

An Oracle White Paper  
September 2012

# Performance with the Oracle Database Cloud

## Multi-tenant architectures and resource sharing

# Table of Contents

Overview .....	3
Performance and the Cloud .....	4
Performance and resources .....	4
Performance and resource sharing .....	5
Resource sharing in the Oracle Database .....	6
Specifying resource usage .....	7
Limiting resource usage .....	7
Database Resource Manager .....	7
Instance caging .....	8
Consumer groups .....	8

## Overview

Performance is a key factor in the success of any information system. Moving to a Cloud-based solution shapes the delivery of performance in several ways, including how performance is evaluated and managed.

This paper will examine the performance characteristics of Cloud-based solutions, explore the ways that computing resources are managed to deliver performance from a shared Cloud resource, and explain how the Oracle Database Cloud uses Oracle technology to provide performance across the entire Database Cloud tenant community.

## Performance and the Cloud

The Cloud provides a broad array of powerful benefits – elimination of capital expenditures, reduction of operating expenses, extremely rapid provisioning for Cloud services and nearly instant deployment of applications. The Oracle Database Cloud Service delivers all of these advantages, along with a powerful rapid application development environment, RESTful Web Service access to your data, and a set of packaged applications which can be deployed with a handful of clicks to provide immediate business benefit.

The Oracle Database Cloud, like all Cloud solutions, delivers these advantages through the use of a shared resource model for implementation and the Internet for communication. This architecture model has implications for the overall performance of your Database Cloud Service solutions.

The first implication stems from the use of the Internet. The Internet introduced an enormous number of great features and capabilities that changed the overall computing landscape. Along with these benefits came a significant amount of latency in communication – a latency that has become accepted as part of the new computing environment.

It is not unusual to experience some delay in receiving pages over the Internet. You have probably retrieved a page today that took 5 seconds to fully appear in your browser. This latency is mainly due to the time required to move information from a server somewhere on the Internet to your browser – not because of any inherent slowness in that server's operations itself. In other words, the overhead of communication is primarily responsible for the overall speed of the Cloud-based applications.

One of the truisms about performance is that the perception of performance is always based on expectations met or unmet. The expectation of response time for a Cloud-based application is not necessarily the same as the response you would get from a local server – you won't get your pages as fast on a consistently basis regardless of the speed of the originating server.

The second key aspect of Cloud-based offering is that resources are shared among multiple users. This sharing is the key to the cost and labor savings you get with a Cloud solution. To fully understand how the Oracle Database Cloud Service delivers great performance, it helps to review the way that performance and resource usage are connected.

## Performance and resources

As a starting point, consider how performance and resources are linked. For any computer system, there are two determinants to the overall performance of the system. The first is the underlying capabilities of the server system. No operation can complete faster than the CPU can execute the instructions required for the operation. As CPU chips have become faster, computers have been able to do more work faster. This situation is absolutely true only

when everything the CPU needs is available to it. Since operations inevitably involve data, the amount and speed of memory, data transfer and disk can also have an effect on performance.

In the real world, very few operations have unlimited resources available to them, simply because computer software has been created to share the underlying resources among many processes. In the modern computing world, shared resources do not necessarily mean poor performance. However, oversubscribed resources, which result in processes simply waiting to do their work, do lead to poor performance.

You can reduce the possibility of poor performance due to oversubscribed resources in two basic ways. First of all, you can increase the amount of resources available – a greater supply means less contention for resources. The Oracle Database Cloud has taken this approach by using Exadata and Exalogic to implement the Database Cloud Services – the most powerful computing platform in existence for the Oracle software stack.

The second way to reduce oversubscription of resources is to increase the efficiency of resource sharing, which is described in the next sections of this paper.

## Performance and resource sharing

Resource sharing has been at the core of computing systems for more than five decades. Part of the order-of-magnitude improvement in performance has come from faster components, but also from the ability to share resources more efficiently through software.

Resource sharing is primarily implemented through software, for the simple reason that hardware resources, at their atomic level, cannot be shared. Although a phrase like “the CPU is 50% busy” is common, it is also misleading. You might assume from this statement that the CPU is working at 50% of its capability across time. But CPUs are all-or-nothing resources. What this phrase really means is that the CPU is 100% busy 50% of the time. When you understand resource usage in this way, you can realize how the ability to share resources efficiently through software is all important.

A similar situation exists for other computing resources, such as memory. A single memory bit is either used or not used, so using memory for the information that is most frequently used is the key to improved resource utilization and avoiding the oversubscription that leads to impaired performance. Software systems implement this type of sharing through the use of intelligent algorithms developed over years of operation of real world production systems.

All Cloud computing systems use some form of resources sharing, just like all computer systems. In this sense, a term like “multi-tenancy” loses its meaning – shared resources are available to multiple tenants. There is a distinction between different architectures used to implement Cloud systems that has brought this term to prominence, but the distinction has to do with the granularity of a shared resource, rather than the actual fact of multiple tenants sharing the resources.

In terms of a database Cloud architecture, there are three basic levels of granularity –

- A single dedicated machine, which could contain multiple database instances
- Multiple virtual machines on a single physical machine, each of which could also contain multiple database instance
- Multiple user processes sharing a single database instance. In the Oracle Database Cloud, this multi-tenancy is based on schema isolation.

In the first case, there may be sharing of resource between database instances, but each physical machine is isolated in its use of its own resources<sup>1</sup>. In the second case, the smallest unit of sharing is a virtual machine, which carries with it the overhead of the entire stack, from the operating system to the database instance. The third case has the most discrete granularity, with relatively small user processes being the unit of sharing.

The larger the granularity of the unit used for sharing, the less effective the sharing process can be. One way to visualize this is to imagine a glass cylinder, initially empty. If you were to fill it with large rocks, there would be a lot of space left, which would represent computing resources that could not be effectively shared. Smaller rocks would result in less empty space, and sand would pack the cylinder with the least amount of empty space. In a similar manner, the lower granularity of sharing based on user processes can be much more efficient than sharing based on virtual machines.

Oracle has real world proof of the scalability of schema-based multi-tenancy. Oracle has had a free trial service for Application Express, which uses this type of multi-tenancy, in production for 8 years, and the statistics from this live site back up the analogy. As of August, 2012, the site, [apex.oracle.com](http://apex.oracle.com), was running on a server with 8 cores and 32 GBs of memory. This relatively small machine was supporting 14,000 active tenants – not users, but tenants, each with the potential for many users. This size machine would be able support a considerably smaller number of virtual machines.

You could think that this issue is of no concern to users, since this entire infrastructure is invisible in the Cloud. But a more efficient Cloud infrastructure creates efficiencies of scale across the board for the Cloud provider, and this efficiency inevitably drives the acquisition and use of resources and can result in higher costs or lower overall resource availability, which in turn can result in reduced performance.

## Resource sharing in the Oracle Database

---

<sup>1</sup> Oracle Real Application Clusters do allow for sharing across physical nodes, but that technology is equally available to all three of these sharing types.

The Oracle Database, since its introduction more than 30 years ago, has focused on two main tasks – protecting the integrity of data stored within the database and efficiently sharing resources to provide optimal performance for large numbers of users.

The Oracle Database shares resources among many different user processes. The Oracle Database uses many different strategies for utilizing resources, including highly intelligent use of memory cache for rapid access to data, methods to lower the amount of costly I/O from disk, and, in the case of Exadata, moving processing power to storage components to handle data intensive tasks while simultaneously reducing the amount of data flowing through the server, which saves resources throughout the processing chain.

The Oracle Database Cloud Service takes advantage of all of these performance approaches. Each Oracle Database Cloud Service tenant is handled by a single user process, so the same features that Oracle has developed over decades to provide enterprise class performance for large numbers of users are also utilized in the development and deployment of applications and RESTful Web Services.

## Specifying resource usage

Many other Cloud-based services include multiple options when purchasing a service, based on resources like CPU cycles and memory usage. The Oracle Database Cloud Service uses the enterprise strength resource sharing that has been developed, refined and improved for the Oracle Database over the past 30 years – resource sharing that has made the Oracle Database the leading enterprise-strength database in the world. There is no need to specify CPU cycles or memory, since the Oracle Database already implements sophisticated resource sharing between user processes, allowing for maximum efficiency for all types of operations.

The challenge for tenants of the Oracle Database Cloud is not efficient resource sharing – the Oracle Database handles that. But there is still a potential issue that stems from resource usage – what happens if one user asks for too many resources and affects the resource availability and subsequent performance of other users?

## Limiting resource usage

In order to support a large tenant community, the Oracle Database Cloud Service uses Oracle technology to limit the amount of resources that an individual tenant can use at any one time. The Database Cloud Service limits resources in a way that allow for graceful completion of tasks for long-running operations while still providing the best performance for normal operations.

## Database Resource Manager

The primary tool used by the Oracle Database Cloud Service to limit resource usage is Database Resource Manager. The Database Cloud Service uses two features of Database Resource Instance to protect the performance integrity of tenants – instance caging and consumer groups.

## **Instance caging**

The Database Cloud Service is built using multiple database instances on each Exadata Database Machine. As the name implies, instance caging limits the amount of resources that an individual Oracle Database instance can use. Instance caging guarantees that no one instance can monopolize too many resources for its tenants.

## **Consumer groups**

The multi-tenant architecture of the Oracle Database Cloud means that each instance supports many individual tenants. Each tenant process is always a part of a consumer group and each consumer group is given an upper limit for the percentage of overall resources the group can use. In addition, each consumer group is given a priority, used in deciding how to allocate resources among groups.

Each database instance in the Database Cloud has three basic tenant groups. All processes start out in the consumer group with the highest priority and the largest amount of resources available. If a tenant process exceeds the specified resource limit, such as the amount of CPU time used, the process is moved to the next lower priority consumer group. This next lower group allows for a much larger amount of resource usage, but at a lower priority. If a tenant process exceeds the limits for this consumer group, the process is bumped to the next lower group, with a lower priority and a much higher limit on resource usage. If a tenant process should exceed this limit, the process could be terminated. However, in using a similar consumer group scheme with apex.oracle.com, virtually no processes have been terminated.

The net effect of the use of consumer groups is to insure that most tenant processes are serviced quickly. If a process requests a significantly larger amount of resources, the process will still run, but with a slower response time so as to not adversely affect other tenant processes.

Database Resource Manager consumer groups are implemented in an intelligent way. Although each consumer group has an upper limit on the amount of resources a tenant process can consume, this limit is only relevant if resources are oversubscribed – if the total requests for a resource are greater than the amount of available resources can handle. Consumer group limits are only applied once a resource is oversubscribed. This approach means that consumer groups can exceed their defined limit of a resource if it will not harm other groups by reducing performance through oversubscription, while still insuring that computing resources which are not oversubscribed can be used to their maximum amount.



Oracle Cloud Computing  
September 2012  
Author: Rick Greenwald

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](http://oracle.com)



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

Copyright © 2012, 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. 0110