



An Oracle White Paper
June 2011

Introduction to Resource Management in Oracle Solaris and Oracle Database

Introduction.....	3
Resource Management Overview.....	4
What Is Resource Management?.....	4
Resource Management Use Cases.....	5
Role of Resource Managers in Server and Workload Consolidation.....	7
Oracle Solaris Resource Manager.....	8
Oracle Database Resource Manager.....	8
Comparing and Contrasting the Resource Managers.....	9
Resource Management Control Mechanisms.....	10
Choosing Between the Resource Managers.....	10
Resource Management in a High-Availability Environment.....	11
Oracle Solaris Resource Manager with Oracle Solaris Cluster.....	11
Oracle Database Resource Management with Oracle Solaris Cluster..	12
Oracle RAC in Oracle Solaris Cluster.....	12
Resource Management Configuration Example.....	12
Resources.....	13

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 1 of a four-part series. It provides an overview of Oracle Solaris Resource Manager and Oracle Database Resource Manager and how they can be used to manage system resources effectively with different types of applications. This paper also discusses resource management in high-availability environments. Although most of the topics in this paper are discussed from a consolidated point of view, resource management features and the content in this paper are equally applicable, with few exceptions, in consolidated and isolated environments running Oracle Solaris and Oracle Database.

For more details about Oracle Solaris Resource Manager and Oracle Database Resource Manager, plus a case study that provides examples that demonstrate their features, see the other parts of this series:

- Part 2: "[Effective Resource Management Using Oracle Solaris Resource Manager](#)"
- Part 3: "[Effective Resource Management Using Oracle Database Resource Manager](#)"
- Part 4: "[Resource Management Case Study for Mixed Workloads and Server Sharing](#)"

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.

Resource Management Overview

Modern computing environments have to provide a flexible response to the varying workloads that are generated by different applications on a system. A workload is an aggregation of all processes of an application or group of applications. Without resource management, the system that manages the resources, be it an operating system or a database management system, responds to workload demands by adapting to new requests dynamically. This default behavior generally means that all activity on the computing system is given equal access to resources, which is undesirable, especially in consolidated environments.

A common problem on any system is runaway processes. A runaway process stops responding to the system and runs uncontrollably. A buggy program that spawns new processes in an infinite loop is an example of a runaway process. Faulty runaway processes can disrupt progress and can create inconvenience to all the workloads that are sharing the system resources.

Consolidated environments pose another challenge in the form of system and performance management to maintain service level agreements (SLAs) specified for different workloads. A service level agreement establishes a user-oriented view of the workload mix and the levels of service required. For example, an SLA for a graduate student registration process might specify "90th percentile response time of under 5 seconds for the enroll class transaction with up to 1500 users online during the peak load when the registration opens up for a new semester."

This is not the complete list, but it reiterates the need for resource management functionality to manage the available resources on a server.

The rest of this paper and [part 2](#) and [part 3](#) of this series introduce various resource management techniques and the functionality offered by Oracle Solaris Resource Manager and Oracle Database Resource Manager. Since the focus is resource management in Oracle Solaris and Oracle Database, unless otherwise stated, hereafter, *resource management* refers to the resource management functionality provided in both Oracle Solaris and Oracle Database. Although the words *system* and *server* are used interchangeably in many occasions, in the context of this paper, *system* usually refers to software that manages allocated resources with a clearly defined purpose, for example, Oracle Solaris or Oracle Database.

What Is Resource Management?

In simple words, resource management is the ability to minimize cross-workload performance compromises and the facilities to monitor resource usage and utilization. The main objective of resource management is to balance the workloads on a server for operational efficiency with minimal overhead. Resource management allows controlling how applications use available system resources. Efficient resource management can help organizations improve resource utilization and reduce total cost of ownership through server consolidation.

Resource management features enable treating workloads individually, and the resource management facilities permit modifying the default behavior of the underlying system with respect to different workloads. *Behavior* primarily refers to the set of decisions made by the system algorithms when an application presents one or more resource requests to the system.

Resource management facilitates the following:

- Allocating computing system resources, such as processor time or I/O bandwidth
- Monitoring and adjusting resource allocations as necessary
- Generating extended accounting information for analysis, billing, and capacity planning

Using resource management functionality, users and system administrators can do the following:

- Manage multiple workloads that are competing for shared resources
- Isolate workloads from each another
- Offer resources to workloads on a priority basis
- Restrict or deny access to a specific resource
- Treat certain allocations collectively rather than through isolated mechanisms

A system configuration that uses resource management facilities can serve different objectives, such as the following.

- Prevent an application, process, or session from consuming resources indiscriminately
- Change the priority of workloads based on external events
- Maximize system utilization with balanced resource allocations among a set of workloads

Identifying the cooperating and conflicting workloads is one of the key requirements in planning a resource management configuration that poses the least compromise to the service goals of the business.

Resource Management Use Cases

The ability to perform system and resource management with ease and accuracy is critical to maintaining application and database performance, scalability and availability. Use resource management to ensure that deployed applications meet the required response times and achieve service level objectives.

Resource management can also increase system resource utilization. By categorizing and prioritizing usage, reserve capacity can effectively be used during off-peak periods, often eliminating the need for additional processing power.

Resource management is ideal for environments that consolidate a number of applications on a single server. Instead of running each workload on a separate system with full access to that system's resources, use resource management facilities to segregate workloads within a system. Resource management helps lower overall total cost of ownership by controlling several dissimilar applications on a single system.

Use resource management features in any system that has a large, diverse user base, such as a brokerage firm or an educational institution. For example, in large brokerage firms, traders intermittently require faster access to execute a query or to perform a calculation. By allocating a proportionately larger amount of processing power to the traders' work, the traders get the responsiveness that they need.

Resource management is also ideal for supporting thin-client systems. These platforms provide stateless consoles and input devices, such as smart cards. The actual computation is done on a shared server resulting in a time-sharing type of environment. Use resource management features to isolate the users on such a shared server. Then, it is possible to prevent users from generating excessive load and impacting other users on the shared system.

Use resource management features to meter computing resources usage for billing and capacity planning. For example, modern services such as cloud computing require resource usage tracking per client and per application on daily, weekly, monthly, and annual basis. This helps the clients choose a vendor cloud based on cost and reliability. Also, it helps the service provider determine the capacity requirements to meet ever-changing demands and priorities.

Role of Resource Managers in Server and Workload Consolidation

The push for consolidation of workloads and servers onto fewer systems created new demands on system and performance management. This challenge has been addressed in Oracle Solaris in the form of Oracle Solaris Resource Manager.

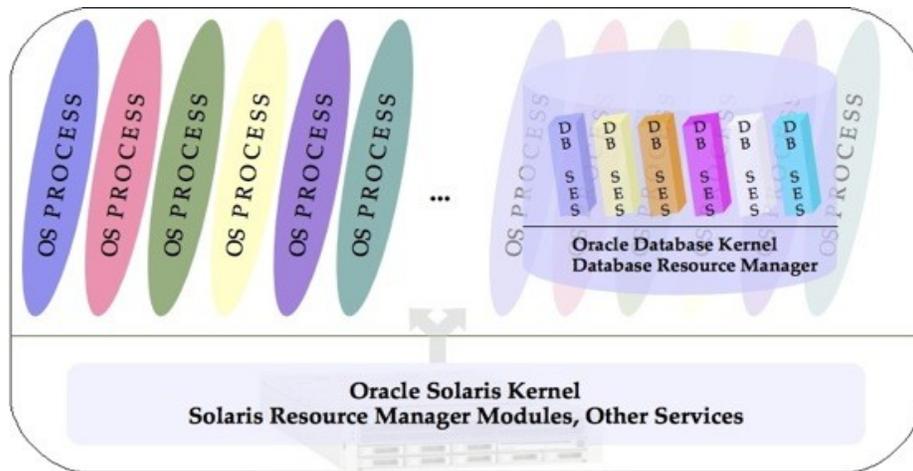


Figure 1. High-Level Depiction of Resource Managers in Oracle Solaris

Oracle Solaris Resource Manager provides the ability to allocate and control major system resources. It also implements administrative policies that enforce access to and capping of system resources for different users. This approach works well when the related processes that constitute an application run completely under the control of the operating system. However, an Oracle Database instance creates its own runtime environment on top of Oracle Solaris, manages database resources, and schedules database jobs independently and transparently to the underlying operating system. In other words, the operating system is largely unaware of the activity that takes place inside the Oracle database management system. Oracle Database Resource Manager fills this gap and helps consolidate multiple database workloads by providing more control over resource management decisions to the Oracle Database server.

For example, in a consolidated banking application, more resources can be allocated with higher priority to the ATM transactions during peak hours to ensure faster response to ATM users. During off-peak hours, the priority and resource allocation for the ATM transactions can be lowered to let other applications perform batch processing, such as generating monthly bank statements. The resource management functionality in Oracle Solaris and Oracle Database can be used to achieve this kind of dynamic functionality.

Oracle Solaris Resource Manager

Historically, all processes have been provided equal access to system resources by the Oracle Solaris kernel. The default time-sharing scheduler attempts to allow each process on the system to have relatively equal CPU access. While this approach works in many cases, it fails to recognize that some activities in an enterprise are more important than others. With the help of the Oracle Solaris Resource Manager facility, system administrators can implement resource allocation policies to guarantee predictable service levels to enterprise applications and users.

Oracle Solaris Resource Manager is a resource control mechanism for allocating and controlling major system resources, such as CPU bandwidth and memory, among various users or workloads. This fine-grained capability enables multiple applications to be consolidated onto a single server, thus improving resource utilization and lowering the Total Cost of Ownership (TCO).

With the help of Oracle Solaris Resource Manager, system administrators can define workloads and partition and allocate system resources to meet predefined Service Level Agreements (SLAs), while maintaining the overall quality of service (QoS) and keeping system resources busy. In addition, Oracle Solaris Resource Manager allows administrators to monitor resource usage, identify users or applications that use more resources than they should, and compile more accurate data over time for capacity planning and billing.

Oracle Database Resource Manager

By default, Oracle Solaris attempts to be fair to all users and workloads by employing a round-robin style time-sharing algorithm for allocating CPU time. However, Oracle Database is presented to Oracle Solaris as a black box. Hence, it is recommended to use Oracle Database Resource Manager to manage database resources within an Oracle Database instance.

Introduced in Oracle8i, Oracle Database Resource Manager is a database resource management mechanism that allows delineating logically distinct units of a workload and partitioning and allocating system resources among those units. The features of Oracle Database Resource Manager enable the prioritization of database operations by aligning the allocation of system resources to the business objectives and priorities of the enterprise. Also, it is possible to balance one user's resource consumption against other users and distribute system resources among tasks of varying importance and priorities to achieve overall enterprise goals.

Careful use of Oracle Database Resource Manager can increase the overall efficiency of the available system resources, which reduces the need to increase a server's processing capacity to maintain acceptable response times. For example, Oracle Database Resource Manager can be configured to manage the CPU load at granular levels with the help of the Instance Caging feature in Oracle Database.

Comparing and Contrasting the Resource Managers

Oracle Solaris Resource Manager and Oracle Database Resource Manager integrate resource management functionality into Oracle Solaris and Oracle Database, respectively. Both of these resource managers are software modules that facilitate resource management, but they do not run as standalone processes. To keep it simple and easy to understand, all the resource management functionality is collectively referred as "Resource Manager" in both Oracle Solaris OS and Oracle Database.

Both the resource managers allow categorizing and prioritizing resource allocation and usage to align the objectives and priorities of an enterprise, and they enable administrators to create effective resource configurations to fully utilize reserve capacity on a server. Consequently, the need for additional processing power can be eliminated, and multiple workloads can be consolidated onto fewer servers.

From an operating system perspective, Oracle Solaris Resource Manager perceives any Oracle Database instance as a black box, whereas Oracle Database Resource Manager focuses on that black box exclusively. Oracle Database Resource Manager functionality is embedded in an Oracle Database module, hence Oracle Solaris has no visibility into the activity of Oracle Database or Oracle Database Resource Manager. Due to this, Oracle Solaris treats an Oracle Database instance as a group of related processes (see Figure 1). If you imagine Oracle Database as an operating system, Oracle Database Resource Manager performs almost similar functionality as that of Oracle Solaris Resource Manager but to a lesser extent. Oracle Database Resource Manager uses different terminology and controls fewer similar resource types, but it work towards a common goal of increased resource utilization and improved quality of service with fewer hardware resources.

Oracle Database Resource Manager includes a scheduling mechanism that keeps track of CPU time and I/O bandwidth consumption and performs scheduling decisions at fixed time intervals, whereas Oracle Solaris Resource Manager does not include a job scheduling mechanism. It merely controls how a process runs on its host system rather than when or where it runs.

Oracle Solaris Resource Manager functionality is available on all recent versions of Oracle Solaris by default, whereas Oracle Database Resource Manager requires an Oracle Database Enterprise Edition license to implement nondefault maintenance plans.

Performance investigations have shown that there is no measurable overhead on a workload when running under Oracle Solaris Resource Manager or Oracle Database Resource Manager.

Resource Management Control Mechanisms

In general, resource control mechanisms can be divided into at least three categories: constraints, scheduling, and partitioning. Table 1 maps the resource control mechanism with the features implemented in Oracle Solaris Resource Managers and Oracle Database Resource Manager.

TABLE 1. MAPPING RESOURCE CONTROL CATEGORIES TO THE IMPLEMENTED FEATURES IN RESOURCE MANAGERS

RESOURCE CONTROL MECHANISM	ORACLE SOLARIS RESOURCE MANAGER	ORACLE DATABASE RESOURCE MANAGER
Constraint	Resource Controls	Resource Allocation Directives
Scheduling	Solaris Scheduler	Oracle Scheduler
Partitioning	Resource Pool, Containers	Instance Caging

Each of the resource control mechanism categories and the corresponding features in Oracle Solaris Resource Manager and Oracle Database Resource Manager are discussed in [part 2](#) and [part 3](#) of this series.

Choosing Between the Resource Managers

There are two ways to manage resources in a database:

- The *black box* approach where the whole instance is managed as a single entity
- Using a *resource management* facility to manipulate resources within an Oracle Database instance

Database resource management can be somewhat simplified by viewing the database instance as a black box and managing resources around it using Oracle Solaris Resource Manager. This strategy is useful in server consolidation because the main objective is to isolate one workload from another.

If the requirement is fine-grained resource management within Oracle Database, use Oracle Database Resource Manager.

In general, it is not recommended to run Oracle Solaris Resource Manager and Oracle Database Resource Manager concurrently without proper understanding and planning. Because neither of them are aware of each others' existence, both resource managers try to control resource allocation in a manner that causes unpredictable behavior and instability to the workloads running on the server.

The following guidelines help maximize predictability when both Oracle Solaris Resource Manager and Oracle Database Resource Manager are used in the same environment.

- If multiple database instances are running on a single server, use Oracle Solaris Resource Manager to allocate and distribute resources among them, and use Oracle Database Resource Manager to control resource distribution within each instance of the database.
- Do not use process priority management commands, such as `nice(1)`, on Oracle Database processes
- Do not change the resource allocations for different database instances frequently

Resource Management in a High-Availability Environment

Oracle Solaris Cluster software provides high-availability (HA) support for applications and databases running in a cluster of servers. It prevents loss of service in the event of a single-point failure through hardware redundancy, hardware and software failure detection, failover of applications, and automatic restart of data services.

All resource management features for controlling system resources from Oracle Solaris and Oracle Database are supported in an Oracle Solaris Cluster environment. Resource management features can be used in any valid Oracle Solaris Cluster topology.

Oracle Solaris Resource Manager with Oracle Solaris Cluster

Oracle Solaris Cluster has some specific integrations with Oracle Solaris Resource Manager.

Oracle Solaris Resource Controls Under Oracle Solaris Cluster

Oracle Solaris resource management features, such as CPU shares and resource pools, can be applied to Oracle Solaris Cluster Data Services using the `Resource_project_name` property. If this property is not specified, the `default` Oracle Solaris project is used.

When using Oracle Solaris resource management features, it is the responsibility of the system administrator or user to implement the Oracle Solaris projects or resource controls as required on all the nodes where the resource is made available for different workloads.

Oracle Solaris Containers and Zone Clusters Under Oracle Solaris Cluster

When managing applications in Oracle Solaris native zones, the zone is created and configured using Oracle Solaris zone management tools, such as `zonecfg(1M)` and `zoneadm(1M)`. Hence, the Oracle Solaris Resource Manager configuration for the zone would apply transparently to applications managed by Oracle Solaris Cluster. In case of a zone cluster, the zone-level resource management configuration for a zone cluster is defined using the `clzonecluster` tool with syntax that is similar to that of the `zonecfg(1M)` command.

Oracle Database Resource Management with Oracle Solaris Cluster

Except for a few features, such as Data Guard, the majority of Oracle Database features, including Oracle Database Resource Manager, are transparent to Oracle Solaris Cluster. It is the responsibility of the database administrator to configure the required database initialization parameters on all nodes where the database instance is running.

Oracle RAC in Oracle Solaris Cluster

Starting with Oracle Real Application Clusters (Oracle RAC) 10g Release 2, databases configured with Oracle Solaris Cluster RAC agents are also required to register with Cluster Ready Services (CRS) or Oracle Clusterware. Oracle RAC databases and instances are typically administered using the Server Control Utility, SRVCTL. When CRS or Oracle Clusterware starts the database instance using its own resource configuration without knowledge of the existence of an Oracle Solaris project, Oracle Solaris resource management support in Oracle Solaris Cluster is not applicable. However, in Oracle 9i RAC or single-instance databases of any Oracle release, Oracle Solaris Cluster controls the databases directly with no dependency on CRS or Oracle Clusterware. Hence, it is possible to make use of Oracle Solaris projects for resource management on Oracle Solaris.

For more information, see the man page of `clzonecluster(1CL)`.

Resource Management Configuration Example

Refer to part 4 of this series, “[Resource Management Case Study for Mixed Workloads and Server Sharing](#),” for a comprehensive example that shows how to consolidate multiple applications and databases on a single server using Oracle virtualization technologies and the resource management features found in Oracle Solaris and Oracle Database.

Resources

Here are resources referenced earlier in this document:

- Part 2 of this series, “Effective Resource Management Using Oracle Solaris Resource Manager”:
<http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-055-solaris-rm-419384.pdf>
- Part 3 of this series, “Effective Resource Management Using Oracle Database Resource Manager”:
<http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-056-oracledb-rm-419380.pdf>
- Part 4 of this series, “Resource Management Case Study for Mixed Workloads and Server Sharing”:
<http://www.oracle.com/technetwork/articles/servers-storage-admin/o11-057-mixed-wl-rm-419381.pdf>

And here are some additional resources:

- “Zones and Containers FAQ: Resource Management, Performance”:
<http://hub.opensolaris.org/bin/view/Community+Group+zones/faq#HSection3ResourceManagementPerformance>
- “Solaris Administrator's Quick Reference” by Jialong He:
http://www.cheat-sheets.org/saved-copy/Solaris_quickref.pdf
- “Estimating a Data Center's Electrical Carbon Footprint”:
http://www.apcmedia.com/salestools/DBOY-7EVHLH_R0_EN.pdf
- Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture* (second edition) by Richard McDougall and Jim Mauro (ISBN-13: 978-0131482098): <http://www.amazon.com/gp/product/0131482092/>
- Resource Management* by Richard McDougall, Adrian Cockcroft, Enrique Vargas, Evert Hoodendoorn, and Tom Bialaski (ISBN-13: 978-0130258557):
<http://www.amazon.co.uk/Resource-Management-Blueprints-Richard-McDougall/dp/0130258555>



Introduction to Resource Management in Oracle
Solaris and Oracle Database

June 2011, Revision 1.0

Author: Giri Mandalika

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



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

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Hardware and Software
Engineered to Work Together

