An Oracle White Paper
June 2011

Resource Management Case Study for Mixed Workloads and Server Sharing
Introduction

Application workloads on a server need to be balanced for the system efficiency. Granular resource management is a necessity to achieve the anticipated performance and service levels in any environment, including virtual and consolidated environments. Without good resource management, faulty runaway workloads can bring progress to a halt causing unwanted delays to priority jobs.

In addition, efficient resource management helps organizations economize by consolidating servers. Server consolidation is one of the effective ways to maximize return on investment (ROI) by cutting unnecessary costs on underutilized servers in a datacenter. Resource management allows controlled allocation of resources to different workloads. An OS process and an active database session are examples of a basic unit of workload.

This paper is part 4 of a four-part series. It introduces various virtualization technologies and hardware platforms offered by Oracle that are suitable for server consolidation. A hypothetical resource management case study is provided with examples to demonstrate different features of Oracle Solaris Resource Manager and Oracle Database Resource Manager.

For more details about Oracle Solaris Resource Manager and Oracle Database Resource Manager and how they can be used to manage system resources effectively in a consolidated environment, see the other parts of this series:

- Part 1: "Introduction to Resource Management in Oracle Solaris and Oracle Database"

The target audience of this paper is Oracle Solaris System Administrators and Oracle Database Administrators. For the sake of simplicity, the acronym "CPU" was used in many places in reference to virtual processors and hardware threads.
Oracle Hardware Platforms and Virtualization Technologies

For various reasons, the traditional IT approach has been one dedicated system per workload. However, such an approach led to overprovisioned and underutilized hardware assets in datacenters and server farms. Also, over the past few years, many datacenters have grown at a significant pace causing global concern over their energy consumption and carbon emission levels. Datacenter owners are now under pressure more than ever for economic, environmental, and political reasons to reduce the energy consumption of their datacenters.

One of the most cost-effective ways to enhance server utilization and to reduce datacenter footprint and energy consumption is to consolidate multiple workloads onto a small number of physical servers using virtualization and other means. With the widespread availability of sophisticated technologies and the variety of features built into the operating systems, database management systems, and other software, it is possible to achieve the required levels of performance, often called a Service Level Agreement (SLA), even in a consolidated virtual environment.

An SLA is a negotiated agreement between two parties, usually between the customer and the service provider. In corporate-level SLAs, system managers might act as the service provider while the end users are the customers. The service provider and customer set up as SLA that establishes a user-oriented view of the workload mix and the service levels required. For example, an SLA might state "80 percentile response time under 5 seconds for the Pricing Process transaction with up to 250 concurrent users during the peak hour."

Maintaining and adhering to SLAs is crucial in consolidated and Web hosting environments, such as cloud computing, to provide predictable service levels to customers.

Oracle Offers the Ideal Consolidation Platform

Oracle Corporation offers a powerful consolidation platform with a rich set of hardware and software technologies for server, storage, and desktop virtualization, which can be used separately or together to reduce IT costs and increase the simplicity of the operational infrastructure. These technologies provide several forms of partitioning as a basis for consolidation. This document revolves around virtualizing enterprise-class servers, so it focuses on server virtualization.

Server Virtualization

Server virtualization technologies enhance consolidation strategies by helping organizations create administrative and resource boundaries between applications on a system to meet performance and security requirements.
Dynamic Domains

Dynamic Domains technology is available on Oracle’s Sun SPARC Enterprise M-series servers, and it allows a single server to be divided into multiple electrically isolated partitions for the purpose of workload isolation. Each domain runs its own instance of Oracle Solaris. These instances do not need to be the same version. Hardware and software failures are contained within a domain, which increases availability and provides a reliable, secure platform for running multiple applications simultaneously.

Dynamic Reconfiguration technology helps reduce or eliminate downtime when adding, removing, repairing, upgrading, or dynamically reconfiguring the hardware components.

Dynamic Domains can be used alone or with Oracle Solaris Containers to provide a flexible and powerful combination of hardware-based and software-based system virtualization.

Oracle VM

Oracle VM is a family of products that work together to facilitate virtual environment creation and management. Consisting of Oracle VM Server and the integrated Oracle VM Manager, which is a browser-based management console, Oracle VM makes it easy to create and manage virtual server pools running on servers across an enterprise.

Oracle VM Server for SPARC

Oracle VM Server for SPARC (previously known as Logical Domains), a software partitioning and server virtualization technology, was built for Oracle servers based on the sun4v architecture. It provides a full virtual machine that runs an independent Oracle Solaris instance and contains a wide range of virtualized devices.

Oracle's Sun SPARC Enterprise T-series servers deliver built-in, no-cost virtualization through logical domains. A logical domain is a discrete logical grouping with its own Oracle Solaris OS, hardware resources, and identity within a physical server. Each logical domain can be created, destroyed, reconfigured, and rebooted independently without requiring a power-cycle of the server.

Oracle VM Server for x86

Oracle VM Server for x86, which is free to download, provides an intuitive graphical interface for creating and managing virtual server pools running on x86 and x64 systems. This virtualization software fully supports Oracle and non-Oracle applications as well as Oracle Solaris, Linux, and Microsoft Windows as guest operating systems. A wide range of Oracle products, including Oracle Database, Oracle Fusion Middleware, Applications Unlimited, and Oracle Linux are certified with Oracle VM Server for x86.
Oracle Enterprise Manager

Oracle Enterprise Manager provides a comprehensive management solution for managing virtual machines, operating systems, and application software (where applicable) from a single product. Oracle VM Management Pack provides integrated, in-depth health and performance monitoring, configuration management, and lifecycle automation for virtual and physical infrastructures for maximum efficiency.

Oracle VM Manager

Oracle VM Manager provides a feature-rich graphical interface for creating and managing Oracle VM environments. With Oracle VM Manager, administrators can load balance across resource pools and automatically reduce or eliminate outages associated with server downtime.

Oracle VM Templates

Oracle VM Templates facilitate rapid software deployment by providing preinstalled and preconfigured software images. Templates are available for download for many key Oracle products including Oracle Database and Oracle Fusion Middleware.

Oracle Solaris Containers

Oracle Solaris Containers are an integral part of the Oracle Solaris 10 operating system and they provision many secure, isolated runtime environments for individual applications using software defined boundaries. All containers run under a single Oracle Solaris kernel enabling fine-grained control over system resources within a consolidated server. In addition, applications can be managed independently of each other.

An Oracle Solaris Container is an Oracle Solaris zone that is bound to a resource pool. A resource pool is a logical entity that owns a subset of the system resources, such as CPU and memory. The resource management features in Oracle Solaris enables system administrators to create resource pools and allocate resources to multiple applications within a single container or to assign and isolate resources to specific containers.

Applications running on the older Solaris 8 and 9 operating systems can run in containers on Oracle Solaris 10 giving organizations access to the latest hardware and operating system advancements without impacting their existing investments in applications.

Oracle's Sun SPARC Enterprise M-series servers support Oracle Solaris Containers in Dynamic Domains, and Oracle's Sun SPARC Enterprise T-series servers support Oracle VM Server for SPARC technologies.

Oracle Solaris Containers are integrated into Oracle Solaris 10. Hence, the Containers partitioning technology is readily available at no additional cost for Oracle Solaris 10 customers to consolidate multiple workloads on SPARC and x86 and x64 systems.
Storage Virtualization

The following storage virtualization technologies from Oracle can help organizations eliminate redundant data, reduce bandwidth requirements, gain flexibility, and better utilize existing infrastructure to reduce space, power, and cooling requirements.

• **Oracle Solaris ZFS** decouples the file system from physical storage allowing more efficient use of storage devices. By automatically allocating space from a shared pool of storage resources when needed, ZFS simplifies storage management. No-cost volume management functionality has been integrated into Oracle Solaris 10.

• **Oracle Exadata Storage Server** offers extreme I/O performance with tens of gigabytes of raw I/O bandwidth making it ideal for storage consolidation. I/O scheduling can be performed using the built-in I/O Resource Manager to keep the I/O devices in Exadata storage cells busy and efficient.

• **Network File System (NFS)** enables the transparent sharing of data and programs between heterogeneous systems and allows access to files without regard to their physical location.

• **Oracle Database 11g Direct NFS Client** integrates NFS client functionality in Oracle Database software. Through this integration, the I/O path between Oracle Database and the NFS server is optimized to provide significantly superior performance. In addition, Direct NFS Client simplifies, and in many cases automates, the performance optimization of the NFS client configuration for different database workloads.

• **Oracle Automatic Storage Management** is integrated into Oracle Database 10g and later releases and provides a simple storage management interface that is consistent across all server and storage platforms for greater management flexibility and efficiency.

Desktop Virtualization

Oracle provides the following set of client-based and server-based desktop virtualization solutions that transcend the limitations of conventional desktop computing to deliver secure, anytime, anywhere access to IT resources from any device:

• **Sun Ray Clients** are ideal for displaying server-hosted virtual desktops. With no moving parts and no local operating system to manage, Sun Ray Clients provide a cost-effective, highly functional thin-client alternative to desktop and laptop computers and reduce many of the problems associated with traditional desktop deployments.

• **Oracle VM VirtualBox** lets client systems run multiple operating environments at the same time to get the most flexibility and utilization out of systems.
• IT staff can reduce the overhead associated with managing individual desktop operating systems by using Oracle Virtual Desktop Infrastructure to standardize on virtual desktop images that can be used across organizations. Users get the flexibility of accessing the same desktop environment from many different client devices and locations.

• Oracle Secure Global Desktop delivers secure access to centralized, server-hosted Windows, UNIX, mainframe, and midrange applications from a variety of clients, including Windows PCs, Mac OS X systems, Oracle Solaris workstations, Linux PCs, thin clients, and more.

Enterprise Servers
Oracle's SPARC and x64 servers offer reliability, availability and serviceability, superior scalability and price/performance, efficient power and cooling, and the broadest offerings in virtualization and consolidation.

Sun SPARC Enterprise T-series Servers
Oracle's Sun SPARC Enterprise T-series servers are environmentally efficient servers that provide built-in hardware virtualization capabilities and deliver superior throughput performance. Multithreaded processors in T-series systems encompass the techniques of both Chip Multiprocessors (CMP) and hardware multithreading in a single chip: CMP with multiple cores per processor and multiple threads per core to improve the instructions processed per cycle. Oracle VM Server for SPARC technology is built into these systems to allow the partitioning of hardware resources, including individual CPU threads for complete isolation between operating system instances.

Combined with embedded cryptographic acceleration, floating-point unit (FPU), I/O, integrated 10 gigabit Ethernet (10 GbE), eco-friendly server design for low power consumption and heat dissipation, and built-in no cost virtualization technology, Sun SPARC Enterprise T-series servers have fewer parts, better reliability and security, and compatibility with the SPARC v9 specification. In addition, they accommodate more users, applications, and workloads making them an ideal powerhouse for server and workload consolidation.

Sun SPARC Enterprise M-series Servers
Powered by SPARC64 processors, Oracle's Sun SPARC Enterprise M-series servers offer high reliability, availability, and serviceability (RAS) as well as superior vertical scalability. Sun SPARC Enterprise M-series servers provide a built-in virtualization capability known as Dynamic Domains. With this technology, administrators can partition a system along hardware boundaries into domains that are electrically fault-isolated from one another.

These powerful servers incorporate many key RAS features, such as Automatic System Recovery (ASR), optional multipathing support to the storage subsystems and networks, and hot-swappable power supplies.
The ability to partition the system into sub-board level domains and the ability to assign or reassign hardware resources to each domain on demand using Dynamic Reconfiguration technology coupled with powerful RAS features make these systems ideally suited for consolidating applications and optimally virtualizing and utilizing resources. Mission-critical workloads, large single instance databases, and consolidation projects are good examples of workloads that might benefit from Sun SPARC Enterprise M-series servers.

Sun x64 Servers

Oracle's Sun x64 servers run 32- and 64-bit operating systems and applications unmodified. Built around Intel and AMD enterprise-class processors, Sun x64 servers and blade systems are designed to deliver the performance and memory capacity that is necessary when consolidating many existing workloads onto a single server. Virtual machine monitors such as Oracle VM Server for x86, Xen, Microsoft Virtual Server, and VMware Infrastructure can be used to virtualize the hardware to run multiple operating systems on a single server. For example, a single system running Oracle VM Server for x86 can support multiple guest operating systems including the Oracle Solaris, Linux, and Microsoft Windows operating systems.

Oracle Enterprise Manager Ops Center

Oracle Enterprise Manager Ops Center is a datacenter lifecycle management tool that provides the ability to discover, provision, patch, manage, and monitor the assets in one or more datacenters from a Web browser. The remote management capabilities are designed to help increase availability and utilization and minimize downtime.

The Web console displays a consolidated view of datacenter resources, including physical systems (x86 and SPARC), operating systems, Oracle Solaris Containers, and logical domains.

Oracle Solaris Operating System

Oracle Solaris is the single most important piece of software that has majority of the previously mentioned virtualization technologies built into it. Technologies such as Oracle Solaris ZFS, Oracle Solaris Containers, Resource Manager, Logical Domain Manager run on top of Oracle Solaris.

For customers facing challenging business and technical requirements, such as lowering costs, simplifying system administration, and maintaining high service levels, Oracle Solaris is the ideal choice. Its innovative, built-in features deliver industry-leading reliability, availability, and security.
Resource Management Case Study

This case study demonstrates how to consolidate multiple applications and databases onto a single server using Oracle virtualization technologies and the resource management features found in Oracle Solaris and Oracle Database. The configurations shown here should not be treated as the Oracle recommended configurations for a consolidation environment. The objective is to show various resource management features with examples. These examples are strictly for the purpose of demonstration.

You are strongly encouraged to refer to part 1 of this series, “Introduction to Resource Management in Oracle Solaris and Oracle Database” for an overview of resource management features in Oracle Solaris and Oracle Database.

This case study assumes that a fictitious organization is consolidating Oracle's Siebel Financial Services, Oracle's PeopleSoft Human Capital Management applications, and the corresponding databases on a single Sun SPARC Enterprise T5240 Server from Oracle. The server has two 8-core 1.6-GHz UltraSPARC T2 Plus processors (128 virtual processors) and 64 GB of physical memory.

The workload characteristics of the enterprise applications are discussed briefly in the following set of tables along with the existing and proposed hardware configurations.

Application #1 : Siebel Financial Services

Table 1 describes the characteristics of the deployed application modules in the fictional organization.

<table>
<thead>
<tr>
<th>APPLICATION MODULE</th>
<th>CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siebel Financial Services eSales</td>
<td>Moderately busy with maximum CPU utilization under 40%</td>
</tr>
<tr>
<td>Siebel Financial Services Call Center</td>
<td>Moderately busy with maximum CPU utilization around 25%</td>
</tr>
<tr>
<td>Oracle Database for Siebel Financial Services</td>
<td></td>
</tr>
</tbody>
</table>
Table 2 lists the existing and proposed hardware configurations.

**Table 2. Deployed and Proposed Hardware Configurations for Siebel Financial Services**

<table>
<thead>
<tr>
<th>Deployed Hardware</th>
<th>Proposed Hardware</th>
</tr>
</thead>
</table>
| • Siebel Financial Services application modules on one Sun SPARC Enterprise T5220 server from Oracle: Number of cores: 8. Number of virtual CPUs (vCPU): 64. RAM: 32 GB. | • Replace one Sun SPARC Enterprise T5220 Server and one Sun SPARC Enterprise M3000 server from Oracle with a single Sun SPARC Enterprise T5240 server.  
  • Siebel Financial Services application modules: Number of cores: 6. Number of virtual CPUs: 48. RAM: 16 GB.  
  • Resource allocation breakdown for Web server: Number of cores: 1. Number of vCPUs: 8. RAM: 16 GB shared with application server.  
  • Resource allocation breakdown for application server: Number of cores: 5. Number of vCPUs: 40. RAM: 16 GB shared with Web server.  
  • Oracle Database for Siebel Financial Services: Number of cores: 2. Number of vCPUs: 16. RAM: 16 GB. |
| • Oracle Database for Siebel Financial Services on one Sun SPARC Enterprise M3000 server: Number of cores: 4 Number of vCPUs: 8 RAM: 32 GB |  |

Table 3 lists the proposed virtualization options for better resource utilization.

**Table 3. Choice of Virtualization for Siebel Financial Services**

<table>
<thead>
<tr>
<th>Tier</th>
<th>Proposed Virtualization Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Global zone of host OS (Oracle Solaris 10)</td>
</tr>
<tr>
<td>Application</td>
<td>Local zone (Containers) running under host OS (Oracle Solaris 10)</td>
</tr>
<tr>
<td>Database</td>
<td>• Guest (logical) domain running under host OS (Oracle Solaris 10)</td>
</tr>
<tr>
<td></td>
<td>• Required OS: Oracle Solaris 10 9/10 for the memory Dynamic Reconfiguration (DR) capability</td>
</tr>
<tr>
<td></td>
<td>• Instance caging within Oracle Database</td>
</tr>
</tbody>
</table>
Table 4 is the resource management plan devised for balancing the performance and the overall resource utilization.

**TABLE 4. RESOURCE MANAGEMENT REQUIREMENTS FOR SIEBEL FINANCIAL SERVICES**

<table>
<thead>
<tr>
<th>TIER</th>
<th>ORACLE SOLARIS RESOURCE MANAGER</th>
<th>ORACLE DATABASE RESOURCE MANAGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>• Restrict physical memory usage to 4 GB.</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>• Limit per-process light-weight processes (LWPs) to 1200.</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>• Isolate Web and application servers using Oracle Solaris zones technology.</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>• Allocate 40 vCPUs to the zone hosting the application server.</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>• Increase shared memory segment maximum size from the default value.</td>
<td>• Implement a resource plan for the requirements outlined in Table 5 and Table 6.</td>
</tr>
<tr>
<td></td>
<td>• Run the database in Fair Share Scheduler (FSS) scheduling class.</td>
<td>• This environment will be shared by another Oracle Database instance (for PeopleSoft HCM), so throttle the number of vCPUs to 16 for this instance.</td>
</tr>
</tbody>
</table>

Table 5 and Table 6 show the database resource consumption caps that will be implemented as a resource plan in Oracle Database. Database users with equal weights get the same priority. Higher weight means higher priority.

**Note:** DOP = Degree of Parallelism; IOPS = Number of I/O operations per second; UNL = Unlimited
TABLE 5. SIEBEL DATABASE RESOURCE CONSUMPTION LIMITS

<table>
<thead>
<tr>
<th>USER GROUP</th>
<th>WEIGHT</th>
<th>CPU RESOURCE ALLOCATION</th>
<th>NUMBER OF ACTIVE SESSIONS</th>
<th>DOP</th>
<th>MAX EXECUTION TIME (CPU TIME)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive</td>
<td>100</td>
<td>UNL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divisional and Regional Manager</td>
<td>75</td>
<td>UNL</td>
<td>UNL</td>
<td>8</td>
<td>10 min</td>
</tr>
<tr>
<td>Line-of-Business Manager,</td>
<td>50</td>
<td>25%</td>
<td>UNL</td>
<td>4</td>
<td>5 min</td>
</tr>
<tr>
<td>Sales/Call Center Representative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervisor</td>
<td>25</td>
<td>15%</td>
<td>10</td>
<td>2</td>
<td>3 min</td>
</tr>
<tr>
<td>Staff</td>
<td>10</td>
<td>10%</td>
<td>5</td>
<td>2</td>
<td>2 min</td>
</tr>
<tr>
<td>Everyone else</td>
<td>5</td>
<td>5%</td>
<td>3</td>
<td>2</td>
<td>90 sec</td>
</tr>
</tbody>
</table>

TABLE 6. SIEBEL DATABASE RESOURCE CONSUMPTION LIMITS CONTINUED

<table>
<thead>
<tr>
<th>USER GROUP</th>
<th>UNDO</th>
<th>I/O BANDWIDTH</th>
<th>IOPS</th>
<th>MAX IDLE TIME</th>
<th>ACTION WHEN LIMITS EXCEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive</td>
<td>UNL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Divisional and Regional Manager</td>
<td>10 GB</td>
<td>UNL</td>
<td>UNL</td>
<td>UNL</td>
<td>Downgrade to next level</td>
</tr>
<tr>
<td>Line-of-Business Manager,</td>
<td>5 GB</td>
<td>5 GB</td>
<td>10000</td>
<td>60 min</td>
<td>Downgrade to next level</td>
</tr>
<tr>
<td>Sales/Call Center Representative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervisor</td>
<td>2 GB</td>
<td>4 GB</td>
<td>8000</td>
<td>30 min</td>
<td>Cancel query</td>
</tr>
<tr>
<td>Staff</td>
<td>1 GB</td>
<td>2 GB</td>
<td>5000</td>
<td>15 min</td>
<td>Kill session</td>
</tr>
<tr>
<td>Everyone else</td>
<td>512 MB</td>
<td>1 GB</td>
<td>4000</td>
<td>10 min</td>
<td>Kill session</td>
</tr>
</tbody>
</table>

The workload characteristics of the second enterprise application, PeopleSoft Human Capital Management, are discussed briefly in the following set of tables along with the existing and proposed hardware configurations.
Application #2: PeopleSoft Human Capital Management

Table 7 lists the characteristics of the deployed application modules.

<table>
<thead>
<tr>
<th>APPLICATION MODULE</th>
<th>CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PeopleSoft Human Resources</td>
<td>• Moderately busy during day with maximum CPU utilization &lt; 30%</td>
</tr>
<tr>
<td></td>
<td>• Mostly quiet after 10 p.m. with average CPU utilization under 2%</td>
</tr>
<tr>
<td>PeopleSoft Payroll for North America</td>
<td>• Semi-monthly payroll processing</td>
</tr>
<tr>
<td></td>
<td>• Extremely busy only for few hours once in every 15 days after 10 p.m.</td>
</tr>
<tr>
<td></td>
<td>with average CPU utilization around 50% (this includes process scheduler and</td>
</tr>
<tr>
<td></td>
<td>database's CPU activity)</td>
</tr>
<tr>
<td></td>
<td>• Mostly idle for the rest of the days with almost no CPU consumption</td>
</tr>
<tr>
<td>Oracle Database for PeopleSoft HCM</td>
<td>• Moderately busy during day with maximum CPU utilization &lt; 15%</td>
</tr>
<tr>
<td></td>
<td>• Mostly quiet after 10 p.m. with average CPU utilization under 2% except</td>
</tr>
<tr>
<td></td>
<td>when processing payroll</td>
</tr>
<tr>
<td></td>
<td>• Extremely busy only for few hours once in every 15 days after 10 p.m.</td>
</tr>
<tr>
<td></td>
<td>during payroll processing with average CPU utilization around 50% (this</td>
</tr>
<tr>
<td></td>
<td>includes process scheduler and database's CPU activity)</td>
</tr>
</tbody>
</table>
Table 8 shows the existing and proposed hardware configurations.

<table>
<thead>
<tr>
<th>DEPLOYED HARDWARE</th>
<th>PROPOSED HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No payroll processing:</strong></td>
<td><strong>Number of cores: 6. Number of vCPUs: 48. RAM: 16 GB.</strong></td>
</tr>
<tr>
<td><strong>Web server:</strong></td>
<td><strong>Number of cores: 1. Number of vCPUs: 8. RAM: 16 GB shared with application server.</strong></td>
</tr>
<tr>
<td><strong>Application server:</strong></td>
<td><strong>Number of cores: 5. Number of vCPUs: 40. RAM: 16 GB shared with Web server.</strong></td>
</tr>
<tr>
<td><strong>Payroll processing:</strong></td>
<td><strong>Number of cores: 2. Number of vCPUs: 16. RAM: 16 GB.</strong></td>
</tr>
<tr>
<td><strong>Web server:</strong></td>
<td><strong>Number of cores: 1. Number of vCPUs: 8. RAM: 8 GB shared with application server.</strong></td>
</tr>
<tr>
<td><strong>Application server:</strong></td>
<td><strong>Number of cores: 1. Number of vCPUs: 8. RAM: 8 GB shared with Web server.</strong></td>
</tr>
</tbody>
</table>

PeopleSoft HCM modules on one Sun SPARC Enterprise TS220 server (Web and application servers):
Number of cores: 8. Number of vCPUs: 64. RAM: 32 GB.

Oracle Database for PeopleSoft HCM and PeopleSoft HCM Process Scheduler on one Sun SPARC Enterprise M4000 server from Oracle:
Number of cores: 8. Number of vCPUs: 16. RAM: 32 GB.
Processor type: SPARC64 VI dual-core processor.

**No payroll processing:**
Number of Cores: 2. Number of vCPUs: 16. RAM: 16 GB.
- **Oracle Database for PeopleSoft HCM:**
  Number of cores: 2. Number of vCPUs: 16. RAM: 16 GB.
- **Process Scheduler:**
  Number of cores: 0. Number of vCPUs: 0. RAM: 0 GB.

**Payroll processing:**
Number of cores: 6. Number of vCPUs: 48. RAM: 24 GB.
- **Oracle Database for PeopleSoft HCM:**
  Number of cores: 5. Number of vCPUs: 40. RAM: 16 GB.
- **Process Scheduler:**
  Number of cores: 1. Number of vCPUs: 8. RAM: 8 GB.
Table 9 lists the proposed virtualization options for better resource utilization.

<table>
<thead>
<tr>
<th>TIER</th>
<th>PROPOSED VIRTUALIZATION OPTION</th>
</tr>
</thead>
</table>
| Web     | • Guest (logical) domain running Oracle Solaris 10 9/10 for the memory DR capability  
|         | • Resource pool (pset, project) in global zone                      |
| Application | • Guest (logical) domain running Oracle Solaris 10 9/10 for the memory DR capability  
|           | • No resource pool in global zone                                   |
| Database | • Guest (logical) domain running Oracle Solaris 10 9/10 for the memory DR capability  
|          | • Instance caging within Oracle Database                             |
Table 10 shows the resource management plan devised for balancing the performance and the overall resource utilization.

<table>
<thead>
<tr>
<th>TIER</th>
<th>ORACLE SOLARIS RESOURCE MANAGER</th>
<th>ORACLE DATABASE RESOURCE MANAGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>Restrict the number of vCPUs to 8.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Application</td>
<td>• Increase max message queue IDs to 256.</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>• During payroll processing, dynamically move 32 vCPUs and 8 GB memory from this domain to the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>other guest domain where Oracle Database for PeopleSoft HCM and PeopleSoft Process Scheduler are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>running.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Use a crontab entry in control domain to dynamically change CPU and memory allocations before</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and after the payroll processing.</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>• Increase shared memory segment maximum size from the default value.</td>
<td>• This environment will be shared</td>
</tr>
<tr>
<td></td>
<td>• Run the database in Fair Share Scheduler (FSS) scheduling class.</td>
<td>by another Oracle Database</td>
</tr>
<tr>
<td></td>
<td>• During payroll processing, bind top Oracle shadow processes to vCPUs in 1:1 fashion for</td>
<td>instance (for Siebel Financial</td>
</tr>
<tr>
<td></td>
<td>improved performance.</td>
<td>Services) so throttle the</td>
</tr>
<tr>
<td></td>
<td>• After payroll processing is completed, move 32 vCPUs and 8 GB memory dynamically back to the</td>
<td>number of vCPUs to 16 for this</td>
</tr>
<tr>
<td></td>
<td>guest domain where they were moved from.</td>
<td>instance.</td>
</tr>
<tr>
<td></td>
<td>• This environment will be shared by another Oracle Database instance (for Siebel Financial</td>
<td>• Enable default maintenance</td>
</tr>
<tr>
<td></td>
<td>Services) so throttle the number of vCPUs to 16 for this instance.</td>
<td>plan in order to enable instance</td>
</tr>
<tr>
<td></td>
<td>• Using Oracle Scheduler, schedule a job to increase CPU_COUNT from 16 to 40 during payroll</td>
<td>caging feature.</td>
</tr>
<tr>
<td></td>
<td>processing and back to 16 when done.</td>
<td>• Using Oracle Scheduler,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>• Restrict memory usage to 4 GB and the number of vCPUs to 8 using a processor set.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Scheduler</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 1 shows the pictorial representation of the hypothetical consolidation scenario that is being considered.

Figure 1. Existing and Proposed Hardware Configurations for a Fictional Organization
Figure 2 is a graphical representation of the dynamic reconfiguration of CPU and memory resources before and after payroll processing.

Figure 2. Proposed Dynamic Reconfiguration in a PeopleSoft Payroll Environment

Sample Implementation for the Hypothetical Consolidation Scenario

The following sections have the implementation details for the consolidation scenario outlined previously. For the benefit of the reader, plenty of comments are embedded in the examples.

Task 1. Create a Guest Domain for Oracle Database

Perform the following as an Oracle Solaris system administrator in the host operating system environment (control domain). These examples assume that Oracle VM for SPARC software is already installed in the global zone of the host operating system.

/* Create guest domain to run Oracle Database for Siebel and PeopleSoft applications */
# ldm add-domain oradb

/* Add 32 CPUs or 4 cores to the guest domain */
# ldm add-vcpu 32 oradb
/* Add 32 GB memory to the guest domain */
# ldm add-memory 32G oradb

/* Add two crypto devices to the guest domain */
# ldm add-crypto 2 oradb

/* Add nxge1 network device to the guest domain */
# ldm add-vsw net-dev=nxge1 guest-vsw1 primary
# ldm add-vnet vnet2 guest-vsw1 oradb

/* Specify the devices to be exported by the virtual disk server as a virtual disk to the guest domain */
# fstyp /dev/dsk/c0t1d0s2
ufs

# fstyp /dev/dsk/c2t4d1s6
ufs

# fstyp /dev/dsk/c2t4d2s6
ufs

# ldm add-vdsdev /dev/dsk/c0t1d0s2 vol1@primary-vds0
# ldm add-vdsdev /dev/dsk/c2t4d1s6 datavol@primary-vds0
# ldm add-vdsdev /dev/dsk/c2t4d2s6 redovol@primary-vds0

/* Add the virtual disks created above to the guest domain */
# ldm add-vdisk vdisk1 vol1@primary-vds0 oradb
# ldm add-vdisk vdisk2 datavol@primary-vds0 oradb
# ldm add-vdisk vdisk3 redovol@primary-vds0 oradb

/* Specify the boot disk for the guest domain */
# ldm set-var boot-device=vdisk1 oradb

/* Set auto-boot property to true for the guest domain */
# ldm set-variable auto-boot\?=true oradb

/* Bind resources to the guest domain */
# ldm bind-domain oradb

/* Verify the resource bindings for the oradb guest domain */
# ldm list-bindings oradb

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>FLAGS</th>
<th>CONS</th>
<th>VCPU</th>
<th>MEMORY</th>
<th>UTIL</th>
<th>UPTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>oradb</td>
<td>bound</td>
<td>----v-</td>
<td>15000</td>
<td>32</td>
<td>32G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Resource Management Case Study for Mixed Workloads and Server Sharing

**MAC**

00:14:4f:f8:ad:29

**HOSTID**

0x84f8ad29

...  

**CORE**

<table>
<thead>
<tr>
<th>CID</th>
<th>CPUSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>(48, 49, 50, 51, 52, 53, 54, 55)</td>
</tr>
<tr>
<td>7</td>
<td>(56, 57, 58, 59, 60, 61, 62, 63)</td>
</tr>
<tr>
<td>8</td>
<td>(64, 65, 66, 67, 68, 69, 70, 71)</td>
</tr>
<tr>
<td>9</td>
<td>(72, 73, 74, 75, 76, 77, 78, 79)</td>
</tr>
</tbody>
</table>

**VCPU**

<table>
<thead>
<tr>
<th>VID</th>
<th>PID</th>
<th>CID</th>
<th>UTIL</th>
<th>STRAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>48</td>
<td>6</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>49</td>
<td>6</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>6</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>29</td>
<td>77</td>
<td>9</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>78</td>
<td>9</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>79</td>
<td>9</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

**MAU**

<table>
<thead>
<tr>
<th>ID</th>
<th>CPUSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>(48, 49, 50, 51, 52, 53, 54, 55)</td>
</tr>
<tr>
<td>7</td>
<td>(56, 57, 58, 59, 60, 61, 62, 63)</td>
</tr>
</tbody>
</table>

**MEMORY**

<table>
<thead>
<tr>
<th>RA</th>
<th>PA</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xe000000</td>
<td>0x40e000000</td>
<td>16G</td>
</tr>
<tr>
<td>0x40e000000</td>
<td>0x80e000000</td>
<td>16G</td>
</tr>
</tbody>
</table>

**VARIABLES**

- auto-boot?=true
- boot-device=vdisk1

**VSW**

<table>
<thead>
<tr>
<th>NAME</th>
<th>MAC</th>
<th>NET-DEV</th>
<th>ID</th>
<th>DEVICE</th>
<th>LINKPROP</th>
<th>DEFAULT-VLAN-ID</th>
<th>PVID</th>
<th>VID</th>
<th>MTU</th>
<th>MODE</th>
</tr>
</thead>
</table>

21
### Resource Management Case Study for Mixed Workloads and Server Sharing

<table>
<thead>
<tr>
<th>guest-vsw1</th>
<th>00:14:4f:8:f8:f6:5c nxge1</th>
<th>0</th>
<th>switch00</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINKPROP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vnet2@oradb</td>
<td>00:14:4f:fb:e3:78</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### NETWORK

<table>
<thead>
<tr>
<th>NAME</th>
<th>SERVICE</th>
<th>ID</th>
<th>DEVICE</th>
<th>MAC</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnet2</td>
<td>guest-vsw1@primary</td>
<td>0</td>
<td>network@0</td>
<td>00:14:4f:fb:e3:78</td>
<td></td>
</tr>
</tbody>
</table>

### DISK

<table>
<thead>
<tr>
<th>NAME</th>
<th>VOLUME</th>
<th>TOUT</th>
<th>ID</th>
<th>DEVICE</th>
<th>SERVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>vdisk1</td>
<td>vol1@primary-vds0</td>
<td>0</td>
<td>disk0</td>
<td>primary</td>
<td></td>
</tr>
<tr>
<td>vdisk2</td>
<td>datavol@primary-vds0</td>
<td>1</td>
<td>disk1</td>
<td>primary</td>
<td></td>
</tr>
<tr>
<td>vdisk3</td>
<td>redovol@primary-vds0</td>
<td>2</td>
<td>disk2</td>
<td>primary</td>
<td></td>
</tr>
</tbody>
</table>

### VCONS

<table>
<thead>
<tr>
<th>NAME</th>
<th>SERVICE</th>
<th>PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>oradb</td>
<td>primary-vcc0@primary</td>
<td>15000</td>
</tr>
</tbody>
</table>

```bash
# ldm list

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>FLAGS</th>
<th>CONS</th>
<th>VCPU</th>
<th>MEMORY</th>
<th>UTIL</th>
<th>UPTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>active</td>
<td>-n-cv-</td>
<td>SP</td>
<td>48</td>
<td>16G</td>
<td>0.0%</td>
<td>45m</td>
</tr>
<tr>
<td>oradb</td>
<td>bound</td>
<td>-----v-</td>
<td>15000</td>
<td>32</td>
<td>32G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

/* Start the guest domain oradb */

# ldm start-domain oradb
LDom oradb started

# ldm list

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>FLAGS</th>
<th>CONS</th>
<th>VCPU</th>
<th>MEMORY</th>
<th>UTIL</th>
<th>UPTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>active</td>
<td>-n-cv-</td>
<td>SP</td>
<td>48</td>
<td>16G</td>
<td>100%</td>
<td>51m</td>
</tr>
<tr>
<td>oradb</td>
<td>active</td>
<td>-t--v-</td>
<td>15000</td>
<td>32</td>
<td>32G</td>
<td>3.1%</td>
<td>2s</td>
</tr>
</tbody>
</table>
/* Install Oracle Solaris as a network install. 
   In this example, we install Oracle Solaris 10 10/09. */

# telnet localhost 15000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

~Connecting to console "oradb" in group "oradb" ....

{0} ok boot net - install
Boot device: /virtual-devices@100/channel-devices@200/network@0 File and args: -
install
Requesting Internet Address for 0:14:4f:fb:e3:78
SunOS Release 5.10 Version Generic_141444-09 64-bit
Copyright 1983-2009 Sun Microsystems, Inc.  All rights reserved.
Configuring devices.
...

# hostname
oradb

# cat /etc/release
Oracle Solaris 10 9/10 s10s_u9wos_14a SPARC
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
Assembled 11 August 2010

# psrinfo -pv
The physical processor has 16 virtual processors (0-15)
UltraSPARC-T2+ (chipid 0, clock 1582 MHz)
The physical processor has 16 virtual processors (16-31)
UltraSPARC-T2+ (chipid 1, clock 1582 MHz)

# prtconf | grep Mem
Memory size: 32768 Megabytes
Task 2. Create a Guest Domain for the PeopleSoft HCM Application

Execute the following as an Oracle Solaris system administrator in the control domain of the host operating system environment. This logical domain will have PeopleSoft Web and application server domains running.

```
# ldm add-domain psfthcm
# ldm add-vcpu 48 psfthcm
# ldm add-memory 16G psfthcm
# ldm add-crypto 1 psfthcm
# ldm add-vsw net-dev=nxge2 guest-vsw2 primary
# ldm add-vnet vnet3 guest-vsw2 psfthcm

# fstyp /dev/dsk/c1t0d0s2
ufs

# ldm add-vdsdev /dev/dsk/c1t0d0s2 psftboot@primary-vds0
# ldm add-vdisk vdisk4 psftboot@primary-vds0 psfthcm
# ldm set-variable auto-boot\?=false psfthcm
# ldm bind-domain psfthcm

# ldm list-bindings psfthcm
NAME           STATE     FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
psfthcm        bound     ------  15001   48    16G
...

# ldm start-domain psfthcm
LDom psfthcm started

# ldm list
NAME         STATE      FLAGS   CONS    VCPU  MEMORY   UTIL  UPTIME
primary      active     -n-cv-  SP      48    16G       99%  11h 12m
oradb        active     -t----  15000   32    32G      3.1%  12m
psfthcm       active     -t----  15001   48    16G       99%  0s

# telnet localhost 15001
~Connecting to console "psfthcm" in group "psfthcm" ....

(0) ok boot net - install
Boot device: /virtual-devices@100/channel-devices@200/network@0  File and args: - install
Requesting Internet Address for 0:14:4f:fa:3c:ef
SunOS Release 5.10 Version Generic_141444-09 64-bit

# hostname
psfthr

# cat /etc/release
Oracle Solaris 10 9/10 s10s_u9wos_14a SPARC
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
Assembled 11 August 2010

# psrinfo -pv
The physical processor has 48 virtual processors (0-47)
UltraSPARC-T2+ (chipid 0, clock 1582 MHz)

# prtconf | grep Mem
Memory size: 16384 Megabytes

**Task 3: Create an Oracle Solaris Container for Siebel Financial Services Application**

Perform the following as an Oracle Solaris system administrator in the global zone of the host operating system environment. This container will have the Siebel application server running.

# hostname
ben16

# zoneadm list -cv
ID NAME             STATUS     PATH                           BRAND    IP
0 global           running    /                              native   shared

# dladm show-dev | grep nxge3
nxge3           link: up        speed: 1000  Mbps       duplex: full

# ifconfig nxge3
ifconfig: status: SIOCGLIFFLAGS: nxge3: no such interface

/* preparatory steps for zone creation */

# mkdir -p /zones/siebapp
# chmod 700 /zones/siebapp

/* Create a "whole root" local zone for Siebel application server */
/* Configure 40 CPUs and exclusive IP using nxge3 NIC */
# zonecfg -z siebelapp
siebelapp: No such zone configured
Use 'create' to begin configuring a new zone.

zonecfg:siebelapp>
zonecfg:siebelapp> create
zonecfg:siebelapp>
zonecfg:siebelapp> set zonepath=/zones/siebelapp
zonecfg:siebelapp> set autoboot=true
zonecfg:siebelapp> set ip-type=exclusive
zonecfg:siebelapp>
zonecfg:siebelapp> add net
zonecfg:siebelapp:net> set physical=nxge3
zonecfg:siebelapp:net> end
zonecfg:siebelapp>
zonecfg:siebelapp> remove inherit-pkg-dir dir=/usr
zonecfg:siebelapp> remove inherit-pkg-dir dir=/sbin
zonecfg:siebelapp> remove inherit-pkg-dir dir=/lib
zonecfg:siebelapp> remove inherit-pkg-dir dir=/platform
zonecfg:siebelapp>
zonecfg:siebelapp> add dedicated-cpu
zonecfg:siebelapp:dedicated-cpu> set ncpus=40
zonecfg:siebelapp:dedicated-cpu> set importance=1
zonecfg:siebelapp:dedicated-cpu> end
zonecfg:siebelapp>
zonecfg:siebelapp> set limitpriv="default,dtrace_proc,dtrace_user"
zonecfg:siebelapp>
zonecfg:siebelapp> verify
zonecfg:siebelapp> commit
zonecfg:siebelapp> exit

# /* Install siebelapp non-global zone */

# zoneadm -z siebelapp install

# zoneadm list -cv

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>-</td>
<td>siebelapp</td>
<td>installed</td>
<td>/zones/siebelapp</td>
<td>native</td>
<td>excl</td>
</tr>
</tbody>
</table>

/* Boot the non-global zone, siebelapp */

# zoneadm -z siebelapp boot

/* Configure the non-global zone, siebelapp, including network services */
Resource Management Case Study for Mixed Workloads and Server Sharing

# zlogin -C -e siebelapp
.. Configuring network interface addresses: nxge3.
.. ben16-1.mydomain.com console login: root
Password:
Aug 17 18:15:27 ben16-1.mydomain.com login: ROOT LOGIN /dev/console
Oracle Corporation SunOS 5.10 Generic Patch January 2005

# psrinfo -pv
The physical processor has 40 virtual processors (0-39)
UltraSPARC-T2+ (chipid 0, clock 1582 MHz)

# prtconf | grep Mem
prtconf: devinfo facility not available
Memory size: 16384 Megabytes

# cat /etc/release
Oracle Solaris 10 9/10 s10s_u9wos_14a SPARC
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
Assembled 11 August 2010

# ifconfig -a
lo0: flags=2001000849 mtu 8232 index 1
   inet 127.0.0.1 netmask ff000000
nxge3: flags=1000843 mtu 1500 index 2
   inet 10.6.xx.xx netmask ffffff00 broadcast 10.6.xx.xx
      ether 0:21:28:76:cf:7f

Task 4. Limit Siebel Web Server Resources

Perform the following as an Oracle Solaris system administrator in the global zone of the host operating system environment.

# zoneadm list -cv
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>STATUS</th>
<th>PATH</th>
<th>BRAND</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>global</td>
<td>running</td>
<td>/</td>
<td>native</td>
<td>shared</td>
</tr>
<tr>
<td>2</td>
<td>siebelapp</td>
<td>running</td>
<td>/zones/siebapp</td>
<td>native</td>
<td>excl</td>
</tr>
</tbody>
</table>

# psrinfo -pv
The physical processor has 48 virtual processors (0-47)
UltraSPARC-T2+ (chipid 0, clock 1582 MHz)
# prtconf | grep Mem
Memory size: 16384 Megabytes

# cat /etc/release
Oracle Solaris 10 9/10 s10s_u9wos_14a SPARC
Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
Assembled 11 August 2010

/* Web server requirements:
1. Limit all Web server processes to consume no more than 4 GB memory.
2. Limit the number of LWPs per process to 1200. */

/* Create a project with limitations of 4 GB physical memory and
1200 LWPs for the Sun Java System Web Server */

# projadd -p 125 -c "Siebel Web Tier" \
> -K "rcap.max-rss=4GB" \
> -K "task.max-lwps=(priv,1200,deny)" sjswebserv

/* Create a new user and associate the newly created project */

# groupadd admin

# useradd -c "Web Server Admin" -d /export/webserv -s /bin/ksh \
> -g admin -u 5678 -K project=sjswebserv wsadmin

/* Enable resource capping daemon in global zone */

# svcadm -v enable rcap
svc:/system/rcap:default enabled.

**Task 5. Enable Oracle Solaris Resource Management in Oracle Database Guest Domain**

Perform the following as an Oracle Solaris system administrator in the global zone of the guest domain
that was created for Oracle Database.

-----------------------------------------------
**FSS Scheduling Class**
-----------------------------------------------

/* Check the current scheduling class */

% dispadmin -d
dispadmin: Default scheduling class is not set
% ps -cafe | grep dbw1
oracle  6174  6142   TS  49 02:19:07 pts/1       0:00 grep dbw1
oracle  2138     1   TS  59 13:59:44 ?       0:06 ora_dbw1_siamst
oracle  2221     1   TS  59 14:00:51 ?       0:08 ora_dbw1_HRHX

/* Set the default scheduler for the system to be the FSS */

/* The following command won't take effect until a domain reboot */
# dispadmin -d FSS

# dispadmin -d
FSS     (Fair Share)

/* Let's change scheduling class of all running processes to FSS
so a reboot can be avoided */

# priocntl -s -c FSS -i all

# ps -cafe | grep dbw1
oracle  2138     1  FSS  29 13:59:44 ?       0:06 ora_dbw1_siamst
oracle  2221     1  FSS  29 14:00:51 ?       0:08 ora_dbw1_HRHX

----------------------------------------------------------------------------
Project project.max-shm-memory
----------------------------------------------------------------------------

/* Check the default value for the maximum shared memory segment */

% prtconf | grep Mem
Memory size: 32768 Megabytes

% prctl -n project.max-shm-memory -i process $$
process: 6572: -su
NAME    PRIVILEGE    VALUE    FLAG   ACTION                       RECIPIENT
project.max-shm-memory
privileged      7.70GB      -   deny                                 -

/* Create a new project that allows larger shared memory segments */

# projadd -p 150 -c "Oracle Database" \
> -K "project.max-shm-memory=(privileged,24G,deny)" orcldatabase

/* Associate the new project with the "oracle" OS user */
```bash
# usermod -K "project=orcldatabase" oracle
# su - oracle

% id -p
uid=5000(oracle) gid=98194051(dba) projid=150(orcldatabase)
```

---

Script for Process Binding

---

/* Check the current number of CPUs */

```bash
% uname -X | grep NumCPU
NumCPU = 32
```

/* Under normal circumstances, 16 CPUs are allocated for the PeopleSoft database. During payroll processing, a few more CPUs will be dynamically added to this domain to increase the CPU count to 40 for the PeopleSoft database.

The following "procbind" script assumes that 32 job streams process payroll in parallel. Run the "procbind" script as "root" user. */

```bash
% cat procbind.sh
```

```
#!/bin/bash

prstat -u oracle -n 50 1 1 | grep -v PID | head -32 | awk '{ print $1 }' > /tmp/oraproc.txt

let count=0;
for i in `cat /tmp/oraproc.txt`
do
echo pbind -b $count $i
    pbind -b $count $i
    let count=$count+2;
done

echo "bound processes:"
pbind
```

/* Corresponding unbind script */

```bash
% cat unbind.sh
```
for i in `cat /tmp/oraproc.txt`
do
    echo "pbind -u $i .."
pbind -u $i
done
echo "bound processes:"
pbind

Task 6. Enable Oracle Database Resource Management in Oracle Database Guest Domain

Perform the following as an Oracle database administrator in the global zone of the guest domain that was created for Oracle Database.

% ORACLE_SID=siamst
% export ORACLE_SID

% sqlplus / as sysdba

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> BEGIN
  2  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
  3
  4  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'EXECUTIVE',
  5          COMMENT        => 'Consumer Group (CG) for Executives',
  6          MGMT_MTH       => 'RUN-TO-COMPLETION');
  7
  8  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'DIV_RGNL_MGR',
  9          COMMENT        => 'CG for Division and Regional Managers',
 10          MGMT_MTH       => 'RUN-TO-COMPLETION');
 11
 12  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'LOBMGR_REPS',
 13          COMMENT        => 'CG for LOB Mgrs, Sales and Call Center Reps',
 14          MGMT_MTH       => 'ROUND-ROBIN');
 15
 16  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'SUPERVISOR',
 17          COMMENT        => 'CG for Supervisors',
 18
 19  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'DIV_RGNL_MGR',
 20          COMMENT        => 'CG for Division and Regional Managers',
 21          MGMT_MTH       => 'RUN-TO-COMPLETION');
 22
 23  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'LOBMGR_REPS',
 24          COMMENT        => 'CG for LOB Mgrs, Sales and Call Center Reps',
 25          MGMT_MTH       => 'ROUND-ROBIN');
 26
 27  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'SUPERVISOR',
 28          COMMENT        => 'CG for Supervisors',
 29
 30  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'DIV_RGNL_MGR',
 31          COMMENT        => 'CG for Division and Regional Managers',
 32          MGMT_MTH       => 'RUN-TO-COMPLETION');
 33
 34  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'LOBMGR_REPS',
 35          COMMENT        => 'CG for LOB Mgrs, Sales and Call Center Reps',
 36          MGMT_MTH       => 'ROUND-ROBIN');
 37
 38  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'SUPERVISOR',
 39          COMMENT        => 'CG for Supervisors',
 40
 41  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP ( CONSUMER_GROUP => 'DIV_RGNL_MGR',
 42          COMM
Resource Management Case Study for Mixed Workloads and Server Sharing

22          MGMT_MTH       => 'ROUND-ROBIN');
23
24  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (  
25          CONSUMER_GROUP => 'TECHSTAFF',
26          COMMENT        => 'CG for Technical Staff',
27          MGMT_MTH       => 'ROUND-ROBIN');
28
29  DBMS_RESOURCE_MANAGER.CREATE_PLAN (  
30          PLAN           => 'SIEBEL_RESOURCE_PLAN',
31          COMMENT        => 'Plan that supports 5 levels of Siebel users',
32          MGMT_MTH       => 'RATIO');
33
34  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (  
35          PLAN                    => 'SIEBEL_RESOURCE_PLAN',
36          GROUP_OR_SUBPLAN        => 'EXECUTIVE',
37          COMMENT                 => 'Executives',
38          SWITCH_GROUP            => NULL,
39          SWITCH_ESTIMATE         => FALSE,
40          MAX_IDLE_TIME           => NULL,
41          MAX_IDLE_BLOCKER_TIME   => NULL,
42          MGMT_P1                 => 100,
43          SWITCH_FOR_CALL         => TRUE,
44          MAX_UTILIZATION_LIMIT   => NULL);
45
46  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (  
47          PLAN                    => 'SIEBEL_RESOURCE_PLAN',
48          GROUP_OR_SUBPLAN        => 'DIV_RGNL_MGR',
49          COMMENT                 => 'Division and Regional Managers',
50          PARALLEL_DEGREE_LIMIT_P1  => 8,
51          SWITCH_GROUP            => 'LOBMGR_REPS',
52          SWITCH_TIME             => 600,
53          SWITCH_ESTIMATE         => FALSE,
54          MAX_EST_EXEC_TIME       => 240,
55          UNDO_POOL               => 10485760,
56          MAX_IDLE_TIME           => NULL,
57          MAX_IDLE_BLOCKER_TIME   => NULL,
58          MGMT_P1                 => 75,
59          SWITCH_FOR_CALL         => TRUE,
60          MAX_UTILIZATION_LIMIT   => NULL);
61
62  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (  
63          PLAN                    => 'SIEBEL_RESOURCE_PLAN',
64          GROUP_OR_SUBPLAN        => 'LOBMGR_REPS',
65          COMMENT                 => 'LOB Mgrs, Sales N Call Center Reps',
66
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>PARALLEL_DEGREE_LIMIT_P1 =&gt; 4,</td>
</tr>
<tr>
<td>67</td>
<td>SWITCH_GROUP =&gt; 'SUPERVISOR',</td>
</tr>
<tr>
<td>68</td>
<td>SWITCH_TIME =&gt; 300,</td>
</tr>
<tr>
<td>69</td>
<td>SWITCH_ESTIMATE =&gt; FALSE,</td>
</tr>
<tr>
<td>70</td>
<td>MAX_EST_EXEC_TIME =&gt; 120,</td>
</tr>
<tr>
<td>71</td>
<td>UNDO_POOL =&gt; 5242880,</td>
</tr>
<tr>
<td>72</td>
<td>MAX_IDLE_TIME =&gt; 3600,</td>
</tr>
<tr>
<td>73</td>
<td>MAX_IDLE_BLOCKER_TIME =&gt; 3600,</td>
</tr>
<tr>
<td>74</td>
<td>MGMT_P1 =&gt; NULL,</td>
</tr>
<tr>
<td>75</td>
<td>SWITCH_IO_MEGABYTES =&gt; 5120,</td>
</tr>
<tr>
<td>76</td>
<td>SWITCH_IO_REQS =&gt; 10000,</td>
</tr>
<tr>
<td>77</td>
<td>SWITCH_FOR_CALL =&gt; TRUE,</td>
</tr>
<tr>
<td>78</td>
<td>MAX_UTILIZATION_LIMIT =&gt; 25);</td>
</tr>
</tbody>
</table>

```
80  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
81      PLAN => 'SIEBEL_RESOURCE_PLAN',
82      GROUP_OR_SUBPLAN => 'SUPERVISOR',
83      COMMENT => 'Supervisors',
84      ACTIVE_SESS_POOL_P1 => 10,
85      PARALLEL_DEGREE_LIMIT_P1 => 2,
86      SWITCH_GROUP => 'CANCEL_SQL',
87      SWITCH_TIME => 180,
88      SWITCH_ESTIMATE => FALSE,
89      MAX_EST_EXEC_TIME => 60,
90      UNDO_POOL => 2097152,
91      MAX_IDLE_TIME => 1800,
92      MAX_IDLE_BLOCKER_TIME => 1800,
93      MGMT_P1 => NULL,
94      SWITCH_IO_MEGABYTES => 4096,
95      SWITCH_IO_REQS => 8000,
96      SWITCH_FOR_CALL => TRUE,
97      MAX_UTILIZATION_LIMIT => 15); |
```

```
99  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
100     PLAN => 'SIEBEL_RESOURCE_PLAN',
101     GROUP_OR_SUBPLAN => 'TECHSTAFF',
102     COMMENT => 'Technical Staff',
103     ACTIVE_SESS_POOL_P1 => 5,
104     PARALLEL_DEGREE_LIMIT_P1 => 2,
105     SWITCH_GROUP => 'KILL_SESSION',
106     SWITCH_TIME => 120,
107     SWITCH_ESTIMATE => FALSE,
108     MAX_EST_EXEC_TIME => 30,
109     UNDO_POOL => 1048576, 
```
MAX_IDLE_TIME           => 900,
MAX_IDLE_BLOCKER_TIME   => 900,
MGMT_P1                 => NULL,
SWITCH_IO_MEGABYTES     => 2048,
SWITCH_IO_REQS          => 5000,
SWITCH_FOR_CALL         => TRUE,
MAX_UTILIZATION_LIMIT   => 10);

DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
    PLAN                    => 'SIEBEL_RESOURCE_PLAN',
    GROUP_OR_SUBPLAN        => 'OTHER_GROUPS',
    COMMENT                 => 'Default Group',
    ACTIVE_SESS_POOL_P1     => 3,
    PARALLEL_DEGREE_LIMIT_P1 => 2,
    SWITCH_GROUP            => 'KILL_SESSION',
    SWITCH_TIME             => 90,
    SWITCH_ESTIMATE         => FALSE,
    MAX_EST_EXEC_TIME       => 20,
    UNDO_POOL               => 524288,
    MAX_IDLE_TIME           => 600,
    MAX_IDLE_BLOCKER_TIME   => 600,
    MGMT_P1                 => NULL,
    SWITCH_IO_MEGABYTES     => 1024,
    SWITCH_IO_REQS          => 4000,
    SWITCH_FOR_CALL         => TRUE,
    MAX_UTILIZATION_LIMIT   => 5);

-- Assuming there are only a few execs, division and regional mgrs,
-- resource consumer group mapping will be based on the Oracle User ID

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (
    ATTRIBUTE       => DBMS_RESOURCE_MANAGER.ORACLE_USER,
    VALUE           => 'JERRY_CEO',
    CONSUMER_GROUP  => 'EXECUTIVE');

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (
    ATTRIBUTE       => DBMS_RESOURCE_MANAGER.ORACLE_USER,
    VALUE           => 'GEORGE_DIVMGR',
    CONSUMER_GROUP  => 'DIV_RGNL_MGR');

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (
    ATTRIBUTE       => DBMS_RESOURCE_MANAGER.ORACLE_USER,
    VALUE           => 'ELAINE_REGNMGR',
    CONSUMER_GROUP  => 'EXECUTIVE');
-- Assuming the application that LOB managers access was hosted on
-- a host called ben50.mydomain.com, consumer group mapping
-- will be based on the CLIENT_MACHINE attribute

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(
        ATTRIBUTE       => DBMS_RESOURCE_MANAGER.CLIENT_MACHINE,
        VALUE           => 'ben50.mydomain.com',
        CONSUMER_GROUP  => 'LOBMGR_REPS');

-- Assuming all the DB requests from Sales and Call Center reps
-- emanate from a service called SIAMST, consumer group mapping
-- can be done based on the SERVICE_NAME attribute

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(
        ATTRIBUTE       => DBMS_RESOURCE_MANAGER.SERVICE_NAME,
        VALUE           => 'SIAMST',
        CONSUMER_GROUP  => 'LOBMGR_REPS');

-- Assuming all the database requests from the supervisor emanate from
-- the REPORTS module, consumer group mapping can be done based
-- on the MODULE_NAME attribute

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(
        ATTRIBUTE       => DBMS_RESOURCE_MANAGER.MODULE_NAME,
        VALUE           => 'REPORTS',
        CONSUMER_GROUP  => 'SUPERVISOR');

-- Assuming technical staff always connect to the database using
-- sqlplus, consumer group mapping will be based on the CLIENT_PROGRAM
-- attribute

DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(
        ATTRIBUTE       => DBMS_RESOURCE_MANAGER.CLIENT_PROGRAM,
        VALUE           => 'sqlplus',
        CONSUMER_GROUP  => 'TECHSTAFF');

-- To resolve conflicting mapping rules, create mapping rule priorities.
-- EXPLICIT parameter represents explicit calls to switch CGs.
-- The priorities assigned must be unique integers from 1 to 10,
-- where 1 represents the highest priority.
EXPLICIT => 1,
ORACLE_USER => 2,
SERVICE_NAME => 3,
CLIENT_MACHINE => 4,
MODULE_NAME => 5,
CLIENT_PROGRAM => 6,
SERVICE_MODULE => 7,
SERVICE_MODULE_ACTION => 8,
MODULE_NAME_ACTION => 9,
CLIENT_OS_USER => 10);

DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();

-- Grant GEORGE_DIVMGR, ELAINE_REGNMGR user to switch to LOBMGR_REPS resource group (RG). Restrict those users in such a way that they won't be able to grant switch privileges for the LOBMGR_REPS group to other users.

DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP (GRANTEE_NAME => 'GEORGE_DIVMGR', CONSUMER_GROUP => 'LOBMGR_REPS', GRANT_OPTION => FALSE);
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP (GRANTEE_NAME => 'ELAINE_REGNMGR', CONSUMER_GROUP => 'LOBMGR_REPS', GRANT_OPTION => FALSE);

-- Since SUPERVISOR and TECHSTAFF consumer groups are low-priority groups, allow anyone to switch to those consumer groups

DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP (GRANTEE_NAME => 'PUBLIC', CONSUMER_GROUP => 'SUPERVISOR', GRANT_OPTION => FALSE);
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP (GRANTEE_NAME => 'PUBLIC', CONSUMER_GROUP => 'TECHSTAFF', GRANT_OPTION => FALSE);

END;
/
PL/SQL procedure successfully completed.

SQL> SELECT USERNAME, INITIAL_RSRC_CONSUMER_GROUP
2  FROM DBA_USERS
3  WHERE USERNAME IN ('JERRY_CEO', 'GEORGE_DIVMGR', 'ELAINE_REGNMGR')
4  /

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>INITIAL_RSRC_CONSUMER_GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEORGE_DIVMGR</td>
<td>DIV_RGNL_MGR</td>
</tr>
<tr>
<td>ELAINE_REGNMGR</td>
<td>DIV_RGNL_MGR</td>
</tr>
<tr>
<td>JERRY_CEO</td>
<td>EXECUTIVE</td>
</tr>
</tbody>
</table>

SQL> SELECT * FROM DBA_RSRC_GROUP_MAPPINGS;

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>VALUE</th>
<th>CONSUMER_GROUP</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE_USER</td>
<td>GEORGE_DIVMGR</td>
<td>DIV_RGNL_MGR</td>
<td></td>
</tr>
<tr>
<td>ORACLE_USER</td>
<td>ELAINE_REGNMGR</td>
<td>DIV_RGNL_MGR</td>
<td></td>
</tr>
<tr>
<td>ORACLE_USER</td>
<td>JERRY_CEO</td>
<td>EXECUTIVE</td>
<td></td>
</tr>
<tr>
<td>ORACLE_USER</td>
<td>SYS</td>
<td>SYS_GROUP</td>
<td></td>
</tr>
<tr>
<td>ORACLE_USER</td>
<td>SYSTEM</td>
<td>SYS_GROUP</td>
<td></td>
</tr>
<tr>
<td>SERVICE_NAME</td>
<td>SIAMST</td>
<td>LOBMGR_REPS</td>
<td></td>
</tr>
<tr>
<td>CLIENT_MACHINE</td>
<td>BEN50.MYDOMAIN.COM</td>
<td>LOBMGR_REPS</td>
<td></td>
</tr>
<tr>
<td>MODULE_NAME</td>
<td>REPORTS</td>
<td>SUPERVISOR</td>
<td></td>
</tr>
<tr>
<td>CLIENT_PROGRAM</td>
<td>SQLPLUS</td>
<td>TECHSTAFF</td>
<td></td>
</tr>
<tr>
<td>ORACLE_FUNCTION</td>
<td>BACKUP</td>
<td>BATCH_GROUP</td>
<td></td>
</tr>
<tr>
<td>ORACLE_FUNCTION</td>
<td>COPY</td>
<td>BATCH_GROUP</td>
<td></td>
</tr>
<tr>
<td>ORACLE_FUNCTION</td>
<td>DATALOAD</td>
<td>ETL_GROUP</td>
<td></td>
</tr>
</tbody>
</table>

12 rows selected.

SQL> SELECT NAME FROM V$RSRC_PLAN
2  WHERE IS_TOP_PLAN = 'TRUE'
3  /

no rows selected

SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'SIEBEL_RESOURCE_PLAN';

System altered.
Task 7. Enable Instance Caging for Oracle Database for Siebel Financial Services

Perform the following as an Oracle database administrator in the global zone of the guest domain that was created for Oracle Database.

/* Enable instance caging using "partitioning" approach

   Step 1. Set the CPU_COUNT to desired number.
           16 is used in this example.
   Step 2. Enable the desired resource plan.
           SIEBEL_RESOURCE_PLAN is used in this example.

   Since CPUs will be dynamically moved during PeopleSoft Payroll processing, Oracle Solaris resource pools were not configured */

SQL> alter system set cpu_count = 16 scope = both;
System altered.

SQL> show parameter cpu_count

NAME     TYPE         VALUE
--------- ---------- ----------------------------
cpu_count integer    20

SQL> alter system set resource_manager_plan='SIEBEL_RESOURCE_PLAN' scope=both;
System altered.

SQL> show parameter resource_manager

NAME                  TYPE         VALUE
---------------------- ---------- ----------------------------
resource_manager_cpu_allocation integer    32
Resource Management Case Study for Mixed Workloads and Server Sharing

resource_manager_plan string SIEBEL_RESOURCE_PLAN

SQL> select name from v$rsrc_plan
2   where is_top_plan = 'TRUE' and cpu_managed = 'ON';

NAME
-------------------------------
SIEBEL_RESOURCE_PLAN

Task 8. Enable Oracle Solaris Resource Management in PeopleSoft HCM Guest Domain

Perform the following as an Oracle Solaris system administrator in the global zone of the guest domain that was created for the PeopleSoft HCM application.

/* Global Zone is psfthr.mydomain.com logical domain */

/* Enable resource pool facility */

# svcadm -v enable system/pools:default
svc:/system/pools:default enabled.

# svcadm -v enable system/pools/dynamic:default
svc:/system/pools/dynamic:default enabled.

# svcs *pool*
STATE STIME FMRI
online 0:55:13 svc:/system/pools:default
online 0:55:24 svc:/system/pools/dynamic:default

/* Create a resource pool with a processor set created with 8 CPUs */

# psrinfo -pv
The physical processor has 48 virtual processors (0-47)
UltraSPARC-T2+ (chipid 0, clock 1582 MHz)

# psrset
#

# cat wspool.cfg
create pset webserv-pset { uint pset.min = 4; uint pset.max = 8 }
create pool webserv-pool
associate pool webserv-pool { pset webserv-pset }
# ls /etc/pooladm.conf
/etc/pooladm.conf: No such file or directory

# pooladm -s

# ls -lh /etc/pooladm.conf
-rw-r--r--  1 root    root        7.4K Aug 22 01:07 /etc/pooladm.conf

# poolcfg -f wspool.cfg
# pooladm -c /etc/pooladm.conf
# pooladm -s

# psrset
user processor set 1: processors 0 1 2 3 4 5 6 7

# poolstat

<table>
<thead>
<tr>
<th>id</th>
<th>pool</th>
<th>size</th>
<th>used</th>
<th>load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>webserv-pool</td>
<td>6</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0</td>
<td>pool_default</td>
<td>42</td>
<td>0.00</td>
<td>0.01</td>
</tr>
</tbody>
</table>

/* Create a project by associating the resource pool webserv-pool */

# projadd -p 1234 -c 'Oracle BEA WebLogic Server for PeopleSoft' \
> -K "project.pool=webserv-pool" weblogicproj

/* Create "bea" user by associating the project weblogicproj.  
BEA WebLogic Server is run under the bea user. */

# groupadd bea

# useradd -d /export/bea -s /bin/ksh -g bea -u 1001 \
> -K "project=weblogicproj" bea

# su - bea

$ id -p
uid=1001(bea) gid=98194051(bea) projid=1234(weblogicproj)

/* Create a resource control project for PeopleSoft application server */

# projadd -p 1235 -c 'Solaris Resource Control Project for PeopleSoft' \
> -K "project.max-msg-ids=(privileged,256,deny)" psftappserv
/* Create "psft" user by associating the "psftappserv" project.
PeopleSoft application server runs under the psft user. */

# groupadd psft

# useradd -d /export/psft -s /bin/ksh -g psft -u 1002 \
> -K "project=psftappserv" psft

# su - psft

$ id -p
uid=1002(psft) gid=98194052(psft) projid=1235(psftappserv)

Task 9. Enable Instance Caging for PeopleSoft HCM Database

Perform the following as an Oracle database administrator in the global zone of the guest domain that
was created for Oracle Database.

SQL> alter system set cpu_count = 16 scope=both;

System altered.

SQL> show parameter cpu_count

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu_count</td>
<td>integer</td>
<td>16</td>
</tr>
</tbody>
</table>

SQL> show parameter resource_manager

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_manager_cpu_allocation</td>
<td>integer</td>
<td>32</td>
</tr>
<tr>
<td>resource_manager_plan</td>
<td>string</td>
<td>SCHEDULER[0x3008]:DEFAULT_MAINTENANCE_PLAN</td>
</tr>
</tbody>
</table>

SQL> select name from v$rsrc_plan
    2   where is_top_plan = 'TRUE' and cpu_managed = 'ON';

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_MAINTENANCE_PLAN</td>
</tr>
</tbody>
</table>

Perform the following as an Oracle Solaris system administrator in the global zone of the guest domain that was created for Oracle Database.

/* Objective : Allocate 8 CPUs and 8 GB RAM to Process Scheduler */

/* Create a resource pool */

# pooladm
pooladm: couldn't open pools state file: Facility is not active

# psrset
#

# svcadm enable pools
# pooladm -s
# poolcfg -c "create pool pool_ps"
# pooladm -c

/* Create a processor for the resource pool. 
 Allocate 8 CPUs for the resource pool. */

# poolcfg -f < create pset pset_ps (uint pset.min = 8 ; uint pset.max = 8) 
> associate pool pool_ps (pset pset_ps)
> EOF
#

# pooladm -c

# psrset
user processor set 1: processors 0 1 2 3 4 5 6 7

/* Create a project with limitations of 8 GB physical memory 
 for the PeopleSoft Process Scheduler. 
 Also associate the resource pool, pool_ps, to limit the 
 number of CPUs to 8. */

# projadd -p 345 -c "System Resources for PeopleSoft Process Scheduler" \
> -K "rcap.max-rss=8GB" \
> -K "project.pool=pool_ps" psftprocsched
/* Create a new user and associate the newly created project */
# groupadd psft

# useradd -c "PeopleSoft Process Scheduler" -d /export/psft -s /bin/ksh \
> -g psft -u 1234 -K "project=psftprocsched" psft

# su - psft

$ id -p
uid=1234(psft) gid=98194052(psft) projid=345(psftprocsched)

/* Enable resource capping daemon */

# svcadm -v enable rcap
svc:/system/rcap:default enabled.

/* Disable resource capping daemon and resource pool facility */

# svcadm -v disable rcap
svc:/system/rcap:default disabled.

# svcadm -v disable pools
svc:/system/pools:default disabled.

# psrset
#

/* Assuming the additional resources (8 CPUs and 8 GB RAM) will be
available on 14th and 29th of every month at 09:45 p.m., create
a couple of cron entries to enable resource pool facility and resource
capping daemon at 09:50 p.m. so PeopleSoft Process Scheduler
can be started with these resource allocations.

These facilities will be disabled at 05:50 a.m. after Process Scheduler
goes down, so these resources go back to where they came from. */

# crontab -l root | grep svcadm
50 21 14,29 * * svcadm enable rcap ; svcadm enable pools >/dev/null 2>&1
50 5 15,30 * * svcadm disable rcap ; svcadm disable pools >/dev/null 2>&1
Task 11. Enable Dynamic Reconfiguration of CPU and Memory Resources for PeopleSoft Payroll Processing

Perform the following as an Oracle Solaris system administrator and Oracle database administrator in the global zone of the host operating system as well as in the global zone of the Oracle Database guest domain.

/* Resource allocations before Dynamic Reconfiguration of CPU and memory resources */

/* Host: oradb, a Guest Domain */

# psrinfo -pv
The physical processor has 16 virtual processors (0-15)
  UltraSPARC-T2+ (chipid 0, clock 1582 MHz)
The physical processor has 16 virtual processors (16-31)
  UltraSPARC-T2+ (chipid 1, clock 1582 MHz)

# prtconf | grep Mem
Memory size: 32768 Megabytes

/* Host: psfthr, another guest domain */

# psrinfo -pv
The physical processor has 48 virtual processors (0-47)
  UltraSPARC-T2+ (chipid 0, clock 1582 MHz)

# prtconf | grep Mem
Memory size: 16384 Megabytes

/* Host OS, primary domain */

# psrinfo -pv
The physical processor has 48 virtual processors (0-47)
  UltraSPARC-T2+ (chipid 0, clock 1582 MHz)

# ldm list

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>FLAGS</th>
<th>CONS</th>
<th>VCPU</th>
<th>MEMORY</th>
<th>UTIL</th>
<th>UPTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>active</td>
<td>-n-cv-</td>
<td>SP</td>
<td>48</td>
<td>16G</td>
<td>0.0%</td>
<td>5d 1h 17m</td>
</tr>
<tr>
<td>oradb</td>
<td>active</td>
<td>-n----</td>
<td>15000</td>
<td>32</td>
<td>32G</td>
<td>0.0%</td>
<td>4d 10h 40m</td>
</tr>
</tbody>
</table>
Resource Management Case Study for Mixed Workloads and Server Sharing

psfthcm          active     -n----  15001   16G      0.0%  4d 11h 24m

/* Dynamically transfer 32 CPUs (four UltraSPARC T2 Plus cores) from guest domain "psfthcm" to "oradb" */

# ldm remove-vcpu 32 psfthcm
# ldm add-vcpu 24 oradb

/* Dynamically transfer 8 GB RAM from guest domain "psfthcm" to "oradb" */

# ldm remove-memory --auto-adj 8G psfthcm
# ldm add-memory --auto-adj 8G oradb

# ldm list

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>FLAGS</th>
<th>CONS</th>
<th>VCPU</th>
<th>MEMORY</th>
<th>UTIL</th>
<th>UPTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>active</td>
<td>-n-cv-</td>
<td>SP</td>
<td>48</td>
<td>16G</td>
<td>0.0%</td>
<td>5d 1h 34m</td>
</tr>
<tr>
<td>oradb</td>
<td>active</td>
<td>-n----</td>
<td>15000</td>
<td>64</td>
<td>40G</td>
<td>0.0%</td>
<td>4d 10h 57m</td>
</tr>
<tr>
<td>psfthcm</td>
<td>active</td>
<td>-n----</td>
<td>15001</td>
<td>16</td>
<td>8G</td>
<td>99%</td>
<td>4d 11h 41m</td>
</tr>
</tbody>
</table>

/* Finally, schedule an Oracle Database job to increase the CPU_COUNT to 40 from 16 and back to 16 when the PeopleSoft Payroll processing is done. Schedule times:

09:55 PM on 14th and 29th of every month : set CPU_COUNT = 40
06:00 AM on 15th and 30th of every month : set CPU_COUNT = 16 */

SQL> SELECT INSTANCE FROM V$THREAD;

INSTANCE
----------
HRHX

SQL> BEGIN
   2   DBMS_SCHEDULER.CREATE_JOB (  
   3       job_name        => 'INCREASE_CPU_COUNT',  
   4       job_type        => 'PLSQL_BLOCK',  
   5       job_action      => 'BEGIN EXECUTE IMMEDIATE ''ALTER SYSTEM SET CPU_COUNT=40''; END;',  
   6       start_date      => SYSTIMESTAMP,  
   7       repeat_interval => 'FREQ=MONTHLY; BYMONTHDAY=14,29; BYHOUR=21; BYMINUTE=55;',  
   8       end_date        => NULL,  
   9       enabled         => TRUE,  
   10   )
   11END;
Resource Management Case Study for Mixed Workloads and Server Sharing

10 comments => 'Set CPU_COUNT to 40 on 14th and 29th of every month at 09:55 p.m.');

12 DBMS_SCHEDULER.CREATE_JOB (  
13   job_name => 'REDUCE_CPU_COUNT',  
14   job_type => 'PLSQL_BLOCK',  
15   job_action => 'BEGIN EXECUTE IMMEDIATE ''ALTER SYSTEM SET CPU_COUNT=16''; END;',  
16   start_date => SYSTIMESTAMP,  
17   repeat_interval => 'FREQ=MONTHLY; BYMONTHDAY=15,30; BYHOUR=6; BYMINUTE=00;',  
18   end_date => NULL,  
19   enabled => TRUE,  
20   comments => 'Set CPU_COUNT to 16 on 15th and 30th of every month at 06:00 a.m.');

21 END;
22 /

PL/SQL procedure successfully completed.

SQL> SELECT owner, job_name, enabled  
2   FROM dba_scheduler_jobs  
3   WHERE job_name LIKE '%CPU_COUNT'  
4 /

OWNER   JOB_NAME            ENABLED
-------- --------------------- --------
SYS     INCREASE_CPU_COUNT   TRUE
SYS     REDUCE_CPU_COUNT     TRUE

/* Assuming payroll processing starts at 10 p.m., dynamically move 32 vCPUs and 8 GB memory from psfthcm guest domain to oradb guest domain.

Assuming payroll processing completes by 6 a.m., dynamically bring back 32 vCPUs and 8 GB memory that was lent by psfthcm guest domain to oradb guest domain tentatively.

Create few cron entries to automate this. */
# crontab -l root | grep ldm

55 21 14,29 * * ldm remove-vcpu 32 psfthcm ; ldm remove-memory 8G psfthcm >/dev/null 2>&1
56 21 14,29 * * ldm add-vcpu 32 oradb ; ldm add-memory 8G oradb >/dev/null 2>&1
00 6 15,30 * * ldm remove-vcpu 32 oradb ; ldm remove-memory 8G oradb >/dev/null 2>&1
01 6 15,30 * * ldm add-vcpu 32 psfthcm ; ldm add-memory 8G psfthcm >/dev/null 2>&1

Resources

Here are resources referenced earlier in this document:

• Part 1 of this series, “Introduction to Resource Management Using Oracle Solaris Resource Manager and Oracle Database Resource Manager”:

• Part 2 of this series, “Effective Resource Management Using Oracle Solaris Resource Manager”:

• Part 3 of this series, “Effective Resource Management Using Oracle Database Resource Manager”:

• “Oracle Database 11g Direct NFS Client”:

• Oracle VM Server for x86:

And here are some additional resources:

• “Zones and Containers FAQ: Resource Management, Performance”:
  http://hub.opensolaris.org/bin/view/Community+Group+zones/faq#HSection3ResourceManagementPerformance


Acknowledgments

The author would like to acknowledge the colleagues at Oracle Corporation who contributed to this document:

Allan Packer, Ritu Kamboj, Yousuf Mohammed, Gia-Khanh Nguyen, Burt Clouse, John Snyder, Kevin Thuan Thai, Sumanta Chatterjee, Uday Shetty, Divyen Patel, Sujeet Vasudevan, Jennifer Glore, and Dennis Peng