplumb
```

Configure virtual switch

```
# ifconfig vsw0 10.6.140.204 netmask 255.255.255.0 broadcast + up
```

Modify the hostname file to make this configuration permanent

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

Enable virtual network terminal server daemon

```
# svcadm enable vntsd
# svcs vntsd
STATE          STIME          FMRI
online         Jun_17         svc:/ldoms/vntsd:default
```

### Create a template Oracle VM for SPARC virtual machine

Create a ZFS file system that will be used to create virtual disks for VMs. We can use either a local disk or iSCSI target created on the storage for creating ZFS over it.

Create a ZFS pool on the localdisk.

```
# zpool create ldompool c1t1d0
# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
ldompool      72K   134G   21K    /ldompool
# zpool list
NAME          SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
ldompool     136G  76.5K  136G     0%  ONLINE  -
```

For detailed instructions on creating a ZFS pool on the iSCSI target created on the storage, see Appendix-B.

Create a 25GB ZFS volume for the first disk.

```
# zfs create -V 25g ldompool/disk1
# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
ldompool      25.0G  109G   21K    /ldompool
ldompool/disk1  25G   134G   16K    -
```

Specify the device to be exported by the virtual disk server as a virtual disk to the guest domain.

```
# ldm add-vdiskserverdevice /dev/zvol/dsk/ldompool/disk1 \ vol1@primary-vds0
```

Create a guest domain ld01-db01 with 30 VCPUs and 7GB Memory.

```
# ldm add-domain ld01-db01
# ldm set-vcpu 30 ld01-db01
# ldm set-memory 7G ld01-db01
```

Add a virtual network device vnet1 to the guest domain ld01-db01

```
# ldm add-vnet vnet1 primary-vsw0 ld01-db01
# ldm add-vdisk vdisk1 vol1@primary-vds0 ld01-db01
```

Add the ISO image as a virtual device that will be used as the installation media. This will allow the domain to boot the Oracle Solaris operating system and install:

```
# ldm add-vdiskserverdevice /installvdvd/sol-10-u8-ga-sparc-dvd.iso iso@primary-vds0
# ldm add-vdisk iso iso@primary-vds0 ld01-db01
```

Set auto-boot and boot-device variables for the guest domain.

```
# ldm set-variable auto-boot\?=false ld01-db01
# ldm bind-domain ld01-db01
# ldm start-domain ld01-db01
```

Connect to the guest domain console from control domain and start the installation (the guest domain console can be determined using the “ldm list” command from the primary domain)

```
# telnet localhost 5000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Connecting to console "ldom1" in group "ldom1" ....
Press ~? for control options ..
{0} ok show-disks
a) /virtual-devices@100/channel-devices@200/disk@1
b) /virtual-devices@100/channel-devices@200/disk@0
q) NO SELECTION
Enter Selection, q to quit: a
/virtual-devices@100/channel-devices@200/disk@1 has been selected.
Type ^Y ( Control-Y ) to insert it in the command line. e.g. ok nvalias mydev ^Y
for creating devalias mydev for
/virtual-devices@100/channel-devices@200/disk@1

{0} ok devalias
iso /virtual-devices@100/channel-devices@200/disk@1
vdisk1 /virtual-devices@100/channel-devices@200/disk@0
vnet0 /virtual-devices@100/channel-devices@200/network@0
net /virtual-devices@100/channel-devices@200/network@0 disk /virtual-
devices@100/channel-devices@200/disk@0 virtual-console /virtual-devices/console@1
name aliases
```

Now boot from the virtual iso image appending the :f (this is to specify the slice 6 of the DVD/ISO image). This can also be done using boot iso:f

```
{0} ok boot /virtual-devices@100/channel-devices@200/disk@1:f
```

After the Oracle Solaris installation is done on the guest domain ld01-db01 execute sys-unconfig. This will reset the operating system back to unconfigured and have no name, network settings, naming services etc. It will also halt the guest domain ld01-db01 to allow a snapshot of the base ldom's disk to be taken

```
# sys-unconfig
# ldm stop-domain ld01-db01
```

Remove the ISO disk from the guest domain, and execute the snap shot of the disk image

```
# ldm remove-vdisk iso ld01-db01
# zfs snapshot ldompool/disk1@base
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
ldompool             30.2G  104G   21K    /ldompool
```

```
ldompool/disk1      30.2G   129G   5.19G  -
ldompool/disk1@base  0        -   5.19G  -
```

Clone the gold ldom disk image for the guest domains (ld02-as01, ld03-db02 and ld04-as02)

```
# zfs clone ldompool/disk1@base ldompool/disk2
# zfs clone ldompool/disk1@base ldompool/disk3
# zfs clone ldompool/disk1@base ldompool/disk4
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
ldompool             30.2G  104G   21K    /ldompool
ldompool/disk1       30.2G  129G   5.19G  -
ldompool/disk1@base  0        -   5.19G  -
ldompool/disk2        0     104G   5.19G  -
ldompool/disk3        0     104G   5.19G  -
ldompool/disk4        0     104G   5.19G  -
```

Allocate system resources to the virtual machines. There are a total of 128 VCPUs and 32G on the system, equal amount of VCPUs and Memory are allocated to the guest domains leaving some resources for primary domain. Note that these allocations can be easily adjusted later when the application and system demand have been more accurately sized.

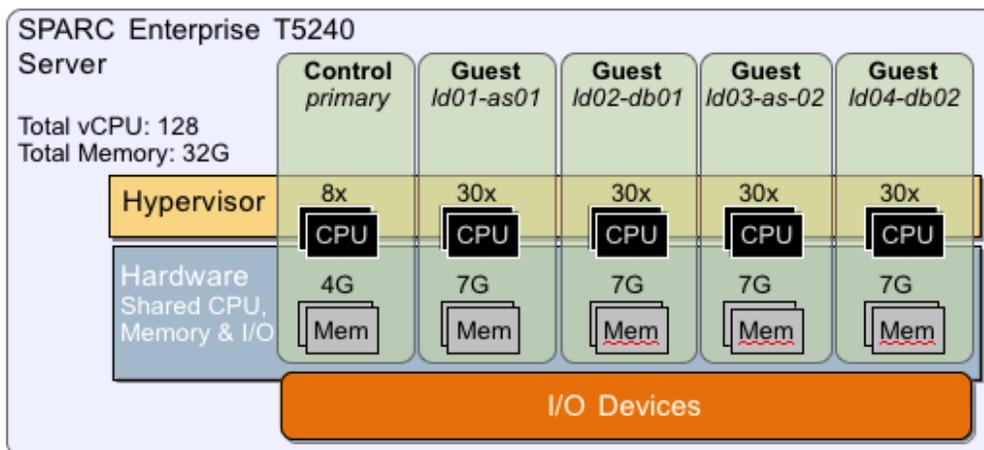


Figure 7: System resource allocation

Create a new guest domain - ld02-as01

```
# ldm add-domain ld02-as01
# ldm set-vcpu 30 ld02-as01
# ldm set-memory 7G ld02-as01
# ldm add-vnet vnet1 primary-vsw0 ld02-as01
# ldm add-vdiskserverdevice /dev/zvol/dsk/ldompool/disk2 \ vol2@primary-vds0
# ldm add-vdisk vdisk1 vol2@primary-vds0 ld02-as01
```

```
# ldm set-variable auto-boot\?=false ld02-as01
# ldm bind-domain ld02-as01
# ldm start-domain ld02-as01
```

Similarly add the ld03-db02 and the ld04-as02 domains with the same resource allocations for CPU and memory.

### Configure virtual network connections

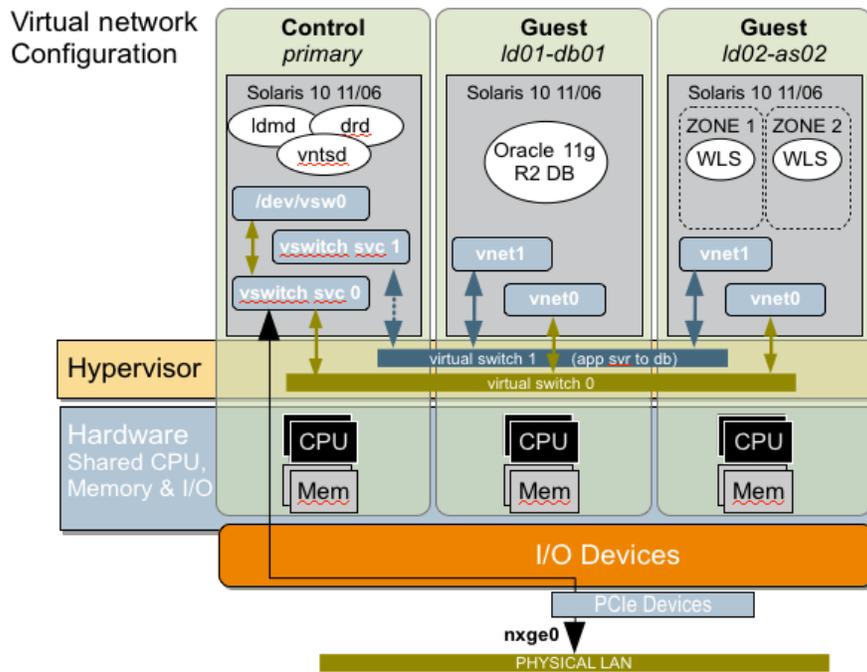


Figure 8: virtual network configuration

Configure a private virtual switch between the guest domains (ld01db01 and ld02-as01) as described by the diagram above by executing the following commands on the primary domain.

```
# ldm add-vsw primary-vsw1 primary
# ldm add-vnet vnet2 primary-vsw1 ld01-db01
# ldm add-vnet vnet2 primary-vsw1 ld02-as01
```

Plumb and configure the secondary interface in the guest domain (ld01-db01)

```
# ifconfig vnet1 plumb
# ifconfig vnet1 192.168.1.100 netmask 255.255.255.0 broadcast + up

# echo "192.168.1.100" > /etc/hostname.vnet1
# echo "192.168.1.100 ld01db01" >> /etc/hosts
# echo "192.168.1.200 ld02as01" >> /etc/hosts
```

Plumb and configure secondary interface in the guest domain (ld02-as01)

```
# ifconfig vnet1 plumb
# ifconfig vnet1 192.168.1.200 netmask 255.255.255.0 broadcast + up
# echo "192.168.1.200" > /etc/hostname.vnet1
# echo "192.168.1.100 ld01db01" >> /etc/hosts
# echo "192.168.1.200 ld02as01" >> /etc/hosts
```

Configure a private virtual switch between guest domains (ld03-db02 and ld04-as02)

```
# ldm add-vsw primary-vsw2 primary
# ldm add-vnet vnet2 primary-vsw2 ld03-db02
# ldm add-vnet vnet2 primary-vsw2 ld04-as02
```

Plumb and configure secondary interface in the guest domain (ld03-db02)

```
# ifconfig vnet1 plumb
# ifconfig vnet1 192.168.2.100 netmask 255.255.255.0 broadcast + up
# echo "192.168.2.100" > /etc/hostname.vnet1
# echo "192.168.2.100 ld03db02" >> /etc/hosts
# echo "192.168.2.200 ld04as02" >> /etc/hosts
```

Plumb and configure a secondary interface in the guest domain (ld04-as02)

```
# ifconfig vnet1 plumb
# ifconfig vnet1 192.168.2.200 netmask 255.255.255.0 broadcast + up
# echo "192.168.2.200" > /etc/hostname.vnet1
# echo "192.168.2.100 ld03db02" >> /etc/hosts
# echo "192.168.2.200 ld04as02" >> /etc/hosts
```

### Create the Oracle Solaris Container(s)

To create a Solaris Container within a guest domain, an additional disk is attached to the guest domain (ld04-as02).

On the primary domain attach additional disk to the guest domain (ld04-as02).

```
# zfs create -V 20G ldompool/additional-disk1-ld04-as02
# ldm add-vdiskserverdevice /dev/zvol/dsk/ldompool/additional-disk1-ld04-as02
vol15@primary-vds0
# ldm add-vdisk vdisk2 vol15@primary-vds0 ld04-as02
```

Create a ZFS pool for containers on the attached disk in the guest domain (ld04-as02).

```
# zpool create zonepool c0d1
# zfs list
NAME          USED  AVAIL  REFER  MOUNTPOINT
zonepool      72K   19.6G   21K    /zonepool
```

Create ZFS file system for the container.

```
# zfs create zonepool/zone1
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
zonepool            102K  19.6G   21K    /zonepool
zonepool/zone1     21K   19.6G   21K    /zonepool/zone1
```

Create a container with the following parameters and install it.

```
zonename: ld02-z01-app01
zonepath:/zonepool/zone1
autoboot: true
networking:
vnet0, address=10.6.140.137
vnet1, address=192.168.2.201
```

For the instructions on creating the zone see the appendix A - Creating and Installing a Zone. Additional zones can be created by using the “zoneadm” command and the “clone” option. In this architecture zone 2 is created by cloning zone 1 after the installation of WebLogic server and the deployment of the SPECjEnterprise2010 application.

Boot the solaris container and configure it.

```
# zoneadm -z ld02-z01-app01 boot
# zlogin -C ld02-z01-app01
```

### Configure filesystems on Oracle Enterprise S7410 storage

For the MedRec application the following LUNs are created on the Oracle Enterprise S7410 storage.

- ora-home-medrec
- vol01-medrec
- vol02-medrec
- vol03-medrec

The screenshot shows the Sun Storage S7410 management interface for the 'oracle-nfs' storage. The 'Filesystems' tab is selected, showing a table of LUNs. The table has three columns: NAME, SIZE, and MOUNTPOINT. There are four rows of data representing the MedRec volumes.

NAME ^	SIZE	MOUNTPOINT
ora-home-medrec	41.9G	/export/ora-home-medrec
vol01-medrec	13.6G	/export/vol01-medrec
vol02-medrec	21K	/export/vol02-medrec
vol03-medrec	991M	/export/vol03-medrec

Figure 9: MedRec volumes on the Sun Storage S7410

For the SpecJEnterprise2010 application following LUNs created on the Oracle Enterprise 7410 storage.

- ora-home-specj
- vol01-specj
- vol02-specj
- vol03-specj

The screenshot shows the Sun Storage S7410 management interface for the 'specj-nfs' storage. The 'Filesystems' tab is selected, showing a table of LUNs. The table has three columns: NAME, SIZE, and MOUNTPOINT. There are four rows of data representing the SPECJEnterprise2010 volumes.

NAME ^	SIZE	MOUNTPOINT
ora-home-specj	8.99G	/export/ora-home-specj
vol01-specj	1.32G	/export/vol01-specj
vol02-specj	160M	/export/vol02-specj
vol03-specj	22K	/export/vol03-specj

Figure 10: SPECJEnterprise2010 volumes on Sun Storage S7410

Mount these volumes on “ld01-db01” virtual machine (note that sql03 is the Sun Storage S7410)

**Note: The mount options here are tested and functional but not performance optimized. See the references “storage and servers” section for NFS and Oracle Database on Sun Storage tuning resources**

```
# mkdir -p /db/vol01
# mkdir -p /db/vol02
# mkdir -p /db/vol03
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/vol01-medrec /db/vol01
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/vol02-medrec /db/vol02
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/vol03-medrec /db/vol03
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/ora-home-medrec /export
```

Mount these volumes on “ld02-db02” virtual machine.

**Note: These mounts and mount options can be added into the Solaris /etc/vfstab of the guest virtual machine so that they are mounted automatically.**

```
# mkdir -p /db/vol01
# mkdir -p /db/vol02
# mkdir -p /db/vol03
# mkdir -p /export/home/oracle
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/ora-home-specj /export/home/oracle
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/vol01-specj /db/vol01
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/vol02-specj /db/vol02
# mount -F nfs -o \ hard,rw,noac,rsize=1048576,wsiz=1048576,suid,forcedirectio \
sql03:/export/vol03-specj /db/vol03
```

## Software installation

Once the SPARC server has been partitioned and the virtualization environment and operating systems established, the WebLogic server and Oracle database were installed and the MedRec and SPECjEnterprise applications deployed.

In this reference architecture the use of command line tools is demonstrated but tools such as the Oracle Enterprise Manager and the Sun Ops Centre facilities do provide alternatives and may be favored especially if there are a significant number of applications being consolidated or a large number of servers in use for consolidation.

### WebLogic Server Installation

The MedRec application is a reasonably light-weight Java EE application and so it is anticipated (for demonstration purposes) that a single instance of WebLogic server will be all that is required to service a potential user load. For this reason WebLogic is installed directly into the ld01-as01 virtual machine without the use of Oracle Solaris Containers. The Oracle Weblogic Application Server can be downloaded from <http://downloads.oracle.com>. The following outlines the installation of Oracle WebLogic Server into a guest virtual machine using the command line.

**Note: The home directory of the WebLogic Server Installation is referred to as \$WL\_HOME**

The general steps are :-

1. Go to the directory where WebLogic 10 binaries are copied
2. Run the installation script

```
bash-2.03# ./server1032_solaris32.bin -mode=console
```
3. On the welcome screen, just press Enter to continue
4. Enter an option for receiving security updates or select option **3** to set bypassing security updates.
5. Select **1** to create a new home directory or **next** to accept the default.
6. Select **2** for Custom installation when prompted for install type
7. Select **Server Examples** to ensure the MedRec application is installed (see below)
8. For the product directory, select the default directory and select **1** to continue
9. The “Install complete” message at the end, marks the end of installation.

## Oracle 11gR2 database installation

The Oracle 11gR2 database is needed for both the MedRec and SPECjEnterprise applications and hence is installed into both the ld01-db01 and the ld02-db02 virtual machines.

### Configure Solaris for Oracle Installation

To prepare for the database installation several Solaris setup steps are required.

Create the Oracle Groups and User Account that will be used to install and maintain the Oracle 11g Release 2 software.

```
# /usr/sbin/groupadd -g 200 oinstall
# /usr/sbin/groupadd -g 201 dba
# /usr/sbin/useradd -u 200 -g oinstall -G dba oracle
```

Verify the user “oracle” has been created

```
# id -a oracle
uid=440(oracle) gid=200(oinstall) groups=201(dba)
```

Set the password on the oracle account

```
# passwd oracle
Changing password for user oracle.
New password: *****
Retype new password: *****
passwd: all authentication tokens updated successfully.
```

By default, one quarter of physical memory is allocated for shared memory. If a greater shared memory parameter is required, then use Solaris 10's Resource Control Facility to set the shared memory kernel parameter, `project.max-shm-memory`. It is well worth the time to research the Solaris “project” facility to understand how server resources can be assigned on a per-user and per-application fashion in Solaris.

Modify the resource control project settings, so that they persist after a system restart:

```
# projmod -sK "project.max-shm-memory=(privileged,8G,deny)" user.oracle
```

Verify that the resource control is active:

```
# su - oracle
$ id -p
uid=100(oracle) gid=100(dba) projid=100(group.dba)
$prctl -n project.max-shm-memory -i process $$
process: 5754: -bash
NAME      PRIVILEGE      VALUE      FLAG      ACTION      RECIPIENT
project.max-shm-memory  privileged      8.00GB      -              deny
```

Set the following C Shell environment variables before starting the installation of Oracle RAC:

```
% setenv ORACLE_BASE /export/oracle
% setenv ORACLE_HOME /export/oracle/app/product/11gR2/db_1
% setenv ORACLE_SID medrecdb
```

### Oracle 11gR2 Software installation

The Oracle 11gR2 database software can now be installed. Set the DISPLAY environment variable to installer GUI. As the oracle user, start the installation using "runInstaller" from the "database" directory:

1. Welcome
  - Click on Next
2. Specify Inventory Directory and Credentials
  - The defaults should be correct
  - Click on Next
3. Select Enterprise Edition
  - Click on Next.
4. Specify Home detail
  - Name: Oracle\_Home
  - Path: /export/oracle/app/product/11gR2/db\_1.
  - Clear the check box for "create starter database,"
  - Click on Next.
5. Specify Hardware Cluster Installation Mode
  - Select Local Installation
  - Click on Select All
  - Click on Next
6. Product-Specific Prerequisite Checks
  - Correct any problems found before proceeding.
  - Click on Next
7. Database Install option
  - Select "install database software only"
  - Click on Next.
8. Summary
  - Click Install and wait until the installation finishes.
9. Execute Configuration Scripts
  - Execute the scripts presented by the installer as root.
  - Click

## Database Creation

The Database Configuration Assistant (DBCA) can be used to create the databases for both the MedRec and SPECjEnterprise2010 applications. The example below demonstrates creating the MedRec database.

```
# $ORACLE_HOME/bin/dbca
• Welcome
  • Click on Next
• Operations
  • Select "Create Database"
  • Click on Next
• Node Selection
  • Select all the nodes in the cluster
  • Click on Next
• Database templates
  1. Select "General Purpose."
  2. Click on Next.
• Database Identification
  1. Global Database Name: medrec
  2. SID: medrec
• Management Options
  • Check "Configure Database with Enterprise Manager."
  • Click on Next.
• Database Credentials
1. Choose password options.
2. Click on Next.
• Storage Options
  1. Select "Filesystem option"
  2. Click on Next.
  3. Specify the location for the creation of the data files. Choose the
     volumes created on the Oracle Enterprise S7410 storage.
  4. Click on Next.
• Recovery Configuration
  • Click on Next.
• Database Content
  • Click on Next.
• Initialization Parameters
  • Set initialization parameters accordingly (i.e 2GB in our test)
  • Click on Next.
• Database Storage
• Click on Next.
• Creation Option
• Check "Create Database"

• Click on Finish
```

## Deploying and running MedRec

The MedRec application suite consists of two main Java EE applications. Each application supports one or more user scenarios for MedRec:

- medrecEar—Patients Web Application
- physicianEar—Physicians Web Application

The medrecEar and PhysicianEar application are deployed to the WebLogic Server instance called MedRecServer. The physicianEar application communicates with the controller components of medrecEar via Web services.

MedRec is part of “Server Examples” bundle that gets installed during the WebLogic custom installation as shown in the WebLogic installation instructions above.

By default the MedRec sample application uses the embedded “derby” database. To switch to the Oracle database installed on ld01-db01 and ld02-db02 virtual machines the steps are :

- modify the JDBC settings of the WebLogic domain running MedRec to point to the Oracle 11g instance on the guest virtual machine (see below)
- create the MedRec schema in the Oracle 11g database instance and populate the database with test data.

**Modify the MedRec-jdbc.xml to match the following :-**

```
# cat <$WL_HOME>/samples/domains/medrec/config/jdbc/MedRec-jdbc.xml>
<jdbc-driver-params>
  <url>jdbc:oracle:thin:@<HOSTNAME>:1521:medrec</url>
  <driver-name>oracle.jdbc.driver.OracleDriver</driver-name>
  <properties>
    <property>
      <name>user</name>
      <value>medrec</value>
    </property>
  </properties>
  <password-encrypted>{3DES}MRDbxG9Avyo=</password-encrypted>
</jdbc-driver-params>
```

**Create the MedRec schema and populate the database**

Two SQL scripts are available in the \$WL\_HOME/ samples/server/medrec/setup/db directory under the WebLogic server installation directory :

- medrec\_oracle.ddl – which contains the SQL statements for creating the tables used by the MedRec application.
- medrec\_oracle.sql - contains the SQL statements for populating the tables with data.

To start the MedRec application :

```
# cd <$WL_HOME>/samples/domains/medrec
# ./startWeblogic.sh &
```

Once the WebLogic server is started the “ Server started in RUNNING mode” will be displayed and MedRec application should be available at <http://<hostname>:7001/start.jsp>. If this address is not accessible then review the messages from the “./startWebLogic.sh” command above.

### Deploying SPECjEnterprise2010

Detailed deployment instructions for the SPECjEnterprise2010 benchmark application is beyond the scope of this document. The basic steps to use SPECjEnterprise2010 as an example application are:

- install WebLogic server in zone1 using the WebLogic installation instructions detailed above.
- build the specj.ear and emulator.ear files using the supplied ant scripts
- add the JMS queues and datasources to the WebLogic instance
- deploy specj.ear and emulator.ear using the ant scripts
- clone zone1 to zone 2 using the clone option of zoneadm and then change the WebLogic host / IP to reflect the new zone IP address.

A full user guide is available for SPECjEnterprise2010 at <http://www.spec.org/jEnterprise2010/docs/UsersGuide.html> and the benchmark is available for order at <http://www.spec.org/order.html> .

## References

### Servers and Storage

- Oracle Sun T-series Servers : <http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/T-series/index.html>
- Sun T5240:  
<http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/T-series/031584.htm>
- Deploying Oracle Databases Using NFS on Sun Storage 7000 Unified Storage System *Sridhar Ranganathan and Jeffrey Wright*
- Oracle Sun Storage S7410:  
<http://www.oracle.com/us/products/servers-storage/storage/unified-storage/031680.htm>

### Oracle Solaris 10 Operating System

- Solaris 10:  
<http://www.oracle.com/us/products/servers-storage/solaris/index.html>
- Solaris ZFS Filesystem:  
<http://www.oracle.com/us/products/servers-storage/storage/storage-software/031857.htm>

### Virtualization

- Oracle VM Server for SPARC:  
<http://www.oracle.com/us/technologies/virtualization/oraclevm/oracle-vm-server-for-sparc-068923.html>
- Beginners Guide to LDoms:  
<http://www.sun.com/blueprints/0207/820-0832.pdf>
- Solaris Containers:  
<http://www.oracle.com/technetwork/systems/containers/index-jsp-140778.html>

### Software and Applications

- Oracle 11gR2 Database:  
<http://www.oracle.com/us/products/database/index.html> Oracle WebLogic 11g:  
<http://www.oracle.com/us/products/middleware/application-server/index.html>
- SPECjEnterprise2010:  
<http://www.spec.org/jEnterprise2010>

## Appendix A – Creating and Installing a Zone

This example shows the commands required to create an Oracle Solaris Container. In this example we will create a container with the name ld02-z01-app01

```
# zonecfg -z ld02-z01-app01
ld02-z01-app01: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:ld02-z01-app01> create
zonecfg:ld02-z01-app01> set zonepath=/zonepool/zone1
zonecfg:ld02-z01-app01> set autoboot=true
zonecfg:ld02-z01-app01> add net
zonecfg:ld02-z01-app01:net> set address=10.6.140.137
zonecfg:ld02-z01-app01:net> set physical=vnet0
zonecfg:ld02-z01-app01:net> end
zonecfg:ld02-z01-app01> add net
zonecfg:ld02-z01-app01:net> set address=192.168.2.201
zonecfg:ld02-z01-app01:net> set physical=vnet1
zonecfg:ld02-z01-app01:net> end
zonecfg:ld02-z01-app01> verify
zonecfg:ld02-z01-app01> info
zonename: ld02-z01-app01
zonepath: /zonepool/zone1
brand: native
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: shared
inherit-pkg-dir:
  dir: /lib
inherit-pkg-dir:
  dir: /platform
inherit-pkg-dir:
  dir: /sbin
inherit-pkg-dir:
  dir: /usr
net:
  address: 10.6.140.137
  physical: vnet0
  defrouter not specified
net:
  address: 192.168.2.201
  physical: vnet1
  defrouter not specified
zonecfg:ld02-z01-app01>commit
zonecfg:ld02-z01-app01>exit
```

```
# chmod 700 /zonepool/zone1
```

Verify that the zone is configured correctly

```
# zoneadm list -cv
ID NAME          STATUS      PATH                BRAND   IP
0 global         running     /                   native  shared
ld02-z01-app01  configured /zonepool/zone1  native  shared
```

Install the operating system in the new zone.

```
# zoneadm -z ld02-z01-app01 install
cannot create ZFS dataset zonepool/zone1: dataset already exists
Preparing to install zone <ld02-z01-app01>.
Creating list of files to copy from the global zone.
Copying <7690> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1139> packages on the zone.
Initialized <1139> packages on zone.
Zone <ld02-z01-app01> is initialized.
The file </zonepool/zone1/root/var/sadm/system/logs/install_log> contains a log of
the zone installation.
```

## Appendix B – Creating an iSCSI target and ZFS pool.

The following example uses the web management interface of the Sun Storage S7410 and is available by browsing to <https://hostname:215> with a supported browser (refer to the storage section in the references for more details on 7000 storage management and configuration).

To create an iSCSI target on Sun Storage S7210 using the web management interface.

Go to Configuration → SAN → iSCSI Targets , Click on the (+) icon to create a new iSCSI target.

Click on the OK after setting target properties. It should create a new iSCSI target as shown below.



To share LUNs only via particular targets build Target Groups. To create a group or add to an existing one, drag the entity from the left to the table on the right. When done adding new target group click on Apply to save the changes.



Create and Share a LUN as an iSCSI target

LUN can be quickly created using the the storage web console.

Go to the Shares → Projects → default → LUNs.

Click on the (+) icon and provide the LUN configuration.

**Create LUN**
**CANCEL** **APPLY**

Project

Name

Volume size

Thin provisioned

Volume block size

Target Group

Initiator Group

LU Number  0  Auto-assign

Operational Status

Target group “targets-0” share this LUN as an iSCSI target.

When done click on the Apply to create a LUN.

Filesystems <span style="float: right;">LUNs 1 Total</span>			
NAME	SIZE	GUID	
ldom-lun	500G	600144F09E0C3A8000004C62AE220003	

Enable iSCSI Data Service, to provide LUN access via the iSCSI protocol; Ensure the iSCSI Data Service is enabled.

Go to Configuration → Services If the service is not online, click the power icon .

Data Services			
Service	Status	Last Update	Actions
NFS	Online	2010-8-5 13:02:21	
iSCSI	Online	2010-8-10 16:04:59	
CIFS	Disabled	2010-8-5 13:00:43	
FTP	Disabled	2010-8-5 13:00:43	
HTTP	Disabled	2010-8-5 13:00:44	
NDMP	Online	2010-8-5 13:02:20	
Remote Replication	Online	2010-8-5 13:01:41	
Shadow Migration	Online	2010-8-5 13:01:41	
SFTP	Disabled	2010-8-5 13:00:44	
Virus Scan	Disabled	2010-8-5 13:00:44	

Mounting an iSCSI device and enabling the iSCSI initiator daemon

```

bash-3.00# svcadm enable svc:/network/iscsi/initiator:default
bash-3.00# svcs svc:/network/iscsi/initiator:default
STATE          STIME         FMRI
online         Aug_05        svc:/network/iscsi/initiator:default
bash-3.00# iscsiadm modify discovery --sendtargets enable

```

Adding Storage IP address as the discovery-address. Provide the appropriate iSCSI target server ip in place of 10.6.140.88

```

bash-3.00# iscsiadm add discovery-address 10.6.140.88

```

Displaying iSCSI target details

```

bash-3.00# iscsiadm list target
Target: iqn.1986-03.com.sun:02:661c5ce0-9a48-4d29-dc88-be12e2937635
  Alias: ldom-target
  TPGT: 2
  ISID: 4000002a0000
  Connections: 1

```

Formatting and Labeling the disk.

```

bash-3.00# format
Searching for disks...done
c5t600144F09E0C3A8000004C62AE220003d0: configured with capacity of 499.91GB
AVAILABLE DISK SELECTIONS:
  0. clt0d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
    /pci@400/pci@0/pci@8/scsi@0/sd@0,0
  1. clt1d0 <SEAGATE-ST914602SSUN146G-0603-136.73GB>
    /pci@400/pci@0/pci@8/scsi@0/sd@1,0
  2. c5t600144F09E0C3A8000004C62AE220003d0 <SUN-SunStorage7410-1.0 cyl 16250 alt
2 hd 254 sec 254>
    /scsi_vhci/ssd@g600144f09e0c3a8000004c62ae220003
Specify disk (enter its number): 2
selecting c5t600144F09E0C3A8000004C62AE220003d0
[disk formatted]
Disk not labeled. Label it now? yes

```

Create a ZFS pool on the iSCSI disk.

```

bash-3.00# zpool create iscsi-ldompool c5t600144F09E0C3A8000004C62AE220003d0s6

bash-3.00# zpool list
NAME          SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
iscsi-ldompool 496G  76.5K  496G    0%  ONLINE  -
ldompool      136G  50.6G  85.4G   37%  ONLINE  -
bash-3.00#

```



Deploying Oracle WebLogic (Java EE)  
applications on Oracle Sun T-series Servers and  
Sun Storage 7000 Unified Storage  
September 2010  
Author: Tom Daly, Ramin Moazeni

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 © 2010, 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. 0410

**SOFTWARE. HARDWARE. COMPLETE.**

