An Oracle White Paper
September 2010

Virtualization Options for Oracle Database Deployments on Sun SPARC Enterprise T-series Systems

ORACLE

Introduction

The combination of Oracle's Sun SPARC Enterprise T-series systems and the Oracle Solaris Operating System provides an ideal platform for Oracle Database deployments. The Oracle Solaris Operating System includes virtualization technology called Oracle Solaris Containers. Oracle Solaris Containers isolate software applications and services using flexible, software-defined boundaries, allowing many private execution environments to be created within a single instance of Oracle Solaris. As part of Oracle Solaris, Containers are available across the entire T-series server line. In addition to the capabilities provided by Oracle Solaris Containers, the T-series servers all feature virtualization capabilities granting hardware isolation called Oracle VM Server for SPARC (previously known as Logical Domains). The ability to virtualize at both the Oracle VM Server for SPARC layer as well as at the Oracle Solaris Container layer offers users a good deal of flexibility when deploying applications and application infrastructure.

This document is intended for IT professionals who want to understand the benefits of deploying Oracle databases into virtualized environments on Sun SPARC Enterprise T -series systems.

Overview of Sun SPARC Enterprise T-series Systems

Oracle offers a variety of Sun SPARC Enterprise T-series systems, ranging from the single socket T5120/T5220 to the 4-socket T5440 servers. Running the Oracle Solaris Operating System, these T-series servers all feature enterprise-class reliability, availability and serviceability, features such as predictive self-healing and hot swap components (internal disks and power supplies) for maximum availability. The T5120 and T5220 feature up to 64 Logical Domains each, while the T5440 features up to 128 Logical Domains each, granting the ability to partition hardware resources of T-series enterprise servers into smaller logical systems. Each Logical Domain runs its own copy of the Oracle Solaris Operating System, and CPUs may be moved between Logical Domains without requiring a reboot or even restarting an Oracle Database. Each Logical Domain in turn is capable of running a few to thousands of virtualized operating system instances using Oracle Solaris Containers.

In addition to the isolation of Logical Domains and the operating system virtualization of Oracle Solaris Containers, Oracle Solaris Resource Manager can be used to precisely control resources available to multiple applications running in a single OS instance with a minimum of administrative effort.

Logical Domains are created at the hypervisor layer, a virtual layer in the firmware of T-series systems, managed from the host control domain. Oracle Solaris Resource Manager offers resource controls without the need to administer another operating system. Oracle Solaris Containers offer a lot of flexibility in resource control as well as the isolation of virtual operating system instances. In many cases, enterprises can benefit by combining more than one of these technologies on a single server.

Figure 1 illustrates the spectrum of isolation/flexibility for Logical Domains, Oracle Solaris Containers, and Oracle Solaris Resource Manager:

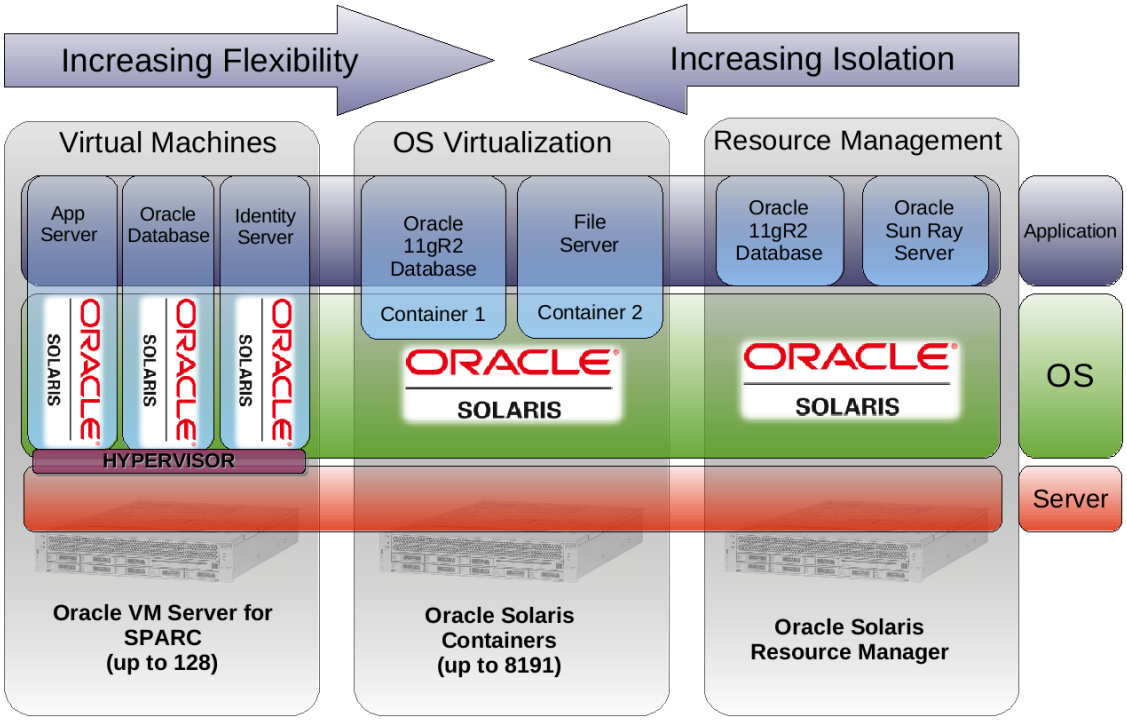

Figure 1 : Spectrum of Isolation and Flexibility for virtualization options

Introduction to Oracle VM Server for SPARC

One of the key features of the Oracle's Sun SPARC Enterprise T-series servers is the ability to partition the available hardware resources into smaller logical systems called Logical Domains. Logical Domains run their own copies of the Oracle Solaris operating system and offer a high level of isolation from other domains in the system because the partitioning occurs at the hypervisor level. Oracle's Sun T-series servers, configured with up to 256 CPUs and 512 GB 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.

Hardware and OS isolation

1. Logical Domains are implemented in a hypervisor, a small piece of firmware that is responsible for resource isolation. Since it is implemented in firmware a small change is needed to the operating system and this change is made in Oracle Solaris. All Input/Output (I/O) resources provided to guest domains

became virtual instead of physical, which means that those resources are not directly connected to the guest domain, but accessed indirectly from a special domain called an I/O service domain, which directly connects to I/O resources. All necessary changes have already been made to the storage and network drivers, so nothing needs to be changed in the user's application. Any user application written for generic Oracle Solaris will work within a logical domain without modification.

2. Each logical domain is the only owner of any CPUs and memory resources assigned to it; it runs its own instance of the Oracle Solaris operating system, installed patches and user accounts. This is in contrast to Oracle Solaris Containers where a single Oracle Solaris kernel is “shared” between many Containers. The Logical domain hypervisor separates the CPUs and memory of each logical domain one from another.

3. Since each logical domain has its own instance of Oracle Solaris, no logical domain can be impacted by a failure in another logical domain. There is however a dependency of a logical domain on an I/O service domain. In the case of failure of an I/O domain all I/O in the logical domain will be stalled until the I/O domain is recovered assuming that no redundancy is configured. It is recommended not to run any user applications in an I/O domain. There are some techniques mentioned later in this paper to increase I/O redundancy using multiple I/O domains.

Certification and licensing

Oracle Database Single Instance and RAC 10g, 11g Release 1 and 2 are certified and supported on Oracle VM Server for SPARC. Consult with your local Oracle sales representative about current licensing details.

Clustered environment deployment options

With server virtualization, it is possible to run multiple virtual machines and operating systems on a single physical system. This capability makes it possible to host multiple Oracle Real Application Clusters 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-2 below.

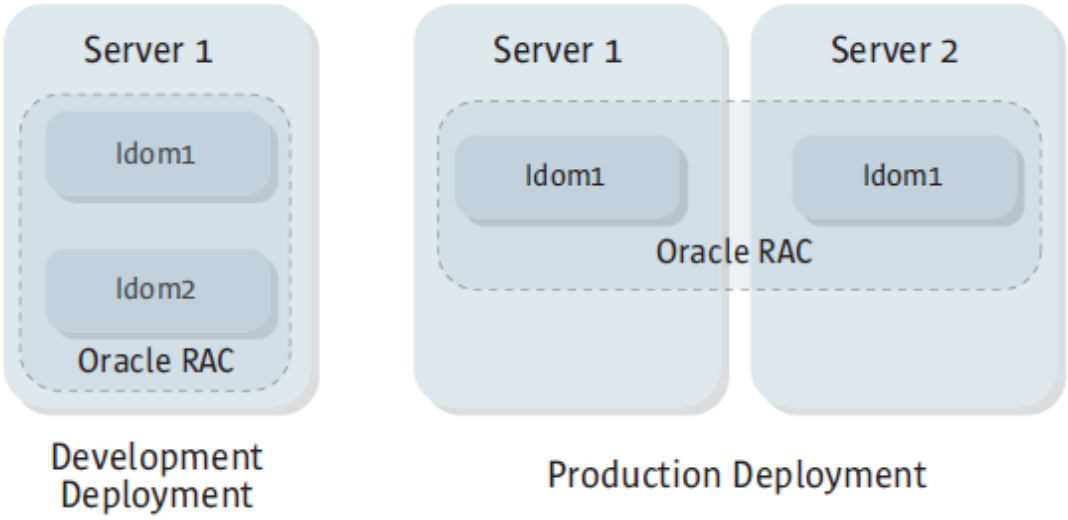

Figure-2: Deployment Options

- Development — All Oracle RAC nodes are located on domains on the same physical server. This variant 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 running a production environment because the physical server is then a single point of failure: if the entire server goes down then all nodes will be down and the entire Oracle RAC cluster will unavailable.

- Production — Oracle RAC nodes are placed on separate physical servers. This variant is recommended for a production environment, because in this configuration nodes are on different physical servers and there is no single point of failure.

Logical domains overview

With Logical Domains, domains have roles which 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 Database 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:

- Control Domain — The control domain is the domain that runs the Logical Domains Manager software, which is used to configure and manage all domains on the platform.

- I/O Domain — An I/O domain is a domain that owns some physical I/O resources, such as physical disks or physical network interfaces, and has direct access to the corresponding physical I/O devices. 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; this limit is currently tied to the number of PCI buses available. For example, a maximum of four I/O domains can be created on the Sun SPARC Enterprise® T5440 server.

- Service Domain — A service domain is a domain that provides virtual device services (such as virtual disk or virtual network service) to some other domains. A service domain is usually also an I/O 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 some service domains. A guest domain does not have any physical I/O devices, but instead uses virtual I/O devices such as virtual disks and virtual network interfaces. Oracle Database can run in a guest domain and use virtual devices.

In addition, a system running Logical Domains will have a primary domain. The primary domain is not a domain role, but is the name of the first domain created on the system. The primary domain initially has the roles of a control domain and I/O domain. Typically, this domain is also used as a service domain and provides virtual disk and network services to guest domains created on the platform.

For more details about domain roles please see: Beginners Guide to LDoms: Understanding and Deploying logical domains for Logical Domains 1.0 release by Tony Shoumack.

Example, creating new domain:

# ldm create ldg1	Create new domain ldg1
# ldm set-vpcu 16 ldg1	Assign 16 threads to domain ldg1
# ldm set-mem 16g ldg1	Assign 16GB of memory to domain ldg1
# ldm add-vdsdev /dev/rdsk/c5t6d0s2 vol1@primary-vds0	Create virtual device, user physical device /dev/rdsk/c5t6d0s2 as backend
# ldm add-vdisk vdisk1 vol1@primary-vds0 ldg1	Attach previously created virtual device as new disk to domain ldg1
# ldm bind ldg1	Bind assigned resources to domain ldg1
# ldm start ldg1	Start domain ldg1

Adding resources to domain:

# ldm add-vcpu 8 ldg1	Add 8 cpu thread from domain ldg1
# ldm add-mem 2g ldg1	Add 2GB from domain ldg1. Reboot required.
# ldm add-vdisk vdisk9 ldg1	Add virtual disk from domain ldg1

Removing resources from domain:

# ldm rm-vcpu 8 ldg1	Remove 8 cpu thread from domain ldg1
# ldm rm-mem 2g ldg1	Remove 2GB from domain ldg1. Reboot required.
# ldm rm-vdisk vdisk9 ldg1	Remove virtual disk from domain ldg1

Dynamic Reconfiguration

Logical domains enable dynamic changes of the resources assigned to a domain. That said, if you change the amount of memory assigned to a domain, you have to reboot that domain in order for the change to take effect. Currently, Oracle Solaris does not support changing memory online in Oracle VM Server for SPARC.

If you add to or remove CPUs from a domain, Oracle Solaris detects this and takes appropriate steps to add CPUs to the operating system or to take CPUs offline and remove them from the operating system. Moreover, you may set specific policies which will allocate more CPUs to a heavily-utilized domain or free CPUs from a domain if CPUs are idle.

Example of cpu policy, which will add or free CPUs in the range of minimum 2 and maximum 16 depending on the load and this policy will work between specified time:

```
# ldm add-policy tod-begin=09:00 tod-end=18:00 util-lower=25 util-
upper=75 vcpu-min=2 vcpu-max=16 attack=1 decay=1 priority=1
name=high-usage ldom1
```

Live Migration of an Oracle instance

Live migration enables a system administrator to move a running system from one physical box to another. Such migration can be done while all applications and Oracle Database are up and running thus keeping memory structures (for example the Oracle SGA) intact and caches warm. A few requirements must be satisfied before starting Live Migration. Firstly, the target system must have available CPUs and memory resources, so they can be assigned to the migrated domain. Additionally, the target system must have access to the same external storage as the source domain. During migration all applications and the operating system is suspended. The time needed to transfer image from source to target depends on the amount of memory assigned. (For details see "Increasing application availability using Sun's Logical Domain Mobility.")

Physical to virtual migration

If you are planning to migrate from an old system to a new one or consolidate multiple systems onto a single system, the P2V (physical to virtual) migration tool will make the task much easier. This tool is bundled with Logical Domains Manager software as a separate package and installed automatically by the installation script. The P2V tool converts existing environments into a virtual system. The source system may be any of the following:

- Any sun4u SPARC system that runs at least the Oracle Solaris 8 Operating System
- Any sun4v system that runs the Oracle Solaris 10 OS, but is not running in a logical domain

(see “Logical Domains 1.3 Administration Guide” for details)

Increasing storage availability

- MPxIO (Oracle Solaris Multiplexed I/O) within a single I/O domain. If the storage is attached to the I/O domain by multiple connections, they may be combined into single MPxIO path to increase availability. Multiple Host Bus Adapters (HBA) should be used to avoid any single HBA dependency. The benefit of this configuration is that it protects from most failures (such as a cable disconnect or HBA failure) while keeping configuration simple. The disadvantage of this configuration is that if an I/O domain fails all guest domains accessing disks through this I/O domain will lose connectivity to the disks.

- MPxIO with Virtual I/O Failover. Each Logical Domain itself offers Virtual I/O Failover. In the simple configuration each physical storage device is assigned to single virtual service in single I/O domain. With Virtual I/O Failover you may assign two virtual disk services each with it's own path to different I/O domains. Virtual disk services are created by assignment to the same multipath group. In the case of failure of one I/O domain Virtual I/O Failover will cause all traffic to switch to another I/O domain. The benefit of this configuration is that it protects not only against HBA or cable failure, but also against the failure of an I/O domain. The disadvantage of this configuration is that four connections are required.

For details see “LDoms I/O best practices storage availability with logical domains” by Peter A. Wilson

Network availability

- IPMP (IP network multipathing) with two I/O Domains. In this configuration two virtual network switches must be created, one in each I/O domain. Two virtual network interfaces are created on top of these virtual network switches and both interfaces are assigned to the guest domain. Within the guest domain these two virtual network interfaces are bonded into an IPMP group. You will need an extra physical network card to build such a configuration, since the four on-board network interfaces belong to a single bus and can't be attached to multiple I/O domains.

- IPMP with single I/O Domain. This case is similar to the previous configuration except that both network switches are attached by a single I/O domain. This case is simple to implement and it can be set up with the built-in network interfaces, although the I/O domain becomes a single point of failure.

- IPMP within a single I/O Domain. Instead of creating two virtual network interfaces for a guest domain and binding them into IPMP group, two physical network interfaces can be bound into an IPMP group within a single I/O domain. A single network interface still needs to be created for the guest domain. Such a configuration will hide the IPMP implementation from the guest domain but will add the necessity to route packets between the guest domain and the IPMP group.

For details see “LDoms I/O best practices network availability with logical domains” by Peter A. Wilson

Performance Impact

When running Oracle Database in a guest domain it is necessary to take into account possible performance impact:

1. An I/O service domain needs CPU cycles to serve I/O requests coming from a guest domain. This can be achieved by assigning one or two cores (8 or 16 CPUs) to I/O domain. Tests have shown that the CPUs system load on an I/O domain with 24 CPUs is around 15% for a storage I/O load of 300 MB/sec for an FC-AL link and a network load around 75 MB/sec simultaneously.

2. There will be additional I/O latency due to the extra processing path, although such latency was not significant in executed tests.

3. It is important to avoid swap activity on the guest domains because swap activity on the guest domains will generate additional I/O requests which must be handled by the control domains.

4. CPU cores should not be shared between domains, especially between two domains running Oracle database instances. If CPUs from the same CPU core are assigned to different domains, then this can reduce the efficiency of these CPUs. The reason for

this is that a single CPU core shares a memory cache between CPUs belonging to this core. Cache Thrashing occurs when two programs are using the cache for different memory pages and the load is very heavy. To avoid this situation always add/remove CPUs in multiplies of 8 (number of CPUs in CPU core on T2 and T2 Plus platform). See the script check_core_assignment.pl in the Appendix. This script can be used to periodically check if any cores are shared between domains. The message "MultiUsage detected" is a signal that listed cores are shared.

ZFS

A ZFS volume can be used as the back-end device for the system virtual disk of any guest domain. For rapid deployment and disk space savings, it is recommended to create a ZFS snapshot of the ZFS volume used as a back-end device for the system virtual disk of an installed but unconfigured domain. Later, additional guest domains can be quickly deployed by utilizing the ZFS snapshot of a system disk of an installed but unconfigured domain. To deploy a new guest domain, all that is required is to make a clone of this ZFS snapshot. This leads to significant time saving as the Oracle Solaris operating system is already installed for the new guest domain. In addition, less disk space is required, as all guest domains on this system will be cloned from one copy and so will not consume disk space for common files. Of course changing a file in one domain does not replicate this change across other domains.

Management

Oracle VM Server for SPARC is supported by Oracle Enterprise Manager with Ops Center installed. With Oracle Enterprise Manager you can manage your systems starting from deployment, patching, starting and up to the Cluster and Database management.

Oracle Solaris Containers

Oracle Solaris Containers provide additional flexibility in virtualizing operating system instances, while still providing many of the isolation features.

Operating System virtualization with Oracle Solaris Containers allows a one-application-per-server deployment model while simultaneously sharing hardware resources. An integral part of the Oracle Solaris Operating System, Oracle Solaris Containers isolate applications and

services using flexible, software-defined boundaries and allow many private execution environments to be created within a single instance of Oracle Solaris. Each environment has its own identity, separate from the underlying hardware. Each behaves as if it is running on its own operating system making consolidation simple, safe and secure.

The number of containers on a system is limited in practice only by memory and disk space, though currently a maximum of 8191 Oracle Solaris Containers can be created for a single operating system image. Each Oracle Solaris Container has a very small CPU and memory overhead, far less than a typical hypervisor-based operating system instance. Oracle Solaris Containers do not restrict CPU and memory resource associations, so an execution environment can allocate many CPUs but limited memory, a lot of memory but few CPUs, or a more balanced pool of resources.

Oracle Solaris Containers can all share CPU resources, can each have dedicated CPU resources, or can each specify a guaranteed minimum amount of resources as well as a maximum. Memory can be shared among all Oracle Solaris Containers, or each can have a specified memory cap. Physical I/O resources such as disk and network can be dedicated to individual Oracle Solaris Containers, shared by some, or shared by all. Regardless of what is shared or dedicated, each virtualized environment will have isolated access to local file system and networking, as well as system and user processes.

For example, consider a consolidation involving multi-tier applications like Learning Management System (LMS), and School Management System (SMS). The application stack has 3 servers: a database server at the back-end, an application server in the middle tier and a web server at the front-end. Each application stack needs 3 servers to host the application. Consider the following existing environment:

1. There are 6 existing production servers for both LMS and SMS applications.

2. The existing development environment has 3 servers, and the pre-production testing environment has 3 servers.

3. The CPU utilization observed on all the above servers is in the range of 10-20%.

Now let's consolidate these servers.

3 productions servers for SMS + 3 production servers for LMS + 6 for development and pre-production environment of LMS and SMS = total 12 servers.

Based on the above environment, all these 12 servers can be consolidated on two T-series Sun SPARC T5240 servers, each with 2 sockets and 16 cores for a total of 128 CPUs.

Following are benefits of such a consolidation based on Oracle Solaris Containers:

1. A smaller number of OS instances to manage: only 2 from 2 servers.

2. An isolated, secure environment which looks like an independent system.

3. Different applications administrators will have root access to only their application environment.

Let's look at what we could do to consolidate 12 servers on two T5240 servers using Oracle Solaris Containers.

First T5240 - Database Server

Further to the above requirement, make one of the T5240s as the Database Server. To host 6 different database servers, let's create 6 Containers (2 Containers for LMS DB and SMS DB x 3 environments, production, pre-production and development = 6 Containers for DB). Assign two cores (16 CPUs) for each Oracle Solaris Container. These Containers act as independent Oracle Database Servers. Create a project for the oracle user, and assign all necessary resource limits, such as the maximum shared memory limit, within the oracle user project. Note that before Oracle Solaris 10, /etc/system would have been used to set these limits.

For faster deployment of all these database servers, host all these containers on a ZFS file system. Create one Container environment with Oracle binaries installed on it. Then clone the same to create other Container environments. This Cloning processes does not require additional disk space, however once these environments are established, if different Oracle Database patches are required for different applications these patches could be applied independently to each container, and the cloned file system space will grows accordingly based on the changes being made in that Container.

Limit the memory of each Container as required using the Container configuration parameter "capped-memory". If it's not set the entire memory is shared among all the Containers.

Based on the resource requirement during run time, cores can be added or removed dynamically from the different Container environments.

Second T5240 - Application and Web Server

In addition to the above environment, allocate one of the T5240s as the Application and Web Server. Let's consolidate Application and Web server in the same Container environment using the Oracle Solaris Fair Share Scheduler (FSS) to ensure that a certain proportion of the available CPU resources is always dedicated to one of the servers. By this process the application and web servers are hosted on the same Container environment.

To host 6 different application servers, let's create 6 Containers [(1 for LMS + 1 for SMS) x 3 environments production, pre-production and development = 6 Containers for Applications]. Assign two cores (16 threads) for each Oracle Solaris Container. These Containers act as independent Oracle Application and Web Servers.

Create a project for the application user, and assign a 60% share of the available CPU for the Application server environment and assign the rest of the 40% share to the web server on the same Container environment. Dynamically increasing or decreasing the CPU count will proportionately give a dedicated share of the CPU resources to the web and application servers within a Container environment.

Since all these Containers are hosted on a ZFS file system, create one environment with Web and Application servers installed without the application itself. And clone the same for all other environments, meaning that deployment can be carried out more quickly.

Limit the memory of each Container as required using the Container configuration parameter "capped-memory". If it's not set the entire memory will be shared among all the Containers.

Suppose based on run time resource requirements, that CPU resources need to be changed dynamically for the production Container environment. First try to manage it within the Container environment by altering the share percentage, or if additional compute power is

required, assign additional cores to the container environment so that the compute power of each share increases proportionately.

Oracle Solaris Containers are ideal for such a consolidation. With the increasing cost and complexity of managing many separate systems, or even the larger granularity of Oracle VM Server for SPARC, it is often advantageous to consolidate multiple applications onto larger, more scalable servers. Oracle Solaris Containers allow more efficient resource utilization with a reduced number of systems.

Dynamic resource reallocation permits unused resources to be shifted to other Oracle Solaris Containers as needed. Security and fault isolation mean that poorly behaved applications no longer require a dedicated and often under-utilized system. With the use of Oracle Solaris Containers, these applications can be safely and securely consolidated with other applications. This allows system administrators to delegate some administrative functions while maintaining overall system security.

If CPU resource distribution needs to be controlled with finer granularity than a CPU, CPU resources can be specified in shares (which are used to express a ratio and can add up to as big a number as one chooses).

For more complete hardware level isolation and greater fault isolation, as well as to host different operating systems with different patch set levels, Oracle VM for SPARC on Sun SPARC Enterprise T-series servers may be a better match, but Oracle Solaris Containers can be created and hosted on top of these environments for even greater control over resource utilization.

For more information about Oracle Solaris Containers please see "System Administration Guide: Solaris Containers-Resource Management and Solaris Zones" on http://docs.sun.com.

Global zone

The Oracle Solaris base operating system is referred as the global zone, and all other soft partitions are created above this base and the same kernel is shared among all other partitions.

Non-global zone

The Oracle Solaris container environment known as a a non-global zone shares the global zone's kernel along with other non-global zones hosted on the same physical server or logical domain.

Certification

Single Instance Oracle Database 10gR2 and 11gR1 are certified to run inside Oracle Solaris Containers on Oracle Solaris for SPARC.

Oracle RAC Database 10gR2 and 11gR1 are certified to run inside Oracle Solaris Containers on Oracle Solaris for SPARC.

The licensing model of Oracle Database is aligned such that the use of Oracle Solaris Containers can provide cost containment. Please contact your local Oracle sales partner for further details.

Storage

MPxIO – the I/O multipathing facility of Oracle Solaris is leveraged in the global zone to provide high availability at the storage level, and these highly available storage LUNs (Logical Unit Number) can be provisioned inside Oracle Solaris Containers. Oracle data files can be hosted using ASM for both single instance Oracle databases or for Oracle RAC. In the case of Oracle RAC, ASM acts like a clustered volume manager to host Oracle data files and manages the high availability of LUNs across different storage boxes or controllers.

Dedicated LUNs are configured for each container environment for the application.

Network

At the network layer, virtualization is achieved by leveraging VLAN (Virtual Local Area Network) tagged NICs (Network Interface Card). With VLAN tagging, a single NIC port can allow a system to connect to multiple VLANs, and therefore multiple networks. Thus VLANs allow a single physical NIC to be divided into multiple logical NICs. High availability of the VLAN tagged NICs is achieved by leveraging Active-Active IPMP group configuration inside Oracle Solaris Containers environment. Since each VLAN tagged NIC is independent, a failure

of one NIC might affect one non-global zone, other non-global zones will not be impacted, and within a virtual environment an IPMP group can be used to provide high availability.

An Active-Active IPMP group configuration simplifies the deployment of Oracle RAC private networking, by assigning 2 IP addresses for both the NICs.

For public networking, although it is an active-active IPMP group, we could assign one IP as a VIP (virtual IP address), and another IP as the container IP hosted on each of the active NICs of the same group.

Performance

When Oracle Database is running inside Oracle Solaris Containers, be aware of the following possible performance implications:

1. I/O - There is no impact on the storage I/O, however understanding of the workload and isolation of dedicated paths is required. If paths are shared then having an idea about the traffic over those LUNs would help in distributing the load among all other available paths.

2. Network I/O may have some impact if the various container environments are leveraging the same physical set of NICs. This potential problem can be overcome by dedicating a physical NIC to each container (“exclusive IP” mode).

3. There will typically be less than 1% overhead on the system compared to running on the application on the global zone.

4. For optimal performance, assign one core (8 CPUs) at a time instead of assigning part of a core for a given Oracle Solaris Container.

ZFS

With many Oracle Solaris Containers deployed on single physical system, ZFS allows quick deployment of each new Oracle Solaris Containers by making a ZFS snapshot of an installed Oracle Solaris Container. To deploy new Oracle Solaris Container, just clone the ZFS snapshot. This saves time each time a new Oracle Solaris container is installed, as well as

saving disk space for all common files, since new deployments are clones of the first snapshot. However any new files or changes to existing files in one Oracle Solaris Container environment apply only to that environment.

Management

Oracle Solaris Containers are supported by Oracle Enterprise Manager with Ops Center installed. Oracle Solaris Containers can be created, deleted, configured, monitored, and migrated using Oracle Enterprise Manager.

Oracle Solaris Resource Manager

The Oracle Solaris Resource Manager is a resource control mechanism that provides the ability to allocate and control major system resources such as CPU, network bandwidth, and memory of various users or applications. This fine-grained ability enables the consolidation of multiple applications onto a single server thus improving the resource utilization and lowering the Total Cost of Ownership (TCO).

To guarantee predictable service levels, Oracle Solaris Resource Manager implements administrative policies that govern the resources that different users or applications can access, and more specifically, the level of consumption of those resources that each user or application is permitted. In other words, using Oracle Solaris Resource Manager, system administrators can define workloads, partition and allocate system resources to different entities in such a way that predefined Service Level Agreements are met while maintaining the overall quality of service and keeping the system resources busy. In addition, Oracle Solaris Resource Manager facility allows administrators to monitor resource usage, so they can identify users or applications that tend to use more resources than they should, and to compile more accurate data over time for capacity planning and billing purposes.

For example, in the case of a consolidated banking application, more resources can be allocated with higher priority to the ATM application during the daytime to ensure faster response to ATM users. During the off-peak hours of ATM activity, the priority and the resource allocation can be lowered in order to let other applications perform batch processing such as generating monthly bank statements.

The basic building blocks of Oracle Solaris Resource Manager are tasks, projects and resource controls. Oracle Solaris Resource Manager facilitates establishing resource limits on a per-process, per-task and per-project basis.

A "task" is a collection of related processes, and a "project" is an administrative identifier that is used to identify related work or to classify a service such as a database instance. A "project" may consist of one or more "tasks" that represent a workload. That is, a "workload" is an aggregation of all processes of an application or group of applications. Every process that runs in the system is associated with a "project" and a "task".

A "resource control" dictates how the Oracle Solaris operating system will manage the controlled resource as well as how the system will react when the imposed resource limit has been reached. For example, a system administrator at a university can limit the number of threads in each task to 50 for all tasks in a project that was created for all undergraduate students, and instruct the OS to kill such tasks when the established limit has been reached. This would help prevent runaway processes from exhausting system resources and in bringing the system to a complete halt.

For more information about the Oracle Solaris Resource Manager and the underlying technology, please refer to the "Resource Management" section in "System Administration Guide: Solaris Containers-Resource Management and Solaris Zones".

Deploying Oracle Database on the Oracle Solaris Platform

The Oracle Solaris Operating System runs across the entire range of Sun SPARC and x86 platforms from entry level servers to 64-processor servers like the Sun SPARC Enterprise M9000 server and 256 thread servers like the Sun SPARC Enterprise T5440.

The Oracle Solaris 10 Operating System introduces new features to enhance manageability, performance and availability. The key new features include Oracle Solaris Containers, Predictive Self-healing, DTrace for advanced observability, ZFS for next-generation volume management and file system support, and user and process rights management. An Oracle database deployment can take advantage of each of these features of Oracle Solaris 10 to enhance the manageability, scalability, availability and security of both single and multiple Oracle database instances – all across multiple platform and processor architectures.

Proven Performance and Scalability

The Oracle database has a proven track record of scaling well both vertically as well as horizontally on the Oracle Solaris 10 platform. Oracle Database 11gR2 with Oracle Real Application Clusters demonstrated excellent horizontal scalability across 12 Sun SPARC Enterprise T5440 servers on Oracle Solaris 10 running the industry-standard TPC-C workload. The Oracle database also scaled well on a single Oracle Sun SPARC Enterprise M9000 running the industry-standard TPC-H data warehousing benchmark. Oracle believes in empowering its customers to use scalability in both horizontal and vertical dimensions to best meet their performance and availability criteria. See References Section below for the results.

Predictive Self-healing: Enhanced Availability

The Oracle Solaris Operating System has implemented predictive self-healing for CPU, memory, and I/O bus nexus components for a variety of hardware platforms incorporating SPARC, AMD Opteron and Intel Xeon processors, exploiting the specific hardware RAS features provided by the underlying system. Additionally, the Oracle Solaris Operating System provides a platform neutral technology, Memory Page Retirement (MPR), to ensure that both the Oracle Solaris Operating System and user applications continue to operate in the face of main memory faults.

Oracle Solaris MPR technology ensures that Oracle database deployments can continue uninterrupted even when the underlying system has memory errors. Consider the scenario of an Oracle database instance deployed on a system that is experiencing memory errors. The diagnosis engine of the Oracle Solaris fault manager, which is continuously examining both correctable errors (CEs) and uncorrectable memory errors (UEs), will see a series of correctable errors in a memory location as an indication of a potentially uncorrectable memory error. If the Oracle database has memory pages that contain CEs then Oracle Solaris MPR will retire those pages from memory without interrupting Oracle processes. If the Oracle database references memory pages that have uncorrectable memory errors, then Oracle Solaris MPR will retire clean pages containing UEs, again without interrupting Oracle processes. In the unlikely case of the Oracle database having dirty memory pages with UEs, the oracle processes will come down. However, if the Oracle Database is configured with Service Management Facility, as explained in the next section, it can restart automatically.

Adding the Oracle database and Oracle listeners as a service to the Oracle Solaris Service Management Facility (SMF) can provide the following advantages:

1) Automatically restarts failed services in dependency order, whether they failed as the result of administrator error, a software bug, or were affected by an uncorrectable hardware error.

2) Provides more information about misconfigured or misbehaving services, including an explanation of why a service isn't running, as well as individual, persistent log files for each service.

3) Delegates the task of managing the Oracle services to Oracle administrators -SMF is integrated with Oracle Solaris Role Based Access Control (RBAC) which ensures that the services can be securely managed by non-root users, including the ability to configure, start, stop, or restart services.

User Rights Management: Enhanced Security

Default installations of the Oracle database can be made more secure by exploiting the user rights management feature of Oracle Solaris 10 security. In a typical Oracle deployment, all Oracle database administrators (DBAs) login as UNIX user oracle. Hence, it is not possible to track the DBA-related activities of an individual user; only the combined activities of all DBAs are tracked by the operating system and the database server. User rights management enables you to create an oracle role and assign it to users with DBA responsibilities. In this scenario, the users will login to the database server system with their regular UNIX logins and assume the oracle role when they need to do any Oracle DBA-related tasks. This approach ensures that multiple Oracle administrators do not share a single login. They login in as individual user and are accountable for their individual actions; yet they have the flexibility to perform all the functions of an Oracle administrator by assuming the oracle role. Complete accountability for individual users can be enforced by enabling auditing of the oracle role; which in turn will provide a detailed description all Oracle DBA-related activities for each individual UNIX user.

DTrace: Enhanced Observability

With the advent of multi-tier architectures today's applications have become very complex. While individual levels of the application tier may have excellent tools for observability and debugging, there are no tools to observe and optimize the entire application stack. This problem becomes even more complicated for observing applications in production which are

likely sensitive to performance impacts. Also, it is not always easy to stop and start these applications to enable debug flags. Adding debug versions of applications into production may not be permitted. Even if permitted, bringing debug versions into production involves expensive and time consuming QA cycles. All of these issues complicate the problem of observation.

DTrace, a Dynamic Tracing framework, was developed to address this very problem. It can be used to observe any or all tiers of the application stack, it is truly dynamic and does not require application code changes or even an application restart. One can observe fully optimized applications using DTrace. The overhead of observation is low and there is no overhead when observation is turned off. Instrumentation can be turned on and off dynamically thus only collecting information when it is needed. DTrace is safe and turns itself off when observation overhead affects system performance.

DTrace can be used to observe applications developed in C, C++, Java, JavaScript, Ruby, PHP, Perl, Python among other programming and scripting languages. Other system layers, like I/O, networking, application and kernel locks, CPU counters etc, can also be observed using DTrace.

DTrace scripts are used to enable and program points of instrumentation. D-script format does not change based on the application tier being observed and a single script can be used to observe multiple tiers at the same time.

DTrace can be used to look at Oracle database processes in isolation or concurrently with any other processes running on the system and can be an invaluable tool for identifying performance bottlenecks and many other real world issues.

Enhanced System V IPC implementation: Ease of Deployment

Prior to Oracle Solaris 10, installing the Oracle database on the Oracle Solaris Operating System required changes to the /etc/system file. Every reconfiguration required a reboot for the changes to take effect. However, the System V IPC implementation in Oracle Solaris 10 no longer needs changes to the /etc/system file. Instead, the new resource control facility is used, which allows changes to become effective immediately, without a system reboot. Furthermore, the default settings of the System V IPC parameters have been set to typical defaults enabling Oracle database instances to run out-of-the-box without requiring special parameters to be set.

Oracle database deployments on Oracle Solaris 10 work out of the box, with no additional system configuration, if the System Global Area (SGA) uses less than 25% of the system's total memory. If the deployment plans to use more than 25% of the systems memory, then the shared memory resource parameter can be dynamically set to the required value using the resource control facility.

Conclusion

Sun SPARC Enterprise T-series systems running Oracle Solaris provide ideal platforms for Oracle Database deployments. The Oracle Solaris Operating System delivers built-in virtualization capabilities across the entire T-series server line in addition to enterprise-class performance, reliability, availability, serviceability, and security. On top of the capabilities provided by the Oracle Solaris Operating System, the T-series servers feature virtualization capabilities granting hardware isolation of Logical Domains. The ability to virtualize at the Logical Domain layer – and at the operating system layer within each domain, in addition to the ability to control resource utilization with Oracle Solaris Resource Manager provides a broad range of options not available on other platforms.

Appendix

1. check_core_assignment.pl script

```
#!/usr/bin/perl

@AllCores = ();

open(DOM, "ldm ls -p|") || die "failed to get domains";

while (<DOM>)
{
  if ( m/DOMAIN\|name=([^\|]*)/ )
  {
    $domain = $1;
    open(CPU, "ldm ls-bindings -p $domain|") || die "failed to get cpus for
$domain\n";
    while (<CPU>)
    {
      if ( m/\|vid=\d*\|pid=(\d*)/ ) {
        $core = int($1 / 8);
        push (@AllCores, $core) unless $seen{$core}++;
         push (@{$Usage[$core]}, $domain) unless $seen{$core, $domain}++;
       }
    }
  }
}

foreach $c (sort {$a <=> $b} @AllCores)
{
       my $mu = 0;

       print "Core $c used by ";
       foreach $k (sort @{$Usage[$c]}) {
              print "$k ";
              $mu++;
       };
       if ($mu > 1) { print "MultiUsage detected"; }
       print "\n";
}
```

References

- Beginners Guide to LDoms: Understanding and Deploying logical domains for Logical Domains 1.0 release by Tony Shoumack
- “Increasing application availability using Sun's Logical Domain Mobility .”
- “LDoms I/O best practices storage availability with logical domains” by Peter A. Wilson
- White Paper: Best Practices for Data Reliability with Oracle VM Server for SPARC
- White Paper: Best Practices for Network Availability with Oracle VM Server for SPARC
- “Logical Domains 1.3 Administration Guide”
- Running Oracle Real Application Clusters (RAC) on Sun™ Logical Domains by Alexandre Chartre, Daniel Dibbets and Roman Ivanov.
- Deploying Oracle Database on the Oracle Solaris Platform
- Best Practices for Running Oracle Databases in Oracle Solaris Containers
- System Administration Guide: Solaris Containers-Resource Management and Solaris Zones
- Sun SPARC® Enterprise Series Servers Configuration Concepts
- Customer references: http://www.sun.com/third-party/global/oracle/success/index.jsp
- Sun SPARC Enterprise T5440 Servers and Sun Storage F5100 Flash Arrays Deliver World Record Database Result on TPC-C Benchmark http://www.oracle.com/us/solutions/performance-scalability/sparc-enterprise-t5440-bmark-073752.html

- Oracle® Database 11g Sets New World Record TPC-H Three Terabyte Non-Clustered Benchmark Result on Sun SPARC Enterprise M9000 Server http://www.oracle.com/us/corporate/press/073155

- “Highly available and Scalable Oracle RAC networking with Oracle Solaris 10 IPMP”, by John MchHug and Mohammed Yousuf

- “Best practices for deploying Oracle RAC in Oracle Solaris Containers virtual environment”, by Mohammed Yousuf

Virtualization Options for Oracle Database Deployments on Sun SPARC Enterprise M-series Systems

September 2010

Author: Roman Ivanov, Mohammed Yousuf

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

SOFTWARE. HARDWARE. COMPLETE.