Sun ORACLE SUN FIRE X86 SERVERS

An Oracle White Paper
September 2010

Deploying Oracle Database on x86 Systems with Oracle Solaris

ORACLE

Introduction

Oracle Solaris 10 is the ideal operating system for use on x86 hardware. Innovative, built-in features deliver breakthrough virtualization and utilization, high availability, advanced security, and industry leading performance.

Oracle Database 11g delivers industry leading performance, scalability, security and reliability on a choice of clustered or single-servers and provides comprehensive features to easily manage the most demanding transaction processing, business intelligence, and content management applications.

This whitepaper introduces the Oracle Solaris Operating System to users of Oracle Database 11g on x86-based platforms. It examines Oracle Solaris features and capabilities that are relevant to database environments, including performance, observability tools, security, compatibility, availability and virtualization options. Highly regarded features of Oracle Solaris on x86 include Oracle Solaris ZFS for next-generation volume management and file system support, Oracle Solaris DTrace for advanced observability, virtualization options including Oracle VM and Oracle Solaris Containers, Predictive Self Healing for continuous availability, and Solaris Trusted Extensions. These features are discussed with focus on the relevance to Oracle Database 11g deployments.

Overview of Solaris 10 on x86 Servers

With a single source code base, Oracle Solaris 10 runs on both x86- and SPARC-based systems, delivering the same features across all platforms. This means that applications developed and optimized on Oracle Solaris 10 run on more than 1,000 system models from leading vendors. The operating system runs on hardware ranging from laptops and single-board computers to datacenter and grid installations, while serving applications ranging from military command-and-control systems to telecommunications switch gear and stock trading.

Oracle Solaris 10 on x86 features which are particularly relevant to Oracle DBAs include:

- Dynamic SGA tuning of Oracle's database memory using Oracle Solaris's Dynamic Intimate Shared Memory (DISM) feature. When running Oracle Database on Oracle Solaris, the amount of physical memory consumed by the SGA can dynamically grow or shrink to meet continually changing resource requirements of the database.
- Oracle Database performance on x86 servers benefits from the use of Solaris 10 large memory pages which are automatically allocated for the database's SGA shared memory. Large pages can reduce the

number of memory pointers by more than 500 times. This reduction in complexity translates into noticeable performance improvements, especially on systems with very large amounts of memory.

- Oracle Solaris 10 security features are critical to enterprise deployments. The system's integrity can be verified by using Oracle Solaris' digitally signed binaries and file verification features. Reduce risk by granting only the privileges needed for users and processes. Simplify administration and increase privacy and performance by using the standards-based key management and cryptographic frameworks in Oracle Solaris. The system can be secured by using dynamic service profiles, including a built-in, reduced exposure network services profile. Access to data can be controlled based on its sensitivity level by using the labeled security technology in Oracle Solaris 10 with Trusted Extensions
- Oracle Solaris ZFS is a general-purpose file system that is designed to span from the desktop to the datacenter. Anyone who has ever lost important files, run out of space on a partition, spent weekends adding new storage to servers, tried to grow or shrink a file system, or experienced data corruption knows the limitations of traditional file systems and volume managers. Oracle Solaris ZFS addresses these challenges efficiently and with minimal manual intervention.
- System administrators, integrators, and developers can use the dynamic instrumentation and tracing capabilities in Oracle Solaris to see what's really going on in the system. DTrace can be safely used on production systems, without modifying applications. It is a powerful tool that gives a comprehensive view of the entire system, from kernel to application, even those running in a Java Virtual Machine. This level of insight reduces the time for diagnosing problems from days and weeks to minutes and hours, and ultimately reduces the time required to fix those problems.
- Oracle Solaris 10 includes the leading Web 2.0 open source packages, optimized and ready to run.
- Oracle Solaris 10 delivers binary compatibility from release to release and source compatibility between SPARC and x86 processors.
- Oracle Solaris 10 also offers powerful built-in virtualization features. With Oracle Solaris Containers, you can maintain a one-application-per-virtual-server deployment model while consolidating dozens or even hundreds of applications onto a single server and OS instance. These features let you share hardware resources while maintaining predictable service levels, increase utilization rates, and cut system and licensing costs while gaining the ability to quickly provision and move workloads from system to system.
- Setting numerous price/performance records since its release, Oracle Solaris 10 unleashes even more power from existing applications. Download the latest Oracle Solaris Studio compilers and developer tools to bring even greater performance to your applications.
- Predictive self-healing is a key feature in Oracle Solaris 10 that helps you increase system and service availability. It automatically detects, diagnoses, and isolates system and software faults before they cause downtime. And it spans the full range from diagnosis to recovery on systems based on SPARC, AMD, and Intel processors.

Oracle is moving fast to differentiate its x86 server products … by leveraging flash memory, Sun's Flash Accelerator, Oracle system-management software, and Oracle Grid Control software, and by building a clustering fabric to tie multiple x86 servers together. The broader view of the software landscape that Oracle brings to the Sun product line is one that:

1. Is working to combine Sun Ops Center systems management with the Oracle Enterprise Manager (OEM) framework, and remote-control for lights-out management via Oracle ILOM

2. Is building solutions optimized for specific solution-sets (e.g., analytics, ERP, data warehouse); this is the integrated stack approach announced by Oracle in January, combining hardware and software into deployable systems

3. "Supports virtualized fabrics that provide virtualized I/O to arrays of scale-out x86 servers, and also supports Oracle's own Oracle Virtual Machine (OVM) hypervisor, leveraging its software IP as it competes with HP, Cisco and IBM in scale-out converged infrastructure

…At the same time, Oracle is advancing its support of scale-out x86 server systems, whether deployed in an array, a grid, or a cluster. Oracle Grid Control software, and Oracle Real Application Clusters (RAC) are key elements of this scale-out approach to enterprise computing. And, addressing the management complexity and IT skill-set issues that many customers have faced in deploying scale-out enterprise systems, Oracle has pre-tested and pre-certified specific solutions that include automation of virtualized networking and workload management.

…Oracle's strategy is differentiated by the way that it is selling a complete end-to-end solution that combines Sun hardware with Oracle database software, Oracle middleware and Oracle applications, offering customers a full solution from a single vendor. This approach provides rapidly deployable solutions for customers — but also full responsibility for all components within the system"

Jean Bozman, Jed Scaramella, IDC Link

x86 Hardware for Oracle Solaris 10

Oracle Solaris on Sun x86 Servers

Oracle offers x86 clustered systems tightly integrated from applications to disk. Oracle uses best-of-breed components to simplify the deployment, management, and support of IT infrastructure and to deliver unmatched performance and scalability.

Oracle's x86 server family ranges from 2-socket to 8-socket systems that run your choice of operating system: Oracle Solaris, Oracle Enterprise Linux, Oracle VM, Red Hat Linux, SUSE Linux, Windows or VMware. These servers are designed to deliver peak performance while reducing energy costs and improving space efficiency with leading performance per watt and compute density.

I

Oracle x86 Cluster

Your Solution for Rapid Business Growth

- Engineered together
- Tested together
- Packaged together
- Certified together
- Deployed together
- Supported together
- Upgraded together

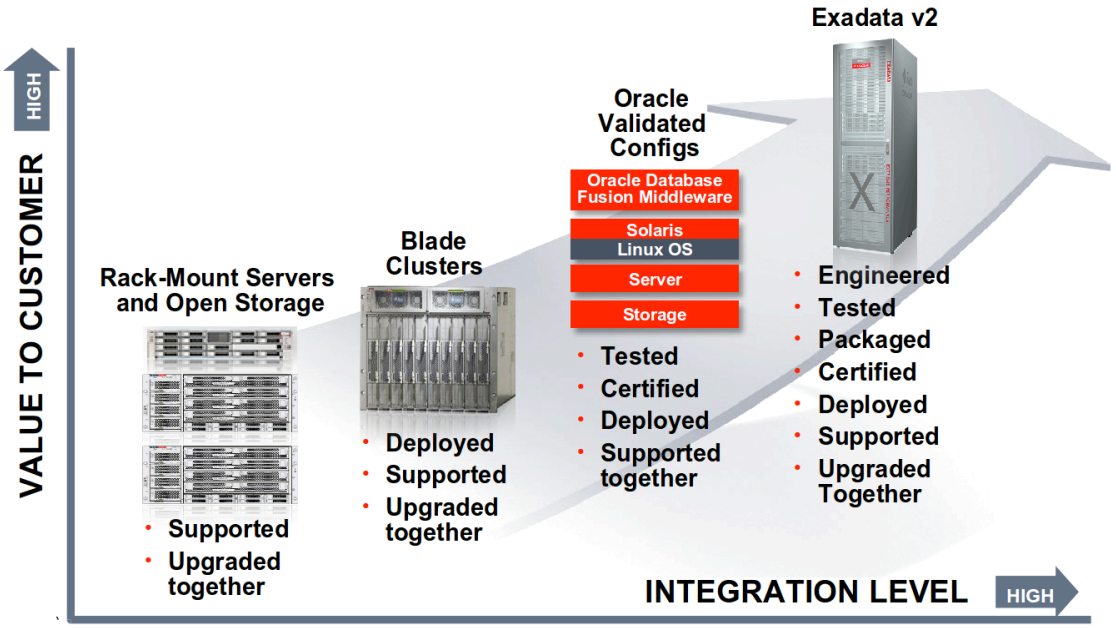
Oracle Virtualized eBusiness Suite Solution

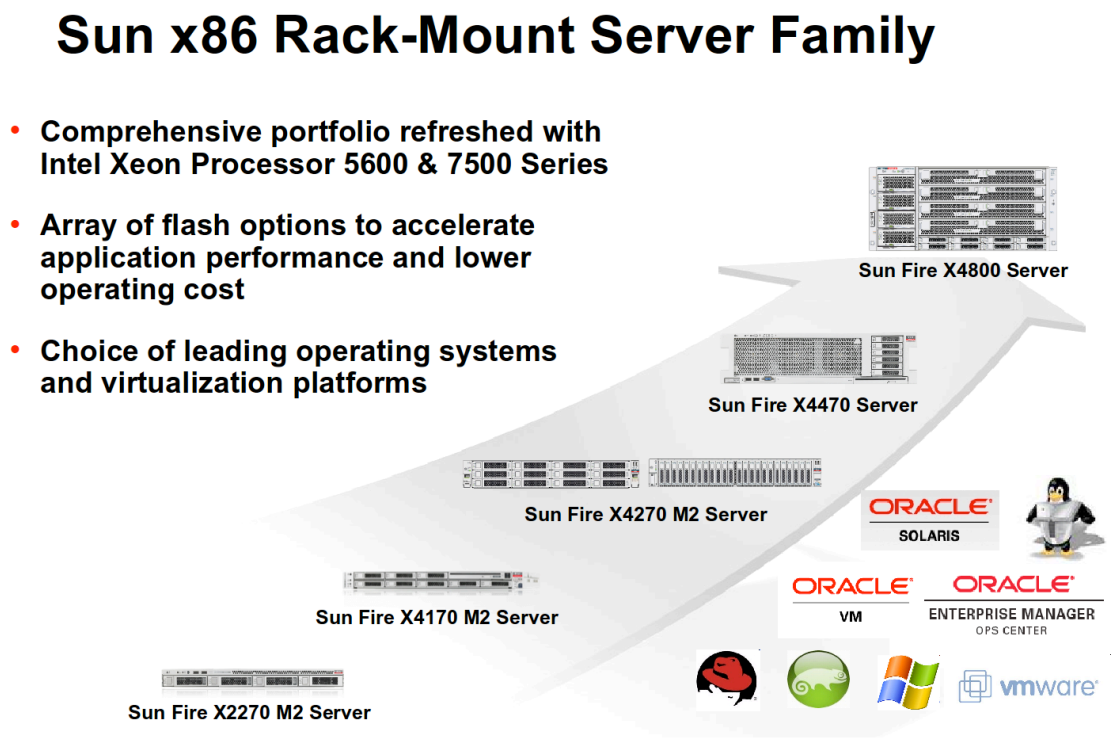
Oracle Exadata Database Machine Version 2

Reduce management complexity | **Performance and scale** | **Operational efficiencies**

Value of Oracle x86 Cluster

High Performing Application-to-Disk Solutions from a Single Vendor





Oracle Solaris x86 on third-party x86 hardware

As of August 2010, there are over 1250 x86 systems on the Solaris Hardware Compatibility List (HCL).[1] A free download of the Hardware Certification Test Suite (HCTS) is available, which enables partners and customers to test and self-certify their hardware platforms, to ensure they can be listed as certified systems on the HCL. Additionally Sun OEM partners, including Dell and HP, offer Solaris as an Operating System choice.

Oracle Solaris Premier Subscription for Non-Oracle Hardware provides Oracle support and the Oracle Solaris license for new and existing customers running Oracle Solaris across multiple x86 platforms. This offer ensures successful Oracle Solaris deployments by providing immediate access to Oracle's

1 "BigAdmin: Hardware Compatibility List (HCL) Overview," http://www.sun.com/bigadmin/hcl/overview.jsp.

deep product expertise and fast and accurate issue resolution—while limiting risk in your IT environment.[2]

Oracle Solaris 10: Optimized for the Intel® Xeon® Processor

Oracle and Intel, as part of a broad strategic alliance, have been working together to ensure that Oracle Solaris is optimized to unleash the power and capabilities of current and future Intel Xeon processors.[3] As a result, Oracle and Intel have made significant advances to optimize Oracle Solaris for Intel Xeon processor-based systems, and are working closely to develop new capabilities that are part of the Intel Xeon processor 5600 series and the Intel Xeon processor 7500 series. Some examples of this include:

- Scalable performance: Oracle Solaris enhances Intel® multicore processor capabilities—up to 8 processor sockets, each with up to 8 cores and 2 threads per core—as well as Intel® Hyper-Threading Technology, Intel® QuickPath Technology (Intel® QPI), and Intel® Turbo Boost Technology. Additional optimizations improve memory and network performance.
- Advanced reliability: The Oracle Solaris Fault Management Architecture (FMA) integrates with the Intel® Machine Check Architecture (MCA) Recovery features, enabling systems to automatically monitor, report, and recover from hardware errors to maintain data integrity and keep mission critical applications and services online.
- Power efficiency and utilization: Oracle Solaris takes advantage of performance-enhanced dynamic power management capabilities of the Intel Xeon processor 5600 and 7500 series.
- Cost-effective virtualization: Enhancing Oracle Solaris to take advantage of the latest Intel® Virtualization Technology (Intel VT) features enables the highest consolidation ratios.

Storage Options for Oracle Solaris x86

Oracle offers a full range of storage solutions, including flash storage arrays, the Sun Flash Accelerator PCIe card, disk arrays, and unified storage arrays.

Disk Storage

Oracle Sun Storage Fibre Channel Arrays provide mission-critical, tier-1 storage that scales in every dimension. Tailor performance profiles to site and application specific needs to achieve higher application performance while reducing power, space, and cooling requirements.

---

2 "Oracle Solaris Premier Subscription for Non-Oracle Hardware," http://www.oracle.com/us/products/servers-storage/solaris/non-sun-x86-081976.html.

3 "oracle-solaris-intel-xeon-bwp-065889.pdf," http://www.oracle.com/us/products/servers-storage/servers/x86/oracle-solaris-intel-xeon-bwp-065889.pdf.

RAID 0, 1, 3, 5, 6 (p+q), and 10 support matches performance and availability to any mix of application requirements. As indicated in the subsequent section on file system recommendations, RAID configurations provided by Oracle Sun Storage Fibre Channel Arrays are fully supported and recommended for use in conjunction with Oracle ASM and provides the best storage solution, in terms of both performance and reliability.

Flash Storage

Oracle provides the industry's most complete Flash-based storage portfolio, from high-performance arrays through Flash-optimized system software and databases, all with Oracle support.

Using Oracle Flash-based storage can reduce I/O service times up to 15x, and accelerate applications up to 2x using Oracle's Flash storage and Flash-optimized database and systems software.

Oracle provides two Flash storage options: the Sun Storage 5100 Flash Array and the Sun Flash Accelerator PCIe Card.

Oracle's Sun Storage F5100 Flash Array is the world's fastest and most power-efficient flash array for accelerating databases and I/O intensive applications. It redefines database performance, cutting transaction times in half and doubling application throughput. The Sun Storage F5100 Flash Array lets you scale your performance and capacity needs efficiently without impacting data availability so you can meet your growing business needs.

The Sun Flash Accelerator PCIe Card improves response times, and reduces I/O latency. Based on Oracle's Sun FlashFire technology, it delivers the I/O performance of over 300 disk drives to eliminate storage I/O bottlenecks and help your servers and applications run faster and more efficiently. In particular, the Sun Flash Accelerator PCIe card can be used in Flash Cache configurations, which are a new feature in Oracle Database 11gR2.

Unified Storage

Oracle Sun Unified Storage 7000 Series Systems combine an innovative storage architecture and file system with storage analytics and management technologies to deliver leadership performance and value.

Oracle Sun Unified Storage Systems simultaneously provide multiple storage interconnects (Gigabit Ethernet, 10GbE, Fibre Channel and InfiniBand), thereby consolidating storage for file and block I/O driven applications onto a single high-capacity, high-performance storage system with reduced administration and the industry's most comprehensive and intuitive analytics environment.

Oracle Sun Unified Storage Systems run the ZFS file system. As mentioned in the following section on file system recommendations, Oracle ZFS is recommended for Oracle Database binaries and fully certified for Oracle Database data files.

Storage analytic and storage management tools are included with Oracle Sun Unified Storage Systems, which enables an integrated view from Oracle Database to Oracle Sun Storage 7000 and provides monitoring of multiple storage systems from a single pane.

File System Recommendations

While the Oracle binaries can be installed on any supported file system, the recommended best practice for Oracle Database deployments on Oracle Solaris is to use Oracle Solaris ZFS for the Oracle Database binaries and Oracle ASM as an integrated volume manager for the Oracle Database data files.

However, if you choose to use Oracle Solaris ZFS for Oracle database deployment, which is certified for Oracle Database 10g and 11g , refer to the resources section for a list of collateral on best practices for using Oracle Solaris ZFS with Oracle Database.

Oracle Binaries – Oracle Solaris ZFS

Oracle Solaris Zettabyte File System (ZFS) technology offers a dramatic advancement in data management with a virtual storage pool design, integrated volume manager, and data services that provide an innovative approach to data integrity.

Oracle Solaris ZFS software enables more efficient and optimized use of storage devices, while dramatically increasing reliability and scalability. Physical storage can be dynamically added or removed from storage pools without interrupting services, providing new levels of flexibility, availability, and performance.

Oracle Solaris ZFS protects all data by 256-bit check sums, resulting in 99.99999999999999999-percent error detection and correction. Oracle Solaris ZFS constantly reads and checks data to help ensure it is correct, and if it detects an error in a storage pool with redundancy (protected with mirroring, Oracle Solaris ZFS RAIDZ, or Oracle Solaris ZFS RAIDZ2), Oracle Solaris ZFS automatically repairs the corrupt data. This contributes to continuous availability by helping to protect against costly and time-consuming data loss due to hardware or software failure, and by reducing the chance of administrator error when performing file system-related tasks.

Oracle Solaris ZFS software optimizes file system reliability by maintaining data redundancy on commodity hardware. It seamlessly and transparently supports new hybrid disk storage pools that include Flash technology for superior application performance.

Oracle Solaris ZFS can also be used to create snapshots and clones: snapshots are read-only, while clones are read-write copies of the file system.

In addition to typical disaster recovery and other business continuity purposes, Oracle Solaris ZFS snapshots are particularly useful as the Oracle Database environment is being built. In case of any configuration errors, the entire filesystem may be reverted back to its previous state.

Deployments to multiple target systems can be accelerated using Oracle Solaris ZFS cloning functionality. Instead of installing multiple systems from scratch, existing instances may be cloned in a fraction of the time.

Oracle Data Files – ASM

As mentioned above, the recommended practice for the Oracle Database data files is to use Oracle Automatic Storage Management (ASM) as an integrated volume manager.

Once disk sets are configured, Oracle ASM will begin to stripe data across all available disk sets. The striping method is configurable to three levels:

- Normal – Oracle ASM will maintain two copies of all data
- High Availability – Oracle ASM will maintain three copies of all data.
- External – Oracle ASM will maintain only one copy of all data. In this scenario, the external storage device is expected to mirror its own data.

Please note that data redundancy can be defined at a much more granular level (down to the file level). Advanced configurations such as this, are beyond the scope of this paper. For further details, please see the *Storage Administrators' Guide.*

For storage systems that do not provide hardware RAID redundancy, it is recommended to present all storage devices to Oracle ASM individually, without the use of third-party volume managers. Since Oracle ASM provides volume management in addition to a file system, including additional volume managers would be an unnecessary duplication of functionality.

For storage systems that provide hardware RAID, it is recommended to mirror the disks using hardware and set the redundancy level of the volume in ASM to "External".

- For maximum performance, the array should use RAID 10 and expose the disk set as one LUN.
- For more efficient disk usage (at a performance cost), RAID 5 is recommended. Again, each disk set would be exposed as one LUN.
- Also compliant with recommended practices, is the use of mirrored disk pairs. Using this method, each disk pair is mirrored and presented to Oracle ASM under its own LUN.

After creating disk groups, Oracle ASM will stipe data across all available disks, creating either one, two, or three copies of data, depending on availability settings.

Finally, it is recommended to separate data files and redo log files into two separate disk groups. Since redo log information has different properties (sequential reads and writes), significant performance gains may be possible.

Since Oracle Database 10g, ASM provides the database administrator with a simple storage management interface, which is consistent across all server and storage platforms. The Automatic Storage Management feature in Oracle Database 11g Release 2 extends ASM functionality to manage **all** data: Oracle database files, Oracle Clusterware files and non-structured general-purpose data such as binaries, externals files and text files.

Improving Performance with Flash Cache

For a database instance, the amount of data that is being actively accessed at any given time is referred to as the working set. For a large database instance, the working set may be small at times, for example, nights or weekends when the load is light and an individual user executes a query against an index and a small amount of data is returned. At other times, the working set of the same database instance may be large, for instance when queries result in full table scans. Optimal performance is obtained when the working set is resident in the buffer cache of the SGA and database performance will degrade as the working set outgrows the buffer cache such that data must be fetched from disk. When the working set

exceeds the buffer cache, Oracle Enterprise Manager tuning alternatives should be explored, for examples, the SQL Access Advisor may recommend indexes which will reduce the size of the working set. When tuning is not able to align the working set with size of the buffer cache, adding RAM and increasing the SGA_MAX_SIZE can result in substantial performance benefits, however RAM is relatively expensive. An affordable alternative to increasing the amount of RAM used by the SGA buffer cache is to use Flash Cache to supplement to the buffer cache.

Consider adding the flash cache when all of the following are true:[4]

- Your database is running on the Solaris or Oracle Enterprise Linux operating systems. The flash cache is supported on these operating systems only.
- The Buffer Pool Advisory section of your Automatic Workload Repository (AWR) report or STATSPACK report indicates that doubling the size of the buffer cache would be beneficial.
- db file sequential read is a top wait event.
- You have spare CPU.

The Oracle 11gR2 database offers a general mechanism to put a flash device to optimal use, by configuring the device as an extra level of database block buffer cache. The configuration is very straightforward: it is sufficient to specify two initialization parameters, one with the name of the flash disk and another one with its size.

*.db_flash_cache_file='/dev/rdsk/c7t55d0s0'

*.db_flash_cache_size=21474836480

The advantage of using flash technology in this way is that it has no impact on the existing setup and operations of a database. Sun Storage F5100 Flash Arrays and Sun Flash Accelerator F20 PCIe Cards are ideal devices for Flash Cache.

See *Oracle Database Administrator's Guide 11g Release 2 (11.2) "*Configuring Database Smart Flash Cache" for more information about Flash Cache tuning and sizing.

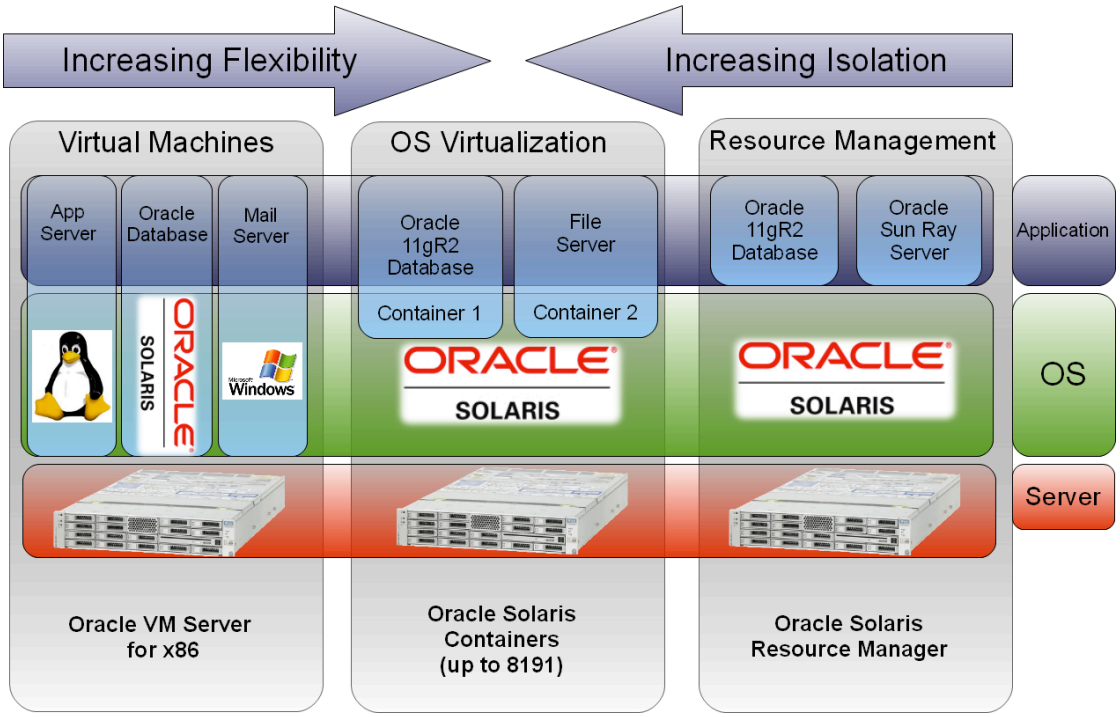
Virtualization Options for Oracle Solaris x86

Oracle Solaris x86 virtualization options exist to meet a broad spectrum of needs. On one end of that spectrum, hypervisor based virtualization provides complete isolation of virtual machines. On the other extreme, Oracle Solaris Resource Manager offers resource controls without the need to administer another operating system. In between these two extremes, Oracle Solaris Containers offer a

[4] "Configuring Database Smart Flash Cache," http://st-doc.us.oracle.com/11/112/server.112/e10595/memory005.htm.

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 Hypervisor-base virtualization, Oracle Solaris Containers, and Oracle Solaris Resource Manager:



Oracle VM Server for x86 and Hypervisor Based Virtualization

Oracle's Sun Fire x86 servers support all of the expected hypervisors: VMware, VirtualBox, Hyper-V and Oracle VM, and Oracle Solaris x86 is supported within the hypervisors. In contrast, Oracle Database is supported natively only on Oracle VM.

Oracle VM is free, next generation, scalable server virtualization software that supports Oracle and non-Oracle applications. Oracle VM provides an easy-to-use graphical interface for creating and managing virtual server pools running on x86-based systems across an enterprise; Solaris, Linux and Windows guests are supported. Part of Oracle VM includes the Xen hypervisor, and Oracle's engineering team contributes heavily to feature development of Xen mainline software.

Oracle VM delivers:

- Leading Performance – Low-overhead architecture with the Xen hypervisor provides scalable performance.

- Latest Hardware Support – Leverage the new hardware features from Intel Xeon and AMD Opteron processors for higher performance and more efficient power management.
- Faster Software Deployment with Oracle VM Templates – Download and import pre-configured virtual machines containing pre-installed Oracle enterprise applications or other software to get up and running quickly.
- Rapid VM Provisioning and Cloning – Sparse file support in OCFS2 enables significantly faster virtual machine provisioning and cloning; allow users more control over data allocation, improving performance and storage efficiency.
- Oracle VM Manager Command Line Interface (CLI) and Web Services API – Enable integration of third party products with Oracle VM Manager.
- Secure Live VM Migration – Completely eliminate service outages associated with planned maintenance or scale up your resources quickly by migrating running VMs to other servers over secure SSL links without interruption.
- High Availability – Reliably and automatically restart failed VMs on other servers in the server pool after unexpected server- or individual VM outage. New, server pool master auto-failover feature eliminates any single point of failure for virtualization management.
- Automatic or Manual Server Pool Load Balancing – Guest VMs are automatically placed on the server with the most resources available in the pool at start-up, or can be started within a user-designated subset of servers.
- Physical-to-Virtual / Virtual-to-Virtual Machine Conversion – Quickly convert existing Linux or Windows physical servers or VMDK or VHD virtual machines to Oracle VM virtual machines to reduce license expenses.
- Virtual CPU Scheduling Priorities and Caps per VM – Control access to CPU between multiple VMs to align with IT/business priorities.
- Virtual Machine I/O Resource Management – Set bandwidth cap for each virtual network interface and prioritize the virtual disks.
- Importing Block Devices as Shared Virtual Disks – Block devices can be assigned to VMs, greatly improving performance of applications such as Oracle Real Application Cluster (RAC).
- Solaris, Linux and Windows Support – Run Solaris, Linux and Windows guest operating systems on Oracle VM.
- Official Certification Based On Real-World Testing – Supported for use with the most sophisticated enterprise workloads under real-world conditions.
- Virtualization and Management: Zero License Costs, Zero License Keys – Include Oracle VM Manager for centralized, browser-based management of your resource pools.
- Affordable, Full-Stack Enterprise-Class Support – Worldwide support from Oracle for the entire virtualization environment and workloads together.

Oracle Solaris Containers

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

Operating System virtualization with Oracle Solaris Containers allows you to maintain the one-application-per-server deployment model while simultaneously sharing hardware resources. An integral part of the Oracle Solaris 10 Operating System, Oracle Solaris Containers isolate software applications and services using flexible, software-defined boundaries and allow many private execution environments to be created within a single instance of Oracle Solaris 10. 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 containers can be created for a single operating system image. Each container has a very small amount of 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. Also, fractions of CPUs can be used as well as whole CPUs, and memory can also be specified in small amounts such as megabytes.

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.

Oracle Solaris Containers are ideal for the consolidation of environments. With the increasing cost and complexity of managing many separate systems, it is often advantageous to consolidate multiple applications onto larger, more scalable servers. Oracle Solaris Containers provide 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.

Oracle Solaris Containers run fine inside a hypervisor: OK to mix and match.

The licensing model of Oracle 11gR2 Database allows CPU capping via Oracle Solaris Containers to contain costs. Please contact your local Oracle sales partner for further details.

If CPU resource distribution needs to be controlled with finer granularity than a CPU thread, 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 information about Oracle Solaris Containers please see *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones* on http://docs.sun.com.

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 ability, as with Oracle Solaris Containers, enables the consolidation of multiple applications onto a single server thus improving the resource utilization and lowering the Total Cost of Ownership (TCO). Control of CPU, Memory, and other resources is as fine grained as with Oracle Solaris Containers, but can be applied to users, or projects rather than an entire container.

To guarantee predictable service levels, 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 Solaris Resource Manager, system administrators can define workloads, and partition and allocate system resources to different entities in such a way that pre-defined Service Level Agreements are met while maintaining the overall quality of service and keeping the system resources busy. In addition, 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. 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*.

Oracle RAC on Oracle Solaris x86

Oracle Database and Oracle RAC enable the creation of horizontally scaled, high-performance, highly available shared databases on clusters of cost-effective industry-standard servers, helping increase return on investment and total cost of ownership.

Oracle RAC is a cluster database with a shared architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches is to provide highly scalable and available solutions for all your business applications. Oracle RAC utilizes Oracle Clusterware for the inter-node communication required in the clustering database environments. Oracle Clusterware is the technology that transforms a server farm into a cluster. A cluster, in general, is a group of independent servers that will cooperate as a single system. Oracle Clusterware is the intelligence in this system that ensures the required cooperation, and is a key component of the Oracle enterprise grid architecture is well.

The cluster interconnect can have a major effect on Oracle RAC performance. In Oracle RAC systems with interconnect intensive workloads, Gigabit Ethernet can often become a bottleneck for high-volume cluster messaging and Oracle Cache Fusion traffic between nodes for many applications. InfiniBand, in contrast, can provide significant advantage in both raw bandwidth and reduced latency compared with Gigabit Ethernet, and typically can provide higher performance than Gigabit Ethernet for Oracle RAC systems

In Oracle RAC technology, Cluster Interconnect maximizes performance with the RDS protocol and InfiniBand networking. Reliable Datagram Sockets (RDS) is a protocol providing reliable datagram services, multiplexing UDP packets over an InfiniBand connection, improving performance to Oracle RAC. InfiniBand networking has a 10X bandwidth advantage and 10X latency reduction versus Gigabit Ethernet.

The combination of Oracle RAC software, RDS, and Oracle Solaris on Sun x86 servers is particularly well suited to the demanding task of clustering. Oracle Sun x86 servers based on multi-core AMD Opteron and Intel Xeon processors are designed particularly to provide the speed, savings, and simplicity for data center optimization.

Tools, Process Monitoring, and Observability

Oracle Enterprise Manager

When monitoring Oracle Database 11g performance on Solaris 10 for x86, the Oracle Enterprise Manager (Enterprise Manager) provides critical feedback with respect to the utilization of system resources. Enterprise Manager is a system management tool that provides centralized management of a database environment. Combining a graphical console, Oracle Management Servers, Oracle Intelligent Agents, common services, and administrative tools, Enterprise Manager provides a comprehensive systems management platform for Oracle products.

The Web-based Enterprise Manager Database Control (Database Control) is the primary tool for managing an Oracle database. It is installed with Oracle Database. You can use Database Control to perform administrative tasks such as:

- Diagnosing, modifying, and tuning the database
- Grouping related targets together to facilitate administration tasks, sharing tasks with other administrators, and scheduling tasks at varying time intervals
- Configuring and managing Oracle Net Services for an Oracle home
- Launching integrated Oracle and third-party tools

Memory Management

Memory management involves maintaining optimal sizes for the Oracle instance memory structures as demands on the database change. Initialization parameter settings determine how SGA and instance PGA memory is managed.

Users may choose to use automatic memory management for both the SGA and PGA, or may choose to use manually manager either or both.

In automatic memory management, Oracle Database manages the SGA and instance PGA memory completely automatically. This method is the simplest and is strongly recommended by Oracle.

The only user-specified controls are the target memory size initialization parameter (MEMORY_TARGET) and optional maximum memory size initialization parameter (`MEMORY_MAX_TARGET`). Oracle Database tunes to the target memory size, redistributing memory as needed between the SGA and the instance PGA. Using automatic memory management, the database automatically adjusts the size of the large pool and database buffer cache depending on which type of jobs are running. If you create your database with DBCA and choose the basic installation option, then automatic memory management is enabled by default.

Performance Diagnostics and Tuning

Oracle database performance problems typically result in unacceptable response times, which are the time to complete specified workloads, or unacceptable throughput, which is the amount of work that can be completed in a specified time. Common problems include:

- CPU bottlenecks
- Undersized memory structures
- I/O capacity issues
- Inefficient or high-load SQL statements
- Unexpected performance regression after tuning SQL statements
- Concurrency and contention issues
- Database configuration issues

The general goal of tuning is usually to improve response time, increase throughput, or both. In general, tuning is the effort to achieve specific, measurable, and achievable tuning goals by using database resources in the most efficient way possible.

The Oracle performance method is based on identifying and eliminating bottlenecks in the database, and developing efficient SQL statements. Applying the Oracle performance method involves the following tasks:

- Performing pre-tuning preparations
- Tuning the database proactively on a regular basis
- Tuning the database reactively when users report performance problems
- Identifying, tuning, and optimizing high-load SQL statements

Oracle Database performance tuning requires the use of advisors. Oracle Database advisors provide specific advice on how to address key database management challenges, covering a wide range of areas including space, performance, and undo management.

Automatic Workload Repository (AWR)

Automatic Workload Repository (AWR) is a repository of historical performance data that includes cumulative statistics for the system, sessions, individual SQL statements, segments, and services. These statistics are the foundation of performance tuning. By automating the gathering of database statistics for problem detection and tuning, AWR serves as the foundation for database self-management.

The database stores recent AWR statistics in the SGA. By default, the MMON process gathers statistics every hour and creates an AWR snapshot (see "Manageability Monitor Processes (MMON and MMNL)"). A snapshot is a set of performance statistics captured at a specific time. The database writes snapshots to the SYSAUX tablespace. AWR manages snapshot space, purging older snapshots according to a configurable snapshot retention policy.

An AWR baseline is collection of statistic rates usually taken over a period when the system is performing well at peak load. Specify a pair or range of AWR snapshots as a baseline. To diagnose problems, use an AWR report to compare statistics captured during a period of bad performance to a baseline.

An automated maintenance infrastructure known as AutoTask illustrates how Oracle Database uses AWR for self-management. By analyzing AWR data, AutoTask can determine the need for maintenance tasks and schedule them to run in Oracle Scheduler maintenance windows. Examples of tasks include gathering statistics for the optimizer and running the Automatic Segment Advisor.

Automatic Database Diagnostic Monitor (ADDM)

Automatic Database Diagnostic Monitor (ADDM) is a self-diagnostic advisor built into Oracle Database. Using statistics captured in AWR, ADDM automatically and proactively diagnoses database performance and determines how identified problems can be resolved. DBAs can also run ADDM manually.

ADDM takes a holistic approach to system performance, using time as a common currency between components. ADDM identifies areas of Oracle Database consuming the most time. For example, the database may be spending an excessive amount of time waiting for free database buffers. ADDM drills down to identify the root cause of problems, rather than just the symptoms, and reports the effect of the problem on Oracle Database overall. Minimal overhead occurs during the diagnostic process.

In many cases, ADDM recommends solutions and quantifies expected performance benefits. For example, ADDM may recommend changes to hardware, database configuration, database schema, or applications. If a recommendation is made, then ADDM reports the time benefit. The use of time as a measure enables comparisons of problems or recommendations.

Besides reporting potential performance issues, ADDM documents areas of the database that are not problems. Subcomponents such as I/O and memory that are not significantly impacting database performance are pruned from the classification tree at an early stage. ADDM lists these subcomponents so that you can quickly see that there is little benefit to performing actions in those areas.

Active Session History (ASH)

Active Session History (ASH) samples active database sessions each second, writing the data to memory and persistent storage. ASH is an integral part of the database self-management framework and is useful for diagnosing performance problems.

Unlike instance-level statistics gathered by AWR, ASH statistics are gathered at the session level. An active session is a session that is using CPU and is not waiting for an event in the idle wait class.

You can use Enterprise Manager or SQL scripts to generate ASH reports that gather session statistics gathered over a specified duration. You can use ASH reports for:

- Analysis of short-lived performance problems not identified by ADDM
- Scoped or targeted performance analysis by various dimensions or their combinations, such as time, session, module, action, or SQL ID

For example, a user reports that the database was slow between 10:00 p.m. and 10:02 p.m, but the two minute performance degradation represents a small portion of the AWR snapshot interval from 10:00 p.m. and 11:00 p.m. and does not appear in ADDM findings. ASH reports can help identify the source of the transient problem.

Application and SQL Tuning

Oracle Database completely automates the SQL tuning process. ADDM identifies SQL statements consuming unusually high system resources and therefore causing performance problems. In addition, AWR automatically captures the top SQL statements in terms of CPU and shared memory consumption. The identification of high-load SQL statements happens automatically and requires no intervention.

SQL Tuning Advisor

Automatic SQL tuning is exposed through SQL Tuning Advisor. SQL Tuning Advisor runs automatically during system maintenance windows as a maintenance task. During each automatic run, the advisor selects high-load SQL queries in the database and generates recommendations for tuning these queries.

SQL Tuning Advisor recommendations fall into the following categories:

- Statistics analysis
- SQL profiling
- Access path analysis
- SQL structure analysis

A SQL profile contains additional statistics specific to a SQL statement and enables the optimizer to generate a better execution plan. Essentially, a SQL profile is a method for analyzing a query. Both access path and SQL structure analysis are useful for tuning an application under development or a homegrown production application.

A principal benefit of SQL Tuning Advisor is that solutions come from the optimizer rather than external tools, thus, tuning is performed by the database component that is responsible for the execution plans and SQL performance. The tuning process can consider past execution statistics of a SQL statement and customizes the optimizer settings for this statement.

SQL Access Advisor

SQL Access Advisor offers advice on how to optimize data access paths. Specifically, it recommends how database performance can be improved through partitioning, materialized views, indexes, and materialized view logs.

Schema objects such as partitions and indexes are essential for optimizing complex, data-intensive queries. However, creation and maintenance of these objects can be time-consuming, and space requirements can be significant. SQL Access Advisor helps meet performance goals by recommending data structures for a specified workload.

The SQL Access Advisor can be run from Enterprise Manager using the SQL Access Advisor Wizard or by invoking the DBMS_ADVISOR package. The DBMS_ADVISOR package consists of a collection of analysis and advisory functions and procedures callable from any PL/SQL program.

Oracle Solaris Tuning and Observability Tools

SOLARIS 10 PERFORMANCE OBSERVABILITY AND DEBUGGING TOOLS

GROUP	SOLARIS COMMANDS
System Statistics	1. acctcom – process accounting 2. busstat – Bus hardware counters 3. cpustat – CPU hardware counters

	4. iostat – IO & NFS statistics
	5. kstat – display kernel statistics
	6. mpstat – processor statistics
	7. netstat – network statistics
	8. nfsstat – nfs server stats
	9. sar – kitchen sink utility
	10. vmstat – virtual memory stats
Process Statistics	11. cputrack - per-processor hw counters
	12. pargs – process arguments
	13. pflags – process flags
	14. pcred – process credentials
	15. pldd – process's library dependencies
	16. plockstat – process lock statistics
	17. psig – process signal disposition
	18. pstack – process stack dump
	19. pmap – process memory map
	20. pfiles – open files and names
	21. prstat – process statistics
	22. ptree – process tree
	23. ptime – process microstate times
	24. pwdx – process working directory
Toolkits	25. DTraceToolkit – DTrace Tools
	26. K9Toolkit – perl perf tools
	27. nicstat – network stats
Process Control	28. pgrep – grep for processes
	29. pkill – kill processes list
	30. pstop – stop processes
	31. prun – start processes
	32. prctl – view/set process resources
	33. pwait – wait for process
	34. preap – reap a zombie process Table text Table text
Kernel Tracing/ debugging	35. dtrace – trace and monitor kernel
	36. lockstat – monitor locking statistics
	37. lockstat -k – profile kernel
	38. mdb – debug live and kernel cores
Process Tracing	39. abitrace – trace ABI interfaces
	40. dtrace – trace the world

41. mdb – debug/control processes
42. truss – trace functions and system calls

Monitoring the Solaris run queue

If a server has more requests for processing than compute cycles, processes are scheduled in the run queue. A large run queue indicates that the need to find more compute cycles (i.e. allocate or obtain more or faster CPU's) or reduce the workload (i.e. SQL tuning). You see the current run queue length with vmstat. Watch the "r" column.

Virtual memory pressure

Solaris will try to use all of a server's RAM efficiently, for example, I/O that is buffered with RAM can be accessed more efficiently than issuing physical I/O operations. As a result, a running system will typically have little free memory because available memory will often have been allocated to I/O buffering. To identify a system with insufficient RAM for the jobs that are being executed, use vmstat and watch the "sr" scan rate column. The scan rate should be at or close to zero. If the scanner kicks in for a short time but returns to zero, virtual memory pressure is not having a significant impact on your performance. If the system is always scanning, you need to kill non-critical processes, reduce the size of your SGA, or add more RAM to the system.

DTrace

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. Oracle administrators can use DTrace probes, in conjunction with Oracle's AWR report, to quickly understand and resolve performance issues on the Oracle Solaris 10 platform.

Free Software for Oracle Solaris 10

The Oracle Solaris 10 Operating System is complemented by a collection of popular, freely available software. This freeware includes in-bound open source and third party components such as utilities, productivity tools, and development tools. Users can now more conveniently leverage the technologies shared on the Internet.

There are two primary sources of freeware that work with the Solaris 10 Operating System:

- Freeware that is included on the Solaris 10 CD in separate and distinct modules, which is being made available as a convenience to our customers
  - technologies that users may expect to find with their operating environment are now included with the Solaris environment
- Freeware that is co-packaged via the Solaris 10 Companion CD
  - other useful and popular technologies are offered as an unsupported value-add CD
  - See Solaris 10 OS Companion Software CD downloads page (URL in Reference, below) for more details.

Proven Performance and Scalability

The Sun Fire x86 Clustered Systems have achieved 14 outstanding world-record results on multiple industry standard benchmarks and prominent application workloads.[5] The Oracle database has a proven track record of scaling well both vertically as well as horizontally on the Oracle Solaris 10 platform. Oracle believes in empowering its customers to use both horizontal and vertical scalability dimensions to best meet their critical performance and availability criteria.

---

5 "Sun x86 Systems | Extreme Scalability, World-class Performance, Energy-efficient | Oracle," http://www.oracle.com/us/solutions/performance-scalability/sun-x86-systems-gateway-082396.html.

Business applications deployed on both single and multiple-instances Oracle databases have consistently demonstrated exceptional performance and scalability running online as well as batch based workloads on the Oracle Solaris 10 platform. For instance, the SAP ERP 6.0 2-tier Sales and Distribution benchmark demonstrated near linear scalability of Oracle Real Application Cluster 10g in an SAP environment when deployed on a four node Sun Blade X6270 cluster running Oracle Solaris 10.

The Oracle database deployment on the Oracle Solaris 10 platform provides customers the flexibility to select the scaling method that best suites the business systems that they are implementing, from scaling out across smaller building blocks, to scaling up on a large SMP configuration and leveraging the Oracle Solaris Containers features to "scale within". It also enhances application performance and scalability on OLTP and batch workloads across x86 systems.

Protect against faults: Enhance uptime

The Oracle Solaris Operating System provides a proven architecture for building and deploying systems and services capable of Predictive Self Healing, which is a cohesive architecture and methodology for automatically diagnosing, reporting and handling software and hardware fault conditions, thereby enhancing the systems availability. Solaris Fault Manager and Solaris Service Management facility (SMF) are the two key components of Predictive Self Healing technology. The following section describes how an Oracle database deployment can take advantage of Oracle Solaris Predictive Self Healing technology and can continue uninterrupted even when there are hardware and software fault conditions.

Protect against hardware faults: Solaris Fault Manager

Solaris Fault Manager monitors data relating to hardware errors and automatically diagnoses the underlying problem. Once diagnosed, Solaris Fault Manager automatically responds by off-lining faulty components such as a CPU, memory region or I/O channel. The net benefit is that the system continues to operate with the remaining system resources, achieving a graceful degradation rather than an undesired disruption of the entire system.

Figure 2 demonstrates the fault management architecture in a simplistic manner. The fault management architecture is divided into three areas: error handlers, diagnosis engines and agents. A fault or defect in hardware is associated with a set of observed symptoms called errors. The error events are dispatched to software components called diagnosis engines designed to diagnose the underlying problems corresponding to those symptoms. The diagnosis engine then produces fault event that is broadcast to any agents deployed on the system that know how to respond to that particular fault.

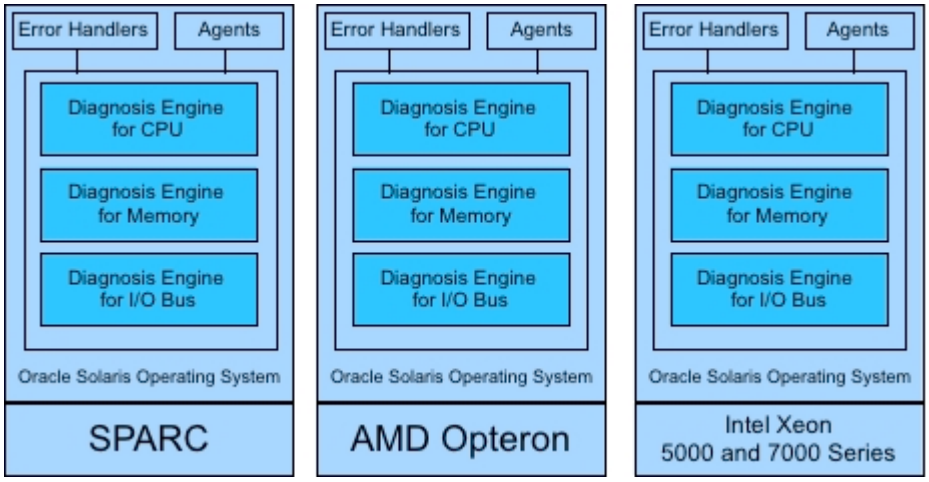


Figure 2: Fault Management Architecture defines hardware specific diagnosis engines.

The Oracle Solaris Operating System has implemented diagnostic engines for CPU, memory, and I/O bus nexus components for a variety of hardware platforms incorporating SPARC, AMD Opteron and Intel Xeon 5000 series and 7000 series processors, exploiting the specific hardware reliability, availability, and serviceability (RAS) features provided by the underlying system.

The Oracle Solaris Operating System running on Intel's Xeon 5000 series and 7000 series processor based system provides diagnosis engines that are completely integrated with Intel's Machine Check Architecture (MCA). Intel's MCA recovery enables the system to detect and correct errors in memory and cache that were previously "uncorrectable" through ECC or other means. MCA accomplishes this by first detecting and containing errors before the data is consumed by an application, then works in conjunction with Solaris to determine the best course of action to keep the system and application running. This advanced recovery capability means that systems based on the Intel Nehalem processor running the Oracle Solaris Operating System will be able to recover and remain running in situations where other x86-based systems would not. Hence, an Oracle database deployment on x86 platforms running Oracle Solaris 10 will provide correct diagnosis and recovery should a hardware fault occur since Solaris Fault Manager has specialized diagnosis engines for specific processor families.

Protect against memory faults: Memory Page Retirement

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. The MPR technique allows memory pages suffering from correctable errors and relocatable clean pages suffering from uncorrectable errors to be removed from use in the virtual memory system without interrupting user applications. It also allows relocatable dirty pages associated with uncorrectable errors to be isolated with limited impact on affected user processes, avoiding an outage for the entire system.

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 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 uncorrectable memory. If the Oracle database has memory pages that contain CEs then 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 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, even in this scenario, if Oracle is configured with Service Management Facility, as explained in the next section, it can restart automatically.

Protect against software faults: Service Management Facility

Service Management Facility is a core part of the Oracle Solaris Predictive Self Healing technology, which provides automatic recovery from software failures as well as administrative errors. With SMF, system administrators can use simple command line utilities to easily identify, observe, and manage both the services provided by the system and the system itself.

A Solaris service is any long-lived software object with a well-defined state, start and stop, and relationship to other services on the system. In Oracle Solaris 10, each software service has an advertised state. Should a failure occur, the system automatically diagnoses it and locates/pinpoints the source of the failure. Failing services are automatically restarted whenever possible, reducing the need for human intervention. Should manual intervention be required, system administrators can quickly identify the root cause of the service's failure and significantly reduce the times-to-repair and recover from said failure.

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

- If the Oracle database service comes down for any reason including administrator error, software error or uncorrectable hardware error, it will be automatically restarted in dependency order.
- If any service from dependency order fails, the Oracle database service will gracefully come down and a complete explanation of why a service isn't running, as well as individual, persistent log files for each service will be available for debugging purposes.
- The task of managing the Oracle services can be delegated to Oracle administrators; SMF is integrated with Solaris RBAC which ensures that the services can be securely managed by non-root users, including the ability to configure, start, stop, or restart services.

Configure Oracle as a service in Service Management Facility

This section describes the steps required for adding Oracle database as an SMF service so it can be automatically restarted in case of any type of failure.

- Create a service manifest file *oracledatabase.xml* in /var/svc/manifest/application/database directory.

- You need to create the directory if it doesn't exist and have the appropriate privileges to perform this action. Appendix B has a sample *oracleDatabase.xml* file that you can tailor to your environment.

- Create a methods script file to define how to start and stop this service.
  - Create a shell script *oracledb* in /lib/svc/method directory and change its permission to 555. This script will have methods to start and stop the Oracle database.
- Validate and import the manifest file into the Solaris service repository to create the service in SMF by issuing the following command
  - svccfg validate /var/svc/manifest/application/database/oracledatabase.xml
  - svccfg import /var/svc/manifest/application/database/oracledatabase.xml

- Enable the service using the following svcadm command.
  - svcadm enable svc:/application/database/oracle
- Verify that the service is online
  - svcs -a | grep oracle
- Monitor and troubleshoot the service
  - You can monitor the log file of this service at /var/svc/log/application-database-oracle:default.log.
  - If the service is in maintenance mode (invoke svcs –x command to list failing services), look at the log file to find the cause. Once you resolve the error, clear the maintenance flag on the service by issuing the following command :

```
# svcs clear /application/database/oracle
```

Table 1 shows all the files associated with the Oracle Solaris SMF service

SMF	FILE LOCATION
Service Identifier (FMRI)	Svc:/application/database/oracle
Service Log	/var/svc/log
Service Manifest	/var/svc/manifest/application/database/oracledatabase.xml
Service Start Method	/lib/svc/method/oracledb

Enhance out-of-box accountability

The Oracle Solaris 10 Operating System provides security features previously only found in Sun's military-grade Trusted Solaris OS. User and Process Rights Management work in conjunction with Oracle Solaris Containers to let you securely host thousands of applications and multiple customers on the same system. Solaris Trusted Extensions is a standard part of Oracle Solaris and allows customers who have specific regulatory or information protection requirements to take advantage of labeling features previously only available in highly specialized operating systems or appliances.

Oracle Solaris provides two resources for auditing: BART (Basic Audit Reporting Tool) and BSM (Basic Security Module). Solaris BSM, when enabled, creates an audit trail for specified users. BART is a file tracking tool that operates entirely at the file system level. BART gives you the ability to quickly, easily, and reliably gather information about the components of the software stack that is installed on deployed systems. Refer to reference section for a list of collateral that describes Oracle Solaris auditing in greater detail.

The following section explains how an Oracle Database installation can be made more secure with enhanced accountability by exploiting the user rights management feature of Oracle Solaris 10.

Track activities of individual DBAs

User rights management reduces security risks by providing privileged users only the capabilities needed to run a select number of commands consistent with their needs rather than granting full super-user access to the system. This increases security by reducing the chances of administrative errors or accidental/malicious use of systems. User rights management, based on Oracle Solaris Role-Based Access Control (RBAC) capabilities, is centrally managed for reduced administration cost and increased flexibility for rapidly changing business requirements. Effective security reduces downtime, raises quality of service, and keeps costs low.

In RBAC, roles are assigned to users. When a user assumes a role, the capabilities of the role are available. Roles get their capabilities from rights profiles. Rights profiles can contain authorizations, privileged commands, and other supplementary rights profiles. Privileged commands are commands that execute with security attributes.

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 DBAs login as the 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 users 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.

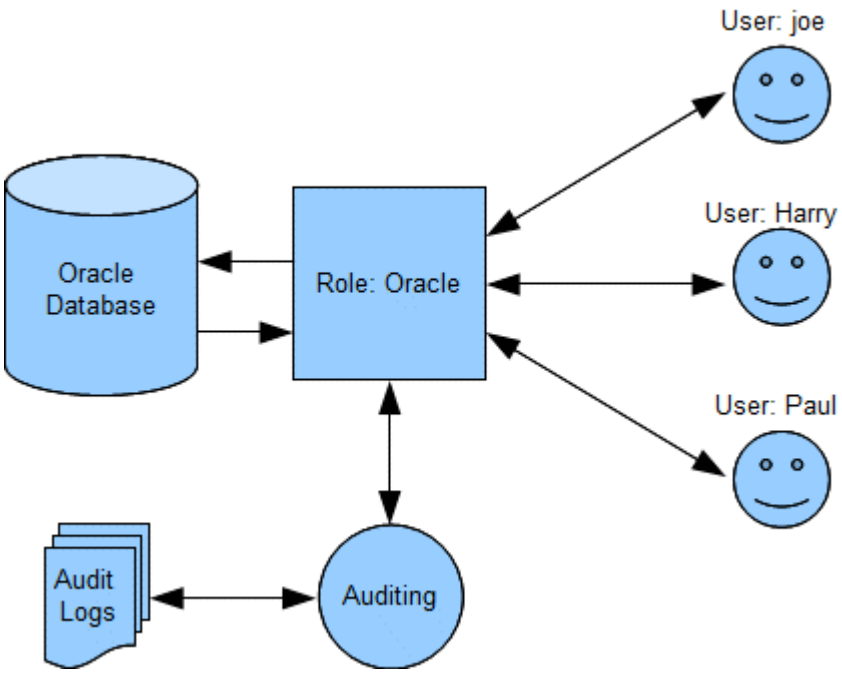


Figure 4 : Using *Oracle* role enhances security and accountability

If additional security is required, the privileges of the UNIX user can be adjusted such that individual UNIX users cannot view Oracle processes. Similarly, the privileges of the *Oracle* role can be adjusted such that they can view only the Oracle processes.

Create an Oracle role

The pre-requisite for creating an *Oracle* role is to define a rights profile for the *Oracle* role, which will define the capabilities of this role. An Oracle administrator would need access to all commands under the $ORACLE_HOME/bin directory. He would need access to commands found in the /usr/bin and /usr/sbin directories. An Oracle database administrator would additionally need authorization to manage Oracle database and listener SMF services, if they exist.

Figure 5 illustrates the relationship between an *Oracle* role and an Oracle database administration rights profile. The *Oracle* role would have the permissions for all the executables under $ORACLE_HOME as well as executables under /usr/bin and usr/sbin. Additionally, it will have the authority to manage Oracle SMF services.

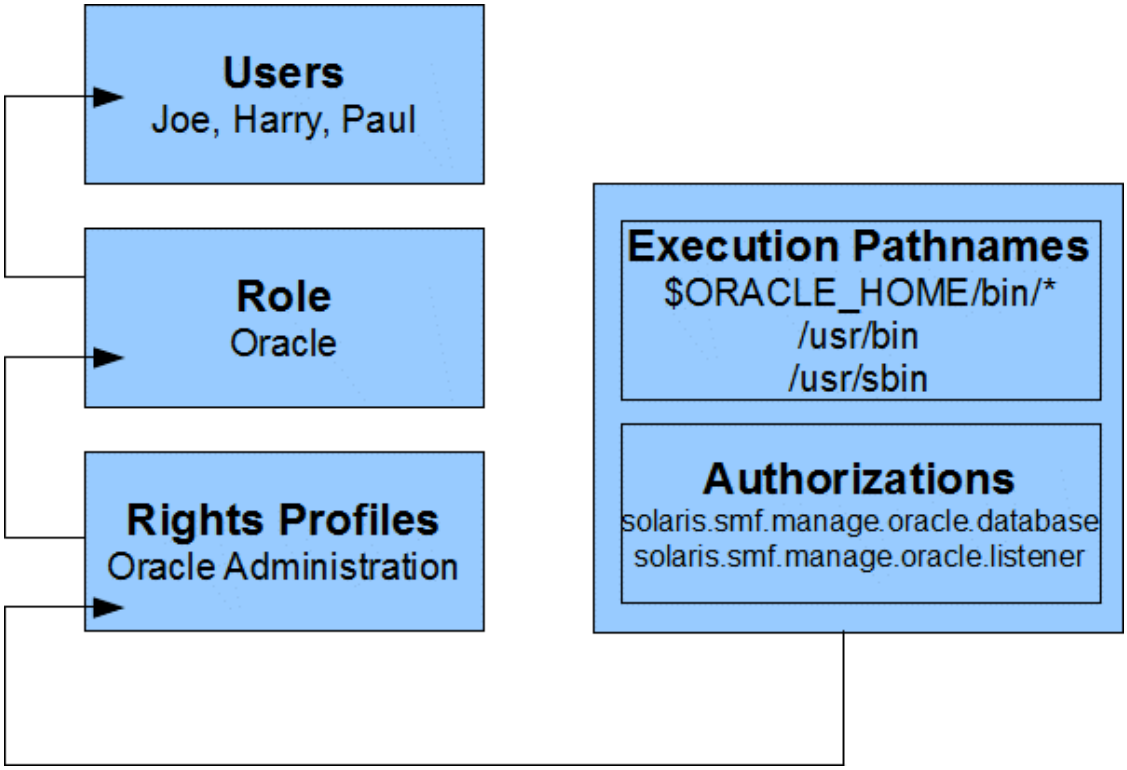


Figure 5 Oracle role has permissions to access Oracle database deployment and control Oracle SMF services

Creating an Oracle role is a two step process, the first step is to create an Oracle database administration rights profile and the second step is to create the role and assign it the Oracle administration rights profile.

Step 1: Create Oracle Administration rights profile

- Start the Solaris Management Console (smc) as Superuser :
  - %/usr/sadm/bin/smc &
- Click on the 'This Computer' icon in the Navigation pane
- Click on System Configuration->Users->Rights
- Click Action->Add Rights. The Add Rights wizard opens.
- Create the Oracle Administration rights profile with the Add Rights wizard by entering the following information in the wizard:

TABLE 2. CREATE ORACLE ADMINSTRATOR RIGHTS PROFILE

TAB	FIELD	VALUE

General	Name	Oracle Administrator
	Description	Rights profile for Oracle DBAs
Commands	Add Directory	Click Add Directory, type $ORACLE_HOME/bin in the dialog box and click OK
	Commands Denied/ Commands Permitted	Move $ORACE_HOME/bin to the Commands permitted column
	Set Security Attributes	Select , click Set Security Attributes and set Effective UID=oracle
Authorizations	Authorizations Excluded/ Authorizations Included	Select Oracle SMF authorization, if configured. Refer to Appendix B for details
Supplementary Rights	Rights Excluded/ Rights Included	No Supplementary rights profiles.

Step 2 Create an Oracle role and associate Oracle administration rights profile with this role

- Start the Solaris Management Console (smc) as superuser
  - /usr/sadm/bin/smc &
- Click on the 'This Computer' icon in the navigation pane
- Click on System Configuration->Users->Administrative Roles
- Click Action->Add Administrative Role. The Add Administrative Role wizard opens.
- Create the Oracle role with the Administrative Role wizard by following these steps
  - Set the role name to Oracle, full role name to Oracle DBA role Description to Role for Oracle DBA. Click Next
  - Set and confirm the role password. Click Next.
  - Select the Oracle Administrator rights profile from the Available Rights column and add it to Granted Rights column. Click Next
  - Add UNIX logins of all Oracle DBAs to the list of users who can assume this role.

Simplify 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. 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 reasonable defaults enabling Oracle database instances to run out-of-the-box without requiring special parameters to be set.

Oracle 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.

Create a project for Oracle Database Installation

By default, the Oracle Solaris OS provides all workloads running on the system equal access to all system resources. Oracle Solaris uses projects facility to identify a workload. Every user in the Oracle Solaris OS system is assigned a default project. Users cannot login to the system unless they are associated with a project. Oracle Solaris 10 provides a resource control facility to set resource limits for projects. The resource control facility provides project wide resource controls to define Oracle Solaris kernel's interprocess communication (IPC) facilities. These resource controls replace the /etc/system tunables and can be set dynamically.

In order to set the shared memory to more than 25% of the system, you need to create a project, assign it to the *oracle* user and set the max-shm-memory resource control to the desired value.

The following command creates a project named oracle, assigns it to user *oracle* and group *dba* and sets max-shm-memory resource to 10 gigabytes:

```
$  projadd -U oracle -G dba -K 'project.max-shm-memory= (privileged, 10G, deny)' oracle
```

You can optionally set the project id with the -p option and comment with the -c option

```
$ projmod -p 100 -c "Project for Oracle database deployment' oracle
```

If an Oracle database is deployed on a non-global zone on Oracle Solaris 10 8/07 update, the System V IPC resource controls are added zone -wide. Hence, these resources can be set during the process of creating the zone or altered on a zone wide basis; there is no need to create a project to set System V IPC variables on a non-global zone.

Consolidate multiple Oracle Database instances

Oracle Solaris Containers

Oracle Solaris Containers, Oracle's operating system level virtualization technology, provide complete, isolated, and secure run time environments for applications. This technology allows application components to be isolated from each other using flexible, software-defined boundaries. Oracle Solaris Containers are designed to provide fine-grained control over resources that the applications use, allowing multiple applications to operate on a single Oracle Solaris 10 OS instance while maintaining specified service levels (Figure 6).

Unlike other commercially-available virtualization solutions, Solaris Containers are included with the Oracle Solaris Operating System at no additional cost. Further, both Oracle Database 10g and 11g have been certified on Oracle Solaris Containers and are fully supported by Oracle. Oracle Solaris Containers can be used to deploy virtualized application environments on x86 platforms, at significant cost savings and much lower risk compared to alternative solutions.

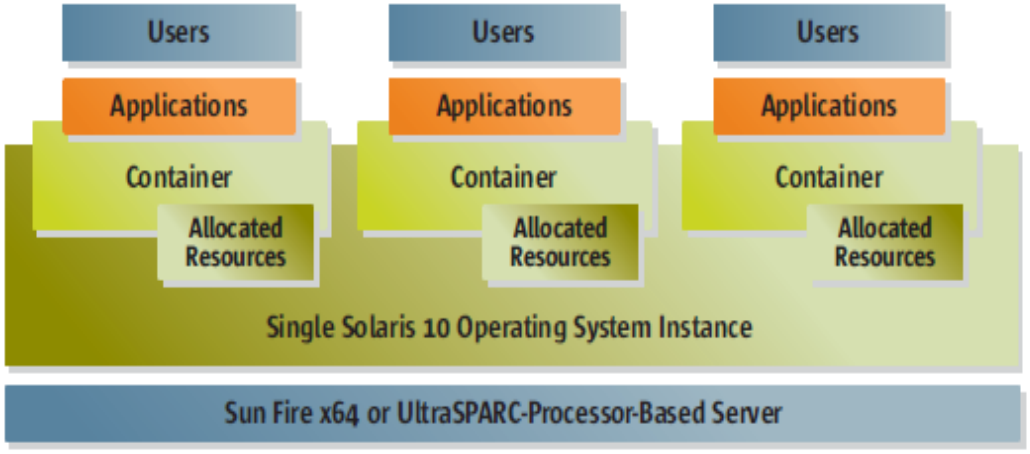


Figure 6 : Solaris Containers enable multiple applications to operate while maintaining specified service levels

Oracle Solaris Containers use Oracle Solaris Resource Manager (SRM) features along with Oracle Solaris Zones software partitioning technology to deliver a virtualized environment that can have fixed resource boundaries for application workloads. For more detailed information about these technologies, see the references section.

Unlike virtual machines, Oracle Solaris Containers provide operating system level virtualization by giving the functionality and isolation of multiple OS instances without requiring multiple physical machines or hypervisor-based virtual machines. Isolation between Oracle Solaris Containers is accomplished by restricting the scope of system calls, rather than the CPU-intensive task of emulating hardware architectures and instruction sets in software. This makes it possible to create hundreds, even thousands, of Oracle Solaris Containers on a single system. Because of this negligible overhead, and unlike physical partitioning or hypervisor-based virtual machines, Oracle Solaris Containers can be created in large numbers. You can create up to 8191 Oracle Solaris Containers in a single system.

Computing resources—CPUs, physical memory, network bandwidth, and more—can be dedicated to a single application one moment and then shared with others in an instant, all without moving applications or rebooting the system or virtual machine where the Oracle Solaris Container resides.

Manage license

By deploying Oracle databases in Oracle Solaris 10 Containers customers can license only the CPUs or cores located in capped Oracle Solaris 10 Containers, since they are recognized as licensable entities, known as hard partitions.

Oracle licensing policy defines hard partitioning as "a physical subset of a server that acts like a self-contained server" (for more details see reference section). The following example (Figure 7) illustrates how an eight processor system can be partitioned into a three processor sub-system using Oracle Solaris Containers technology in the Oracle Solaris 10 OS.

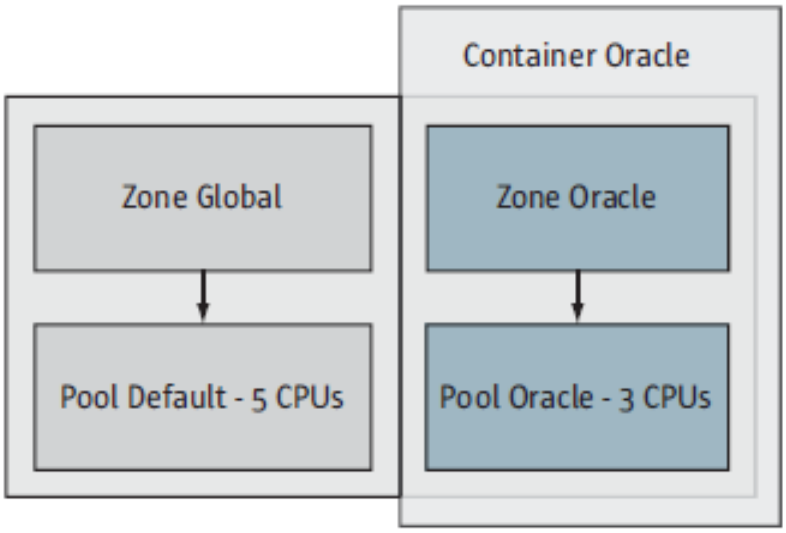


Figure 7: Figure 7 illustrates creation of an Oracle Solaris Container for database deployment.

To create an Oracle Solaris 10 Container that fits the licensing requirements set by Oracle, the Oracle Solaris system administrator needs to create a resource pool with the desired number of CPUs or cores and bind a zone to this resource pool. Alternatively, the administrator may set up a container to use a dynamic pool with a specified CPU maximum limit. The license is driven by the maximum number of CPUs or cores in this pool.

Maintain Quality of Service (QoS)

Customers can consolidate multiple Oracle database instances into separate containers on the same system to enable competing applications, such as online transaction processing (OLTP) and data warehousing applications, to run with predefined resource allocation, changing as business needs change. For example, in Figure 8, the OLTP container is allocated 70 shares of the CPU resources, while the data warehouse is allocated 10 shares, resulting in a 7:1 ratio of CPU resources allocated to each container. Shares allow unused cycles to be used by other applications, or the allocation can be changed dynamically during peak times to provide more CPU resources to either container. In addition, the resources for each container are further subdivided, allocating a portion of resources to each project within the container. This helps ensure that each project always has the resources it requires to function predictably. Database administrators can have complete control over their isolated environment. In addition, a separate project can be created specifically for database administrators in order to limit their access to resources, which can keep other processes from consuming critical CPU resources and negatively affecting the performance of the database.

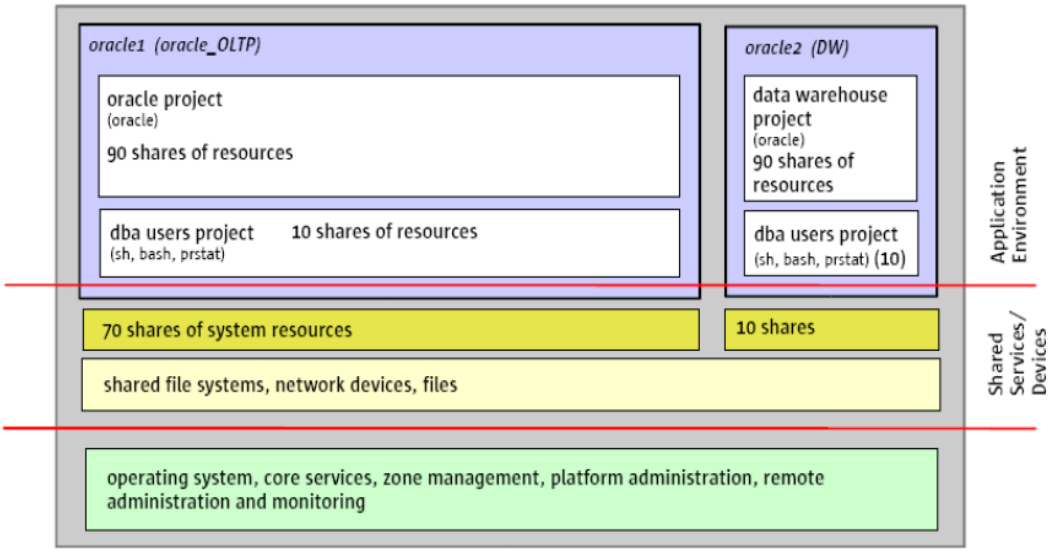


Figure 8 : This figure illustrates consolidation of multiple databases and restriction on database administrator access.

The reference section has a list of collateral that provides step-by-step instructions to set up Oracle Solaris Container for an Oracle database deployment.

Conclusion

The Oracle database has a proven track record of scaling well both vertically as well as horizontally on the Oracle Solaris 10 platform. Additionally, an Oracle database deployment on Oracle Solaris 10 platform can easily take advantage of the unique features found in Oracle Solaris 10 Operating System to enhance the manageability, scalability, availability and security of both single and multiple Oracle

database instances – all across multiple platform and processor architectures. The reference section has a list of collateral that provides detailed information on how Oracle databases deployed on x86 servers can further take advantage of Oracle Solaris 10 platform.

References

The following table contains links to useful information related to this paper

RESOURCES	
Oracle Solaris Operating System	http://www.oracle.com/us/products/servers-storage/solaris/index.html
Platform Choice and Oracle Solaris 10	http://www.oracle.com/us/products/servers-storage/solaris/solaris-10-platform-choice-ds-067317.pdf
Oracle Solaris Operating System— Optimized for the Intel® Xeon® Processor 5600 and 7500 series	http://www.oracle.com/us/products/servers-storage/servers/x86/oracle-solaris-intel-xeon-bwp-065889.pdf
Oracle Solaris OS: Hardware Compatibility Lists	http://www.sun.com/bigadmin/hcl/
General FAQs for Solaris 10	http://www.oracle.com/technetwork/server-storage/solaris/overview/faqs-general-jsp-141628.html
Oracle Solaris : Virtualization	http://www.oracle.com/us/technologies/virtualization/index.htm
Oracle VM FAQ	http://www.oracle.com/us/technologies/virtualization/oraclevm/026952.pdf
Oracle Solaris : Reliability	http://www.oracle.com/us/products/servers-storage/solaris/reliability-066071.html
Oracle Solaris Security	http://www.oracle.com/security/index.html
Oracle Solaris ZFS	http://www.sun.com/bigadmin/topics/zfs/
Configuring Sun Storage 7000 Unified Storage Systems for Oracle Databases	http://wikis.sun.com/display/BluePrints/Configuring+Sun+Storage+7000+Unified+Storage+Systems+for+Oracle+Databases
Oracle® Database Concepts 11*g* Release 2 (11.2), Chapter 18, Concepts for Database Administrators	http://st-doc.us.oracle.com/11/112/server.112/e10713/cncptdba.htm
Oracle Advanced Security Transparent Data Encryption using Sun Crypto Accelerator 6000 PCIe Card	http://wikis.sun.com/download/attachments/210475591/A01-0001-10.pdf
Automating Centralized File Integrity Checks in the Oracle Solaris 10	http://www.sun.com/blueprints/0305/819-2259.pdf

Operating System	
Best Practices for Running Oracle Databases in Solaris Containers	http://wikis.sun.com/display/BluePrints/Best+Practices+for+Running+Oracle+Databases+in+Solaris+Containers
How to create an Oracle Solaris SMF Manifest	http://developers.sun.com/solaris/docs/smf-manifest-053110.pdf
Solaris 10 OS Companion Software CD downloads	http://www.oracle.com/technetwork/server-storage/solaris/overview/index-137343.html

Appendix A: Oracle's Current Sun x86 Servers

Current x86-Based Rackmount Servers

System	Max CPU's	X86 Arch.	Max Memory	Max I/O Slots	Max Internal Disks	Rack Units
Sun Fire X4800	8	Intel	1TB	(8) Express Modules	(8) SAS	5RU
Sun Fire X4640	8	AMD	512GB	(6) PCI-E, (2) PCI-X	(4) SAS	4RU
Sun Fire X4540	2	AMD	128GB	(3) PCI-E	(48) SATA or (8) SSD	4RU
Sun Fire X4470	4	Intel	512GB	(10) PCI-E	(6) SAS or SSD	3RU
Sun Fire X4450	4	Intel	256GB	(6) PCI-E	(8) SAS or (4) SSD	2RU
Sun Fire X4275	2	Intel	144GB	(6) PCI-E	(12) SAS, SATA or (8) SSD	2RU
Sun Fire X4270 M2 (12-Drive)	2	Intel	144GB	(6) PCI-E	(12) SAS, SATA or SSD	2RU
Sun Fire X4270 M2 (24-Drive)	2	Intel	144GB	(6) PCI-E	(24) SAS, SATA or SSD	2RU
Sun Fire X4270	2	Intel	144GB	(6) PCI-E	(16) SAS, SATA or (8) SSD	2RU
Sun Fire X4170 M2	2	Intel	144GB	(3) PCI-E	(8) SAS, SATA or SSD	1RU
Sun Fire X4170	2	Intel	144GB	(3) PCI-E	(8) SAS, SATA or (4) SSD	1RU
Sun Fire X2270 M2	2	Intel	96GB	(1) PCI-E	(4) SATA or SSD, (2) FMods	1RU
Sun Fire X2270	2	Intel	96GB	(1) PCI-E	(4) SATA or SSD	1RU

Current x86-Based Blade Server Modules

Blade Server Modules	Max CPU's	CPU Arch.	Max CPU's	Max Internal Disks	Supported Blade Chassis
Sun Blade X8450	4	Intel	128GB	2	SB8000, SB8000P
Sun Blade X8440	4	AMD	128GB	2	SB8000, SB8000P
Sun Blade X8420	4	AMD	64GB	2	SB8000, SB8000P
Sun Blade X6440	4	AMD	256GB	Compact Flash	SB6048, SB6000
Sun Blade X6275 GbE	4	Intel	192GB	2 SSD	SB6048, SB6000
Sun Blade X6275 IB	4	Intel	192GB	2 SSD	SB6048
Sun Blade X6270 M2	2	Intel	144GB	4 / SSD	SB6000
Sun Blade X6270	2	Intel	144GB	4 / SSD / Compact Flash	SB6048, SB6000

Appendix B: Manifest file to add Oracle database as an SMF service

The reference section has a list of collateral that provides step by step instructions to create the manifest. The following section explains the different sections of the sample Oracle database manifest file.

- Service name:
  - Service name section defines the name of the service as *oracle*. To change the service name, update the following section of the manifest file:
  - `<service_bundle type='manifest' name='oracle'>`
- Dependency section:
  - This section identifies a group of FMRIs upon which the service is dependent. In the sample Oracle SMF manifest file the dependency is set to network, local file system and name-services milestone. If listener and ASM are added to SMF, they should be added as dependency for Oracle database startup.
- Exec method:
  - Defines the methods to be executed for starting, stopping or restarting the service. The sample Oracle manifest file defines these methods in a shell script oracledb. This shell script should be placed in /var/svc/lib directory.
- Method Context:
  - This element combines credential and resource management attributes for execution methods. Edit this section to change the project or resource pool under which Oracle database is deployed.
- Instance name:
  - Defines the instance name. The sample manifest file defines only a single instance. If you want SMF to control multiple instances, this element needs to be replicated for as many instances as required.
- Method Credential
  - Edit this element to update user, group, supp_groups, privileges and limit_privileges attributes
- Method Environment
  - This element defines all the environment variables that are required by methods. This section needs to be edited to suit your environment.
- SMF and RBAC
  - By default, only root user can manage SMF services. The sample manifest allows any user or role that has solaris.smf.manage.oracle authorization to be able to use and configure this service.

- You can configure the oracle user to have authorization to manage this service as under:

Edit the /etc/security/auth_attr file and add the following lines:

- solaris.smf.manage.oracle:::Manage Oracle service states::

Next assign *solaris.smf.manage.oracle* authorization to *oracle* user.

- usermod -A solaris.smf.manage.oracle oracle

The sample Oracle database manifest file for single instance database follows:

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='oracle'>
<service
        name='application/database/oracle'
        type='service'
        version='1'>
        <!--
           Wait for network interfaces to be initialized.
        -->
        <dependency
                name='network'
                grouping='require_all'
                restart_on='none'
                type='service'>
                <service_fmri value='svc:/milestone/network:default' />
        </dependency>
        <!--
           Wait for name-services to be started.
        -->
        <dependency
                name='name-services'
                grouping='require_all'
                restart_on='none'
                type='service'>
                <service_fmri value='svc:/milestone/name-services:default' />
        </dependency>
        <!--
           Wait for all local filesystems to be mounted.
        -->
        <dependency
                name='filesystem-local'
                grouping='require_all'
                restart_on='none'
                type='service'>
                <service_fmri value='svc:/system/filesystem/local:default' />
        </dependency>
        <exec_method
                type='method'
                name='start'
                exec='/lib/svc/method/oracledb start'
                timeout_seconds='60' />
        <exec_method
                type='method'
                name='stop'
                exec='/lib/svc/method/oracledb stop'
                timeout_seconds='60' />
        <exec_method
                type='method'
                name='refresh'
          type='method'
                name='refresh'
```

```
                exec='/lib/svc/method/oracledb refresh'
                timeout_seconds='60' />
        <!--
          action_authorization is needed
          to allow the framework general/enabled property to be changed
          when performing action (enable, disable, etc) on the service.
        -->
        <property_group name='general' type='framework'>
                     <propval name='action_authorization' type='astring'
                        value='solaris.smf.manage.oracle' />
        </property_group>
        <!--
          We define one instances of Oracle database
        -->
        <instance name='default' enabled='false'>
                <method_context project=":default" resource_pool=":default" work
ing_directory=":default">
                <method_credential group="dba" limit_privileges=":default" privi
leges=":default" supp_groups=":default" user="oracle"  />
                <method_environment>
<envvar name='ORACLE_HOME' value='/oracle11gR2/product/11.2.0/dbhome_1'/>
<envvar name='ORACLE_SID' value='bench'/>
                </method_environment>
                </method_context>
        </instance>
        <stability value='Evolving' />
</service>
</service_bundle>

~


<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='oracle'>
  <service name='application/database/oracle' type='service' version='0'>
<!—Wait for network interfaces to be initialized->
 <dependency
       name='network' grouping='require_all' restart_on='none' type='service'>
      <service_fmri value='svc:/milestone/network:default'/>
    </dependency>
<!--wait for all local filesystems to be mountedà
 <dependency name='filesystem-local' grouping='require_all' restart_on='none'
type='service'>
      <service_fmri value='svc:/system/filesystem/local:default'/>
    </dependency>
<exec_method
       name='start'
       type='method'
       exec='/lib/svc/method/oracledb start'
       timeout_seconds='300'>
     </exec_method>
<exec_method
       name='stop'
       type='method'
       exec='/lib/svc/method/oracledb stop'
       timeout_seconds='300'>
     </exec_method>
<exec_method
       name='refresh'
       type='method'
       exec='/lib/svc/method/oracledb start'
       timeout_seconds='60'>
      </exec_method>

    <instance name='default' enabled='false'>
```

```
      <method_context project=':default' resource_pool=':default'
working_directory=':default'>
        <method_credential group='dba' limit_privileges=':default'
privileges=':default' supp_groups=':default' user='oracle'/>
      </method_context>
<method_environment>
        <envvar name='ORACLE_HOME' value='/oracle11gR2/product/11.2.0/dbhome_1'/
>
        <envvar name='ORACLE_SID' value='bench'/>
        </method_environment>
    </instance>

<stability value='Evolving'/>
  </service>
</service_bundle>
```

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

Deploying Oracle Database on x86 Systems
with Oracle Solaris
September 2010
Author: Jeff Taylor

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

SOFTWARE. HARDWARE. COMPLETE.