Oracle® Database

Quality of Service Management User's Guide

12*c* Release 1 (12.1)

E49047-06

November 2015

ORACLE

Oracle Database Quality of Service Management User's Guide , 12c Release 1 (12.1)

E49047-06

Copyright © 2014, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Janet Stern

Contributing Authors: Mark Scardina

Contributors: Troy Anthony, Mark Bauer, Janet Blowney, Annie Flint, Barry Gleason, Roland Knapp, Barb Lundhild, John McHugh, Markus Michalewicz, Ganesh Padmanabhan, Daniel Semler, Peter Sharman, Richard Strohm, Mark Townsend, Li Weng, Douglas Williams

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

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. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface ix

Audience ix
Related Documents ix
Conventions ix

Changes in This Release for Oracle Database Quality of Service Management xi

Changes in Oracle Database QoS Management 12c Release 1 (12.1.0.2) xi
New Features xi
Changes in Oracle Database QoS Management 12c Release 1 (12.1.0.1) xii
New Features xii
Deprecated Features xiii
Desupported Features xiii
Other Changes xiii

1 Introduction to Oracle Database QoS Management

What Is Oracle Database QoS Management? 1-1
Benefits of Using Oracle Database QoS Management 1-2
Overview of Oracle Database QoS Management 1-2
How Does Oracle Database QoS Management Work? 1-3
Overview of Policy Sets 1-5
Overview of Server Pools 1-7
How Server Pools Are Used by Oracle Database QoS Management 1-9
Overview of Performance Classes 1-10
Overview of Performance Policies and Performance Objectives 1-12
How Oracle Database QoS Management Collects and Analyzes Performance Data 1-16
Overview of Recommendations 1-17
What Does Oracle Database QoS Management Manage? 1-22
Managing Database Resources to Meet Service Levels 1-23
High Availability Management and Oracle Database QoS Management 1-26
Overview of Metrics 1-27
Performance Metrics 1-27
Resource Metrics 1-28

Performance Satisfaction Metrics 1-28
Using Metrics to Identify Performance Issues 1-28

2 Supported Workloads and Strategies

Supported Configurations for Oracle Database QoS Management 2-1
Supported Server Pool Configurations 2-1
Supported Database Configurations 2-2
Supported Service Configurations 2-2
Supported Workload and Objective Types 2-3
Strategies for Creating Classifiers for Performance Classes 2-4
Configuration Strategies for Effective Resource Management 2-5
About Resource Bottlenecks 2-5
CPU Resource Bottlenecks 2-5
Configuration Recommendations for Global Cache Resource Bottlenecks 2-6
Configuration Recommendations for I/O Resource Bottlenecks 2-6
Configuration Recommendations for Other Types of Bottlenecks 2-6
Sample Implementation of Oracle Database QoS Management 2-6
Description of the Demo System 2-6
Description of the System Workload 2-7
Initial Oracle Database QoS Management Configuration 2-8
Creating Oracle Database QoS Management Performance Policies for the Demo System 2-9
Managing Service Levels with Oracle Database QoS Management 2-10

3 Installing and Enabling Oracle Database QoS Management

Configuring Oracle Database QoS Management to Manage Oracle Database Workloads 3-1
Installing and Configuring Oracle Grid Infrastructure for a Cluster 3-2
Creating and Configuring Server Pools 3-2
Creating and Configuring an Oracle RAC Database 3-2
Creating Oracle Database QoS Management Administrator Accounts 3-3
Enabling Oracle Database QoS Management 3-4

4 Administering the Oracle Database QoS Management System

Determining If Oracle Database QoS Management is Enabled 4-1
Checking the Enabled Status for a Database 4-1
Checking the Enabled Status for the Cluster 4-1
Monitoring Performance with Oracle Database QoS Management 4-2
Using the Oracle Database QoS Management Dashboard 4-3
Accessing the Oracle Database QoS Management Dashboard 4-3
Enabling Oracle Database QoS Management for a Cluster 4-5
Disabling Oracle Database QoS Management for a Cluster 4-5
Interpreting the Performance Overview Graphs 4-5
Viewing Recommendations 4-7
Viewing Recommendation Details 4-8

Implementing Recommendations 4-9
Administering the Policy Set 4-9
Editing a Policy Set 4-10
Adding Server Pools to a Policy Set 4-13
Modifying Server Pool Settings 4-13
Adding Database Services to a Policy Set 4-14
Updating a Policy Set to Include a New Database 4-14
Managing Performance Classes 4-15
Creating a Performance Class 4-15
Deleting a Performance Class 4-17
Renaming a Performance Class 4-17
Editing an Existing Performance Class 4-18
Specifying the Evaluation Order of the Classifiers 4-19
Managing Performance Policies 4-20
Creating a Performance Policy and Specifying Performance Objectives 4-20
Editing an Existing Performance Policy 4-22
Copying a Performance Policy 4-23
Setting the Current Performance Policy 4-23
Deleting a Performance Policy 4-24
Specifying Authorized Actions for a Performance Policy 4-25
Setting Server Pool Directive Overrides 4-25
Reviewing Performance Metrics 4-26
Viewing Performance Metrics for All Performance Classes 4-26
Viewing Performance Metrics for Individual Performance Classes 4-28
Configuring Alerts for Quality of Service Management Events 4-28
Viewing the Resource Wait Times Breakdown 4-29
Creating Administrative Users for Oracle Database QoS Management 4-30
QOSCTL Utility Reference 4-30
Editing the Resource Plan for Oracle Database QoS Management 4-32

5 Troubleshooting Oracle Database QoS Management

Common Problems 5-1
Cannot Enable Oracle Database Quality of Service Management 5-1
Cannot Enable Oracle Database QoS Management for a Database 5-1
Oracle Database Resource Manager Not Enabled and Resource Plan Errors 5-2
Do Not Have Access to a Server Pool 5-2
Server Pool Is Marked As Unmanageable 5-3
Metrics Are Missing For a Performance Class 5-3
Oracle Database QoS Management is not Generating Recommendations 5-4
Recently Added Server was Placed in the Wrong Server Pool 5-4
RMI Port Conflict Detected 5-4
Locating Log or Trace Files 5-4
Enabling Tracing 5-5

Glossary

Index

List of Figures

1-1 Instance Caging and CPU Slices 1-4
1-2 Elements of an Oracle Database QoS Management Policy Set 1-6
1-3 Diagram of Server Pools, Oracle Databases, and Database Services 1-7
1-4 Server Pools and CPU Slices 1-8
1-5 Baseline Resource Management by Performance Policy 1-14
1-6 Diagram of Oracle Database QoS Management Server Architecture 1-17
1-7 Example of the Analysis for a Recommended Action 1-21
2-1 Illustration of a Sample Workload 2-8
4-1 Performance Satisfaction Metrics for Measure-Only Performance Classes 4-2
4-2 Oracle Database Quality of Service Management Dashboard 4-5
4-3 Example of the Performance Overview Charts 4-5
4-4 Performance Overview Page Showing Performance Classes Not Meeting Their Performance Objectives 4-7

Preface

Welcome to *Oracle Database Quality of Service Management User's Guide*. This document describes the concepts of Oracle Database Quality of Service Management (Oracle Database QoS Management), and instructs you on how to configure, administer, and troubleshoot Oracle Database QoS Management. Oracle Database QoS Management monitors resource use and wait times, and uses this information to manage the resources that are shared across applications. Oracle Database QoS Management makes and implements recommendations that adjust the system configuration to keep the applications running at the specified performance levels.

Audience

Oracle Database Quality of Service Management User's Guide is intended for cluster, database, and system administrators who perform the following tasks:

- Provision and deploy the hardware
- Install the Oracle Grid Infrastructure software and allocate shared resources
- Manage the performance of diverse workloads in a shared environment
- Manage workloads when failures occur

Related Documents

For more information, see the following documents in the Oracle Database 12*c* (12.1) documentation set:

- *Oracle Grid Infrastructure Installation Guide for Linux* or other supported platform
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database Concepts*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in This Release for Oracle Database Quality of Service Management

This chapter lists changes in Oracle Database Quality of Service Management:

- Changes in Oracle Database QoS Management 12*c* Release 1 (12.1.0.2)
- Changes in Oracle Database QoS Management 12*c* Release 1 (12.1.0.1)

Changes in Oracle Database QoS Management 12*c* Release 1 (12.1.0.2)

The following are changes in *Oracle Database Quality of Service Management User's Guide* for Oracle Database 12*c* Release 1 (12.1.0.2):

New Features

- Authorized Automatic Actions

  This features enables you to specify authorized automatic actions that Oracle Quality of Management Service can perform on a per resource and per policy basis. Specifying an authorized automatic action eliminates the requirement for a human response to a recommended action, and thus improves response time to react to a demand surge or node failure that puts SLAs at risk.

  See "Specifying Authorized Actions for a Performance Policy"

- Memory Guard Does Not Require Oracle Database QoS Management to be Active

  With this release, Memory Guard is enabled by default independent of whether you use Oracle Database QoS Management. Memory Guard detects memory stress on a node and causes new sessions to be directed to other instances until the existing workload drains and frees memory. When free memory increases on the node, then services are enabled again to automatically accept new connections.

  See *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Memory Guard.

- Performance-Related Alerts Generated When Running Oracle Database QoS Management in Monitoring Mode

  In the Enterprise Manager Policy Editor, you can now set a non-zero performance objective (PO) for a performance class and also check the Measure-Only box. In earlier releases, the PO value was ignored if Measure-Only was selected, preventing administrators from measuring how well their workload performed against a performance objective.

In this release, if a positive value is specified as a PO and the Measure-Only box is checked, then Oracle Database QoS Management calculates the Performance Satisfaction Metrics (PSMs); if the PO is violated, then Oracle Database QoS Management identifies the resource and generates an alert.

See "Overview of Metrics" for more information.

- Rank Order of Databases in an Active Oracle Database QoS Management Performance Policy Determine Oracle RAC Database Start Order

  When a user created Oracle Database QoS Management policy is active, then the ranked order of the performance classes determines which Oracle RAC databases start first or request real-time LMS process slots. Using the Max(Ranks) across all Performance Classes within a database provides a consistent expression of the expressed business criticality of each database.

- Support for Administrator-Managed Databases in Measure-Only Mode

  Oracle Quality of Management Service supports the measuring and monitoring of database workloads for administrator-managed databases. This significantly enhances real-time monitoring of performance and bottleneck identification without needing a thorough understanding of database schema or database internals.

Changes in Oracle Database QoS Management 12*c* Release 1 (12.1.0.1)

The following are changes in *Oracle Database Quality of Service Management User's Guide* for Oracle Database 12*c* Release 1 (12.1.0.1):

New Features

- MemoryGuard

  MemoryGuard protects a node from eviction or reboot due to memory stress, improving the high availability of the cluster impacted by a high workload load or login storm.

- Support for Server Pool Category definitions and integration with CRS Policy Sets

  Oracle Quality of Management Service supports the use of server pool categorization. This allows you to manage servers dynamically using server pools by identifying servers distinguished by particular attributes which is especially beneficial to clusters made up of heterogeneous node.

  See *Oracle Clusterware Administration and Deployment Guide* for more information about server categories and server pools.

- Support for Monitoring CDBs and PDBS

  Oracle QoS Management can be used with container databases (CDBs) and pluggable databases (PDBs) in measure-only mode..

  See *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for more information about container databases and pluggable databases.

- Custom Resource Plans Supported in Measure-Only Mode

  You can now use custom Database Resource Manager plans with Oracle Quality of Service Management if the database is set for Measure-Only mode.

- Grid Management Repository is a mandatory requirement for Oracle Quality of Service Management

  Oracle Quality of Service Management for Oracle Database 12c requires the Grid Management Repository to be configured when installing or upgrading to Oracle Clusterware 12*c*.

Deprecated Features

- The `-checkpasswd` option of the QOSCTL utility

Desupported Features

Some features previously described in this document are desupported in Oracle Database 12*c* Release 1. See *Oracle Database Upgrade Guide* for a list of desupported features.

Other Changes

- Oracle Enterprise Manager Database Control is not included in this release

  In previous Oracle Database releases, Oracle Enterprise Manager Database Control was the primary database management tool described in this manual. Now, this book describes Oracle Enterprise Manager Cloud Control 12*c* as the primary interface.

1

Introduction to Oracle Database QoS Management

This chapter provides an overview of Oracle Database Quality of Service Management (Oracle Database QoS Management). This chapter includes the following sections:

- What Is Oracle Database QoS Management?
- Benefits of Using Oracle Database QoS Management
- Overview of Oracle Database QoS Management
- What Does Oracle Database QoS Management Manage?
- Overview of Metrics

What Is Oracle Database QoS Management?

Many companies are consolidating and standardizing their data center computer systems. Instead of using individual servers for each application, they run multiple applications on clustered databases. Also, the migration of applications to the Internet has introduced the problem of managing an open workload. An open workload is subject to demand surges, which can overload a system, resulting in a new type of application failure that cannot be fully anticipated or planned for. To keep applications available and performing within their target service levels in this type of environment, you must:

- Pool resources.
- Have management tools that detect performance bottlenecks in real time.
- Reallocate resources to meet the change in demand.

Oracle Database QoS Management is an automated, policy-based product that monitors the workload requests for an entire system. Oracle Database QoS Management manages the resources that are shared across applications, and adjusts the system configuration to keep the applications running at the performance levels needed by your business. Oracle Database QoS Management responds gracefully to changes in system configuration and demand, thus avoiding additional oscillations in the performance levels of your applications.

Oracle Database QoS Management monitors the performance of each work request on a target system. Oracle Database QoS Management starts to track a work request from the time a work request requests a connection to the database using a database service. The amount of time required to complete a work request, or the response time (also known as the end-to-end response time, or round-trip time), is the time from when the request for data was initiated and when the data request is completed. By accurately measuring the two components of response time, which are the time spent using

resources and the time spent waiting to use resources, Oracle Database QoS Management can quickly detect bottlenecks in the system. Oracle Database QoS Management then makes suggestions to reallocate resources to relieve a bottleneck, thus preserving or restoring service levels.

Oracle Database QoS Management manages the resources on your system so that:

- When sufficient resources are available to meet the demand, business-level performance requirements for your applications are met, even if the workload changes.
- When sufficient resources are *not* available to meet the demand, Oracle Database QoS Management attempts to satisfy the more critical business performance requirements at the expense of less critical performance requirements.

Benefits of Using Oracle Database QoS Management

In a typical company, when the response times of your applications are not within acceptable levels, problem resolution can be very slow. Often, the first questions that administrators ask are: "Did we configure the system correctly? Is there a parameter change that fixes the problem? Do we need more hardware?" Unfortunately, these questions are very difficult to answer precisely. The result is often hours of unproductive and frustrating experimentation.

Oracle Database QoS Management provides the following benefits:

- Reduces the time and expertise requirements for system administrators who manage Oracle Real Application Clusters (Oracle RAC) resources.
- Helps reduce the number of performance outages.
- Reduces the time needed to resolve problems that limit or decrease the performance of your applications.
- Provides stability to the system as the workloads change.
- Makes the addition or removal of servers transparent to applications.
- Reduces the impact on the system caused by server failures.
- Helps ensure that service-level agreements (SLAs) are met.
- Enables more effective sharing of hardware resources.

Oracle Database QoS Management helps manage the resources that are shared by applications in a cluster. Oracle Database QoS Management can help identify and resolve performance bottlenecks. Oracle Database QoS Management does not diagnose or tune application or database performance issues. When tuning the performance of your applications, the goal is to achieve optimal performance. Oracle Database QoS Management does not seek to make your applications run faster, but instead works to remove obstacles that prevent your applications from running at their optimal performance levels.

Overview of Oracle Database QoS Management

This section provides a basic description of how Oracle Database QoS Management works, and of how it evaluates the performance of workloads on your system.

- How Does Oracle Database QoS Management Work?

- Overview of Policy Sets
- Overview of Server Pools
- How Server Pools Are Used by Oracle Database QoS Management
- Overview of Performance Classes
- Overview of Performance Policies and Performance Objectives
- How Oracle Database QoS Management Collects and Analyzes Performance Data
- Overview of Recommendations

How Does Oracle Database QoS Management Work?

With Oracle Database, you can use services to manage the workload on your system by starting services on groups of servers that are dedicated to particular workloads. At the database tier, for example, you could dedicate one group of servers to online transaction processing (OLTP), dedicate another group of servers to application testing, and dedicate a third group of servers for internal applications. The system administrator can allocate resources to specific workloads by manually changing the number of servers on which a database service is allowed to run.

Using groups of servers in this way isolates the workloads from each other to prevent demand surges, failures, and other problems in one workload from affecting the other workloads. However, in this type of deployment, you must separately provision the servers to each group to satisfy the peak demand of each workload because resources are not shared.

Oracle Database QoS Management performs the following actions:

1. Uses a policy created by the QoS administrator to do the following:
   - Assign each work request to a Performance Class by using the attributes of the incoming work requests (such as the database service to which the application connects).
   - Determine the target response times (Performance Objectives) for each Performance Class.
   - Determine which Performance Classes are the most critical to your business
2. Monitors the resource usage and resource wait times for all the Performance Classes.
3. Analyzes the average response time for a Performance Class against the Performance Objective in effect for that Performance Class
4. Produces recommendations for reallocating resources to improve the performance of a Performance Class that is exceeding its target response time, and provides an analysis of the predicted impact to performance levels for each Performance Class if that recommendation is implemented.
5. Implements the actions listed in the recommendation when directed to by the Oracle Database QoS Management administrator, then evaluates the system to verify that each Performance Class is meeting its Performance Objective after the resources have been reallocated.

Oracle Database QoS Management and Server Pools

You can use server pools to create groups of servers within a cluster to provide workload isolation. A server can only belong to one server pool at any time. You can create an Oracle Database in a single server pool, or across multiple server pools. Oracle Database QoS Management can make recommendations to move a server from one server pool to another based on the measured and projected demand. Oracle Database QoS Management can also relocate servers to satisfy the Performance Objectives currently in effect.

See Also:

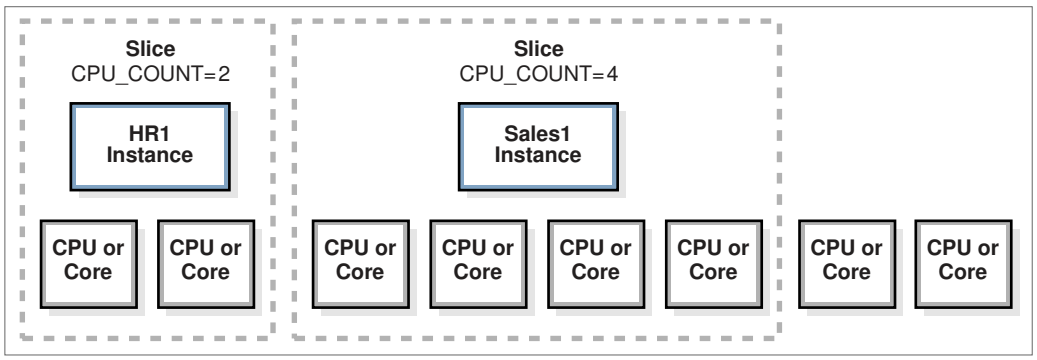
Oracle Clusterware Administration and Deployment Guide for more information about server pools

---

Oracle Database QoS Management and Instance Caging

When multiple database instances share a single server, they must share its CPU, memory, and I/O bandwidth. Instance Caging limits the amount of CPU an Oracle database instance consumes by using the Oracle Database Resource Manager and the `CPU_COUNT` database initialization parameter. When using Oracle Database QoS Management, the sum of the values for `CPU_COUNT` for all instances of the server must be less than or equal to the total number of physical CPUs. Also, each CPU partition, or **slice**, must be uniform in thickness (number of CPUs) for each instance of a database within a server pool. These requirements help to ensure predictable and isolated performance of each database.

Figure 1-1 Instance Caging and CPU Slices



When you implement instance caging, Oracle Database QoS Management can provide recommendations to reallocate CPU resources from one slice to another slice within the same server pool. If you choose to implement the recommendation to modify the instance caging settings, then Oracle Database QoS Management modifies the `CPU_COUNT` parameter uniformly for all the database instances running on servers in the server pool.

Modifying the `CPU_COUNT` parameter and configuring Oracle Database QoS Management so that a Resource Plan is activated enables the Instance Caging feature. When you use Instance Caging to constrain CPU usage for an instance, that instance could become CPU-bound. This is when the Resource Manager begins to do its work, allocating CPU shares among the various database sessions according to the active resource plan.

Oracle Database QoS Management and Services

In an Oracle RAC cluster, Oracle Database QoS Management monitors the server pools and nodes on which the database services are offered. A service can run in only one server pool. If the database spans multiple server pools, then you must create multiple services to access the instances in all server pools.

Workload is monitored for clients and applications that connect to the database using database services that are managed by Oracle Clusterware. The connections must use Java Database Connectivity (JDBC) (thick or thin), or Oracle Call Interface (OCI). Connections should use services with its run-time goal for the load balancing advisory set to `SERVICE_TIME`, and the connection load balancing goal set to `LONG`. For example:

```
srvctl modify service -db db_name -service service_name -rlbgoal SERVICE_TIME
-clbgoal LONG -cardinality UNIFORM
```

You must define the cardinality of database services as follows:

- If the server pool that the service runs in has a maximum size greater than 1 (or UNLIMITED), then set the cardinality of the service to `UNIFORM`.
- If the server pool that the service runs in has a maximum size of 1, then set the cardinality of the service to `SINGLETON`.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for more information about services and managing database workloads

Overview of Policy Sets

The central concept in Oracle Database QoS Management is the Policy Set. A Policy Set enables you to specify your resources, Performance Classes (workloads), and one or more Performance Policies that specify the Performance Objective for each Performance Class. A Policy Set can also specify constraints for resource availability. Oracle Database QoS Management Performance Policies manage the availability of resources systemwide for each Performance Class so that the system is able to satisfy the Performance Objectives you set in the Performance Policy.

When you use Oracle Enterprise Manager to create a new Default Policy for your system, Oracle Database QoS Management provides default classification rules and associated Performance Class names. For example, when you create the initial Policy Set, Oracle Database QoS Management discovers all database services in a cluster, and creates a Performance Class for each service. The Performance Class is named by appending `_pc` to the service name. For example, if the name of a service is `sales`, then the name assigned to the Performance Class for that service is `sales_pc`.

Only one Performance Policy in the Policy Set can be active at any time. You can activate Performance policies to respond to particular requirements by using calendar schedules, maintenance windows, events, and so on. For more information about Performance Policies, see "Overview of Performance Policies and Performance Objectives".

When you create a Policy Set, you specify which server pools in the cluster should be managed by Oracle Database QoS Management. You also define Performance Classes (used to categorize workloads with similar performance requirements). You then

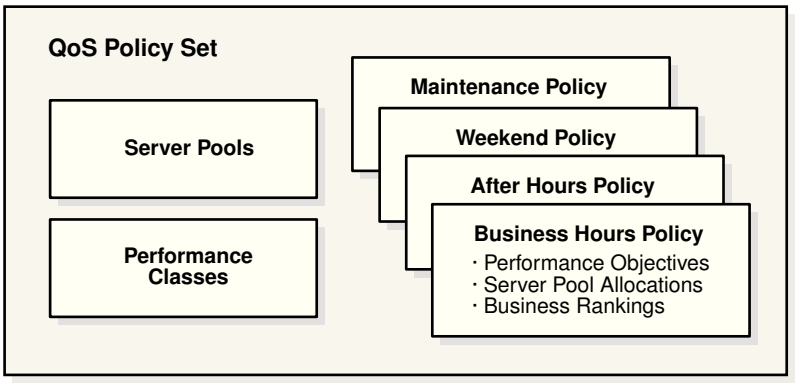
create a Performance Policy to specify which Performance Classes have the highest priority and the Performance Objectives of each Performance Class. To satisfy the Performance Objectives, Oracle Database QoS Management makes recommendations for reallocating resources when needed, and predicts what effect the recommended actions will have on the ability of each Performance Class to meet its Performance Objective.

For example, you could create a policy to manage your application workloads during business hours. The applications used by customers to buy products or services are of the highest priority to your business during this time. You also give high priority to order fulfillment and billing applications. Human resource and enterprise resource planning (ERP) applications are of a lower priority during this time. If your online sales applications experience a surge in demand, then Oracle Database QoS Management may recommend that more resources be allocated to the sales applications and taken away from applications of lesser importance. The recommendation also includes a prediction of the change in performance (positive or negative) for each Performance Class. See "Overview of Recommendations" for more information about recommendations.

A Policy Set, as shown in Figure 1-2, consists of the following:

- The server pools that are being managed by Oracle Database QoS Management
- Performance Classes, which are groups of work requests with similar performance objectives
- Performance policies, which describe how resources should be allocated to the Performance Classes by using:
  - Performance Objectives
  - Performance Class ranks
  - server pool directive overrides

Figure 1-2 Elements of an Oracle Database QoS Management Policy Set



See Also:

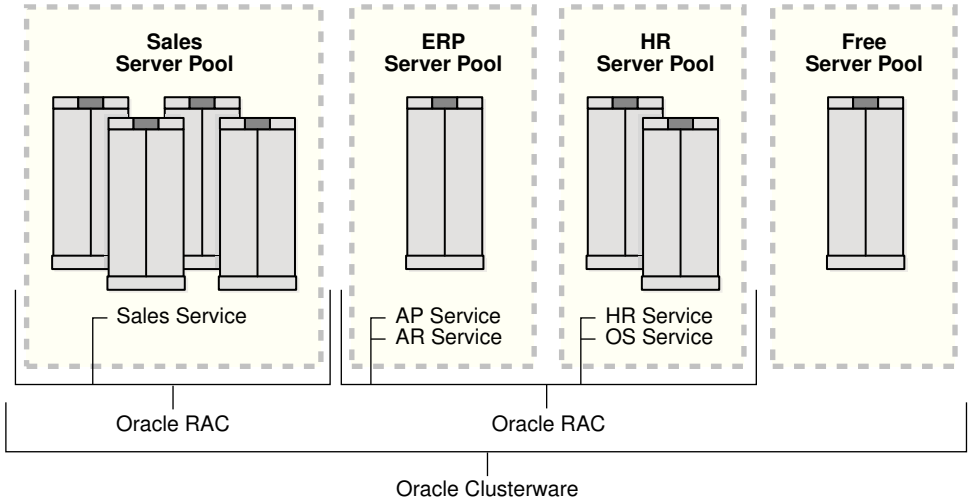
- "Applying Classifiers to Work Requests"
- "Create an Initial Policy Set"
- "Administering the Policy Set"

Overview of Server Pools

When deciding how many clusters to create for your business, you need to compare the possible cost savings through consolidation of servers with the risk that the consolidated workloads will interfere with each other in some significant way. With the introduction of server pools to logically divide a cluster, you can achieve the benefit of physical consolidation and resource agility while maintaining workload isolation.

As the administrator, you can define the workloads that can run in various server pools, as shown in Figure 1-3. Applications that connect to your Oracle RAC database use a service that runs only on the servers currently allocated to that server pool. For example, in Figure 1-3, connections and applications that use the OS service access only the servers in the HR server pool, so that work done by those connections does not interfere with the applications using the Sales service. Oracle Database QoS Management can assist you with managing the resource allocations within each of those groups to meet your service levels, and can redistribute resources automatically to meet changes in your business requirements.

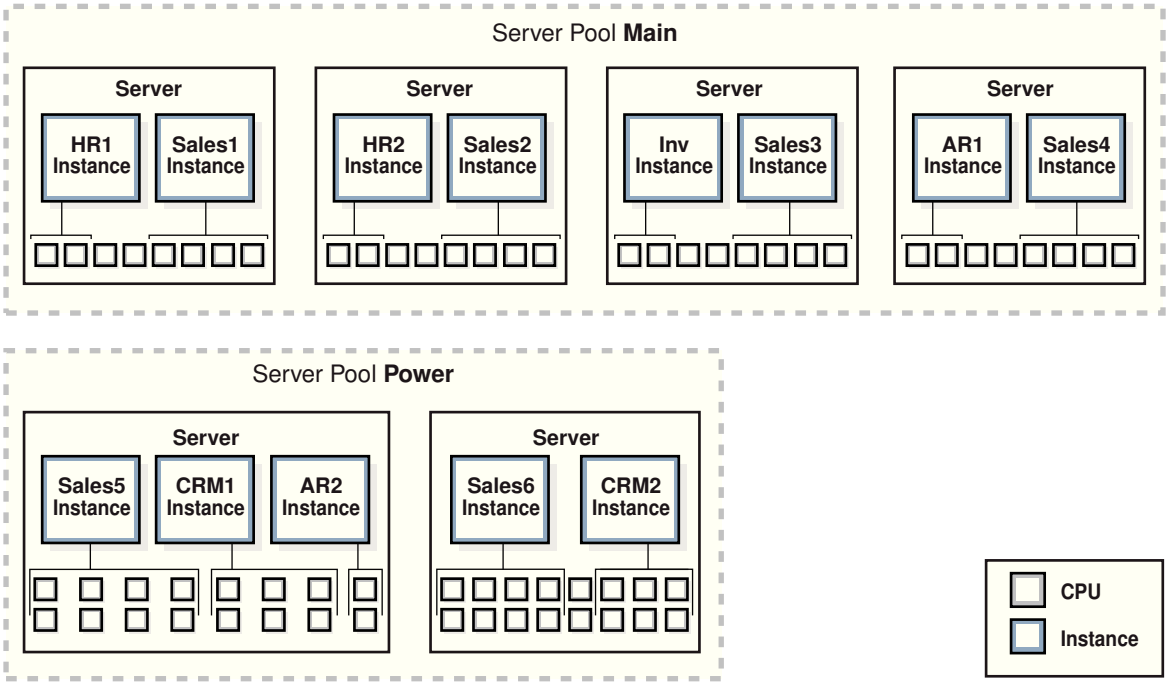
Figure 1-3 Diagram of Server Pools, Oracle Databases, and Database Services



With server pools, you can now create groups of servers that can be managed as a single entity. Databases can be created to run in these server pools. If each server runs only a single instance of a database, then if the database needs more resources, an additional server can be allocated to the server pool. If multiple database instances run on a single server, then they must compete for the shared resources of that server, such as memory and CPU. If one of the database instances is experiencing a much higher workload than the other instances, then that database instance can significantly degrade the performance of the other instances running on the same server.

You can use Instance Caging to limit the amount of CPU an Oracle Database instance consumes. By setting the `CPU_COUNT` parameter to limit the maximum number of CPUs an instance can use, you partition the CPUs among the database instances on a server, thus preventing them from using excessive amounts of CPU resources. The `CPU_COUNT` setting must be the same for each instance of a database within a server pool. Oracle Database QoS Management can monitor the CPU usage among all the database instances in the server pool, and recommend changes to the current settings if needed.

Figure 1-4 Server Pools and CPU Slices



Instead of using a single server pool, Oracle recommends that you manage services by creating multiple server pools, and relocate services between them. Using this configuration provides the following benefits:

- Different types of workloads require different configurations, and have different tuning goals. For example, a customer using your OLTP applications to purchase goods or services expects the shipping and payment information screens to respond quickly. If your applications take too long to process the order, then the customer may lose interest, and your company may lose a sale. By contrast, an employee using an internal HR application is motivated to continue using HR screens, even if they do not respond quickly. If your HR applications take longer than expected to process an online task, the employee is unlikely to quit.

- Applications can have various resource requirements throughout the day, week, or month to meet your business objectives. You can use server pools to divide the resources among the application workloads. To meet the Performance Objectives of a given time period, you could use server pool directive overrides in a Performance Policy to change the default attributes (such as Max or Min) for a server pool.

  For example, if your company has an online tax filing application, then the application must prepare and file the tax statements for your customers before the government-specified deadline. In the timeframe immediately preceding a filing deadline, applications related to tax statement preparation and filing require more resources than they do at other times of the year. To ensure you meet this service requirement, you can create a Performance Policy named QuarterlyFilings to override the standard server pool directive, and specify that when QuarterlyFilings is active, the server pool used by the tax preparation applications should have a minimum of four servers instead of two to handle the additional workload. When the QuarterlyFilings Performance Policy is not in effect, your default Performance Policy is in effect, and the minimum number of servers in that server pool is two.

- Because Oracle Database QoS Management regulates the number of servers that support a workload, application users experience a consistent level of performance, even in the presence of changing demand levels. This prevents performance expectations of your customers from being reset when workload levels change from low to high demand.

  For example, assume your company sells a new consumer product that is in high demand, and your company advertises that they have large quantities of that product for sale at a reduced price. As a result, many new customers create orders for this product, and your OLTP applications must process a rapidly increasing number of transactions (a demand surge occurs). New customers do not know what to expect with regards to the OLTP application peformance. However, existing customers may react negatively if their online shopping experience is impacted by the flood of new customers. Also, if your OLTP application cannot process all the incoming orders, then some of the new customers may quit the application and place their order with a different company, or visit a retail store instead.

  Oracle Database QoS Management helps you to manage the reallocation of available resources to meet the demand surge without sacrificing the quality of service of your other applications.

- Some workloads do not scale well but still benefit from the high availability of a cluster environment. Deploying these workloads in a fixed-size server pool offers both performance manageability and high availability.

  For example, if you run an ERP application in a server pool with a fixed size of one server, then the maximum size of the server pool and the minimum size of the server pool are both set to one. If the server in that server pool fails, then Oracle Clusterware automatically allocates a new server to the server pool to maintain the minimum size of one server. Any instances and services located on the failed server are restarted on the new server, so the applications using these instances and services remains available.

Note:

- *Oracle Clusterware Administration and Deployment Guide* for more information about server pools
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about policy-managed Oracle RAC databases
- *Oracle Database Concepts* for a description of an OLTP system

How Server Pools Are Used by Oracle Database QoS Management

When you first install Oracle Grid Infrastructure for a cluster, a default server pool (the Free pool) is created. All servers are initially placed in this server pool. You should create one or more server pools depending on the workloads that need to be managed. When you create a new server pool, the servers that you assign to that server pool are automatically moved out of the Free pool and placed in the newly created server pool. At this point, you can install a database to run in that server pool, and create database services that are managed by Oracle Clusterware for applications to connect to that database.

For an Oracle RAC database to take advantage of the flexibility of server pools, the database must be created using the policy-managed deployment option, which places the database in one or more server pools. Upgraded Oracle databases are converted directly to administrator-managed databases, and should be separately migrated to policy-managed databases. See *Oracle Real Application Clusters Administration and Deployment Guide* for more information about changing an administrator-managed database to a policy-managed database.

Starting with Oracle Database 12*c* Release 1 (12.1.0.2), you can monitor administrator-managed databases in measure-only or monitor mode.

Caution:

If you use candidate server lists (`server_names` attribute) or categories when creating server pools, then the ability for Oracle Database Quality of Service (QoS) Management to expand the server pool will be limited by those restrictions as non-eligible servers cannot be used.

See Also:

- "Adding Server Pools to a Policy Set"
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX* (or other platforms) for more information about creating a policy-managed database
- "Monitoring Performance with Oracle Database QoS Management"

Overview of Performance Classes

A Policy Set contains Performance Objectives for various Performance Classes, or workloads, that run on your cluster. Oracle Database QoS Management uses a set of classification rules defined in the Policy Set to categorize work requests into a Performance Class. The fundamental classifier used to assign work requests to Performance Classes is the name of the service that is used to connect to the database.

- Performance Class Tags
- Applying Classifiers to Work Requests
- Using Additional Filters for Classifying Work Requests
- Deciding to Create New Performance Classes

Performance Class Tags

The classification of work requests applies a user-defined name (tag) that identifies the Performance Class to which the work request belongs. All work requests that are grouped into a particular Performance Class have the same performance objectives. In effect, the tag connects the work request to the Performance Objective for the associated Performance Class. Oracle Database QoS Management assigns tags to each work request so that every component of the system can take measurements, and provide data to Oracle Database QoS Management for evaluation against the applicable Performance Objectives.

Applying Classifiers to Work Requests

Classification occurs wherever new work enters the system. When a work request arrives at a server, the work request is checked for a tag. If the work request has a tag, then the server concludes that this work request has already been classified, and the tag is not changed. If the work request does not include a tag, then the classifiers are checked, and a tag for the matching Performance Class is attached to the work request.

To illustrate how work requests are classified, consider an application that connects to an Oracle RAC database. The application uses the database service `sales`. The Oracle Database QoS Management administrator specified during the initial configuration of Oracle Database QoS Management that the `sales_pc` Performance Class should contain work requests that use the `sales` service. When a connection request is received by the database, Oracle Database QoS Management checks for a tag. If a tag is not found, then Oracle Database QoS Management compares the information in the connection request with the classifiers specified for each Performance Class, in the order specified in the Performance Policy. If the connection request being classified is using the `sales` service, then when the classifiers in the `sales_pc` Performance Class are compared to the connection request information, a match is found, and the database work request is assigned a tag for the `sales_pc` Performance Class.

Using Additional Filters for Classifying Work Requests

A single application can support work requests of many types, with a range of performance characteristics. By extending and refining the default classification rules, the Oracle Database QoS Management administrator can write multiple Performance Objectives for a single application. For example, the administrator may decide that a web-based application should have separate Performance Objectives for work requests related to logging in, browsing, searching, and purchasing.

Oracle Database QoS Management supports user-defined combinations of connection parameters to map Performance Classes to the actual workloads running in the database. These connection parameters belong to two general classes, and can be combined to create fine-grained Boolean expressions:

- **Configuration Parameters**—The supported configuration parameters are `SERVICE_NAME` and `USERNAME`. Each classifier in a Performance Class must specify the name of a database service. Additional granularity can be achieved by identifying the name of the user that is making the database connection from either a client or the middle tier. The advantage of using these classifiers is that they do not require application code changes to associate different workloads with separate Performance Classes.

- **Application Parameters**—The supported application parameters are `MODULE`, `ACTION`, and `PROGRAM`. These are optional parameters. The values for `MODULE` and `ACTION` must be set within the application. Depending on the type of application, you can set these parameters as follows:

  – OCI—Use `OCI_ATTR_MODULE` and `OCI_ATTR_ACTION`.

  – Oracle Data Provider for .NET (ODP.NET)—Specify the `ModuleName` and `ActionName` properties on the `OracleConnection` object.

  – JDBC—Set `MODULE` and `ACTION` in `SYS_CONTEXT`.

The `PROGRAM` parameter is set or derived differently for each database driver and platform. Consult the appropriate Oracle Database developer's guide for further details and examples.

To manage the workload for an application, the application code makes database connections using a particular service. To provide more precise control over the workload generated by various parts of the application, you can create additional Performance Classes, and use classifiers that include `PROGRAM`, `MODULE`, or `ACTION` in addition to the service or user name. For example, you could specify that all connections to your cluster that use the `sales` service belong to the `sales_pc` Performance Class, but connections that use the `sales` service and have a user name of `APPADMIN` belong to `sales_admin` Performance Class.

See Also:

- *Oracle Call Interface Programmer's Guide*
- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows*
- *Oracle Database JDBC Developer's Guide*

Deciding to Create New Performance Classes

The Performance Classes in use at a particular data center are expected to change over time. For example, you may need to modify the Performance Objectives for one part of your application. In this case you would create a new Performance Class with additional classifiers to identify the target work requests, and update your Performance Policy to add a new Performance Objective for this Performance Class. In other words, you replace a single Performance Objective with one or more finer-grained Performance Objectives, and divide the work requests for one Performance Class into multiple Performance Classes.

Application developers can suggest which Performance Classes to use. Specifically, an application developer can suggest ways to identify different application workloads, and you can use these suggestions to create classifiers for Performance Classes so that each type of work request is managed separately.

You can create additional Performance Classes to specify acceptable response times for different application workloads. For example, a Performance Objective may indicate that a work request performing the `checkout` action for the `sales_pc_checkout` Performance Class should not take more than one millisecond to complete, and a work request performing the `browse` action for the `sales_pc_browse` Performance Class can take 100 milliseconds second to complete.

See Also:

"Managing Performance Classes"

Overview of Performance Policies and Performance Objectives

To manage the various Performance Objectives, you define one or more Performance Policies. A Performance Policy is a collection of Performance Objectives, and a measure of how critical they are to your business. For example, you could define a Performance Policy for normal business hours, another for weekday nonbusiness hours, one for weekend operations, and another to be used during processing for the quarter-end financial closing. At any given time, a single Performance Policy is in effect as specified by the Oracle Database QoS Management administrator. Within each Performance Policy, the criticalness, or ranking, of the Performance Objectives

can be different, enabling you to give more priority to certain workloads during specific time periods.

A Performance Policy has a collection of Performance Objectives in effect at the same time; there is one or more Performance Objectives for each application or workload that runs on the cluster. Some workloads and their Performance Objectives are more critical to the business than others. Some Performance Objectives may be more critical at certain times, and less critical at other times.

- Overview of Performance Objectives
- Overview of Server Pool Directive Overrides
- Overview of Performance Class Ranks
- Oracle Database QoS Management Policy Workload Criticality Determines Database Startup Order

Overview of Performance Objectives

You create Performance Objectives for each Performance Class to specify the target performance level for all work requests that are assigned to each Performance Class. A Performance Objective specifies both a business requirement (the target performance level) and the work to which that Performance Objective applies (the Performance Class). For example, a Performance Objective could specify that work requests in the `hr_pc` Performance Class should have an average response time of less than 0.2 seconds.

Performance Objectives are specified with Performance Policies. Each Performance Policy includes a Performance Objective for each and every Performance Class, unless the Performance Class is marked Measure-Only. In this release, Oracle Database QoS Management supports only one type of Performance Objective, average response time.

The response time for a workload is based upon database client requests. Response time measures the time from when the cluster receives the request over the network to the time the request leaves the cluster. Response time does not include the time required to send the information over the network to or from the client. The response time for all database client requests in a Performance Class is averaged and presented as average response time, measured as database requests for second.

See Also:

"Creating a Performance Policy and Specifying Performance Objectives"

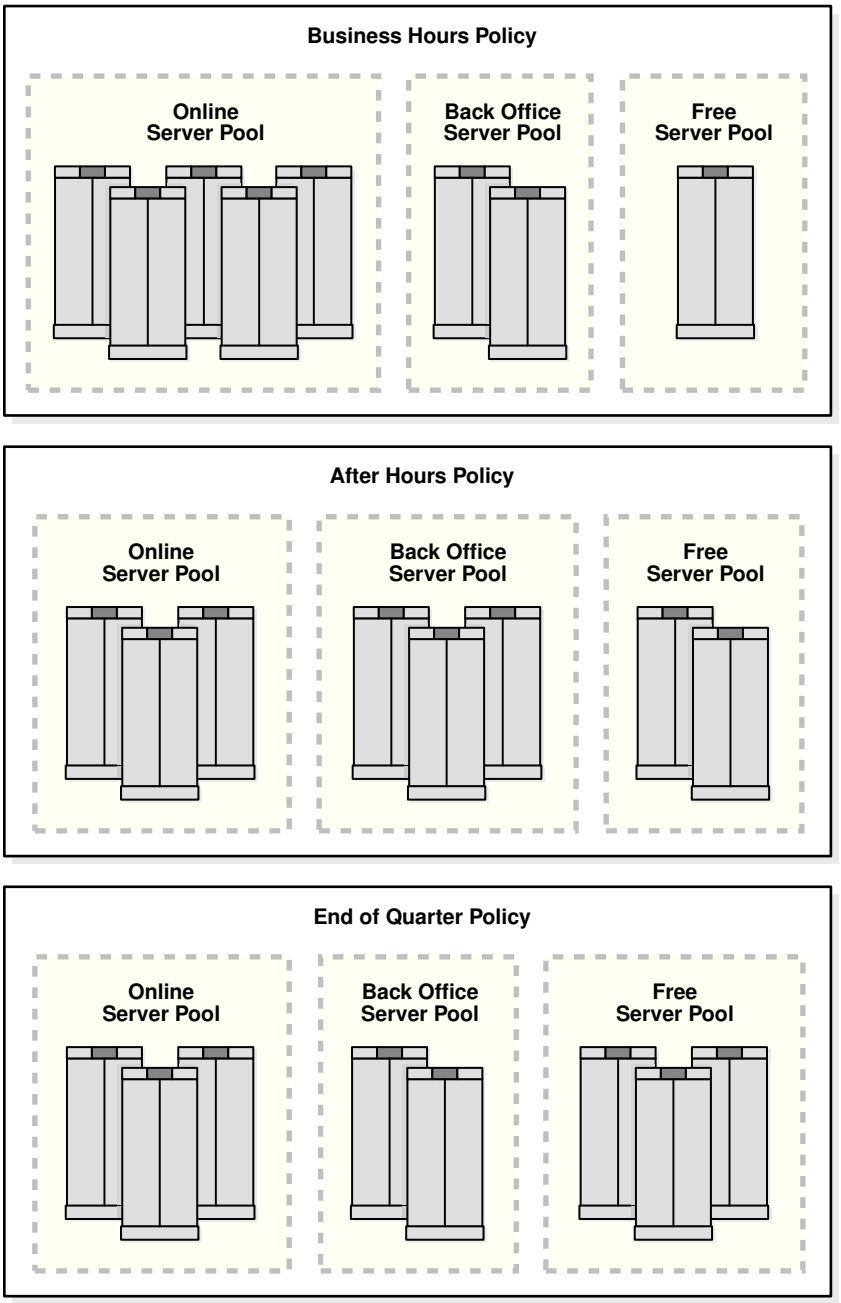
Overview of Server Pool Directive Overrides

A server pool directive override sets the availability properties of Min, Max, and Importance for a server pool when the Performance Policy is in effect.

A Performance Policy can also include a set of server pool directive overrides. Server pool directive overrides serve as constraints on the allocation changes that Oracle Database QoS Management recommends, because the server pool directive overrides are honored during the activation period of the Performance Policy. For example, Oracle Database QoS Management never recommends moving a server out of a server pool if doing so results in the server pool having less than its specified minimum number of servers.

You could create Performance Policies for your system to manage workload based on the time of year or time of day, as shown in Figure 1-5. Under normal conditions, these Performance Policies keep your database workload running at a steady rate. If the workload requests for a database increase suddenly, then a particular server pool may require additional resources beyond what is specified by the Performance Policy.

Figure 1-5 Baseline Resource Management by Performance Policy



For example, assume your business takes orders over the telephone, and creates orders using a sales application. Your telephone sales department is only open during regular business hours, but customers can also place orders themselves over the Internet. During the day, more orders are placed so the sales applications need more resources to handle the workload. This configuration is managed by creating the Business Hours Performance Policy, and specifying that the Back Office server pool

can have a maximum of two servers, enabling Oracle Database QoS Management to move servers to the Online server pool, as needed. After the telephone sales department closes, the workload for the sales applications decreases. To manage this configuration you create the After Hours Performance Policy and specify that the Back Office server pool can have a maximum of four servers, enabling your internal applications to acquire the additional resources that they need to complete their workloads before the next business day.

In this scenario, the Business Hours and After Hours Performance Policies can contain server pool directive overrides. When a Performance Policy contains a server pool directive override, the current settings of Max, Min, and Importance for the specified server pool are overridden while that Performance Policy is in effect. This enables additional servers to be placed in the Sales server pool to give the online sales applications the resources they need and to limit the resources used by the Back Office server pool, so that its workload does not interfere with the Sales workload.

See Also:

"Setting Server Pool Directive Overrides"

Overview of Performance Class Ranks

Within a Performance Policy, you can also assign a level of business criticalness (a *rank*) to each Performance Class to give priority to meeting the Performance Objectives for a more critical Performance Class over a less critical one. When there are not enough resources available to meet all the Performance Objectives for all Performance Classes at the same time, the Performance Objectives for the more critical Performance Classes must be met at the expense of the less critical Performance Objectives. The Performance Policy specifies the *business criticalness* of each Performance Class, which can be Highest, High, Medium, Low, or Lowest.

Note:

Priority access to resources, based on rank, does not apply to single-instance Oracle RAC databases or Oracle RAC One Node.

For example, using the Performance Policies illustrated in Figure 1-5, when the Business Hours Performance Policy is in effect, the sales applications, which access the Online server pool, have the highest rank. If there are not enough resources available to meet the Performance Objectives of all the Performance Classes, then the applications that use the Online server pool will get priority access to any available resources, even if the applications using the Back Office server pool are not meeting their Performance Objectives.

You can have multiple Performance Classes at the same rank. If Oracle Database QoS Management detects more than one Performance Class not meeting its Performance Objective and the Performance Classes are assigned the same rank in the active Performance Policy, then Oracle Database QoS Management recommends a change to give the Performance Class closest to meeting its Performance Objective more resources. After implementing the recommended action, when the Performance Class is no longer below its target performance level, Oracle Database QoS Management performs a new evaluation of the system performance.

See Also:

"Managing Performance Policies"

Oracle Database QoS Management Policy Workload Criticality Determines Database Startup Order

If a user-created Oracle Database QoS Management policy is active, then the ranked order of the performance classes determines the order in which the associated Oracle RAC databases start or request real-time LMS process slots. Using the performance class rankings ensures that mission critical databases running in a consolidated environment have their LMS processes run in real-time, thus eliminating a resource bottleneck within inter-node communication. Because the Oracle Database QoS Management policy specifies the rank of each workload, using the value of Max(Ranks) for each database provides a consistent expression of the expressed business criticality of each database.

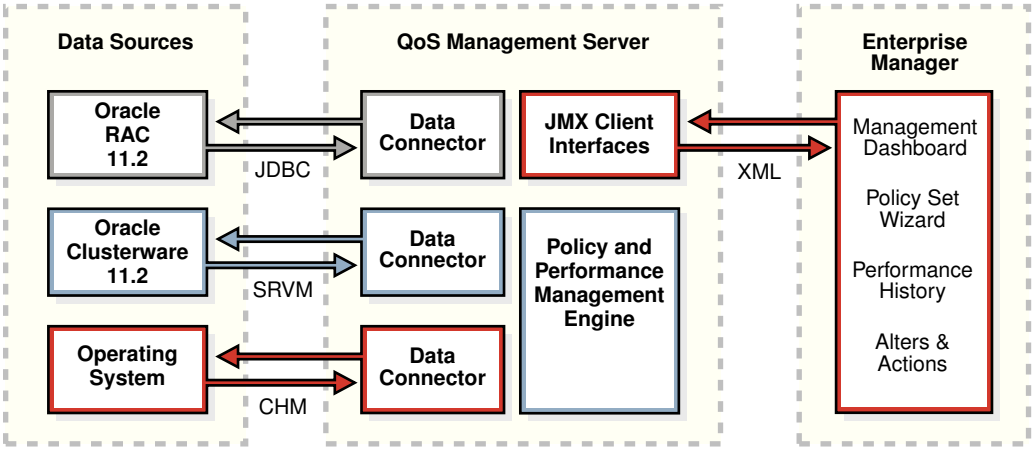
How Oracle Database QoS Management Collects and Analyzes Performance Data

The Oracle Database QoS Management Server retrieves metrics data from each database instance running in managed server pools. The data are correlated by Performance Class every five seconds. The data include many metrics such as database request arrival rate, CPU use, CPU wait time, I/O use, I/O wait time, Global Cache use and Global Cache wait times. Information about the current topology of the cluster and the health of the servers is added to the data. The Policy and Performance Management engine of Oracle Database QoS Management (illustrated in Figure 1-6) analyzes the data to determine the overall performance profile of the system for the current Performance Objectives established by the active Performance Policy.

The performance evaluation occurs once a minute, and results in a recommendation if any Performance Class does not meet its objectives. The recommendation specifies which resource is the bottleneck. Specific corrective actions are included in the recommendation, if possible. The recommendation also includes a listing of the projected impact on all Performance Classes in the system if you decide to implement the recommended action.

Figure 1-6 diagrams the collection of data from various data sources and shows how that information is used by Oracle Enterprise Manager. In this figure, CHM refers to Oracle Cluster Health Monitor and Server Manager (SRVM) is a component of Oracle Clusterware.

Figure 1-6 Diagram of Oracle Database QoS Management Server Architecture



See Also:

- "Using the Oracle Database QoS Management Dashboard"
- "Interpreting the Performance Overview Graphs"
- "Reviewing Performance Metrics"

Overview of Recommendations

If your business experiences periodic demand surges or must support an open workload, then to retain performance levels for your applications you can design your system to satisfy the peak workload. Creating a system capable of handling the peak workload typically means acquiring additional hardware to be available when needed and sit idle when not needed. Instead of having servers remain idle except when a demand surge occurs, you could decide to use those servers to run other application workloads. However, if the servers are busy running other applications when a demand surge hits, then your system may not be able to satisfy the peak workload and your main business applications do not perform as expected. Oracle Database QoS Management enables you to manage excess capacity to meet specific performance goals through its recommendations.

- How Oracle Database QoS Management Generates Recommendations
- Types of Recommendations
- Choosing the Best Recommendation
- Contents of a Recommendation
- Overview of Implementing Recommendations
- Example: How Recommendations Are Generated

How Oracle Database QoS Management Generates Recommendations

When you use Oracle Database QoS Management, your system is continuously monitored in an iterative process to see if the Performance Objectives in the active Performance Policy are being met. Performance data are sent to Oracle Enterprise

Manager for display in the Oracle Database QoS Management Dashboard (the Dashboard) and Performance History pages.

When one or more Performance Objectives are not being met, after evaluating the performance of your system, Oracle Database QoS Management seeks to improve the performance of a single Performance Objective: usually the highest ranked Performance Objective that is currently not being satisfied. If all Performance Objectives are satisfied with capacity to spare for both the current and projected workload, then Oracle Database QoS Management signals "No action required: all Performance Objectives are being met."

See Also:

- "Using the Oracle Database QoS Management Dashboard"
- "Viewing Recommendations"
- "Reviewing Performance Metrics"

Types of Recommendations

If Performance Objectives are not being met for a Performance Class, then Oracle Database QoS Management issues recommendations to rebalance the use of resources to alleviate bottlenecks. Oracle Database QoS Management evaluates several possible solutions and then chooses the solution that:

- Offers the best overall system improvement
- Causes the least system disruption
- Helps the highest ranked violating performance class

The types of recommendations that Oracle Database QoS Management can make are:

- Promoting and Demoting Consumer Groups
- Moving Servers Between Server Pools
- Modifying the CPU Count

Promoting and Demoting Consumer Groups

If Performance Objectives are not being met for a Performance Class, and the Performance Class accesses the same database as other Performance Classes, then Oracle Database QoS Management can recommend consumer group mapping changes. Changing the consumer group mappings gives more access to the CPU resource to the Performance Class that is not meeting is Performance Objective. Oracle Database QoS Management issues consumer group mapping recommendations only for Performance Classes that are competing for resources in the same database and server pool.

Modifying the CPU Count

If you have multiple database instances running on servers in a server pool, Oracle Database QoS Management can recommend that CPU resources used by a database instance in one slice on the server be donated to a slice that needs more CPU resources. If there is a Performance Class that is not meeting its Performance

Objective, and there is another slice on the system that has available headroom, or the Performance Classes that use that slice are of a lower rank, then Oracle Database QoS Management can recommend moving a CPU from the idle slice to the overloaded slice. If this recommendation is implemented, then the `CPU_COUNT` parameter is adjusted downwards for the idle instance and upwards for the overworked instance on all servers in the server pool.

Moving Servers Between Server Pools

Another recommended action that Oracle Database QoS Management can display is to move a server from one server pool to another to provide additional resources to meet the Performance Objectives for a Performance Class. If all the server pools in the cluster are at their specified minimum size, or if the server pool needing the resource is at its maximum size, then Oracle Database QoS Management can no longer recommend removing servers from server pools. In this situation the Dashboard displays "No recommended action at this time."

The minimum size of a server pool is the number of servers that that server pool is required to have. If you add the values for the server pool minimum attribute for each server pool in your cluster, then the difference between this sum and the total number of servers in the cluster represents shared servers that can move between server pools (or *float*) to meet changes in demand. For example, if your cluster has 10 servers and two server pools, and each server pool has a minimum size of four, then your system has two servers that can be moved between server pools. These servers can be moved if the target server pool has not reached its maximum size. Oracle Database QoS Management always honors the Min and Max size constraints set in a policy when making Move Server recommendations.

If you set the minimum size of a server pool to zero and your system experiences a demand surge, then Oracle Database QoS Management can recommend moving all the servers out of that server pool so that the server pool is at its minimum size. This results in the Performance Classes that use that server pool being completely starved of resources, and essentially being shut down. A server pool with a minimum size of zero should only host applications that are of low business criticalness and Performance Classes that are assigned a low rank in the Performance Policy.

Choosing the Best Recommendation

When trying to relieve a resource bottleneck for a particular Performance Class, Oracle Database QoS Management recommends adding more of the resource (such as CPU time) for that Performance Class or making the resource available more quickly to work requests in the Performance Class. The recommendations take the form of promoting the target Performance Class to a higher Consumer Group, demoting competing Performance Classes within the resource plan, adjusting CPU resources shared between different slices in a server pool, or moving servers between server pools.

Implementing a recommended action makes the resource less available to other Performance Classes. When generating recommendations, Oracle Database QoS Management evaluates the impact to system performance as a whole. If a possible recommendation for changing the allocation of resources provides a small improvement in the response time of one Performance Class, but results in a large decrease in the response time of another Performance Class, then Oracle Database QoS Management reports that the performance gain is too small, and the change is not recommended.

Oracle Database QoS Management can issue recommendations that involve a negative impact to the performance of a Performance Class if:

- The negative impact on the Performance Class from which the resource is taken is projected not to cause a Performance Objective violation and a positive impact is projected for the Performance Class that gets better access to resources
- The Performance Class from which the resource is taken is lower ranked, and thus less critical to your business, than the Performance Class being helped

If the resource bottleneck can be resolved in multiple ways, then Oracle Database QoS Management recommends an action that is projected to improve the performance of the highest ranked Performance Class that is violating its objective. You can also view the alternative recommendations generated by Oracle Database QoS Management and see whether the action was recommended for implementation. For example, one possible solution to resolving a bottleneck on the CPU resource is to demote the Consumer Group associated with the Performance Class that is using the CPU the most. By limiting access to the CPU for the work requests in this Performance Class, the work requests in the other Performances Classes for that database get a larger share of the CPU time. However, Oracle Database QoS Management may decide not to recommend this action because the gain in response time for the target Performance Class is too small.

See Also:

- "Viewing Recommendations"
- "Viewing Recommendation Details"

Contents of a Recommendation

The analysis data for a recommendation include the projected change in response time for each Performance Class, the projected change in the Performance Satisfaction Metric (PSM) for each Performance Class, and the reason this action is chosen among other alternative actions, as shown in Figure 1-7. In this example, if you implement the recommended action, then Oracle Database QoS Management predicts that the `sales cart` Performance Class, which has the highest ranking, will have an improvement in response time from 0.00510 seconds for database requests to 0.00426 seconds, which equates to an 11.6% gain in its PSM. The other Performance Classes are not effected by the change because they use a different server pool.

Figure 1-7 Example of the Analysis for a Recommended Action

Recommended Actions

Action Rank 1: Promote sales cart from Consumer Group 2 to Consumer Group 0.

Action Promote sales cart from Consumer Group 2 to Consumer Group 0.

Estimated Time 2 minutes

Rationale All potential single mapping changes have been analyzed. Changes evaluated and rejected are listed below.

Evaluation The beneficiary's PSM value is expected to change by 11.565 percentage points. The sum of all PSM values is expected to change by -27.265 percentage points. This action is a candidate for recommendation.

Projected Results

	Performance Satisfaction Metric (Last 5 min)		Average Response Time		
Performance Class	Projected (%)	Projected Change (%)	Objective Value (sec)	Current Value (sec)	Projected Value (sec)
Default_pc	100	0.0	0.00000	0.00670	0.00959
hr_pc	68	0.0	0.00800	0.00259	0.00259
sales_pc	-42	-38.8	0.00500	0.00514	0.00856
erp_pc	74	0.0	0.01000	0.00262	0.00262
sales cart	-30	11.6	0.00300	0.00510	0.00426

Implement

See Also:

- "Performance Satisfaction Metrics"
- "Using Metrics to Identify Performance Issues"
- "Viewing Recommendations"

Overview of Implementing Recommendations

Oracle Database QoS Management does not implement the recommendations automatically, however, you can configure Enterprise Manager to generate alerts based upon the duration that a Performance Class has not been meeting its objective. After the Oracle Database QoS Management administrator implements a recommendation, the system performance is reevaluated for the specified settling time before any new recommendations are made.

See Also:

- "Using the Oracle Database QoS Management Dashboard"
- "Viewing Recommendations"
- "Implementing Recommendations"
- "Reviewing Performance Metrics"

Example: How Recommendations Are Generated

Consider a system that has two servers in an Online server pool, and two servers in a Back Office server pool. The Online server pool hosts two workloads: the `sales_pc` Performance Class and the `sales_cart` Performance Class. The minimum size of the Online server pool is two. The Back Office server pool hosts two internal applications: a human resources (HR) application and an enterprise resource planning (ERP) application. The Back Office server pool has a minimum size of one. The `sales_cart` Performance Class has the highest rank and the `erp_pc` Performance Class has the lowest rank. The `sales_pc` Performance Class is ranked higher than the `hr_pc` Performance Class.

In this scenario, if the `sales_pc` workload surges, causing contention for resources and causing the `sales_cart` Performance Class to violate its Performance Objective, then this could lead to a service-level agreement (SLA) violation for the OLTP application. Oracle Database QoS Management issues a recommendation to increase access to the CPU for the `sales_cart` Performance Class at the expense of the `sales_pc` workload, because the `sales_cart` Performance Class is of a higher rank; a higher rank indicates that satisfying the Performance Objective for the `sales_cart` Performance Class is more important than satisfying the Performance Objective for the `sales_pc` Performance Class.

If, after you implement the recommendation, the sales_cart and `sales_pc` Performance Classes are still not satisfying their Performance Objectives, then Oracle Database QoS Management issues a recommendation to increase the number of servers in the Online server pool by moving a server from the Back Office server pool, or a server pool that hosts less critical workloads or workloads with more headroom. In this scenario, a server can be moved from the Back Office server pool, because the Back Office server pool is currently above its minimum size of one. If the Back Office server pool had a minimum size of two, then Oracle Database QoS Management would have to find an available server in a different server pool; Oracle Database QoS Management does not recommend to move a server from a server pool if doing so will cause a server pool to drop below its minimum size.

If you implement the recommended action, and your applications use Cluster Managed Services and Client Run-time Load Balancing, then the application users should not see a service disruption due to this reallocation. The services are shut down transactionally on the server being moved. After the server has been added to the stressed server pool, all database instances and their offered services are started on the reallocated server. At this point, sessions start to gradually switch to using the new server in the server pool, relieving the bottleneck.

Using the same scenario, if the `sales_pc` Performance Class and `hr_pc` Performance Class both require additional servers to meet their Performance Objectives, then Oracle Database QoS Management first issues recommendations to improve the performance of the `sales_pc` Performance Class, because the `sales_pc` Performance Class is ranked higher than the `hr_pc` Performance Class. When the `sales_pc` Performance Class is satisfying its Performance Objectives, then Oracle Database QoS Management makes recommendations to improve the performance of the `hr_pc` Performance Class.

What Does Oracle Database QoS Management Manage?

Oracle Database QoS Management works with Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware. Oracle Database QoS Management operates over an entire Oracle RAC cluster, which can support a variety of applications.

- Managing Database Resources to Meet Service Levels
- High Availability Management and Oracle Database QoS Management

Note:

Oracle Database QoS Management supports only OLTP workloads. The following types of workloads (or database requests) are not supported:

- Batch workloads
- Workloads that require more than one second to complete
- Workloads that use parallel data manipulation language (DML)
- Workloads that query GV$ views at a signification utilization level

Managing Database Resources to Meet Service Levels

Oracle Database QoS Management manages the CPU resource for a cluster. Oracle Database QoS Management does not manage I/O resources, so I/O intensive applications are not managed effectively by Oracle Database QoS Management.

Oracle Database QoS Management integrates with the Oracle RAC database through the following technologies to manage resources within a cluster:

- Database Services
- Oracle Database Resource Manager
- Oracle Clusterware
- Run-time Connection Load Balancing

Oracle Database QoS Management periodically evaluates the resource wait times for all used resources. If the average response time for the work requests in a Performance Class is greater than the value specified in its Performance Objective, then Oracle Database QoS Management uses the collected metrics to find the bottlenecked resource. If possible, Oracle Database QoS Management provides recommendations for adjusting the size of the server pools or making alterations to the consumer group mappings in the resource plan used by Oracle Database Resource Manager.

Database Services

You create database services to provide a mechanism for grouping related work requests. An application connects to the cluster databases using database services. A user-initiated query against the database could use a different service than a web-based application. Different services can represent different types of work requests. Each call or request made to the Oracle RAC database is a work request.

You can also use database services to manage and measure database workloads. To manage the resources used by a service, some services may be deployed on several Oracle RAC instances concurrently, whereas others may be deployed on only a single instance to isolate the workload that uses that service.

In an Oracle RAC cluster, Oracle Database QoS Management monitors the server pools and its nodes, on which the database services are offered. Services are created by the database administrator for a database. For a policy-managed database, the service

runs on all servers in the specified server pool. If a singleton service is required due to the inability of the application to scale horizontally, then the service can be restricted to run in a server pool that has a minimum and maximum size of one.

To use Oracle Database QoS Management for managing performance, create one or more policy-managed databases that run in server pools. If you have administrator-managed databases, then the database instances are placed in the Generic server pool and Oracle Database QoS Management can only monitor these databases.

When you first configure Oracle Database QoS Management, a default Performance Policy is created for each service that is discovered on the server pools being monitored. The name of these default Performance Classes are *service_name*_pc. The workload you want to monitor and manage the resource for must use a database service to connect to the database.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for more information about services

Oracle Database Resource Manager

Oracle Database Resource Manager (Resource Manager) is an example of a resource allocation mechanism; Resource Manager can allocate CPU shares among a collection of resource consumer groups based on a resource plan specified by an administrator. A resource plan allocates the percentage of opportunities to run on the CPU.

Oracle Database QoS Management does not adjust existing Resource Manager plans; Oracle Database QoS Management activates a resource plan named APPQOS_PLAN, which is a complex, multilevel resource plan. Oracle Database QoS Management also creates consumer groups that represent Performance Classes and resource plan directives for each consumer group.

When you implement an Oracle Database QoS Management recommendation to promote or demote a consumer group for a Performance Class, Oracle Database QoS Management makes the recommended changes to the mapping of the Performance Class to the CPU shares specified in the APPQOS_PLAN resource plan. By altering the consumer group, the Performance Class that is currently not meeting its Performance Objective is given more access to the CPU resource.

By default, the APPQOS_PLAN is replaced during the Oracle Scheduler maintenance window. Oracle recommends that you use the APPQOS_PLAN as the plan during those daily windows because this resource plan incorporates the consumer groups from the DEFAULT_MAINTENANCE_PLAN plan. You can force the use of APPQOS_PLAN by running the following commands in SQL*Plus:

```
BEGIN
 DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/

BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
   attribute=>'RESOURCE_PLAN',value=>'APPQOS_PLAN');
END;
/

BEGIN
```

```
 DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Repeat these commands for every weekday, for example, `TUESDAY_WINDOW`.

Note:

Do not edit the `APPQOS_PLAN` except as specified in "Editing the Resource Plan for Oracle Database QoS Management".

See Also:

- "Enabling Oracle Database QoS Management"
- *Oracle Database Administrator's Guide* for more information about Resource Manager

Oracle Clusterware

You must have Oracle Clusterware installed and configured before you can use Oracle Database QoS Management. The cluster administrator should create server pools to be used to deploy policy-managed Oracle RAC databases. Administrator-managed Oracle RAC databases use only the Generic server pool.

When you first configure Oracle Database QoS Management and create the initial Policy Set, you specify which server pools should be managed by Oracle Database QoS Management and which should only be monitored. If you select a server pool to be managed by Oracle Database QoS Management, then Oracle Database QoS Management monitors the resources used by all the Performance Classes that run in that server pool. If a Performance Class is not satisfying its Performance Objective, then Oracle Database QoS Management can recommend moving servers between server pools to provide additional resources where needed.

Oracle Database QoS Management uses Oracle Cluster Health Monitor (CHM) to collect memory metric data for the servers in the cluster.

See Also:

*Oracle Clusterware Administration and Deployment Guide*or more information about policy-managed databases and server pools

Run-time Connection Load Balancing

Run-time connection load balancing enables Oracle Clients to provide intelligent allocations of connections in the connection pool when applications request a connection to complete some work; the decision of which instance to route a new connection to is based on the current level of performance provided by the database instances.

Applications that use resources managed by Oracle Database QoS Management can also benefit from connection load balancing and transparent application failover (TAF). Connection load balancing enables you to spread user connections across all of

the instances that are supporting a service. For each service, you can define the method you want the listener to use for load balancing by setting the connection load balancing goal, using the appropriate SRVCTL command with the `-clbgoal` option. You can also specify a single TAF policy for all users of a service using SRVCTL with the options `-failovermethod`,`-failovertype`, and so on.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about run-time connection load balancing
- "Supported Service Configurations"
- *Oracle Database Net Services Administrator's Guide* for more information about configuring TAF

High Availability Management and Oracle Database QoS Management

Performance management and managing systems for high availability are closely related. Users typically consider a system to be up, or available, only when its performance is acceptable. You can use Oracle Database QoS Management and Performance Objectives to specify and maintain acceptable performance levels.

Oracle Database QoS Management is a run-time performance management product that optimizes resource allocations to help your system meet service-level agreements under dynamic workload conditions. Oracle Database QoS Management provides recommendations to help the work that is most critical to your business get the necessary resources. Oracle Database QoS Management assists in rebalancing resource allocations based upon current demand and resource availability. Nonessential work is suppressed to ensure that work vital to your business completes successfully.

Oracle Database QoS Management is not a feature to use for improving performance; the goal of Oracle Database QoS Management is to maintain optimal performance levels. Oracle Database QoS Management assumes that system parameters that affect both performance and availability have been set appropriately, and that they are constant. For example, the `FAST_START_MTTR_TARGET` database parameter controls how frequently the database writer checkpoints blocks to the data files to minimize instance recovery time. Using a low value for this parameter reduces the amount of time required to recover your database, but the overhead of writing redo log data more frequently can have a negative impact on the performance of your database. Oracle Database QoS Management does not make recommendations regarding the values specified for such parameters.

Management for high availability encompasses many issues that are not related to workload and that cannot be affected by managing workloads. For example, system availability depends crucially on the frequency and duration of software upgrade events. System availability also depends directly on the frequency of hardware failures. Managing workloads cannot change how often software upgrades are done or how often hardware fails.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using Oracle RAC for high availability
- *Oracle Database Performance Tuning Guide*
- *Oracle Database High Availability Overview*

Overview of Metrics

Oracle Database QoS Management bases its decisions on observations of how long work requests spend waiting for resources. Examples of resources that work requests can wait for include hardware resources, such as CPU cycles, disk I/O queues, and Global Cache blocks. Other waits can occur within the database, such as latches, locks, pins, and so on. Although the resource waits within the database are accounted for in the Oracle Database QoS Management metrics, they are not managed or specified by type.

The response time of a work request consists of execution time and a variety of wait times; changing or improving the execution time generally requires application source code changes. Oracle Database QoS Management therefore observes and manages only wait times.

Oracle Database QoS Management uses a standardized set of metrics, which are collected by all the servers in the system. There are two types of metrics used to measure the response time of work requests: performance metrics and resource metrics. These metrics enable direct observation of the wait time incurred by work requests in each Performance Class, for each resource requested, as the work request traverses the servers, networks, and storage devices that form the system. Another type of metric, the Performance Satisfaction Metric, measures how well the Performance Objectives for a Performance Class are being met.

See Also:

"Using the Oracle Database QoS Management Dashboard"

Performance Metrics

Performance metrics are collected at the entry point to each server in the system. They give an overview of where time is spent in the system and enable comparisons of wait times across the system. Data is collected periodically and forwarded to a central point for analysis, decision making, and historical storage. See Figure 1-6 for an illustration of how the system data are collected.

Performance metrics measure the response time (the difference between the time a request comes in and the time a response is sent out). The response time for all database client requests in a Performance Class is averaged and presented as the average response time, measured as database requests per second.

See Also:

"Reviewing Performance Metrics"

Resource Metrics

There are two resource metrics for each resource of interest in the system:

- Resource usage time—measures how much time was spent using the resource for each work request
- Resource wait time—measures the time spent waiting to get the resource

Resources are classified as CPU, Storage I/O, Global Cache, and Other (database waits). The data are collected from the Oracle RAC databases, Oracle Clusterware, and the operating system.

See Also:

"Viewing the Resource Wait Times Breakdown"

Performance Satisfaction Metrics

A useful metric for analyzing workload performance is a common and consistent numeric measure of how work requests in a Performance Class are doing against the current Performance Objective for that Performance Class.

This numeric measure is called the Performance Satisfaction Metric.

Different performance objectives are used to measure the performance of workloads, as shown in the following table:

Workload Type	Performance Objectives
OLTP	Response time, transactions per second
Batch	Velocity, throughput
DSS	Read or cache hit ratio, duration, throughput

Oracle Database QoS Management currently supports only OLTP workloads. For OLTP workloads, you can only configure a response time performance objective.

See Also:

"Interpreting the Performance Overview Graphs"

Using Metrics to Identify Performance Issues

The Oracle Database QoS Management metrics provide the information needed to systematically identify Performance Class bottlenecks in the system. When a Performance Class is violating its Performance Objective, the bottleneck for that Performance Class is the resource that contributes the largest average wait time for each work request in that Performance Class.

The Oracle Database QoS Management metrics are used to find a bottleneck for a Performance Class using the following steps:

1. Oracle Database QoS Management selects the highest ranked Performance Class that is not meeting its Performance Objective.
2. For that Performance Class, wait times for each resource are determined from the collected metrics.
3. The resource with the highest wait time per request is determined to be the bottlenecked resource.

Analyzing the average wait for each database request and the total number of requests for each Performance Class provides the resource wait time component of the response times of each Performance Class. The largest such resource contribution (CPU, Storage I/O, Global Cache, or Other) is the current bottleneck for the Performance Class.

See Also:

"Viewing the Resource Wait Times Breakdown"

2

Supported Workloads and Strategies

This chapter discusses the different ways that you can use Oracle Database Quality of Service Management (Oracle Database QoS Management) to manage the workload on your system and the best practices for configuring your system to effectively use Oracle Database QoS Management.

- Supported Configurations for Oracle Database QoS Management
- Strategies for Creating Classifiers for Performance Classes
- Configuration Strategies for Effective Resource Management
- Sample Implementation of Oracle Database QoS Management
- Creating Oracle Database QoS Management Performance Policies for the Demo System
- Managing Service Levels with Oracle Database QoS Management

Supported Configurations for Oracle Database QoS Management

To use Oracle Database QoS Management, your system must meet certain requirements. Also, your applications and database connections must conform to certain standards.

If you do not use a supported configuration, then Oracle Database QoS Management reports a configuration violation and is disabled.

- Supported Server Pool Configurations
- Supported Database Configurations
- Supported Service Configurations
- Supported Workload and Objective Types

Supported Server Pool Configurations

Oracle Database QoS Management requires that your cluster and databases use server pools. Databases in the Generic pool cannot be managed by Oracle Database QoS Management and are not visible in the user interface. Servers in the Free pool can be used to provide additional resources to managed server pools.

Before Oracle Database QoS Management can manage the resources for your cluster, you must mark server pools as managed in your Performance Policy. If a server pool is not managed by Oracle Database QoS Management, then the server pool is not visible to the Oracle Database QoS Management server. Any workloads or resources in the unmarked server pools are not managed by Oracle Database QoS Management.

You can select server pools to be managed by Oracle Database QoS Management even if they do not contain any database instances, or have a current size of zero. This enables you to preconfigure a server pool for management by Oracle Database QoS Management before the workload is started. Also, if you configure a server pool to have a minimum size of zero, then Oracle Database QoS Management can remove the servers from that server pool to provide resources for higher priority workloads. In this case, even though there are no resources within the server pool for Oracle Database QoS Management to manage, any workloads that run in that server pool are still monitored and, if possible, resources are allocated to that server pool to support the workloads.

Supported Database Configurations

Oracle Database QoS Management only works with Oracle RAC databases of release 11.2.0.2 or higher. For full use of Oracle Database QoS Management features, the databases must be configured as policy-managed databases that run in server pools. The underlying server pools used by the database instances must be marked as managed by Oracle Database QoS Management. For administrator-managed databases in the Generic server pool, Oracle Database QoS Management measures and monitors the workload and response times.

Database services should be created as `UNIFORM` services, meaning the service is offered by every available instance running in the specified server pool. If your application requires a `SINGLETON` service, then, to use Oracle Database QoS Management in management mode, the service must run in a server pool that has a maximum size of one. If you use a `SINGLETON` service in a server pool that has a maximum size greater than one, then Oracle Database QoS Management reports a configuration violation. If you selected measure-only for the management mode for this database, then you can use `SINGLETON` services any configuration.

Oracle Database QoS Management supports multiple databases sharing a server pool. If you have multiple databases using the same server pool, then every database that uses the server pool must have Oracle Database QoS Management enabled. Oracle Database QoS Management also supports Oracle RAC One Node databases (sometimes referred to as *singleton databases*), but these databases must use server pools that have a maximum size of one.

When you create a database, the default value of the `CPU_COUNT` initialization parameter for the database instance is set to the value of the number of physical CPUs on each node that the instance runs on. If you have multiple database instances on the same node, then you must adjust the value of `CPU_COUNT` for each instance so that the sum of `CPU_COUNT` for each instance that runs on the node is equal to or less than the physical number of CPUs on that node. Also, the value of `CPU_COUNT` must be the same for every instance of a database. For example, for the `sales` database, you cannot have `CPU_COUNT` set to four for the `sales1` instance and `CPU_COUNT` set to two for the `sales2` instance if both instances are in the same server pool.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide

Supported Service Configurations

Database services that are managed by Oracle Clusterware are required for Oracle Database QoS Management. All workloads managed by Oracle Database QoS Management must connect to the database using a database service that is managed

by Oracle Clusterware. You cannot use the default database service; the default database service is not managed by Oracle Clusterware.

The services used to connect to the database should be `UNIFORM`. If your application requires a `SINGLETON` service, then, to use Oracle Database QoS Management, the service must run in a server pool that has a maximum size of one.

The Oracle RAC high availability framework monitors the database and its services and sends event notifications using fast application notification (FAN). Oracle Clusterware and Oracle Net Services provide load balancing for services according to rules specified in the service configuration. These rules are:

- The **connection load balancing goal**: Connections are routed to an instance using the current workload for that instance and the type of connection (`LONG` or `SHORT`) to determine which instance can provide the best performance. For Oracle Database QoS Management, the connection load balancing goal should be set to `LONG`, so that if a new server is allocated to a server pool for a service, new connections will migrate faster to the new server. By migrating connections more quickly to a new server, the workload is balanced faster across all the available instances and the response times for workloads improve at a faster rate.

- The **run-time connection load balancing goal**: The load balancing advisory data is used to determine which instance best meets the goal specified for the service. The two goals are `SERVICE_TIME`, for which load balancing advisory data is based on elapsed time for work done in the instance, and `THROUGHPUT`, for which the load balancing advisory data is based on the rate that work is completed in the instance. For Oracle Database QoS Management, the run-time connection load balancing goal should be set to `SERVICE_TIME` for all database services that use server pools except optionally those with a maximum size of one.

Run-time connection load balancing sends advice to connection pools on how to balance connection requests across instances in an Oracle RAC database. The load balancing advisory also provides advice about how to direct incoming work to the instances that provide the optimal quality of service for that work. This minimizes the need to relocate the work later.

To configure the load balancing goals for a service, use the Server Control (SRVCTL) utility, as shown in the following example, or use Enterprise Manager:

```
srvctl modify service -db db_name -service service_name -rlbgoal SERVICE_TIME -
clbgoal LONG
```

Supported Workload and Objective Types

In the initial release of Oracle Database QoS Management, only online transaction processing (OLTP) workloads are supported. The only supported Performance Objective is average response time for database requests. Oracle Database QoS Management is designed to manage open workloads, or a workload in which demand is independent of the response time.

The database requests for your application workload must have an average response time of less than one second, and preferably, an average response time of less than 0.5 seconds. Each database request within a Performance Class should be homogenous with respect to resource usage. If a subset of the database requests in a workload use significantly more resources than the other requests, then you should create a new Performance Class to contain the database requests that require more resources. See "Deciding to Create New Performance Classes".

Oracle Database QoS Management does not support workloads that involve parallel queries. By default, parallel queries run on all available instances of the database, regardless of which service was used to connect to the database; the workload is not contained, or restrained to running on only those instances that offer the service. For a similar reason, Oracle Database QoS Management does not support workloads that involve a significant amount of database requests involving queries to GV$ views.

For a workload to be managed by Oracle Database QoS Management, the database connections must use a database service that is managed by Oracle Clusterware. The client or application that initiates the connection must be a JDBC (thick or thin) client, or an OCI client. Workloads that use a bequeath connection to the database are not managed by Oracle Database QoS Management.

Strategies for Creating Classifiers for Performance Classes

Currently, the only workloads that Oracle Database QoS Management manages are OLTP database workloads. To manage the workload for a database, the incoming work requests must be assigned to a Performance Class. Workload is mapped to a Performance Class using classifiers.

In multi-tier environments, a request from a client is routed to different database sessions by the middle tier or through load balancing, making the tracking of a client across database sessions difficult. Classifiers use session attributes to identify work requests. The attributes used are service name, user name, module, action, and program. See "Using Additional Filters for Classifying Work Requests".

Each classifier must specify one or more service names. If a classifier specifies multiple service names, then when matching the connection data to a Performance Class, the service names are evaluated using an `OR` operation. If any one of the service names specified in the classifier matches the service name in the work request, then the comparison evaluates to `TRUE`.

To set the `MODULE` and `ACTION` attributes, use the `OCIAttrSet()` call. Use the default namespace, `USERENV`, for the application context.

You can also optionally include the UserName and program name in the classifier. The user name is the name of the database user to which the session connects. The program attribute is the name of the client program used to log in to the server.

If the classifier for a Performance Class specifies multiple attributes, then the session attributes are combined using an `AND` operation. If all of the attribute values specified in the classifier match the session attribute values in the work request, then the comparison evaluates to `TRUE`. If you have more than one classifier that uses similar attribute values, then place the classifier with the most fine-grained conditions first. See "Applying Classifiers to Work Requests".

For example, consider the following classifiers:

- `create_invoice_taxes_pc` that specifies the `sales_cart` service, the `ORDER` module, and the `CALCULATE TAX` action
- `create_invoice_pc`, which specifies the `sales_cart` service and the `ORDER` module

The `create_invoice_taxes_pc` classifier should be evaluated before the `create_invoice_pc` classifier. If a work request uses the `sales_cart` service, and is performing the `CALCULATE TAX` action in the `ORDER` module, then the work request is assigned to the `create_invoice_taxes_pc`. If the work request does not have the matching values for all the attributes, then the work request is compared to

the next classifier, for `create_invoice_pc`. If you evaluate the `create_invoice_pc` classifier first, then any work request that uses the `sales_cart` service and the `ORDER` module will be assigned to the `create_invoice_pc` Performance Class, regardless of what action the work request performs.

You can create up to 47 Performance Classes for a cluster. If you have more than 47 services for your cluster, then use more than one service name within classifiers. Once a match is found for a classifier, Oracle Database QoS Management assigns a tag to the matching work request. The tag is based on the classifier that evaluates to `TRUE`. See "Performance Class Tags".

Configuration Strategies for Effective Resource Management

This section discusses key configuration recommendations and requirements for systems that are managed by Oracle Database QoS Management.

- About Resource Bottlenecks
- CPU Resource Bottlenecks
- Configuration Recommendations for Global Cache Resource Bottlenecks
- Configuration Recommendations for I/O Resource Bottlenecks
- Configuration Recommendations for Other Types of Bottlenecks

About Resource Bottlenecks

Oracle Database QoS Management measures use and wait times for CPU, Global Cache, I/O, and other resources to determine where a bottleneck is located.

The target Performance Class and its bottle-necked resource are identified on the Oracle Database QoS Management Dashboard (the Dashboard), however, only the CPU resource is actively managed in this release.

CPU Resource Bottlenecks

A CPU resource bottleneck is detected when there are excessive wait times on the collection of CPU queues running that workload. Oracle Database QoS Management offers recommendations you can implement to relieve the bottleneck.

One solution to this type of bottleneck is to increase the number of opportunities for the workload to run on the CPU. Oracle Database QoS Management implements this solution by assigning the workload to a consumer group that has more CPU shares across the server pool.

Another solution is to provide more CPU resources. If you have multiple instances sharing the CPU resources for each server in the server pool, and you have implemented instance caging, then Oracle Database QoS Management can suggest altering the CPU counts for the instances in the server pool; this solution gives more CPU resources to the workloads that are not meeting performance expectations by taking CPU resources away from an instance that is of lower rank or has the headroom to contribute the resources.

If there is a CPU resource bottleneck that cannot be relieved by adjusting the CPU counts between instances, then Oracle Database QoS Management can recommend moving a new server into the server pool. The server can come from the Free pool,

from a less-stressed server pool, or from a server pool that hosts a less critical workload.

Configuration Recommendations for Global Cache Resource Bottlenecks

A Global Cache resource bottleneck is detected when there is excessive data block movement between database instances. This is usually caused by an application that is not configured properly or is not able to scale horizontally. Configuring the application to run in a server pool with a maximum size of one or partitioning the data can usually relieve the bottleneck.

Oracle Database QoS Management cannot perform either of these actions in this release and does not provide a recommendation that can be implemented for this type of bottleneck.

Configuration Recommendations for I/O Resource Bottlenecks

An I/O resource bottleneck is detected when there are excessive wait times on the storage subsystem. This type of bottleneck is typically caused by either too few disk spindles or not enough network bandwidth on the storage interconnect. To resolve this bottleneck, spread the database files across a higher number of disks, or configure a separate network interface card (NIC) for a dedicated storage interconnect.

Oracle Database QoS Management cannot resolve this type of bottleneck in this release and does not provide a recommendation that can be implemented.

Configuration Recommendations for Other Types of Bottlenecks

The last resource type used to categorize bottlenecks, Other, is used for all other wait times. These database wait times are usually caused by SQL performance issues that result from an application that is not optimized, waiting on latches, and so on. These bottlenecks can be investigated using Oracle Database tuning tools such as Automatic Workload Repository (AWR) and Automatic Database Diagnostic Monitor (ADDM).

Resolving these types of bottlenecks are outside the scope of the run-time system management provided by Oracle Database QoS Management and Oracle Database QoS Management does not provide any recommendations that can be implemented

Sample Implementation of Oracle Database QoS Management

This section describes a sample implementation of Oracle Database QoS Management. The process by which Oracle Database QoS Management manages performance is described.

- Description of the Demo System
- Description of the System Workload
- Initial Oracle Database QoS Management Configuration

Description of the Demo System

The sample implementation uses a four-node cluster running on Linux.

The nodes are named `test_rac1` to `test_rac4`. In normal operation, each node does the following:

Node	Purpose	Services
`test_rac1`	Runs Oracle Grid Infrastructure for a cluster and the first database instance for the `backoffice` database	`HR` and `ERP`
`test_rac2`	Runs Oracle Grid Infrastructure for a cluster and the second database instance for the `backoffice` database	`HR` and `ERP`
`test_rac3`	Runs Oracle Grid Infrastructure for a cluster and the first database instance for the `online` database	`Sales` and `Sales_Cart`
`test_rac4`	Runs Oracle Grid Infrastructure for a cluster and the second database instance for the `online` database	`Sales` and `Sales_Cart`

The cluster is logically divided into two server pools with the following constraints:

Name	Min Size	Max Size	Current Size	Importance
`backoffice`	1	-1	2	1
`online`	1	-1	2	2
`Free`	0	-1	0	0

The server pool constraints as shown here guarantee that at least one server is allocated to each of the server pools (and the databases that run in those server pools) and the remaining servers can be shared on a demand basis to manage service levels. The `online` server pool hosts the most critical workloads, because it has the highest value for Importance. If a server failure occurs, then maintaining the minimum size for the `online` server pool takes priority over maintaining the minimum size of the other server pools.

Description of the System Workload

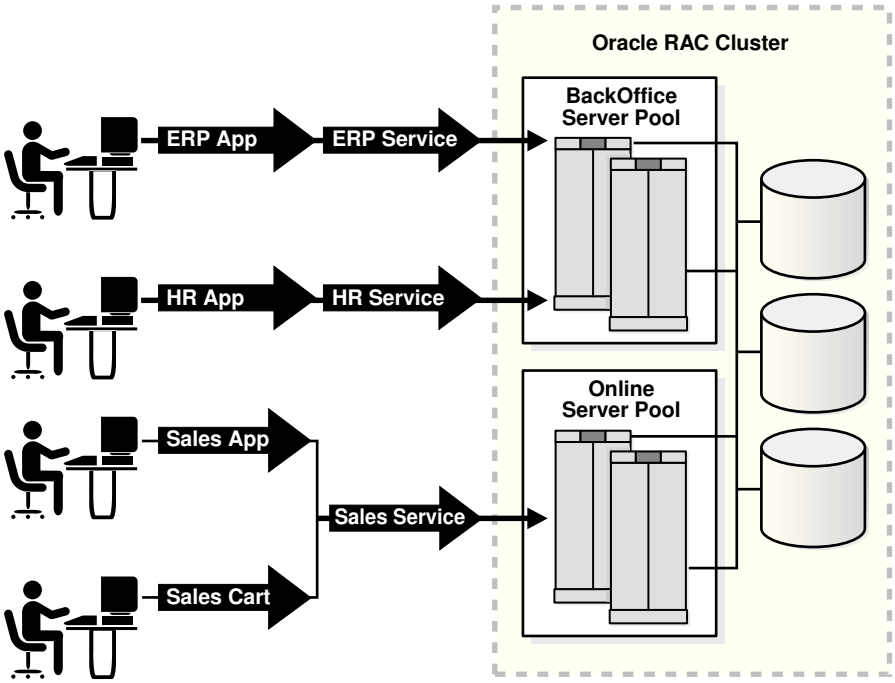
This release of Oracle Database QoS Management focuses on managing OLTP workloads, which are the type most likely to have an open workload (workloads for which demand remains constant even as system performance degrades) and be vulnerable to outages due to workload surges. For this demonstration, we assume there is a combination of internal and external workloads hosted in the same cluster so the resources can be shared.

There are four types of workloads demonstrated for this demo system, as illustrated in Figure 2-1:

- An ERP application based on J2EE that connects to the database instances in the `backoffice` server pool using the `ERP` service
- An internal HR application based on Oracle C Interface (OCI) that connects to the database instances in the `backoffice` server pool using the `HR` service
- An external Sales application based on J2EE that connects to the database instances in the `online` server pool using the `Sales` service

- An external Sales checkout application (Sales Cart) based on J2EE that connects to database instances through a specific database user in the `online` server pool using the `Sales` service

Figure 2-1 Illustration of a Sample Workload



By using two server pools the workloads and their dependent databases are logically separated but can readily share resources between them.

Initial Oracle Database QoS Management Configuration

At first, there is no Oracle Database QoS Management configured for this system. Using Oracle Enterprise Manager Cloud Control, there are two configuration workflows to complete to enable Oracle Database QoS Management for the cluster. The first workflow configures each database for Oracle Database QoS Management and the second workflow configures and enables Oracle Database QoS Management for the cluster.

See Also:

"Enabling Oracle Database QoS Management" for details on enabling Oracle Database QoS Management

---

After you create a default Policy Set, using the database services that are discovered automatically, Oracle Database QoS Management can be fine-tuned to align the workloads with their respective service-level agreements or objectives.

Creating Oracle Database QoS Management Performance Policies for the Demo System

In this section, the sample implementation of Oracle Database QoS Management is further evolved to include creating and activating Performance Policies and refining them with additional Performance Classes.

Because the default Performance Policy is created by discovering the database services in measure-only mode, the default Performance Policy can initially be activated to test how all of the workloads perform in the cluster. The Dashboard displays both the resource use and wait times that comprise the average response time for each Performance Class during different periods of demand. These numbers can serve to help understand the minimum response times achievable with the allocated resources

If your workloads peak at different times on a regular basis or your service-level agreements (SLAs) are variable based upon time, day of week, and so on, then create additional measure-only Performance Policies that change the size of the server pools to evaluate the minimum resources required for your workloads. In this demonstration, for the Sales application, the workload that uses the `online` server pool requires a minimum of two servers. The `backoffice` server pool requires only one server to satisfy the workload requests. If both server pools currently contain two servers, then you can enable the `online` server pool to take a server from the `backoffice` server pool, if needed, by setting the minimum size of the `backoffice` server pool to one. You would use a server pool directive override in the "Business Hours" Performance Policy to specify the minimum size of one for the `backoffice` server pool.

You could interpret the minimum size of a server pool as the number of servers *owned* by that server pool. If the sum of the minimum sizes of all the server pools in the cluster is less than the number of servers in the cluster, then the extra servers are referred to as *floaters*, which are shared by all server pools. For example, if your cluster has 15 servers, three server pools, and a minimum size of four for each server pool, then your system has three floaters.

After the Performance Policies have been run in measure-only mode, Performance Objectives can be added to each Performance Class. The Performance Objectives can be ranked based upon how critical the maintenance of that Performance Objective is to your business. Performance Objectives should be set to maximize headroom above the observed response times but below the response times required to meet SLAs. Maintaining at least 50% headroom is a good starting point to support trading off resources should a Performance Class experience a workload surge. For example, if a Performance Class has an average response time of two milliseconds (ms), then the Performance Objective could be set to three ms: two ms response time and an additional one ms which corresponds to the 50% headroom.

Although service-based classifiers can provide for easy configuration, you may want to define more than one Performance Objective for a service. For example, the `sales` service can contain many different workloads, such as Browse Products, Add Customer Account, Sales Cart and Browse Orders. Because the Sales Cart workload generates revenue, you may want this workload to have a shorter response time than the other workloads. You must create a separate Performance Class and associated classifiers to specify specific Performance Objectives for the different workloads.

On the Define Classifier page in the Policy Set wizard, a sales cart performance classifier can be defined by entering `sales` as the Service Name and if the application can set `MODULE` or `ACTION`, enter an appropriate value, otherwise configure a separate

`USERNAME` from the middle tier. As soon as this new Performance Class is defined, the Performance Class appears automatically in all of the Performance Policies in measure-only mode. The new Performance Class is given the lowest rank by default. Use these values initially to test the performance of your system. After the average performance levels can be determined, a Performance Objective and rank for this Performance Class can be set within each Performance Policy.

Managing Service Levels with Oracle Database QoS Management

The implementation of Oracle Database QoS Management is completed by actively managing the service levels, which means responding to alerts, reviewing and implementing recommendations, and tracking results. This section describes the actions you would perform on the demo system.

After all the workloads run and the Dashboard displays the performance of the demo system, you need to be alerted should a workload surge or failure cause a Performance Objective to stop being met. The Performance Satisfaction Metric (PSM) normalizes all of the objectives and provides a quick way to observe the health of the system. By observing the PSM Trend indicator you can see how well a Performance Class is meeting its objective over the last five minutes, and problems can be observed. Performance Objective violations produce recommendations that state how resources should be reallocated to relieve the bottleneck. Details and projections are available for further analysis of the bottleneck and possible solutions. If the recommendation is an action that can be implemented by Oracle Database QoS Management, then an **Implement** button is displayed.

Performance Objective violations of short duration are tolerated in most SLAs. Therefore, Enterprise Manager alerts can be configured by Performance Class specifying the duration of continuous violation. These alerts are configured on the Database alert page, but can be defined for all Performance Classes in the cluster.

An audit log of policy changes, violations and actions is available in the Oracle Grid Infrastructure home in the `oc4j/j2ee/home/log/dbwlm/auditing` directory on the server that hosts the Oracle Database QoS Management server. To determine which server is hosting the Oracle Database QoS Management server (the OC4J container), enter the following command at the operating system prompt:

```
srvctl status oc4j
```

3

Installing and Enabling Oracle Database QoS Management

This chapter describes the tasks you must complete to install and configure Oracle Database QoS Management on your system. Some of the tasks in this section must be performed by the cluster administrator.

Note:

Oracle Clusterware Administration and Deployment Guide for information about the cluster administrator

Configuring Oracle Database QoS Management to Manage Oracle Database Workloads

Configuring Oracle Grid Infrastructure for a cluster and Oracle RAC databases to work with Oracle Database QoS Management involves the following general tasks:

1. The database administrator (DBA) requests access to a server pool to be used for the database. The cluster administrator creates a server pool for the DBA and grants access to this server pool to the DBA. If the cluster administrator user is the same as the DBA user, then the server pool can be created at the time DBCA is run by selecting the Policy-managed option within DBCA. The server pool can also be created after installation by using Server Control (SRVCTL).

   The minimum size of this server pool is the number of database instances. If the maximum size of the server pool is greater than the minimum size, then new instances can be added to the database to handle peak workloads or to accommodate growth.

2. The DBA creates an Oracle RAC database in the allocated server pool by selecting the Policy-managed option within DBCA.

3. The DBA creates database services that are managed by Oracle Clusterware and assigns each service to a server pool. The application users connect to the database using these services.

4. The DBA enables the database for Oracle Database QoS Management using Enterprise Manager Cloud Control.

The initial configuration tasks for the Oracle Database QoS Management administrator are covered in more detail in the following sections:

- Installing and Configuring Oracle Grid Infrastructure for a Cluster

- Creating and Configuring Server Pools
- Creating and Configuring an Oracle RAC Database
- Enabling Oracle Database QoS Management

Installing and Configuring Oracle Grid Infrastructure for a Cluster

The installation and configuration of Oracle Grid Infrastructure for a cluster is not covered in this book. Refer to *Oracle Grid Infrastructure Installation Guide* for your platform for information about installing and configuring Oracle Grid Infrastructure for a cluster.

Creating and Configuring Server Pools

By default, a server pool called the Free pool is created during Oracle Grid Infrastructure installation. To create server pools for your Oracle RAC database, you can use SRVCTL or Oracle Enterprise Manager.

When you use DBCA to create an Oracle RAC database, you select policy-managed for the database, and choose the server pools which the database instances should run in.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about server pools
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about using SRVCTL to create a server pool
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX* for information about using DBCA to create an Oracle RAC database

If you use a cluster administrator that is separate from the database administrator, then only the cluster administrator user can create server pools. The cluster administrator then grants privileges on the server pools to the operating system user that owns the Oracle RAC installation. See *Oracle Clusterware Administration and Deployment Guide* for more information.

Note:

When creating a server pool for use with Oracle Database QoS Management, do not configure the `SERVER_NAMES` attribute (the `-servers` option of `srvctl add svrpool` or `srvctl modify svrpool` commands) for the server pool; Oracle Database QoS Management does not support server pools that use candidate lists.

Creating and Configuring an Oracle RAC Database

The steps for creating and configuring an Oracle RAC database are not covered in this book. Refer to *Oracle Real Application Clusters Administration and Deployment Guide* for information on creating an Oracle RAC database using server pools. When creating a database, the DBA should choose to create a policy-managed database.

After you have created the databases, perform the following steps to configure the databases for use with Oracle Database QoS Management:

- Modifying Database Initialization Parameters
- Creating Database Services

Modifying Database Initialization Parameters

The CPU_COUNT parameter for each database instance that runs in a server pool must be set to the same value if the database is managed by Oracle Database QoS Management. Also, on each server, the sum of the values for CPU_COUNT for all database instances running on that server must be less than or equal to the physical CPU count. For example, if you have a server with eight CPUs, and there are two database instances running on this server, then, for the databases to be managed by Oracle Database QoS Management, the CPU_COUNT parameter for each database instance must be set so that the values of the CPU_COUNT parameters for all instances on the server add up to eight. For example, you could have CPU_COUNT=3 on one instance and CPU_COUNT=5 on the other instance, or CPU_COUNT=6 on one instance and CPU_COUNT=2 on the other instance.

Note:

By default, the CPU count of each database that is started on a server is set to the number of physical CPUs installed for that server.

If you are running more than one database in a server pool, then using the default settings for CPU_COUNT will cause Oracle Database QoS Management to report a violation. To avoid this error, manually configure the CPU_COUNT value in the SPFILE using either Oracle Enterprise Manager or SQL*Plus, as shown in the following example, where *n* is the number of CPUs that should be used by the database instances:

```
ALTER SYSTEM SET cpu_count=n SCOPE=BOTH SID='*';
```

CPU_COUNT is a dynamic parameter that is not set by default. It should be set to the maximum number of CPUs that the database instance should utilize at any time. The sum of the values of CPU_COUNT for all instances on a server should not exceed the number of physical CPUs for that server. As a best practice, Oracle recommends using a minimum value of 2 for CPU_COUNT.

Creating Database Services

Applications and users connect to the database using services. For information about creating services for your Oracle RAC database, refer to *Oracle Real Application Clusters Administration and Deployment Guide*.

Creating Oracle Database QoS Management Administrator Accounts

Before logging in to the Oracle Database QoS Management Dashboard (the Dashboard), you must create an Oracle Database QoS Management administrative user. The operating system user associated with this account must be a cluster administrator user to initially set this up.

The administrative user for the Oracle Database QoS Management server is referred to as the QoS Admin user. This user has access to all the features of the Oracle Database

QoS Management server, including checking and changing the account password for the QoS Admin user. You can have multiple QoS Admin users.

To create a QoS Admin user account, perform the following steps:

1. As the cluster administrator user, log in to the node that is hosting the Oracle Database QoS Management server (OC4J container). This can be determined by using the following command from the Oracle Grid Infrastructure home:

   ```
   srvctl status oc4j
   ```

2. Stop the OC4J resource using the following command:

   ```
   srvctl stop oc4j
   ```

3. Log on as a CRS Administrator user and enter the following command:

   ```
   qosctl qosadmin -setpasswd qosadmin
   ```

   After you enter this command, you are prompted to enter the password of the default QoS Admin user one or more times.

   If you want to use a different user name, then you would enter the following command:

   ```
   qosctl qosadmin -adduser username
   ```

   In this example:

   - `qosadmin` is the name of the default QoS Admin user.
   - *`username`* is the name of the QoS Admin user you are creating. You are prompted to enter a password for this user

4. Restart the OC4J resource with the following command:

   ```
   srvctl start oc4j
   ```

See Also:

"Creating Administrative Users for Oracle Database QoS Management" for a complete description of the QOSCTL utility and its commands

Enabling Oracle Database QoS Management

You can enable Oracle Database QoS Management for two different levels:

- Managing the cluster
- Managing individual databases that run on the cluster

If you have multiple databases running on the same cluster, you can determine which databases are managed by Oracle QoS Management. To manage a database, all the databases that use the same server pool must be enabled for Oracle Database QoS Management; otherwise a violation is signalled when you try to access the Dashboard for the database.

To enable Oracle QoS Management for your system, perform the following steps:

1. Enable Oracle QoS Management at the Database Level
2. Create an Initial Policy Set
3. Enable Oracle QoS Management at the Cluster Level

Enable Oracle QoS Management at the Database Level

1. Log in to Oracle Enterprise Manager Cloud Control as the database administrator.
2. From the Database targets page, select the database you want to modify.
3. Select **Availability**, then **Enable / Disable Quality of Service Management**.
4. Enter the Cluster and Database credentials, then click **Login**.

   Note:

   To complete this step, you must specify the login information for both a SYSDBA and a cluster administrator account.

   The Enable/Disable QoS Management screen is displayed.

5. You are prompted to enter a password for the APPQOSSYS user. Choose a password and enter it in the Password and Confirm Password fields, then click OK.

   When you provide a password, the following actions take place:

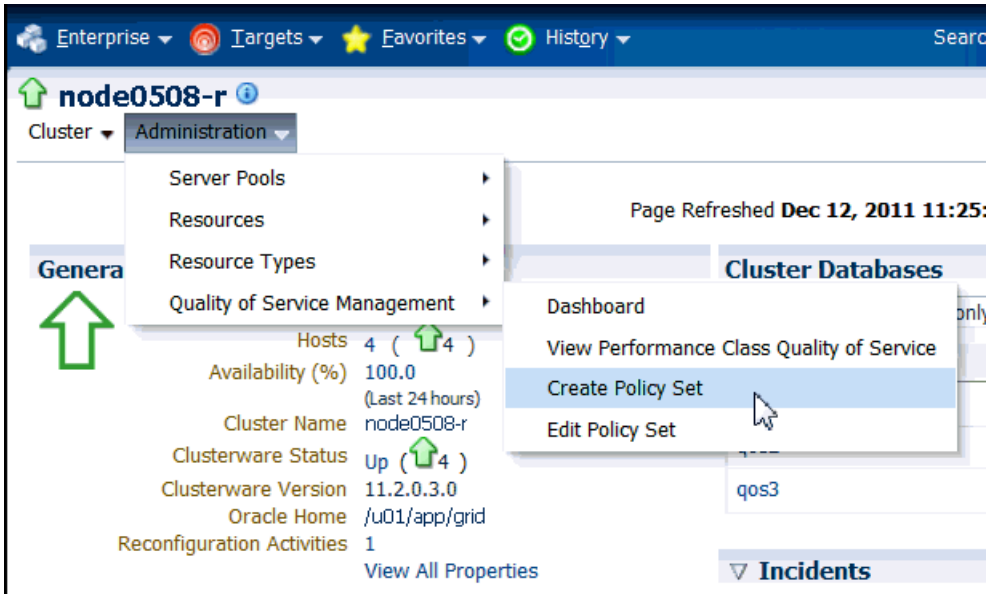
   - The APPQOSSYS account, which enables the Oracle Database QoS Management server to connect to the database, is unlocked and the new password is set.
   - The credentials are written to an Oracle Wallet stored in the Oracle Cluster Registry to enable Oracle Database QoS Management to log in to the database.

6. `APPQOS_PLAN` is set as the active Oracle Database Resource Manager plan for all actively managed databases, so that Oracle Database QoS Management can adjust CPU access for Performance Classes. The APPQOS_PLAN is not required for databases where all their Performance Classes are checked Measure-Only.

   By default, the `APPQOS_PLAN` is replaced during the Oracle Scheduler maintenance window. Oracle recommends that you use the `APPQOS_PLAN` as the plan during those daily windows because this plan incorporates the consumer groups from the `DEFAULT_MAINTENANCE_PLAN` plan. See "Oracle Database Resource Manager" for more information.

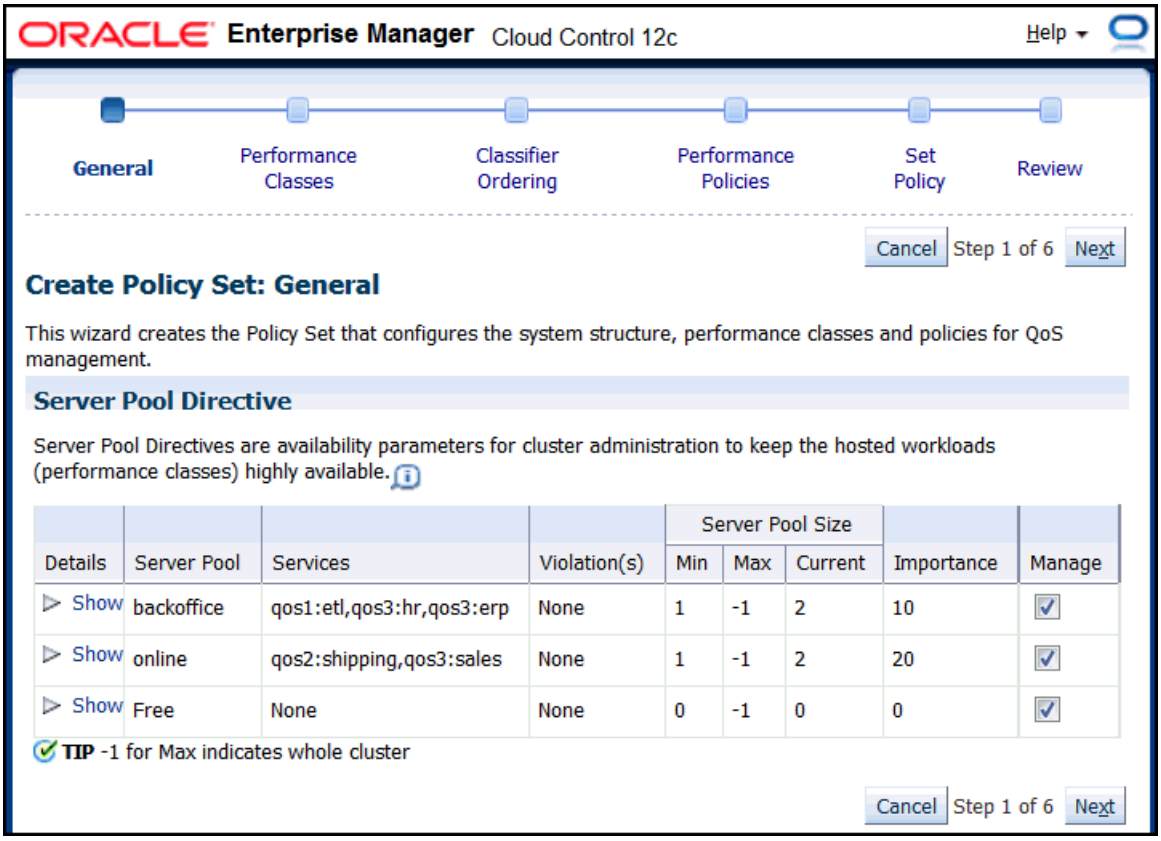
Create an Initial Policy Set

1. On the Oracle Enterprise Manager Cloud Control All Targets page, select the cluster on which your cluster database that has Oracle Database QoS Management enabled runs.
2. Select **Administration**, then **Quality of Service Management**, then **Create Policy Set**.



3. Log in to the Oracle Database QoS Management Server using the QoS Management administrator password (default user name is `qosadmin`).

4. On the first page of the Create Policy Set wizard, check the Manage box next to the server pools that represent your database. For example, `online` and `backoffice`. Click **Next**.



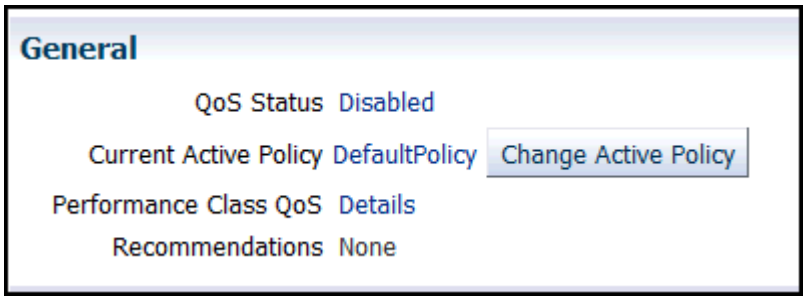
5. To get started with Oracle Database QoS Management, accept the defaults for your initial configuration and click **Next** on each page of the wizard to use the Default

Policy Settings. On the fifth step, click **Set Policy** to set the DefaultPolicy as the Chosen Active Policy, then click **Next**.

On the last step of the Create Policy Set Wizard, click **Submit Policy Set**.

Enable Oracle QoS Management at the Cluster Level

1. Using Oracle Enterprise Manager Cloud Control, on the All Targets page, select the cluster on which your cluster database that has Oracle Database QoS Management enabled runs.
2. Select **Administration**, then **Quality of Service Management**, then **Dashboard**.
3. Log in as the Oracle Database QoS Management user (for example, `qosadmin`).
4. On the Dashboard page, the General section shows the current status of Oracle Database QoS Management. On a new system, the status is Disabled. Click the link **Disabled** next to the status to enable Oracle Database QoS Management for this cluster.



4

Administering the Oracle Database QoS Management System

This chapter describes the basic administrative tasks you perform when using Oracle Database QoS Management to manage performance of your Oracle RAC cluster.

- Determining If Oracle Database QoS Management is Enabled
- Monitoring Performance with Oracle Database QoS Management
- Using the Oracle Database QoS Management Dashboard
- Administering the Policy Set
- Managing Performance Classes
- Managing Performance Policies
- Reviewing Performance Metrics
- Creating Administrative Users for Oracle Database QoS Management
- Editing the Resource Plan for Oracle Database QoS Management

Determining If Oracle Database QoS Management is Enabled

Oracle Database QoS Management can be enabled at three different levels: the cluster, server pool, and database. If you have multiple databases within the same Oracle RAC cluster, they might not all be managed by Oracle Database QoS Management.

Checking the Enabled Status for a Database

To determine if your database is managed by Oracle Database QoS Management, perform the following steps:

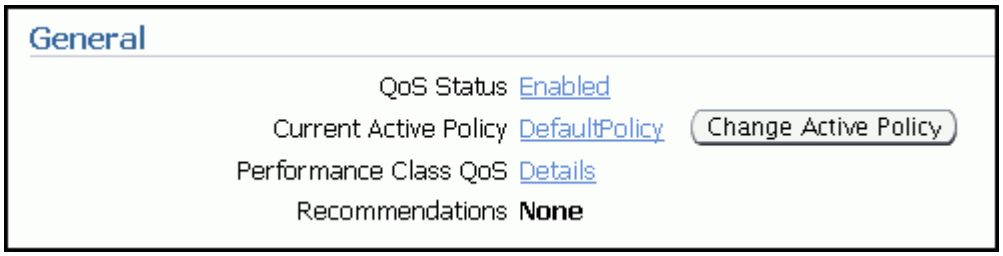
1. Log in to Cloud Control and select the database target to check.
2. From the target's menu, select **Cluster Database**, then **Target Information**.
3. In the Target Information window, at the bottom, the value for QoS Status should be Active if Oracle Database QoS Management is enabled for this database.

Checking the Enabled Status for the Cluster

Using Oracle Enterprise Manager Cloud Control, you can determine whether Oracle Database QoS Management is enabled for your cluster.

To determine if a cluster is managed by Oracle Database QoS Management, perform the following steps:

1. Log in to Cloud Control and select the cluster target to check.
2. From the target's menu, select **Administration**, then **Quality of Service Management**, and then **Dashboard**.
3. If prompted, log in as the QoSAdmin user.
4. On the Dashboard, in the General section, check the value for QoS Status. If the status value is Enabled, then the cluster is being managed by Oracle Database QoS Management.

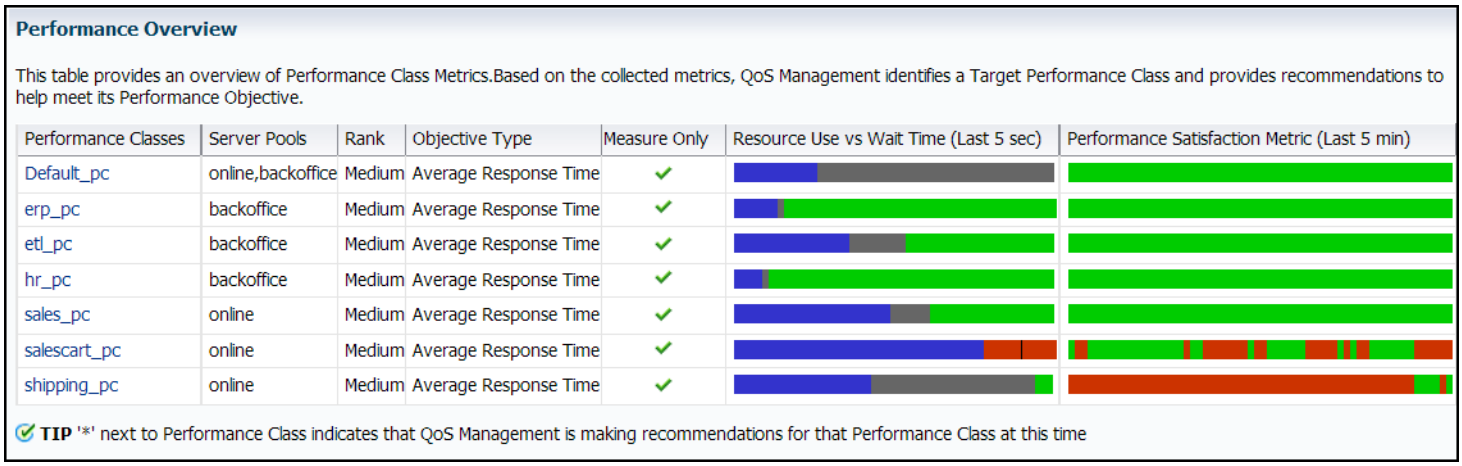


Monitoring Performance with Oracle Database QoS Management

After you have enabled Oracle Database QoS Management and created a default policy set, as described in "Enabling Oracle Database QoS Management", you can start to use Oracle Database QoS Management to monitor the performance of your system.

In measure-only or monitor mode, all the Performance Classes in user-defined Performance Policies have the measure-only box checked. You can set Performance Objectives, and Oracle Database QoS Management displays the Performance Satisfaction Metric (PSM) on the dashboard. If the response time of the system exceeds the Performance Objective specified, the PSM bar changes to red and an optional alert generated, as shown in Figure 4-1. Oracle Database QoS Management does not make recommendations if the measure-only check box is selected.

Figure 4-1 Performance Satisfaction Metrics for Measure-Only Performance Classes



Running Oracle Database QoS Management in measure-only or monitor mode allows you to understand how various workloads perform when sharing resources. Measure-only mode and monitor mode (available with Oracle Database 12c Release 12.1.0.2) assists you in determining the baseline Performance Objectives to use for each Performance Class. You can also use this mode to identify performance bottlenecks in your system.

Starting with the Oracle Database 11*g* Release 2 (11.2.0.3) release, you can use Oracle Database QoS Management with Oracle RAC on systems in measure-only or monitor mode.

If you are using Oracle RAC on an Exadata system, then you can use Oracle Database QoS Management in active management mode. In active management mode, you specify Performance Objectives for each Performance Class, and Oracle Database QoS Management can make recommendations for improving performance when a Performance Class is not meeting its objectives.

See Also:

- "Reviewing Performance Metrics"
- "Managing Performance Classes"
- "Creating a Performance Policy and Specifying Performance Objectives"

Using the Oracle Database QoS Management Dashboard

The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.

- Accessing the Oracle Database QoS Management Dashboard
- Enabling Oracle Database QoS Management for a Cluster
- Disabling Oracle Database QoS Management for a Cluster
- Interpreting the Performance Overview Graphs
- Viewing Recommendations
- Viewing Recommendation Details
- Implementing Recommendations

Accessing the Oracle Database QoS Management Dashboard

The Dashboard has four main sections:

- General
- Performance Overview
- Recommendations
- Resource Wait Time Breakdown

The General section of the Dashboard gives you a quick overview of the system. This section lists the QoS Status (Enabled or Disabled), the Current Active Policy, a link for checking the Performance Class details, and a notification for available recommendations. There is also a button that enables you to quickly change the current active policy.

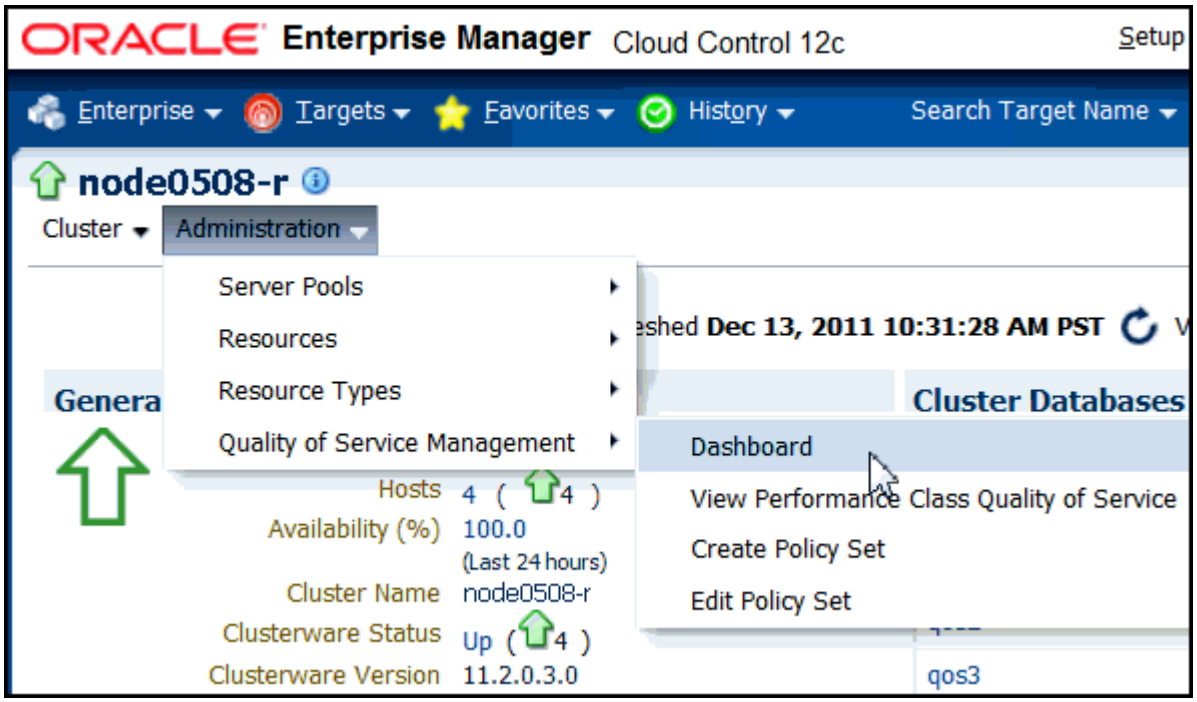
In the Performance Overview section, there is a table that lists the Performance Classes, the server pools where work is occurring, their rank, the Performance Objective being measured, and whether the Performance Objectives are being

monitored only, or are being monitored and managed. For each Performance Class there are bar graphs that provide an overview of the Performance Class metrics. See "Interpreting the Performance Overview Graphs" for more information.

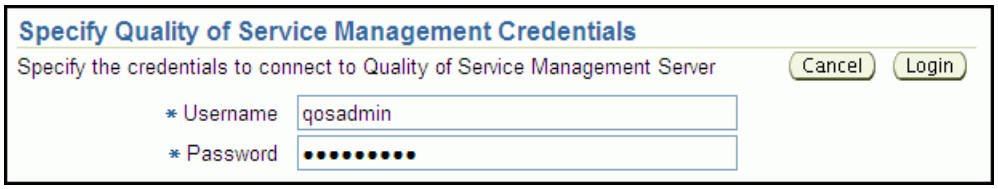
In the Recommendation section you can view the recommendations that are available when a Performance Class is not meeting its Performance Objectives. You can also view any violations that prevent the recommendations from being made.

At the bottom of the Dashboard is the Resource Wait Times Breakdown section. This section contains a table that provides a breakdown of resource wait times by Performance Class. For each Performance Class, the bottlenecked resource is the one that has the most wait time. This data is used by QoS Management to produce Recommendations. The data can also be used to make manual adjustments to the system. If you expand each Performance Class listed in the table, then you can see the server pools for that Performance class, and the resource wait times for each server pool.

1. Log in to Cloud Control and select the cluster target to check.
2. From the target's menu, select **Administration**, then **Quality of Service Management**, and then **Dashboard**.

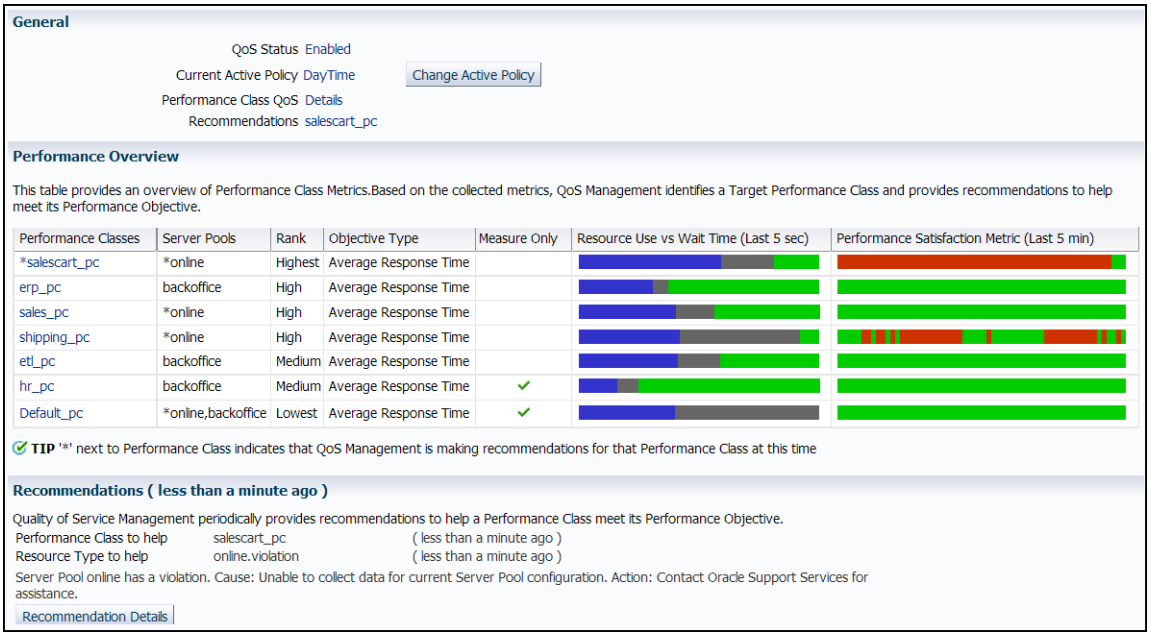


3. A login screen appears, prompting you for the Oracle Database QoS Management administrator (QoSAdmin) credentials. After typing in the username and password, click **Login**.



4. The Quality of Service Management Dashboard page, shown in Figure 4-2, is displayed after the correct credentials are entered.

Figure 4-2 Oracle Database Quality of Service Management Dashboard



Enabling Oracle Database QoS Management for a Cluster

1. On the Dashboard, in the General section, next to QoS Status, click **Disabled**.
2. On the Enable / Disable Quality of Service Management page, click **Enable QoS Management**.

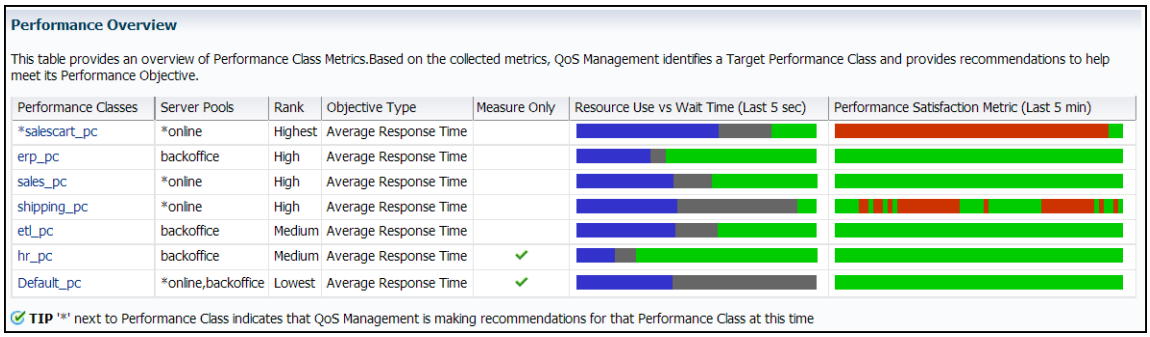
Disabling Oracle Database QoS Management for a Cluster

1. On the Dashboard, in the General section, next to QoS Status, click **Enabled**.
2. On the Enable / Disable Quality of Service Management page, click **Disable QoS Management**.

Interpreting the Performance Overview Graphs

On the Dashboard, in the Performance Overview section, there is a list of the current Performance Classes, some basic information about each Performance Class, and two bar graphs that show the most recent trend for the performance metrics for that class, as shown in Figure 4-3.

Figure 4-3 Example of the Performance Overview Charts



In Figure 4-3, there are bar graphs for the `salescart_pc`, `erp_pc`, `sales_pc`, `hr_pc`, `shipping_pc`, `etl_pc`, and `Default_pc` Performance Classes. There are two bar graphs displayed for each Performance Class. The two bar graphs are titled Resource Use vs. Wait Time (Last 5 sec.) and Performance Satisfaction Metric (Last 5 min.). The Resource Use vs. Wait Time bar graph has three sections of varying size that illustrate the resource use, wait and headroom portions of the Performance Objective. The Performance Satisfaction Metric bar graph is displayed as a single bar, growing from left to right, with the green and red segments representing the five second time slices that were exceeding or violating the Performance Objective respectively. If you place your cursor over a section of the bar graph, then a description of that measurement appears by your cursor.

Resource Use vs. Wait Time

The Resource Use vs. Wait Time chart is refreshed only when you refresh the page contents. In this chart:

- The blue section represents the portion of the average time spent for all database requests by that Performance Class using resources in the last five seconds
- The gray section represents the portion of the average time spent for all database requests by that Performance Class waiting on resources in the last five seconds
- The green section represents the headroom for that Performance Class (proportion of the average time for all database requests below the specified Performance Objective) in the last five seconds
- If a Performance Class is not meeting its Performance Objectives, then the gray and green sections disappear and the resource wait time is shown in red with a line to indicate where the Performance Objective is relative to the actual response time.
- If you place your mouse cursor over any section of this bar graph, then the actual values of Use, Wait and Headroom are displayed

The point between the gray and the green sections of the bar is the Performance Objective value. If you set this value below the resource use time, then you will never meet that objective. When configuring the Performance Objectives for a Performance Class, you must set the Performance Objective high enough to produce sufficient headroom (shown in green) to be able to share resources between Performance Classes to meet service levels as demand changes.

If red appears in this chart for a Performance Class, then you know that the Performance Class is not meeting its Performance Objectives. Oracle Database QoS Management issues a recommendation and an action to implement, if possible, to correct the problem. The recommendations generated by Oracle Database QoS Management occur once each minute, so they correspond to an earlier time than the current Performance Overview charts.

Performance Satisfaction Metric

The Performance Satisfaction Metric chart changes to show red and green lines for specific five second samples. Using this chart you can spot trends in the performance of your system.

For the Performance Satisfaction Metric bar graph:

- The red section represents the periods of time the Performance Class was not meeting its Performance Objectives during the sampling period

- The green section represents the periods of time the Performance Class was meeting its Performance Objectives during the sampling period

Viewing Recommendations

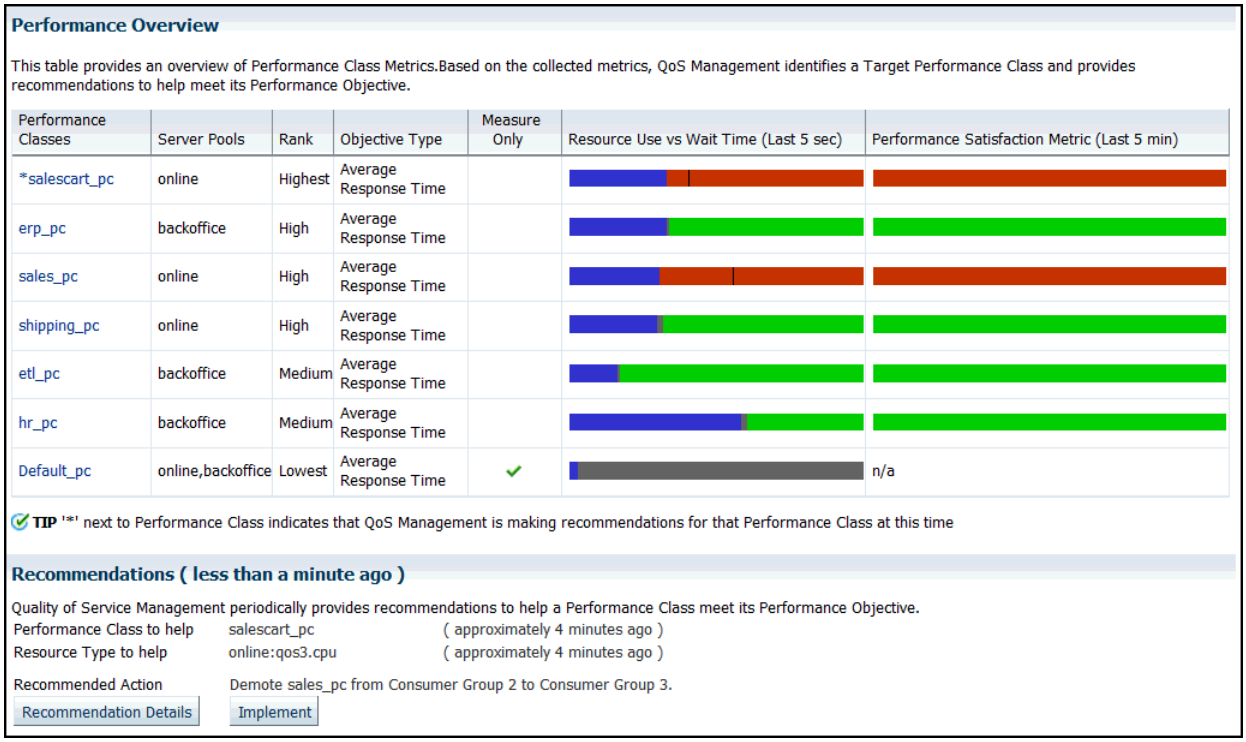
When viewing recommendations, there are three possible results.

1. If the Performance Classes are meeting their Performance Objective, then a recommendation is displayed which states "No action required: all Performance Objectives are being met."
2. If Oracle Database QoS Management determines that a Performance Class is not meeting its Performance Objective and has a recommended action for improving performance, then the Dashboard places an asterisk ('*') in front of the Performance Class name in the Performance Overview chart and displays a Recommended Action.
3. If more than one Performance Class is not meeting its Performance Objective as shown in Figure 4-3, then only the target Performance Class displays an asterisk and a recommendation. If the recommendation has an associated action, then an **Implement** button appears, which you can click to have the action implemented.

   If there are no recommended actions, then you can click the Details button to view the results of the latest analysis. Oracle Database QoS Management shows what possible actions could be taken, and why these actions were not chosen.

Example 4-1 Example Output For Performance Overview Page and Recommended Actions

Figure 4-4 Performance Overview Page Showing Performance Classes Not Meeting Their Performance Objectives



Performance Overview

This table provides an overview of Performance Class Metrics.Based on the collected metrics, QoS Management identifies a Target Performance Class and provides recommendations to help meet its Performance Objective.

Performance Classes	Server Pools	Rank	Objective Type	Measure Only	Resource Use vs Wait Time (Last 5 sec)	Performance Satisfaction Metric (Last 5 min)
*salescart_pc	online	Highest	Average Response Time			
erp_pc	backoffice	High	Average Response Time			
sales_pc	online	High	Average Response Time			
shipping_pc	online	High	Average Response Time			
etl_pc	backoffice	Medium	Average Response Time			
hr_pc	backoffice	Medium	Average Response Time			
Default_pc	online,backoffice	Lowest	Average Response Time	✓		n/a

TIP '*' next to Performance Class indicates that QoS Management is making recommendations for that Performance Class at this time

Recommendations (less than a minute ago)

Quality of Service Management periodically provides recommendations to help a Performance Class meet its Performance Objective.

Performance Class to help salescart_pc (approximately 4 minutes ago)
Resource Type to help online:qos3.cpu (approximately 4 minutes ago)

Recommended Action Demote sales_pc from Consumer Group 2 to Consumer Group 3.

Recommendation Details Implement

Viewing Recommendation Details

If Oracle Database QoS Management has generated a recommendation, then you can click the **Recommendation Details** button to view more information about why the recommendation was made, and the expected performance improvements to be gained if you implement the recommendations.

Recommended Actions

Action: Rank 1: Demote sales_pc from Consumer Group 2 to Consumer Group 3.

Action: Demote sales_pc from Consumer Group 2 to Consumer Group 3.

Estimated Time: 2 minutes

Rationale: All potential single mapping changes have been analyzed. Changes evaluated and rejected are listed below.

Evaluation: The beneficiary's PSM value is expected to change by 21.283 percentage points. The sum of all PSM values is expected to change by -2.669 percentage points. This action is a candidate for recommendation.

Projected Results

	Performance Satisfaction Metric (Last 5 min)		Average Response Time		
Performance Class	Projected (%)	Projected Change (%)	Objective Value (sec)	Current Value (sec)	Projected Value (sec)
Default_pc	100	0.0	1.00000	0.17330	Unknown
etl_pc	83	0.0	0.10000	0.01726	0.01726
hr_pc	39	0.0	0.04500	0.02729	0.02729
sales_pc	-73	-24.0	0.09000	0.17682	0.33397
erp_pc	65	0.0	0.08000	0.02773	0.02773
shipping_pc	68	0.0	0.10000	0.03187	0.03187
salescart_pc	-42	21.3	0.07300	0.19838	0.12569

Implement

You can select different recommendations using the Action drop-down list. Oracle Database QoS Management tries to provide the best recommendation to the QoS administrator, but you could decide that a different action would produce better results. If you select a different action, then the information in the Recommendation Actions and the Situation Analysis sections are updated to reflect the impact of the alternate recommendation. You cannot implement an alternate recommendation if that recommendation has been rejected by the Oracle Database QoS Management System for not providing enough benefit to the system as a whole.

Recommended Actions

Action: Alt: Demote Default_pc from Consumer Group 2 to Consumer Group 3.

Action: Demote Default_pc from Consumer Group 2 to Consumer Group 3.

Estimated Time: 2 minutes

Rationale: n/a

Evaluation: The beneficiary's PSM value is expected to change by 0.321 percentage points. The sum of all PSM values is expected to change by 0.788 percentage points. This action is not worth doing: it does not help the beneficiary enough.

The Recommended Actions page also has a section called Situation Analysis. If you display the output in this section, then you can see a description the projected impact implementing the Recommendation will have on the Performance Classes and the server pool, as shown in the following screenshot.

Situation Analysis

Donor Performance Classes	Donor Server Pools
Quality of Service Management could help salescart_pc at the expense of sales_pc: sales_pc is another PC using resource cpu in Server Pool online. sales_pc's Performance Objective is of lesser rank than salescart_pc's Performance Objective.	Quality of Service Management could move servers from Server Pool backoffice to Server Pool online: The current size of Server Pool backoffice is larger than its configured minimum size.
Quality of Service Management could help salescart_pc at the expense of Default_pc: Default_pc is another PC using resource cpu in Server Pool online. Default_pc is not currently violating its Performance Objective. Default_pc's Performance Objective is of lesser rank than salescart_pc's Performance Objective.	

Implementing Recommendations

If there is a Recommended Action, and you decide to implement the action, then simply click the **Implement** button on the Dashboard or the Details page.

After you have implemented a recommendation, the display on the Dashboard changes to show that there is an action in progress. No new recommendations are displayed until the system has settled after the resource change. You can determine the amount of time you have to wait before any new recommendations are made available by viewing the Details page before you implement a recommendation.

After implementing a Recommended Action, view the Performance Overview charts on the Dashboard to determine if the Performance Class is now meeting its Performance Objectives.

See Also:

"Specifying Authorized Actions for a Performance Policy"

Administering the Policy Set

Whether you are configuring the Oracle Database QoS Management system for the first time, or want to create a new Policy Set, you use the Create Policy Set wizard to create your Policy Set. You use the Edit Policy Set wizard to modify your existing Policy Set.

Note:

If you decide to create a new Policy Set, then all existing Performance Policies and user-added Performance Classes must be re-created. The changes you make to the stored Policy Set are not saved until you click **Submit Policy Set** in the last page of the Policy Set wizard.

- Editing a Policy Set
- Adding Server Pools to a Policy Set
- Modifying Server Pool Settings

- Adding Database Services to a Policy Set
- Updating a Policy Set to Include a New Database

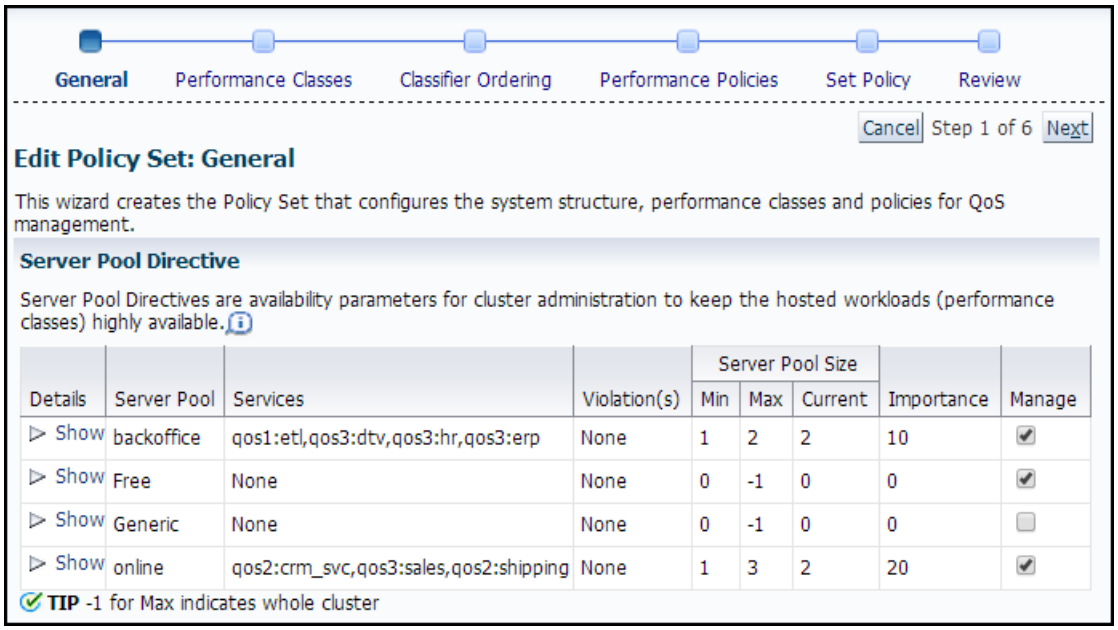
Editing a Policy Set

To modify an existing Policy Set, perform the following steps:

1. From the cluster target menu, select **Administration**, then **Quality of Service Management**, then **Edit Policy Set**.

   The Policy Set Editor wizard is started.

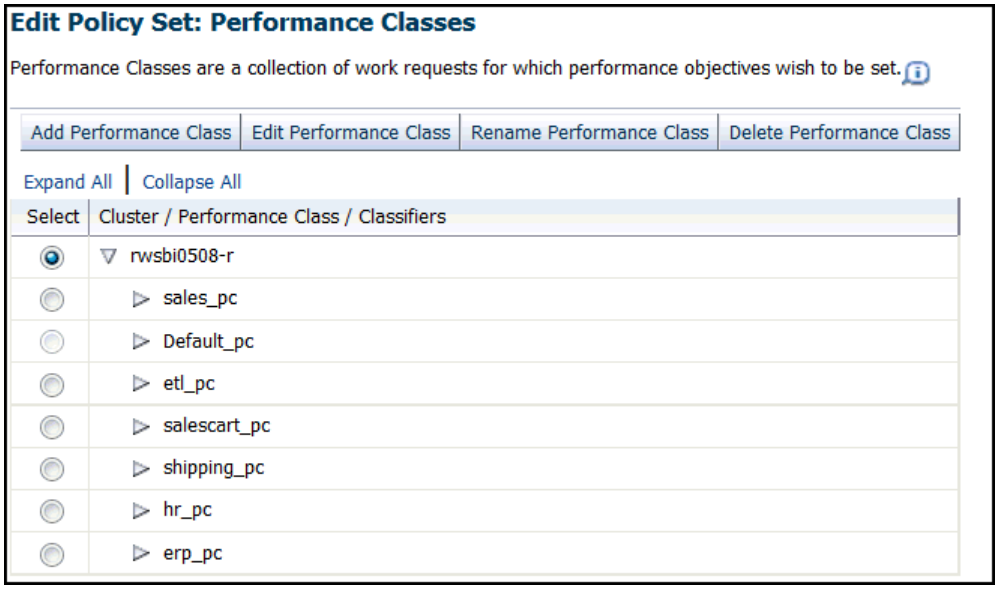
2. On the first page of the Policy Set Editor wizard you can view the current server pool settings.



   This page lets you specify which server pools are managed by Oracle Database QoS Management. If the Manage box for a server pool is unchecked, then none of the servers, databases or workloads that use that server pool are displayed or managed by Oracle Database QoS Management. The Details column displays any configuration violation details and the corrective action to take to enable the server pool to be managed.

   When finished, click **Next**. The Policy Set Editor: Performance Classes page appears.

3. The second page enables you to create, edit, rename, or delete Performance Classes for the system. You can create a new Performance Class, or use the Edit Performance Class button to modify the classifiers for an existing Performance Class.

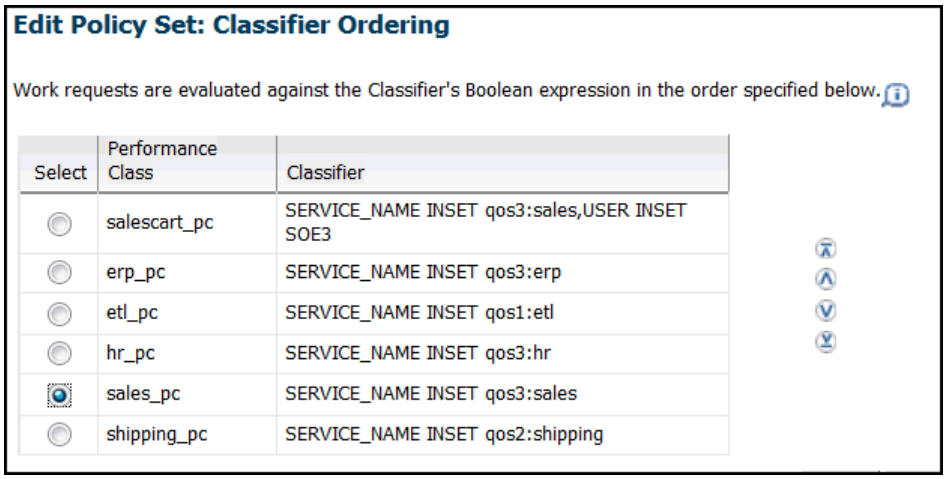


You can use the Expand All link to show the classifiers for each Performance Class, or expand an individual Performance Class entry to show the classifiers for only that Performance Class.

Select the Performance Class you want to edit, and then click the appropriate action button (Edit, Rename, or Delete). If you want to create a Performance Class for this Policy Set, then click the Add Performance Class button. See "Creating a Performance Class" for more information on creating a Performance Class.

When you have finished, click **Next**.

4. After creating or modifying the Performance Classes, the next step is to set the order in which the classifiers are evaluated.

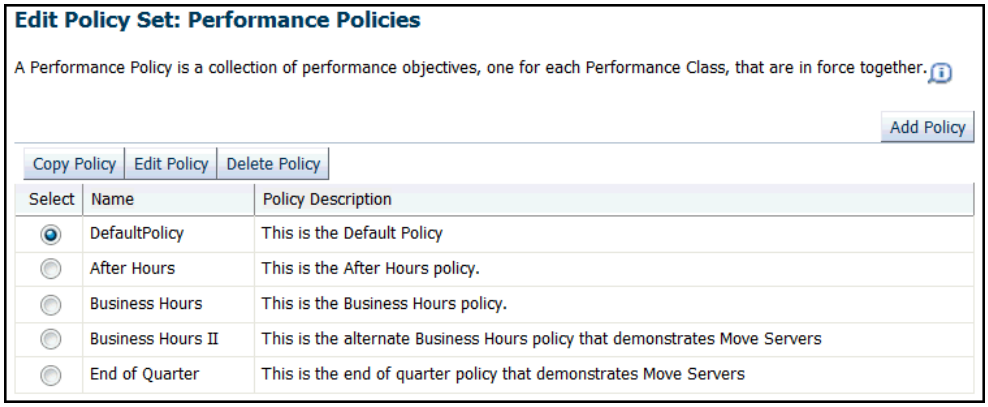


This step is very important, because the classifiers determine which Performance Class a work request is placed into. Select a Performance Class and use the arrow keys to the right of the classifiers to move the entries up and down in the list.

As a work request enters the system, the work request is evaluated against the classifiers. The first expression that evaluates to TRUE determines which Performance Class the work request belongs to. You should put the more fine-

grained classifiers at the top, and the more generic classifiers at the bottom of the list.

5. After you have set the order for your Performance Classes, the next step is to edit the Performance Policies and set the Performance Objectives.



On this page you can:

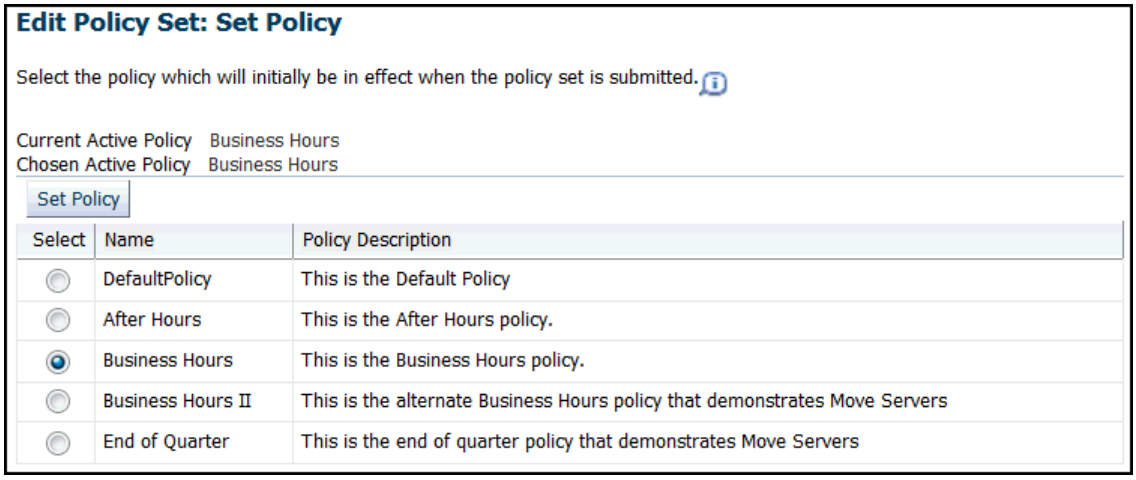
- Create a new Performance Policy
- Edit, copy, or delete an existing Performance Policy
- Specify which resource allocation methods can be automatically implemented by QoS Management.

Note:

The DefaultPolicy Performance Policy cannot be modified, but the policy can be copied and used as the basis for a new Performance Policy.

Refer to the section "Managing Performance Policies" for details on how to perform the tasks on this page.

6. After you have configured the Performance Policies, you must choose which one to put into effect immediately after the Policy Set is submitted.



Select the preferred Performance Policy, then click **Set Policy.** When finished, click **Next**.

7. Before you submit a Policy Set to the Oracle Database QoS Management system, you are asked to review the Policy Set configuration. If the changes are what you intended, then click **Submit Policy Set**. If you must modify any of the displayed settings, then click the **Back** or **Cancel** button.

   After clicking Submit Policy Set, you are returned to the Dashboard.

Adding Server Pools to a Policy Set

Server pools are created by the cluster or database administrator, using either Oracle Enterprise Manager Cloud Control or Server Control (SRVCTL). Refer to *Oracle Clusterware Administration and Deployment Guide* for instructions on how to create a server pool.

After the cluster administrator has created a server pool, you can add the server pool to the Oracle Database QoS Management system. See "Creating a Performance Policy and Specifying Performance Objectives".

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about modifying the size of a server pool
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring a recently allocated server to be a part of an existing Oracle RAC database.

Modifying Server Pool Settings

To modify the server pool settings, you configure a server pool directive override for a Policy Set.

When you configure server pool directive overrides, the new parameters you specify for the server pool are used instead of the parameters specified at the time of server pool creation.

Note:

You can change the minimum and maximum settings for the listed server pools, but you cannot create additional server pools using this interface. Any changes made here, after they are submitted, alter the current server pool properties set in the Manage Server Pools section of Oracle Enterprise Manager Cloud Control.

- You can modify the server pool parameters manually using the Manage Server Pool pages of Oracle Enterprise Manager Cloud Control or SRVCTL.

  You should not use this method of altering the server pool configuration when you have Oracle Database QoS Management enabled. If you use both server pool directive overrides and manually change the server pool configuration, then the

server pool directive overrides specified for the current Policy Set will override the manual settings and can result in confusion.

See Also:

"Setting Server Pool Directive Overrides"

Adding Database Services to a Policy Set

By creating additional services, you can monitor your workload performance at a finer level, by limiting use of the new services to specific applications or users. To add database services to a Policy Set, you must create or modify Performance Classes for each new database service.

1. The database administrator uses SRVCTL or Oracle Enterprise Manager Cloud Control to create database services that are managed by Oracle Clusterware. You can use the Availability menu on the database target page of Enterprise Manager Cloud Control to create services for an Oracle RAC database.

   See Also:

   Oracle Database 2 Day + Real Application Clusters Guide for more information on using Enterprise Manager Cloud Control to create services

2. The QoS Administrator adds one or more Performance Classes to the Policy Set or modifies the existing Performance Classes to include the new database services

   Note:

   Database services appear in Oracle Database QoS Management as `db_name:service_name`, so the service names must be unique within a database.

Updating a Policy Set to Include a New Database

If a new policy-managed database is added to your cluster, you may want Oracle Database QoS Management to manage or monitor the workload on this database.

1. The database administrator first requests servers to host the cluster database from the cluster or system administrator.

2. The cluster or database administrator creates the new server pools with the specified minimum number of nodes. The cluster or database administrator could also decide to grant access to existing server pools instead of creating new ones.

3. The database administrator creates a new database on the allocated server pools. The database administrator must ensure that the new database is configured correctly for management by Oracle Database QoS Management. See "Supported Database Configurations" for details.

4. The database administrator uses SRVCTL or Oracle Enterprise Manager Cloud Control to create database services that are managed by Oracle Clusterware for the new database.

5. The database administrator enables the database for Oracle Database QoS Management from the Database target page of Cloud Control.
6. Using the Edit Policy Set link in Cloud Control, the QoS Administrator adds the new server pools to the Policy Set, adds one or more Performance Classes to the Policy Set, or modifies the existing Performance Classes to include the new database services. See "Editing a Policy Set" for more information about editing a policy set.
7. After you successfully submit the new Policy Set, the new database is monitored and managed by Oracle Database QoS Management.

Managing Performance Classes

Each policy set contains one or more performance classes. Each performance class defines a type of workload on your servers.

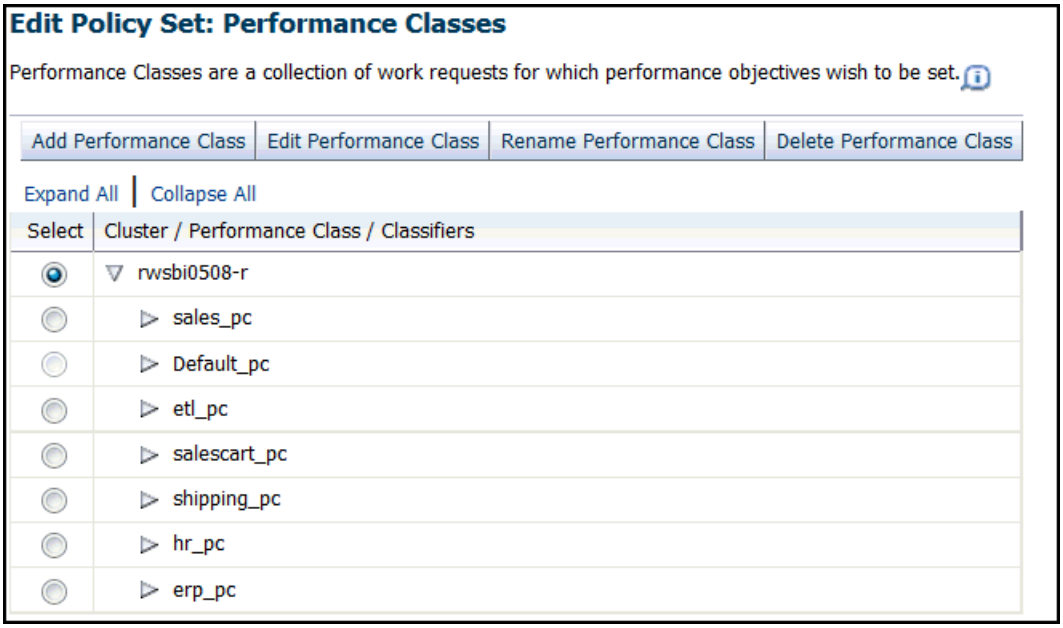
- Creating a Performance Class
- Deleting a Performance Class
- Renaming a Performance Class
- Editing an Existing Performance Class
- Specifying the Evaluation Order of the Classifiers

Creating a Performance Class

To create or edit the Performance Classes and the classifiers they use, perform the following tasks:

1. Start the Policy Set Editor wizard. From the cluster target page in Oracle Enterprise Manager Cloud Control, select **Administration**, then **Quality of Service Management**, then **Edit Policy Set**.
2. Go to the second screen in the wizard.

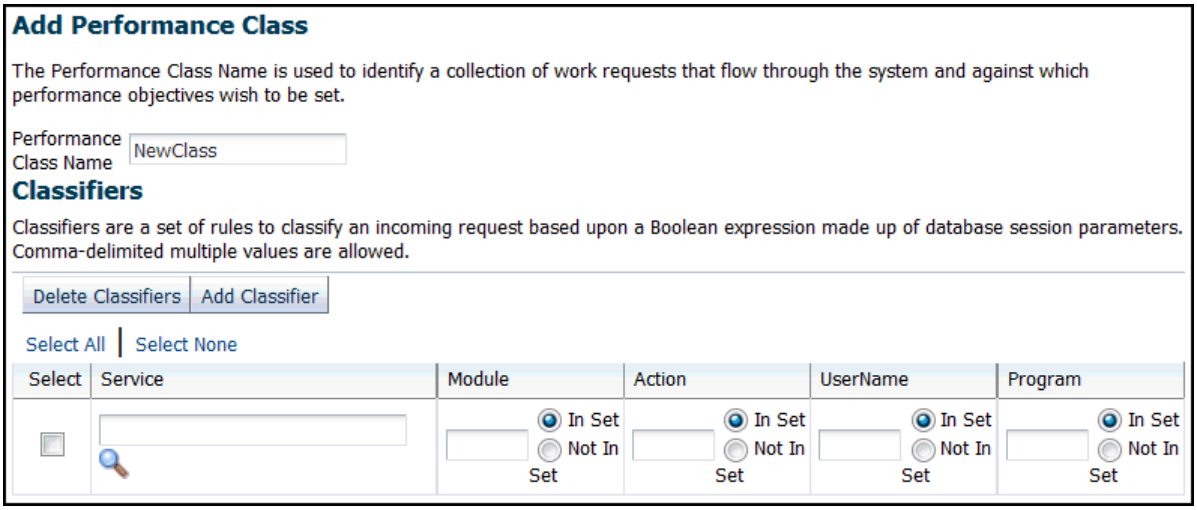
   On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed. If this is the first time configuring the system, then a Performance Class for each database service is shown along with a Default Performance Class.



To specify a classifier for a work request, you must specify at least one service name. If you specify multiple services, then use a comma-delimited list. Optionally, you can also specify any of the following filters:

- A comma-delimited list of module names and whether the work request uses a module in this list
- A comma-delimited list of actions, and whether the work request performs an action in this list
- A comma-delimited list of user names, and whether the work request uses a user name in this list
- A comma-delimited list of programs, and whether the work request is running a program in this list

3. Click **Add Performance Class**, and the Performance Class creation page is displayed. In the Performance Class Name text field, enter a name for the Performance Class.



4. In the Classifiers section, enter information to define a rule for classifying work requests. First select a database service, then specify matching values (In Set) for the module, action, UserName, or program name that is associated with the work request using the specified database service. You can also specify exclusion values (Not In Set) for these attributes.

   If you want to add multiple classifiers for the Performance Class, then click the **Add Classifier** button and enter in the appropriate information.

   Note:

   When evaluating a classifier for a Performance Class, all of the specified values are compared to the work request attributes using an AND operation; if you specify multiple classifiers for the Performance Class, then the results of each classifier evaluation for that Performance Class are combined using an OR operation.

5. After you have defined all the classifiers for the Performance Class, click the **Next** button until you reach the end of the wizard. Review the information you specified, then click **Submit Policy Set**.

Deleting a Performance Class

You can delete a Performance Class that is no longer needed.

1. Start the Edit Policy wizard.
2. Go to the second screen in the wizard.

   On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed.
3. Select a Performance Class and click **Delete Performance Class**.
4. Advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Renaming a Performance Class

You can rename a Performance Class using the Edit Policy Set wizard.

1. Start the Edit Policy wizard.
2. Go to the second screen in the wizard.

   On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed.
3. Select a Performance Class and click **Rename Performance Class**.
4. On the Rename Performance Class page, enter the new name of the Performance Class, then click **OK**.
5. Advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Editing an Existing Performance Class

When editing Performance Classes, you can create, edit, rename, or delete performance classes for the system. You can use the Edit Performance Class button to modify the classifiers for an existing performance class.

1. Start the Policy Set Editor wizard.
   a. From the cluster target page in Oracle Enterprise Manager Cloud Control, select **Administration**.
   b. Select **Quality of Service Management**.
   c. Select **Edit Policy Set**.
2. Go to the second screen in the wizard.

   On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed.
3. Select the Performance Class you want to modify and click **Edit Performance Class**.
4. When you are finished making changes, click **OK**.
5. Advance to the last page of the Policy Set Editor and click **Submit Policy Set**.

Adding Classifiers

If you want to add a classifier to a Performance Class, then perform the following steps:

1. Start the Policy Set Editor wizard.
2. Go to the second screen in the wizard, the Edit Policy Set: Performance Classes page.
3. Select the Performance Class to modify, then click **Edit Performance Class**.
4. In the Classifiers section on the Edit Performance Class page, click the **Add Classifier** button and enter in the appropriate information. When finished, click **OK**.
5. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Changing Classifiers

If you want to modify one or more classifiers for a Performance Class, then perform the following steps:

1. Start the Policy Set Editor wizard.
2. Go to the second screen in the wizard, the Edit Policy Set: Performance Classes page.
3. Select the Performance Class for which you want to change the classifiers, then click **Edit Performance Class**.

4. In the Classifiers section on the Edit Performance Class page, modify the classifier information, then click **OK**.
5. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Deleting Classifiers

To delete one or more classifiers for a Performance Class, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Go to the second screen in the wizard, the Edit Policy Set: Performance Classes page.
3. Select the Performance Class for which you want to delete the classifiers, then click **Edit Performance Class**.
4. In the Classifiers section on the Edit Performance Class page, select the classifiers you want to delete, then click **Delete Classifiers**. When finished, click **OK**.
5. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Specifying the Evaluation Order of the Classifiers

The classifiers generate Boolean expressions that are evaluated each time a work request enters the system. The first classifier that evaluates to TRUE determines the Performance Class for that work request. To ensure that the work requests are put in the correct Performance Classes, you must be careful in specifying the order in which the classifiers are evaluated.

To set the order of evaluation for the classifiers, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Proceed to the third page in the wizard, which is titled Edit Policy Set: Classifier Ordering.
3. Use the arrow keys to the right of a classifier to move the classifier up or down in the list. The classifiers for the Performance Classes at the top of the list are evaluated first. If the work request does not match the classifiers for that Performance Class, then evaluation continues with the next Performance Class in the list, until there are no further evaluations to be made. If a work request matches the classifiers for a Performance Class, then the work request is associated with that Performance Class and evaluation stops.

   For proper classification of work requests, you should put the Performance Classes with the strictest classifiers at the top of the list, and the Performance Classes with most lax classifiers near the bottom of the list. The `Default_pc` Performance Class, which has the most general classifiers, should always be at the bottom of the list.
4. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Managing Performance Policies

- Creating a Performance Policy and Specifying Performance Objectives
- Editing an Existing Performance Policy
- Copying a Performance Policy
- Setting the Current Performance Policy
- Deleting a Performance Policy
- Specifying Authorized Actions for a Performance Policy
- Setting Server Pool Directive Overrides

Creating a Performance Policy and Specifying Performance Objectives

To create and configure a Performance Policy, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Proceed to the fourth page in the wizard, which is titled Edit Policy Set: Performance Policies.
3. Click the **Add Policy** button to create a Performance Policy. The Create Policy page appears.

Add Policy

Policy Name

Policy Description

Performance Definition for Policy:

Specify the business ranking and performance objectives for the Performance Classes below. By checking "Measure Only" a Performance Class will be monitored but not managed by QoS Management.

Performance Class	Rank	Objective Type	Objective Value (sec)	Current Value (sec)	Measure Only
sales_pc	Medium	Average Response Time	0	0	✔
Default_pc	Medium	Average Response Time	0	0	✔
etl_pc	Medium	Average Response Time	0	0	✔
salescart_pc	Medium	Average Response Time	0	0	✔
shipping_pc	Medium	Average Response Time	0	0	✔
hr_pc	Medium	Average Response Time	0	0	✔
erp_pc	Medium	Average Response Time	0	0	✔

Authorized Actions

Select types of resource allocation actions that may be automatically implemented by QoS Management.

☐ Promote or Demote a Performance Class Consumer Group. ☐ Move a CPU between databases within a server pool. ☐ Move a server between server pools.

Server Pool Directive Override

Server Pool Directive Overrides change the server pool availability properties when the associated policy is in effect.

	Server Pool Size						
	Min		Max			Importance	
Server Pool	Current Value	Override	Current Value	Override	Current	Current Value	Override
backoffice	1	1	2	2	2	10	10
Free	0	0	-1	-1	0	0	0

For each Performance Policy you must specify a unique name. You can also provide a description of the policy and its intent. Then you must configure the Performance Classes for the policy.

To configure the Performance Classes, you must do the following:

- Set the rank for each Performance Class, from highest to lowest. A higher rank gives that Performance Class higher priority when there is contention for resources.
- Specify a value for the Performance Objective.

  The Performance Objective value is the appropriate length of time in seconds in which the work request, or database request, should complete, for example, 0.008 seconds, or eight milliseconds.

Note:

You should not use the service-level agreements (SLAs) or target response times as the Performance Objective values. Instead, choose a value that is reasonable, sustainable, and greater than your target response time. Using a higher value gives you time to implement the recommendations from Oracle Database QoS Management regarding the allocation of resources before the Performance Objective is exceeded.

You can also select whether the Performance Class is only monitored, not managed. If you select Measure Only, then Oracle Database QoS Management monitors the Performance Class, but does not provide any recommendations for improving performance.

4. Optional: Specify which actions can be performed automatically by QoS Management. See "Specifying Authorized Actions for a Performance Policy".
5. Optional: Configure server pool directive overrides. See "Setting Server Pool Directive Overrides".
6. Advance to the end of the Edit Policy Set wizard. Click **Submit Policy Set** to make the change permanent.

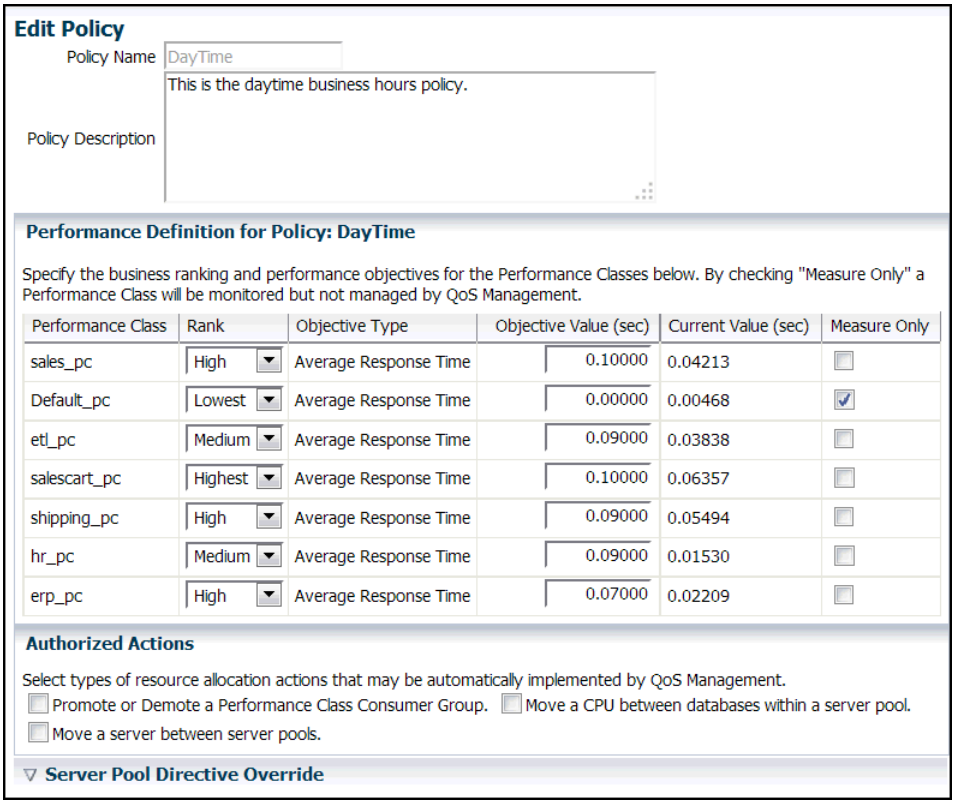
See Also:

"Monitoring Performance with Oracle Database QoS Management"

Editing an Existing Performance Policy

On the Edit Policy page, you can change the rank of each Performance Class, or change the Measure Only setting for a Performance Class. You can also select the type of resource allocation actions that can be automatically implemented by QoS Management and set server pool directive overrides.

1. On the Edit Policy Set: Performance Policies page, select the Performance Policy you want to modify and click **Edit**.



2. Specify the new ranks or objective values for the Performance Classes, or enable or disable the Measure Only setting for a Performance Class.

3. When you have finished making your changes, click **OK** to return to the Policy Set Editor wizard.

4. Click **Next** until you reach the end of the wizard. Review your changes, then click **Submit Policy Set**.

See Also:

- "Monitoring Performance with Oracle Database QoS Management"
- "Setting Server Pool Directive Overrides" for more information on server pool directive overrides

---

Copying a Performance Policy

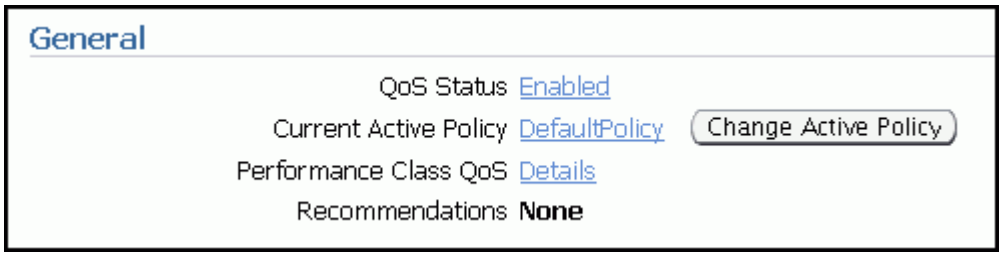
Instead of creating a new Performance Policy, you can instead copy an existing Performance Policy. A copied Performance Policy is identical to the original Performance Policy. You can then simply rename and edit the copy instead of re-creating all the details in a new Performance Policy.

Setting the Current Performance Policy

A Performance Policy is a collection of Performance Objectives, one for each Performance Class, that are in force at the same time. There are multiple ways to modify the active Performance Policy for Oracle Database QoS Management.

Changing the Active Performance Policy from the Dashboard

1. On the Dashboard page, in the General section, click the button **Change Active Policy**.



2. On the Set Policy page, select the Performance Policy you want to use, then click **OK**.

Changing the Active Performance Policy from the Policy Set Editor Wizard

You can change the Performance Policy that will be active when you submit the Policy Set to Oracle Database QoS Management.

1. Start the Policy Set Editor wizard.
2. Proceed to the fifth page in the wizard, which is titled Edit Policy Set: Set Policy.
3. Select the Performance Policy you want enforced, and click **Set Policy**.

At the end of the Policy Set Editor wizard, you can review the settings you specified, then click **Submit Policy Set** to configure Oracle Database QoS Management.

Changing the Active Performance Policy using QOSCTL

Many Oracle Database QoS Management policies are calendar based. You can switch the active performance policy automatically through a job scheduler such as Enterprise Manager Cloud Control, Task Scheduler or CRON by using QOSCTL to set the active policy.

1. Log in to the operating system user as the Clusterware administrator.
2. At the command line, or within a script, use the `qosctl` command with the `-activatepolicy` option. The command must use the following syntax:

   ```
   qosctl qos_admin_username -activatepolicy policy_name
   ```

   If the Performance Policy name includes spaces, then you must enclose the Performance Policy name within double quotes, for example:

   ```
   qosctl qosadmin -activatepolicy "Business Hours"
   ```

See Also:

"QOSCTL Utility Reference" for more information about QOSCTL.

Deleting a Performance Policy

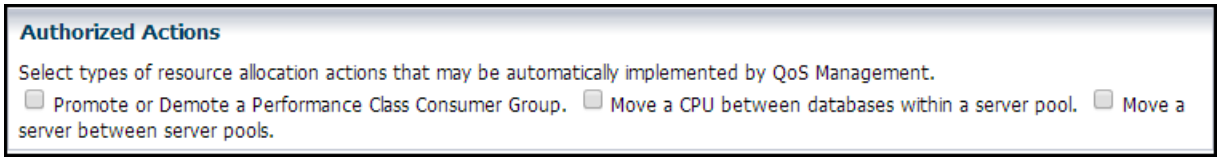
To delete a Performance Policy, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Proceed to the fourth page in the wizard, which is titled Edit Policy Set: Performance Policies.
3. Click the **Delete Policy** button to delete a Performance Policy.
4. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Specifying Authorized Actions for a Performance Policy

You can use the appropriate check boxes to specify which of the following actions can be implemented automatically by QoS Management:

- Promote or demote a performance class consumer group
- Move a CPU between databases within a server pool
- Move a server between server pools



If you do not authorize any of these actions, then QoS Management does not implement any changes to the active system until you review the current Recommendations for a Performance Class and click the **Implement** button.

Setting Server Pool Directive Overrides

A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools. For example, if you are expecting a surge in demand, such as during an advertised sale period, then you could use a server pool directive override to allocate more resources to the accounting applications.

Server pool directive overrides should be used only when necessary. Instead of using a server pool directive override, you should monitor the system over time and modify the server pool settings as needed. Using a server pool directive override can result in unexpected changes in resource allocations. For example, assume you have server pools named `webapps`, `HR`, and `payroll`. You create a server pool directive override to increase the minimum server pool size for the `payroll` server pool. When the server pool directive override is active, a server could be removed from the `HR` or `webapps` server pool to satisfy the higher minimum server requirement of the `payroll` server pool.

See Also:

"Modifying Server Pool Settings" for more information about configuring the server pool settings

---

1. Start the Policy Set Editor wizard.

2. Proceed to the fourth page in the wizard, which is titled Edit Policy Set: Performance Policies.

3. Click the **Edit Policy** button to edit a Performance Policy.

   The Edit policy page appears.

4. Expand the Server Pool Directive Override section, if necessary.

Server Pool Directive Override

Server Pool Directive Overrides change the server pool availability properties when the associated policy is in effect.

	Server Pool Size						
	Min		Max			Importance	
Server Pool	Current Value	Override	Current Value	Override	Current	Current Value	Override
backoffice	1	1	-1	-1	2	10	10
Free	0	null	-1	null	0	0	0
online	1	1	-1	-1	2	20	20

5. To set server pool directive overrides, perform the following:

   - To override the current value for the minimum number of servers in a server pool, enter a new value in the Min:Override field for that server pool. Valid values are 0 to the maximum number of servers in that server pool.
   - To override the current value for the maximum number of servers in a server pool, enter a new value in the Max:Override field for that server pool. Valid values are from the minimum number of servers in that server pool to the size of the cluster.
   - To override the current value for the Importance of the server pool, enter a new value in the Importance:Override field for that server pool. Valid values are 0 to 1000; higher values indicate greater importance.

6. After you have finished entering the server pool directive override values, click **OK** to implement the changes.

7. Advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Reviewing Performance Metrics

You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.

- Viewing Performance Metrics for All Performance Classes
- Viewing Performance Metrics for Individual Performance Classes
- Configuring Alerts for Quality of Service Management Events
- Viewing the Resource Wait Times Breakdown

Viewing Performance Metrics for All Performance Classes

After you have configured Oracle Database QoS Management, a short period of time is required for Oracle Database QoS Management to gather performance data and evaluate the performance of the system. After this period of time has passed, you can

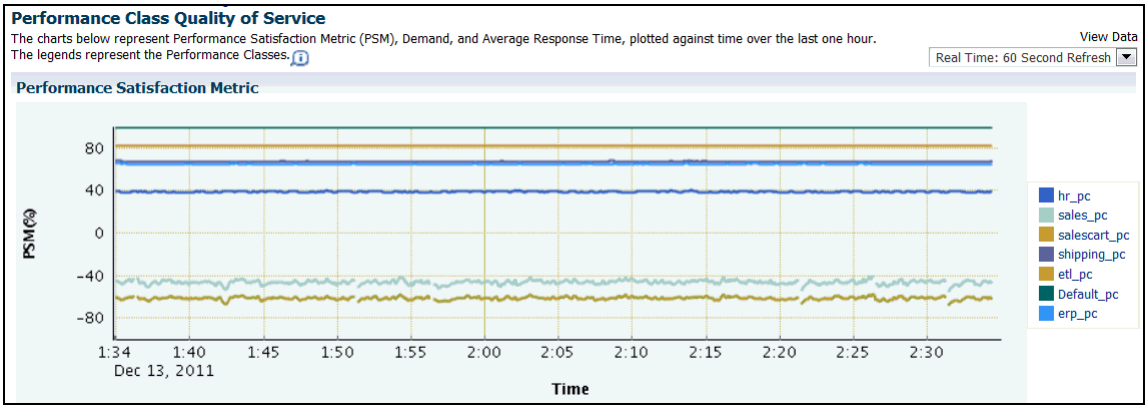
view the performance metrics for your system. To view the current performance metrics, perform the following steps:

1. Log in to Oracle Enterprise Manager Cloud Control as the cluster administrator. Go to the cluster target page.

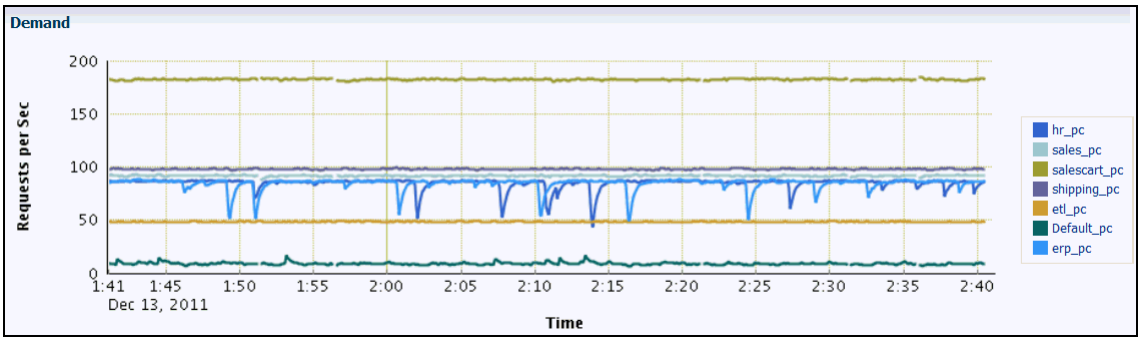
2. From the cluster target menu, select **Administration**, then **Quality of Service Management**, then View Performance Class Quality of Service.

3. The Performance Class Quality of Service page displays three charts measuring the current performance of each Performance Class that is being monitored:

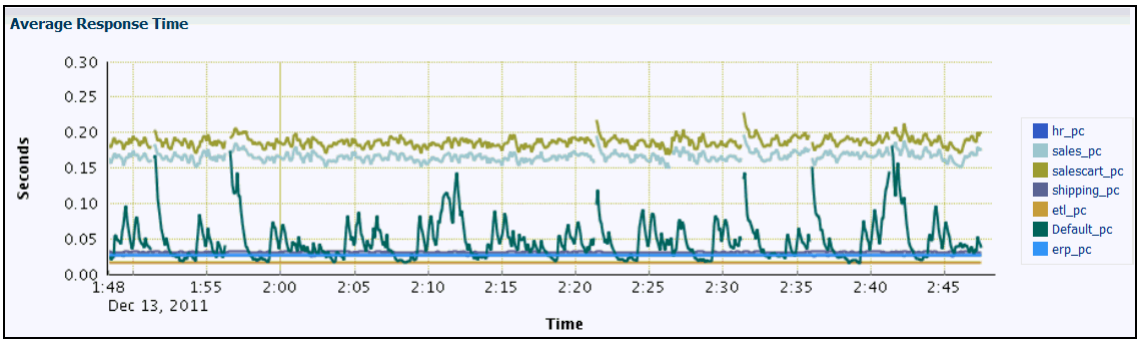
   a. The Performance Satisfaction Metric chart



   b. The Demand chart



   c. The Average Response Time chart



Viewing Performance Metrics for Individual Performance Classes

After you have configured Oracle Database QoS Management, and a short period of time has passed, you can view the performance metrics for a specific Performance Class. To view the current performance metrics for a Performance Class, perform the following steps:

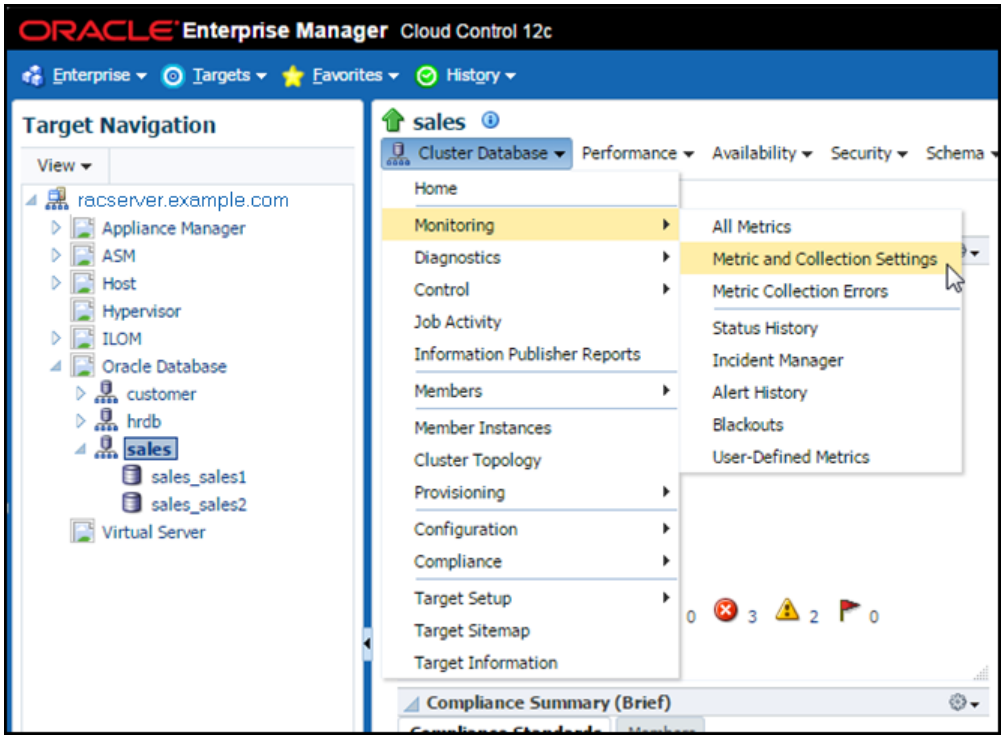
1. Log in to Oracle Enterprise Manager Cloud Control as the cluster administrator.
2. Select the cluster target page that is configured for QoS Management.
3. From the cluster target menu, select **Administration**, then **Quality of Service Management**, then **View Performance Class Quality of Service**.
4. On the right-hand side of any graph, in the legend box, click the link that corresponds to the Performance Class for which you want to view the performance metrics.

Configuring Alerts for Quality of Service Management Events

It is not convenient or efficient to require constant manual monitoring of the Quality of Service Management Dashboard. Instead you can use the Enterprise Manager Cloud Control notification system for reporting negative Performance Satisfaction Metrics (PSMs) that persist for user-specified times.

Both warning and critical levels can be alerted based upon specified durations for each performance class by setting up alert thresholds and notifications. The alerts are configured against the databases that offer the services being monitored

1. Start Enterprise Manager Cloud Control.
2. Navigate to the database for which you want to configure the alert.
3. From the **Cluster Database** menu select **Monitoring** and then select | **Metric and Collection Settings**.



4. Under the **Metrics** tab, edit the metric **QoS Management – Performance Satisfaction Metrics**.

   For example, if Quality of Service Management uses the database service `sales_svc_pc`, then under Negative PSM Duration (seconds), for that service, you might configure a warning alert when the duration is more than 120 seconds and a critical alert if the duration is more than 180 seconds.

The configured alerts will appear on the Database Home page of Enterprise Manager if a violation is detected.

Viewing the Resource Wait Times Breakdown

At the bottom of the Dashboard is the Resource Wait Times Breakdown table. This table provides breakdown of resource wait times by Performance Class. For each Performance Class, the bottlenecked resource is the one that has the most wait time. This data is used by Oracle Database QoS Management to produce recommendations. You can also use this data to make manual adjustments to your system.

Resource Wait Times Breakdown

This table provides breakdown of resource wait times by Performance Class.For each performance class, the bottlenecked resource is the one that has the most wait time.This data is used by QoS Management to produce Recommendations. The data can also be used to make manual adjustments to the system.

Expand All | Collapse All

Performance Class/Server Pool	CPU (sec)	Global Cache (sec)	IO (sec)	Other (sec)
rwsbi0508-r				
salescart_pc	0.121691	0.000000	0.000000	0.000755
erp_pc	0.000850	0.000000	0.000000	0.000116
sales_pc	0.112963	0.000000	0.000000	0.000214
shipping_pc	0.001970	0.000000	0.000000	0.000095
etl_pc	0.000451	0.000000	0.000000	0.000040
hr_pc	0.000886	0.000000	0.000000	0.000114
Default_pc	0.051329	0.000000	0.000000	0.012530
online	0.051284	0.000000	0.000000	0.012375
backoffice	0.000045	0.000000	0.000000	0.000155

Creating Administrative Users for Oracle Database QoS Management

Oracle Database QoS Management provides a command line utility named QOSCTL to help you manage users. This utility is installed on each node of the cluster but runs properly only if executed as the Oracle Clusterware administrator user on the same node on which the Oracle Database QoS Management server (the OC4J container) is running. The correct node to run the utility on can be determined by issuing the following command:

```
srvctl status oc4j
```

The QOSCTL utility supports the creation of Oracle Database QoS Management administrative users. The account information is stored in the local `system-jazn-data.xml` file with encrypted credentials. The account information is also stored in the Oracle Clusterware Repository (OCR) to support failover of the Oracle Database QoS Management Server.

QOSCTL Utility Reference

QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.

Purpose

The QOSCTL utility supports the creation of Oracle Database QoS Management administrative users.

Security Requirements

To use the QOSCTL utility, you must be logged in as an Oracle Clusterware administrator user.

When you enter the user name of an OC4J administrator user, you do not include the password. When you have submit the command for execution, you are prompted for the password associated with the specified *qos_admin_user*.

Displaying Help for the QOSCTL Utility

To display the help for the `qosctl` utility, use the following command:

```
qosctl -help
```

Syntax

The following code example shows the general format of QOSCTL commands:

```
qosctl qos_admin_user command
```

In place of the *qos_admin_user* argument, you would put the user name of an OC4J administrator user. Do not specify the password on the command line.

When configuring the initial accounts for your Oracle Database QoS Management system, the command uses operating system authentication.

Commands

Table 4-1 Summary of Commands for the QOSCTL Utility

Command Syntax	Description
`-activatepolicy` *policy_name*	Sets the current policy. This command enables you to change policies through a CRON or scheduling job. Policy names that include spaces must be encapsulated in double-quotes.
`-adduser` *username*	Adds the specified user and prompts for a password. This user is automatically granted the role required to use the Oracle Database QoS Management application and execute any of these commands.
`-listusers`	Lists the users that are authorized to run the Oracle Database QoS Management application
`-remuser` *username*	Removes the specified user account and all associated permissions **Note:** This command is not recoverable. Use caution, because you can delete your own account.
`-setpasswd` *username*	Updates the password of a specified user. The QOSCTL utility prompts you for the current password for the user. You must supply the correct value for the old password to change the password for a user. If the password for a user has been forgotten or lost, then you should remove the user account and create a new account for the user, with a new password.

Table 4-1 (Cont.) Summary of Commands for the QOSCTL Utility

Command Syntax	Description
`-help`	Displays the syntax for QOSCTL commands.

Editing the Resource Plan for Oracle Database QoS Management

Oracle Database QoS Management activates a resource plan named `APPQOS_PLAN`, which is a complex, multilevel resource plan. Oracle Database QoS Management also creates consumer groups that represent Performance Classes and resource plan directives for each consumer group.

You can modify some of the sections of the APPQOS_PLAN, but only as specified in the following table:

Plan Component	Modifiable?	Description
General	DO NOT EDIT	This section must not be edited as it is key to the QoS Management modeling and recommendation engine. Any changes will be detected and an database error reported on the database via the QoS Management dashboard.
Parallelism	Limited edits allowed	This section may be edited however all values set for the `ORA$APPQOS*` consumer groups must be the same within each column due to the reason stated above.
Thresholds	Limited edits allowed	This section may be edited however all values set for the `ORA$APPQOS*` consumer groups must be the same within each column due to the reason stated above. In addition, none of the "Switch to *" actions should be selected for these groups.
Idle Time	Limited edits allowed	This section may be edited however all values set for the `ORA$APPQOS*` consumer groups must be the same within each column due to the reason stated above.

See Also:

- "Oracle Database Resource Manager"
- *Oracle Database Administrator's Guide*
- *Oracle Database PL/SQL Packages and Types Reference*

5

Troubleshooting Oracle Database QoS Management

This chapter describes some problems you might encounter when using Oracle Database QoS Management and how you can resolve them. This chapter also describes how to locate the trace or log files for Oracle Database QoS Management.

- Common Problems
- Locating Log or Trace Files
- Enabling Tracing

Common Problems

After the initial configuration, Oracle Database Quality of Service Management is an automated system. As a result, most of the problems you might encounter are related to configuring Oracle Database QoS Management. The following sections illustrate the most common problems, and how to resolve them:

- Cannot Enable Oracle Database Quality of Service Management
- Cannot Enable Oracle Database QoS Management for a Database
- Oracle Database Resource Manager Not Enabled and Resource Plan Errors
- Do Not Have Access to a Server Pool
- Server Pool Is Marked As Unmanageable
- Metrics Are Missing For a Performance Class
- Oracle Database QoS Management is not Generating Recommendations
- Recently Added Server was Placed in the Wrong Server Pool

Cannot Enable Oracle Database Quality of Service Management

Before you can enable Oracle Database QoS Management within a cluster, you must first create a Policy Set. To create a Policy Set, refer to "Create an Initial Policy Set".

Cannot Enable Oracle Database QoS Management for a Database

For a database to be managed by Oracle Database QoS Management, the database must be compliant, enabled, and Oracle Database QoS Management must be able to access APPQOSSYS database user:

- A *compliant* database is an Oracle RAC database that is running Oracle Database release 11.2.0.2 or greater.

- For a database to be *enabled*, Oracle Database QoS Management must be able to connect to the database. The connection is configured when you select the **Enable Quality of Service Management** link in the Oracle Enterprise Manager Cloud Control. When you select this link, you are prompted for the cluster credentials and the password for the APPQOSSYS account in the database. By default the APPQOSSYS account is expired. When you submit a password, the account is unlocked.
- If the password for the APPQOSSYS user is changed through other methods or the account is locked, then Oracle Database QoS Management is disabled for this database until this condition is corrected by selecting the **Enable Quality of Service Management** link again.

See Also:

"Enabling Oracle Database QoS Management for a Cluster" for more information on how to enable Oracle Database QoS Management.

Oracle Database Resource Manager Not Enabled and Resource Plan Errors

Oracle Database QoS Management installs a special Oracle Database Resource Manager plan, APPQOS_PLAN, whenever a database is enabled for management by Oracle Database QoS Management. Oracle Database QoS Management requires this resource plan to move Performance Classes to different levels of CPU scheduling. No other plan can be active while Oracle Database QoS Management is enabled on this database. If a resource plan is not enabled for the Oracle RAC database, then an error results when trying to enable the database for management by Oracle Database QoS Management. After a resource plan is enabled, then enabling the database for management by Oracle Database QoS Management succeeds.

To check that the required APPQOS_PLAN is active on the target database, connect using SQL*Plus and issue the following statement:

```
SQL> show parameter resource_manager_plan
```

You should see output similar to the following:

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
resource_manager_plan                string      APPQOS_PLAN
```

Do Not Have Access to a Server Pool

Server pools can be managed separately from the database by configuring special operating system groups. By default, the user that installed Oracle Grid Infrastructure for a cluster can perform operations on server pools. If you use a separate operating system for the Oracle Database installation, then execute permissions on a server pool must be granted to the database software owner before a database can be deployed in that server pool. To grant this permission, you must use the CRSCTL utility to modify the server pool ACL attribute. A database administrator can also create server pools using Server Control (SRVCTL) or Oracle Enterprise Manager.

See Also:

Oracle Clusterware Administration and Deployment Guide for information about `crsctl` commands

Server Pool Is Marked As Unmanageable

A server pool is marked as unmanageable if Oracle Database QoS Management is not able to properly measure or predict the performance of Performance Classes deployed in that server pool. A server pool can be considered unmanageable under the following conditions:

1. The servers in the server pool have different physical CPU counts.
2. The CPU count for every database instance could not be retrieved.
3. The sum of the configured CPU counts for all database instances on a server is greater than its physical CPU count.
4. Singleton services are deployed in a server pool with a maximum size larger than one.
5. Oracle Database QoS Management is unable to collect the metrics for all Performance Classes, or the metrics collected do not contain valid data.
6. The server pool has a database that is not enabled for Oracle Database QoS Management.

If a server is added to a server pool, then Oracle starts up instances and services on the new server for the policy-managed databases in the server pool. When an instance is started on the new server, Oracle checks the SPFILE of the existing instances for the `CPU_COUNT` setting, and uses this value for the new instance. By default, if there is no setting for `CPU_COUNT` in the SPFILE, then the instances on the new server will be started with `CPU_COUNT` set to the number of physical CPUs of the server. This is not a supported configuration and will result in the server pool being marked as unmanageable. Also, if you modify the `CPU_COUNT` parameter but do not store the change in the SPFILE, then the `CPU_COUNT` parameter might be set to the wrong value on the newly started instance resulting in a configuration violation.

Make sure the databases and database instances that you want to be managed by Oracle Database QoS Management conform to the requirements documented in "Supported Database Configurations".

Metrics Are Missing For a Performance Class

Metrics are captured by Oracle Database QoS Management by querying the managed database instances in the cluster. Metrics are not be displayed under the following circumstances:

1. The database requests (work being done) do not match the classifiers defined for a specific Performance Class and "No demand" is displayed.
2. A more general classifier of a Performance Class is evaluated first. As a result, the Performance Class with the more specific classifier is shown as having "No demand". For example, if the `sales_pc` Performance Class uses the classifier `service=Sales` and the `sales_search` Performance Class uses the classifier `(service=Sales and action=Search)`, then, if the `sales_pc` Performance

Class was listed first in the Edit Policy Set Performance Class Order screen, then any work request that uses the `sales` service would be placed in the `sales_pc` Performance Class, including those that are performing the `Search` action. The `sales_search` Performance Class does not generate any metrics and "No demand" is displayed for that Performance Class.

3. The database instance is over-utilized and the metrics query reaches the time out limit and "Incomplete data" is displayed.
4. The database is not producing mutually consistent data for the Performance Class and "Nonconforming data" is displayed. For example a Performance Class that is classifying database requests that exceed one second will cause this response.
5. The metrics collected through the query do not pass the sanity verification checks performed by Oracle Database QoS Management and "Nonconforming data" is displayed.

Oracle Database QoS Management is not Generating Recommendations

Recommendations are generated once a minute when logged into the Oracle Database QoS Management Dashboard (the Dashboard). Recommendations do not appear on the Dashboard in the following cases:

1. Oracle Database QoS Management is disabled.
2. An action is in progress implementing a recommendation.
3. During the first minute of enabling Oracle Database QoS Management, or when submitting a Policy Set or activating a Policy Set.

Recently Added Server was Placed in the Wrong Server Pool

Servers are moved within a cluster by Oracle Clusterware, or as directed by an administrator. When a new server joins the cluster, Oracle Clusterware places the server in a server pool according to the placement algorithm and the state of the server pool attributes of Min, Max and Importance. See the *Oracle Clusterware Administration and Deployment Guide* for a complete description of this placement process.

RMI Port Conflict Detected

When you install Oracle Grid Infrastructure for a cluster, the default port for the Oracle Application Server Containers for J2EE (OC4J) resource is set to 23792. This can cause a port conflict with Java Remote Method Invocation (RMI). You should set the `OC4J_PORT` environment variable in your environment to the number of an available port, then restart the OC4J resource using the following commands:

```
srvctl modify oc4j -rmiport port
srvctl stop oc4j
srvctl start oc4j
```

Locating Log or Trace Files

The log files for Oracle Database QoS Management are located in the Oracle Grid Infrastructure home directory:

- The Oracle Database QoS Management server operations log file is located at `Grid_home/oc4j/j2ee/home/log/dbwlm/auditing/log.xml`

- The Oracle Database QoS Management server trace log file is located at `Grid_home/oc4j/j2ee/home/log/dbwlm/logging/log.xml`
- The OC4J standard out log file and the trace file for the Server Manager (SRVM) component of Oracle Clusterware is located at `Grid_home/oc4j/j2ee/home/log/oc4j_<timestamp>.out`
- The OC4J standard error log file is located at `Grid_home/oc4j/j2ee/home/log/oc4j_<timestamp>.err`
- The start and stop log file for the DBWLM component of Oracle Database QoS Management is located at `Grid_home/oc4j/j2ee/home/log/dbwlm.log`

Enabling Tracing

A default level of tracing is set at installation time. Finer-grained tracing may be enabled under the direction of Oracle Support Services.

Glossary

action

A database session parameter that is set by an application to identify the action associated with a database request.

affinity

The word 'affinity' is used to describe any strategy that is expected to increase the probability that a work request finds the required data cached in the instance to which the work request is routed.

aggregation

Aggregation is the process of taking a collection of measurements, and combining them to produce an aggregate measure. For example, counting all the work requests that are completed by a given server in a given Performance Class is a form of aggregation. Totaling the CPU time used by all the work requests in a given Performance Class handled by a particular server during a time interval is another form of aggregation.

application

An application is software that runs on a system and provides one or more services to a user or a group of users. Oracle CollabSuite, Oracle Email, Oracle CRM, and Oracle Financials are all examples of applications. CollabSuite is an example of an application that provides multiple services.

An application usually consists of multiple components; there may be a database component, a J2EE component, a client PC component, a batch component, a web component, a Web Services component, and so on.

Automatic Provisioning

Automatic Provisioning attempts to automate, as much as possible, the activities involved in re-tasking a piece of hardware. For example, taking a piece of hardware that has been running with one operating system and one set of application components, and re-deploying the hardware with a different operating system and a different set of application components.

average response time

The average of the response times for all work requests for a Performance Class for a given time period, specified in seconds.

bottleneck

A component or resource that limits the performance of an application or an entire system.

capacity planning

Capacity planning is the act of determining the amount and type of hardware needed to service projected peak user loads. Capacity planning is often done as part of a larger capital equipment budgeting cycle, and usually involves making load projections months into the future.

classifiers

Value matching rules that are applied to attributes of the work request to map work requests to Performance Classes.

closed workload

The amount of work performed in a system in which a fixed number of users interact with the application and each of these users issues a succession of requests. A new request from a user is triggered only after the completion of a previous request by the same user. A user submits a request, waits for the response of that request, thinks for a certain time and then sends a new request. The average time elapsed between the response from a previous request and the submission of a new request by the same user is called the "think time".

A closed workload is also referred to as a session-based workload.

clusterware

Any software that enables groups, or clusters, of connected computers to operate or be controlled as a unit.

conditioned data

Conditioned data is created from raw data in a post-processing step of some kind. Taking averages, removing outliers, filtering, and parameter estimation procedures are all examples of the kind of post-processing that may be used to create conditioned data from raw data.

database services

A database service is a user-created service that is managed by Oracle Clusterware and serves as a database session connection point. A database service may be offered on one or more Oracle RAC instances, and managed on a for-instance basis (for starting and stopping the service).

demand

Demand is a measure of the amount of work being presented to the system and is usually measured in work requests or requests per second.

elapsed time

An elapsed time measurement (also known as a *wall clock time measurement*) is a measurement of a single, contiguous time interval. The elapsed time interval begins at some time `t1` and ends at another time `t2`, where both times are read from the same clock. The elapsed time measurement is the value of (`t2 - t1`).

end-to-end response time

The expression *end-to-end response time* includes all time spent and all work done from the time a user request is received (for example, from clicking the Submit button in a browser), until the response is sent back to the user in its entirety. End-to-end response time includes time spent in application servers, Oracle Database, Oracle Automatic Storage Management, and traversing the internal networks of the data center.

entry point

The entry point is the initial point of contact between a work request and the Oracle Database QoS Management system. Work requests are initially classified and tagged at their entry point.

fair share scheduling

Fair share scheduling attempts to fairly allocate a resource such as a CPU among a collection of users, ensuring that each user gets a specified share of the available resource. Lottery based scheduling is one kind of fair share scheduling.

Free pool

A server pool that contains servers that are not assigned to any other server pool.

headroom

When a Performance Class is meeting its Performance Objectives, headroom refers to the difference between the actual response times and the required response times, or the surplus in performance.

layer

Layer and tier are synonymous.

Layer Active Time

Layer Active Time is the cumulative time that a work request is actively doing work at a layer, excluding time spent waiting on layers below. Layer Active Time includes time spent executing at the layer, and time spent waiting for layer local resources, such as the CPU, locally connected disks, memory, and so on.

Layer Response Time

Layer Response Time is the elapsed time for a work request to be completely handled by a specific layer. The layer response time includes the time spent executing the work request, and the time spent waiting for local and remote resources and servers.

layer visit

Often, a single work request from an end user (for example, clicking a link in a browser) causes several requests to arrive at various layers of the system. Each time a request is handled by a layer is called a layer visit.

load shedding

Load shedding refers to the act of rejecting some work requests, so that other work requests may complete successfully. Rejecting requests gracefully may require modifications to your applications. For example, you might want the end user to see a customized rejection page. Alternatively, you might want to store information from the work request so you can reply to the requester at a later time.

lottery based scheduling

Lottery based scheduling is a scheduling algorithm that uses random numbers to apportion resources (such as a CPU) among a collection of users, according to a pre-set distribution.

maintenance window

A contiguous time interval during which automated maintenance tasks are run. Maintenance windows are Oracle Scheduler windows that belong to the window group named `MAINTENANCE_WINDOW_GROUP`.

memory pressure

A state indicating that there is a limited amount of available memory on a server.

metric

A metric is something that can be measured.

module

Module is the database session parameter that is set by an application, generally to identify the application module making the database request.

open workload

Work performed in a system in which new work requests to an application come from outside the system being managed. The work requests are independent of each other and the work request arrival rate is not influenced by the response time for previous requests, or the number of requests that have already arrived and are being processed. The number of work requests the system may be asked to execute at any given time can range from zero to infinity. The system's resources or servers perform various

activities to process a work request and the work request leaves the system when processing is complete.

Open workloads are also referred to as request-based workloads.

Oracle Grid Infrastructure for a cluster

A term assigned to the software stack comprising Oracle's generic Clusterware, Oracle Automatic Storage Management (Oracle ASM), Oracle RAC agents, and the Oracle RAC database management infrastructure layer.

Oracle Database Resource Manager

Oracle Database Resource Manager is a software component available with the Oracle Database; Oracle Database Resource Manager enables an administrator to establish Resource Plans that control how various resources (such as the CPU) may be allocated to consumer groups, which are collections of work requests. The intent is very similar to Oracle Database QoS Management's Performance Class.

performance bottleneck

Oracle Database QoS Management attempts to identify performance bottlenecks due to Performance Classes waiting too long for required resources, such as CPU, Global Cache, or I/O.

Performance Class

A Performance Class is a group of related work requests. Performance Objectives are written for a Performance Class. All work requests that are grouped into a particular Performance Class have the same performance objective.

Performance Class ranks

The Performance Class rank represents the business criticalness of each Performance Class in a set of Performance Objectives that are in effect at a given time. When there are not enough resources available to service all applicable Performance Classes at the same time, Oracle Database QoS Management works to meet the Performance Objectives for the Performance Classes that are highest ranked at the expense of Performance Classes with a lesser rank. For example, Performance Classes with an rank of Lowest are sacrificed if necessary to ensure that Performance Classes of higher rank (Highest, High, Medium and Low) continue to meet their Performance Objectives.

performance objectives

Performance objectives refers to business level objectives for the system. A performance objective includes both Performance Objectives and availability objectives.

Performance Objectives

A Performance Objective defines a level of performance that is optimal for business purposes for a given Performance Class. For a particular Performance Class, a Performance Objective specifies the target average response time for that workload.

In high load situations, work of lower business criticalness may be deliberately starved for resources by the Oracle Database QoS Management system so that more important work can meet its Performance Objectives; in this circumstance the user might receive a "Server Busy" message instead of just experiencing very poor response times.

Performance Policy

A Performance Policy is a collection of Performance Objectives and Performance Class ranks that are intended to be in force at the same time. A Performance Policy must include at least one Performance Objective and Performance Class rank for each Performance Class, unless the Performance Class is marked Measure-Only. A Performance Policy optionally includes server pool directive overrides to set a baseline configuration of server resources for the time period in which the policy is active.

Performance Satisfaction Metric

A normalized numeric value that indicates how well a particular Performance Objective is being met, and which enables Oracle Database QoS Management to compare the performance of the system for widely differing Performance Objectives.

Policy Set

A Policy Set is a wizard-generated XML document that governs the operation of Oracle Database QoS Management. A Policy Set specifies server pools and their hosted Performance Classes, the collection of Performance Policies that specify the Performance Objectives for each Performance Class, and the server pool directive overrides for each Performance Policy.

program name

Program name is a database session attribute set by an application that is generally used to identify the program making the database request.

raw data

Raw data is data that has not been post-processed in any way. Counts, totals, and individual sample values are examples of raw data.

resource

A resource is a shared item that has limited quantity that is required to process a request. For example, CPU Time, threads, memory, I/O devices, slots in queues, network bandwidth, and temp space are all resources. Servers typically provide resources.

resource allocation control

A resource allocation control (also informally known as a *knob*) is a parameter, or collection of parameters, to a resource allocation mechanism. Examples of a resource allocation control include:

- A Consumer Group for Oracle Database Resource Manager
- The number of servers in a server pool

resource allocation mechanism

A resource allocation mechanism is something that gives an external entity such as a person or Oracle Database QoS Management the ability to control how some collection of resources are allocated. Oracle Database Resource Manager is an example of a Resource Allocation Mechanism.

resource metric

A resource metric is a metric that can be measured for any resource. Examples include Resource Usage Time and Resource Wait Time.

Resource Usage Time

Resource usage time is the cumulative time that a work request had exclusive use of a resource.

resource use

Resource use is a measurement that accumulates a specified set of elapsed time measurements into a single number. For example, a measurement of the CPU time spent on a given work request on a given server is a resource measurement: the specified work request uses the CPU for many separate intervals of time as the work request is processed.

resource wait time

Resource wait time is the cumulative time spent waiting for a resource by a work request that is ready to use that resource.

response time

The time between the server receiving a transaction request and sending out a response after committing or aborting the transaction.

rogue work

A work request that uses significantly more resources than expected; for example, the work request may be in a non-terminating loop. In some systems, facilities are provided to stop or re-prioritize rogue work.

routing

Routing is the act of choosing the path that a work request takes through the system. This includes all choices made when picking an entity in another tier of the system to which to pass a work request.

server

A server is a shared computer, typically not dedicated to a single user. A server can be as simple as a single CPU blade, or as complex as a large server with many CPUs sharing memory.

server pools

A server pool is a collection of servers created by the cluster administrator using either Enterprise Manager Cloud Control or the Server Control (SRVCTL) utility. Server pools are contained within tiers; each service is assigned to run in a specific server pool.

server pool importance

A number from 0 to 1000 (0 being least important) that ranks a server pool among all other server pools in a cluster.

server pool maximum

The maximum number of servers that the server pool should contain.

server pool minimum

The minimum number of servers that the server pool should contain.

server pool directive overrides

High availability guidelines for the cluster administrator server to keep the cluster highly available.

service

A service provides a well-recognized value to a user (client) or group of users. A service is provided to an application, and runs on a system. For example, CollabSuite provides a set of services such as Files, Calendar, Web Conferences, and so on. See also database services.

Operational management decisions, such as the hours of operation, capacity planning, provisioning, placement, and so on, are made on a service-by-service basis.

service placement

The activities of starting, stopping, or relocating a database service

singleton services

Services within a server pool that has a size of one.

system

A shared collection of servers and their associated infrastructure (networks, firewalls, storage systems, and so on) over which a workload management instance operates.

system metric

System metrics are metrics that help us to connect the things that are happening at the different layers. They provide a framework within which the rest of the analysis can be done. Examples include request counts, Layer Response Time, Layer Active Time, and so on.

All tiers of the system must provide the same set of system metrics.

tag

When a work request is received by the system, an attempt is made to classify the type of work requested. The objective of classification is to determine which Performance Objective applies to this particular work request. The result of classification is a tag (the Performance Class name) that is carried with the work request as it proceeds through the system. The tag enables the work request to be associated with the Performance Objective for the workload (Performance Class).

tier

A tier is a logical processing component within a system. The tiers are stacked on top of each other to provide the end-to-end processing stack for a system. WebCache, OHS, OC4J, Oracle Database and Oracle Automatic Storage Management are examples of tiers.

There may be multiple entities in a given tier providing either redundancy or distinct functionality. For example, a system might include two OHS instances for higher availability and two databases, one for CRM, and the other for ERP.

uniform services

Services that must be offered on every node of a server pool.

UserName

The OCI_ATTR_USERNAME or the Oracle Database user that is used to authenticate to the database.

work request

A work request is the smallest atom of work that a user can initiate. A work request can be an HTTP request, a SOAP request, a SQL statement sent to the database, or the execution of a process. A work request arrives at a layer, perhaps from the outside world, perhaps from another layer. The work request is processed, and a response is generated; the response is sent back to the requester.

Index

A

access control
  support for Oracle ASM, *xii*
ACL attributes, *5-2*
ACTION, *1-11*, *2-4*
activating
  Performance Policies, *4-24*
active resource plan, *5-2*
adding
  classifiers to a Performance Class, *4-18*
  Performance Classes, *2-9*
  servers to server pools, *5-4*
  services to the Policy Set, *4-14*
administrator-managed databases
  support for, *xii*, *1-9*
administrators
  cluster, *3-2*, *3-3*, *5-2*
  creating for Oracle Database QoS Management, *4-30*
  creating the QoSAdmin user, *3-4*
  database, *5-2*
  initial configuration tasks, *3-1*
  OC4J, *4-30*
  Oracle Database QoS Management, *3-3*
alerts
  in Oracle Enterprise Manager, *1-17*, *2-10*
alternative recommendations, *1-20*, *4-8*
application contexts, *2-4*
applications
  classifying work requests, *1-11*
  definition, *1-22*
  parameters used by classifiers, *1-11*
  resource requirements, *1-8*
  supported connections, *2-4*
APPQOS_PLAN
  editing, *4-32*
APPQOSSYS user, *3-5*, *5-1*
audit log, *2-10*
authorized actions, *xi*, *4-20*, *4-25*
Automatic Database Diagnostic Monitor (ADDM), *2-6*
Automatic Workload Repository (AWR), *2-6*
average response times
average response times *(continued)*
  and Performance Classes, *2-9*
  and Performance Objectives, *1-13*, *2-3*
  definition, *1-13*, *1-27*

B

batch jobs, *1-13*
bequeath connections, *2-4*
bottlenecks
  and Performance Classes, *2-5*, *4-29*
  and Performance Objectives, *1-28*
  and recommendations, *2-10*
  CPU resource, *2-5*
  for Performance Classes, *1-28*
  Global Cache resource, *2-6*
  I/O resource, *2-6*
  locating using metrics, *1-28*
  Other resource, *2-6*
  resolving, *1-19*
business objectives, *1-8*

C

calendars, *1-5*
candidate server lists, *3-2*
CHM, *xii*
classification rules
  defaults, *1-11*
classifiers
  adding, *4-18*
  and multiple services, *2-4*, *4-16*
  and Performance Classes, *2-4*, *4-17*
  and service names, *1-12*
  and session attributes, *2-4*
  and tags, *2-5*
  creating, *4-15*
  default_pc, *4-19*
  deleting, *4-19*
  evaluation order, *5-3*
  for the default Performance Class, *4-19*
  modifying, *4-18*

classifiers *(continued)*
multiple attributes, 2-4
order of evaluation, 2-4
session attributes, 2-4
specifying evaluation order, 4-10, 4-19
clients
Java Database Connectivity (JDBC) clients, 2-4
name of program, 2-4
Oracle Call Interface (OCI), 2-4
supported connections, 2-4
tracking across database sessions, 2-4
cluster administrator, 3-2, 3-3, 4-30, 5-2
configuration recommendations, 2-5
configuration violations, 2-1, 2-2
configuring
load balancing goals, 2-2
Oracle Grid Infrastructure for a cluster, 3-1
Oracle RAC databases, 3-1
Performance Objectives, 4-20
server pool directive overrides, 4-13, 4-22, 4-25
server pools, 2-2
connecting to a database, 2-2
connection load balancing, 2-2
connection pools, 2-2
connections
bequeath, 2-4
load balancing, 2-2
load balancing at run time, 2-2
to databases, 2-4
consumer groups
recommendations, 1-18
copying Performance Policies, 4-23
CPU partition, 1-4
CPU resource bottlenecks
resolving, 2-5
CPU_COUNT
and multiple databases, 2-2, 3-3
default value, 2-2, 3-3
Create Policy Set wizard, 3-5
creating
initial Policy Set, 3-5
Oracle Database QoS Management administrator user, 4-30
Performance Classes, 4-15
Performance Policies, 4-20
QoSAdmin user, 3-4
server pools, 3-2, 4-13
services, 3-3
CRSCTL, 5-2

D

Dashboard
accessing, 4-3
analysis result for recommendations, 4-7
and recommendations, 1-17
Dashboard *(continued)*
and resource bottlenecks, 2-5
and workload surges, 2-10
description of General section, 4-3
implementing recommendations, 4-9
logging in, 3-3
main sections, 4-3
Performance Overview, 4-3, 4-5
Performance Satisfaction Metric chart, 4-5
QoS Status, 4-3
recommendations not generated, 5-4
Recommendations section, 4-3
Resource Use vs. Wait Time chart, 4-5
Resource Wait Times Breakdown, 4-3, 4-29
Situation Analysis, 4-8
viewing recommendation details, 4-8
viewing recommendations, 4-7
database
startup order determined by rank, 1-16
Database Configuration Assistant (DBCA), 3-2
database connections, 1-5
database parameters, 1-26
Database Resource Manager *See* Oracle Database Resource Manager
See Oracle Database Resource Manager
database services *See* services
See services
database sessions
tracking client requests, 2-4
databases
adding services, 4-14
and CPU_COUNT, 2-2, 3-3
compliant, 5-1
connecting with a service, 2-2
connections, 2-4
default service, 2-2
enabling Oracle Database QoS Management, 5-1
Oracle RAC One Node, 2-2
policy-managed, 1-9, 2-2, 5-1
services, 2-2
sharing a server pool, 2-2
singleton, 2-2
startup order, xii
upgrading, 1-9
defaults
and classifiers, 4-19
and CPU_COUNT, 2-2, 3-3
database services, 2-2
namespace, 2-4
OC4J_PORT, 5-4
password for QoSAdmin user, 4-30
Performance Policy, 2-9, 4-10
Policy Set, 1-5, 2-8
QoSAdmin password, 3-4
rank, 2-9
deleting

deleting *(continued)*
classifiers for a Performance Class, *4-19*
Performance Classes, *4-17*
Performance Policies, *4-24*
demand
changing levels of, *1-8*
surges, *1-1*, *1-3*, *1-8*, *1-17*
disabling
Oracle Database QoS Management for a cluster, *4-5*

E

Edit Policy Set wizard, *4-9*, *4-10*, *4-15*, *4-17*, *4-18*
enabling
Oracle Database QoS Management, *3-4*
Oracle Database QoS Management for a cluster, *2-8*, *3-7*, *4-5*, *5-1*
Oracle Database QoS Management for a database, *5-1*
tracing, *5-5*
Enterprise Manager
integration with Oracle Database QoS Management server, *1-16*
evaluating classifiers, *4-17*
evaluation order of classifiers, *4-19*

F

fast application notification (FAN), *2-2*
FAST_START_MTTR_TARGET, *1-26*
floaters, *2-9*
Free pool, *1-9*, *2-1*, *3-2*

G

Generic pool, *2-1*
Global Cache resource bottlenecks
resolving, *2-6*
goals
and services, *2-2*
LONG, *2-2*
SEVICE_TIME, *2-2*
SHORT, *2-2*
THROUGHPUT, *2-2*
Grid Infrastructure Management Repository, *xii*
GV$ view queries, *2-4*

H

headroom
maximizing, *2-9*
high availability, *1-26*

I

I/O resource bottlenecks
I/O resource bottlenecks *(continued)*
resolving, *2-6*
implementing
alternative recommendations, *4-8*
automatic actions, *xi*
recommendations, *4-9*
initial Policy Set, *3-5*
initialization parameters
CPU_COUNT, *2-2*, *3-3*
instance caging, *1-4*
IP addresses
IPv6 support, *xii*

J

Java Database Connectivity
clients, *2-4*
JDBC *See* Java Database Connectivity
See Java Database Connectivity

L

LMS processes, *1-16*
load balancing
configuring goals for a service, *2-2*
connections, *2-2*
run-time connection, *2-2*
load balancing advisory
data, *2-2*
goal, *1-5*, *2-2*
log files, *5-4*

M

maintenance windows, *1-5*
managed server pools
and Oracle RAC database instances, *2-2*
managing
server pools, *3-5*, *5-3*
users, *4-30*
measure-only
Performance Classes, *1-13*, *4-21*, *4-22*
Performance Policies, *2-9*
measure-only mode, *xi*, *2-9*
memory pressure, *xi*
MemoryGuard, *xi*, *xii*
metrics
collecting, *1-16*
for performance, *1-27*
for Performance Classes, *4-3*
for resources, *1-28*
not displayed, *5-3*
performance satisfaction, *1-28*
minimum response times, *2-9*
minimum size of server pools, *2-9*
modifying
classifiers for a Performance Class, *4-18*

modifying *(continued)*
Performance Classes, *4-18*
Performance Policies, *4-22*
server pool settings, *4-13*
MODULE, *1-11*, *2-4*
multi-tier environments, *2-4*

N

namespaces
default, *2-4*

O

OC4J
administrator, *4-30*
default port, *5-4*
OC4J container
locating, *3-4*
OC4J_PORT environment variable, *5-4*
OCI *See* Oracle Call Interface
See Oracle Call Interface
OCIAttrSet function, *2-4*
ODP.NET, *1-11*
OLTP workloads, *2-3*
open workloads, *1-1*
Oracle ASM
access control, *xii*
patching, *xiii*
role separation, *xii*
Oracle Call Interface
clients, *2-4*
Oracle Cluster Health Monitor, *xii*
Oracle Clusterware Repository, *3-5*, *4-30*
Oracle Database QoS Management
administrator, *3-1*, *3-3*
changing active Performance Policy, *4-24*
Dashboard, *4-3*
See also Dashboard
See also Dashboard
Dashboard sections, *4-3*
disabling for a cluster, *4-5*
enabling for a cluster, *2-8*, *3-4*, *3-7*, *4-5*
enabling for a database, *3-4*
log files, *5-4*
selecting server pools, *4-10*
server hosting the OC4J container, *3-4*
status, *4-3*
trace files, *5-4*
troubleshooting, *5-1*
using the Dashboard, *4-3*
viewing cluster management status, *4-1*
viewing current status, *3-7*
Oracle Database QoS Management administrator
creating, *3-4*
default password, *3-4*
Oracle Database Resource Manager
Oracle Database Resource Manager *(continued)*
enabling, *5-2*
Oracle Enterprise Manager alerts, *2-10*
Oracle Enterprise Manager Dashboard, *1-17*
See also Dashboard
See also Dashboard
Oracle RAC databases
creating, *3-2*
supported releases, *2-2*
Oracle RAC high availability framework, *2-2*
Oracle RAC One Node databases, *2-2*
Oracle Real Application Clusters (Oracle RAC)
and Oracle Database QoS Management, *1-23*
Oracle Wallet, *3-5*
overloaded systems, *1-19*
overrides *See* server pool directive overrides
See server pool directive overrides

P

parallel queries, *2-4*
patching
Oracle ASM, *xiii*
performance class
ranks, *1-16*
Performance Classes
adding, *2-9*
adding classifiers, *4-18*
adding services, *4-14*
and alerts, *2-10*
and classifiers, *2-4*
and homogenous requests, *2-3*
and services, *1-11*
changing the rank, *4-22*
classifier ordering, *5-3*
classifiers, *2-4*
configuring Performance Objectives, *4-5*
creating, *4-15*
creating classifiers, *4-15*
default, *4-19*
default names, *1-5*
default rank, *2-9*
deleting, *4-17*
deleting classifiers, *4-19*
evaluating classifiers, *4-17*
improvements in performance, *1-19*
Incomplete data, *5-3*
list of, *4-3*
managing work requests separately, *1-12*
maximum, *2-5*
measure-only, *xi*, *1-13*, *4-5*
metrics, *4-3*
metrics missing, *5-3*
modifying, *4-18*
modifying classifiers, *4-18*
multiple classifiers, *4-17*

Performance Classes *(continued)*
No demand, *5-3*
Nonconforming data, *5-3*
projected impact of recommendations, *1-16*
ranks, *1-15*
relation to Performance Objectives, *1-13*
renaming, *4-17*
satisfying Performance Objectives, *1-16*
setting evaluation order of classifiers, *4-10*
specifying evaluation order of classifiers, *4-19*
specifying ranks, *4-20*
suggested by application developer, *1-12*
unable to collect metrics, *5-3*
used in Performance Policies, *2-9*
viewing performance metrics, *4-5*, *4-26*, *4-28*
viewing resource wait times, *4-3*
wait times for resources, *4-29*
performance metrics, *1-27*
performance objectives
importance, *1-13*
supported types, *2-3*
Performance Objectives
and services, *2-9*
configuring, *4-5*
definition, *1-13*
managing, *1-12*
ranks, *1-17*, *2-9*
relation to Performance Classes, *1-13*
specifying, *4-20*
specifying values, *4-21*
Performance Policies
activating, *4-24*
and server pools, *2-1*
changing active, *4-24*
copying, *4-23*
creating, *4-20*
creating additional, *2-9*
default, *2-9*
Default policy, *4-10*
definition, *1-12*
deleting, *4-24*
measure-only, *2-9*
modifying, *4-22*
ranking Performance Classes, *1-15*
refining with Performance Classes, *2-9*
scheduling the active policy, *4-24*
server pool directive overrides, *1-13*
performance projections, *2-10*
Performance Satisfaction Metric (PSM), *1-28*, *2-10*, *4-5*, *4-21*
performance tuning, *2-6*
Policy and Performance Management engine, *1-16*
policy changes
logs, *2-10*
Policy Set
active policy, *4-3*
Policy Set *(continued)*
adding server pools, *4-13*
administrating, *4-9*
creating a new Policy Set, *4-9*
current policy, *4-3*
default, *1-5*
initial, *1-5*
modifying, *4-10*
Policy Set wizard
Define Classifier page, *2-9*
Policy Sets
contents, *1-5*
creating, *3-5*
default, *1-11*, *2-8*
default settings, *3-5*
policy-managed databases, *2-2*, *5-1*
PROGRAM, *1-11*, *2-4*

Q

QoS administrator
initial configuration tasks, *3-1*
qosadmin, *3-3*
QoSAdmin user
default password, *3-4*, *4-30*
qosctl, *3-4*
QOSCTL utility
syntax, *4-30*

R

ranking Performance Objectives, *2-9*
ranks
and database startup order, *xii*
and Performance Classes, *1-15*, *4-22*
default, *2-9*
Performance Classes, *4-20*
recommendations
alternative, *1-20*
alternatives, *1-19*, *4-8*
analysis results, *4-7*
choosing a different action, *4-8*
consumer group mappings, *1-18*
generating, *1-19*
implementing, *1-17*, *1-19*, *4-9*
implementing alternative recommendations, *4-8*
interval, *4-5*
iterations, *1-22*
negative impacts, *1-19*
not being displayed, *4-9*
not generated, *5-4*
reallocating servers, *1-19*
section of the Dashboard, *4-3*
Situation Analysis, *4-8*
viewing, *4-3*, *4-7*
viewing details, *4-8*

Recommended Actions page, 4-8
removing servers from a server pool, 2-2
renaming
    Performance Classes, 4-17
requirements
    system, 2-1, 2-5
resolving
    CPU resource bottlenecks, 2-5
    Global Cache resource bottlenecks, 2-6
    I/O resource bottlenecks, 2-6
    Other resource bottlenecks, 2-6
resource bottlenecks
    resolving, 2-6
Resource Manager *See* Oracle Database Resource Manager
    See Oracle Database Resource Manager
resource plans
    APPQOS_PLAN, 3-5, 5-2
    editing, 4-32
    used by Oracle Database QoS Management, 1-24
resource types
    bottlenecked, 4-29
    CPU, 2-5
    Global Cache, 2-6
    I/O, 2-6
    OC4J, 5-4
    Other, 2-6
    viewing wait times, 4-3
Resource Use vs. Wait Time chart, 4-5
resources
    allocating, 1-3
    bottlenecks
        and recommendations, 1-16
    increasing CPU access, 1-22
    resolving bottlenecks, 1-19
    starved, 1-19
    types of, 1-28
    usage, 1-28
    wait times, 1-28
    waits, 1-27
response time, 1-13
response times
    average per request, 2-3
    definition, 1-27
    minimum achievable, 2-9
RMI port conflict, 5-4
role separation
    and Oracle ASM, xii
run-time connection load balancing, 2-2
run-time connection load balancing goal, 1-5

S

satrtup order for databases, xii
scheduling
    active performance policy, 4-24
server pool directive overrides
server pool directive overrides *(continued)*
    configuring, 4-13, 4-25
    specifying, 4-22
    valid values, 4-25
server pools
    accessing, 3-1, 5-2
    adding to a Policy Set, 4-13
    adjusting size of, 1-23
    and multiple databases, 2-2
    and recommendations, 1-19
    and services, 1-23
    attributes, 1-13
    candidate server lists, 1-9, 3-2
    configuring, 2-2
    constraints, 2-6
    creating, 3-2, 4-13
    directive overrides, 1-13
    execute permission, 5-2
    Free pool, 1-9, 3-2
    increasing minimum pool size, 4-25
    list of, 4-3
    managed, 2-1, 2-2
    marked as managed, 3-5
    maximum size of one, 2-2, 5-3
    minimum size, 1-18, 2-9, 3-1
    minimum size of one, 2-2
    modifying settings, 4-13
    modifying using Oracle Enterprise Manager, 4-13
    privileges, 3-2
    removing servers from, 2-2
    selecting for management, 4-10
    SERVER_NAMES attribute, 3-2
    servers not added, 5-4
    unmanageable, 5-3
    unmanaged, 2-1
    using multiple server pools, 1-8
    viewing current settings, 4-10
SERVER_NAMES attribute, 3-2
servers
    added to wrong server pool, 5-4
    hosting the Oracle Database QoS Management server, 2-10
service levels, 1-13
SERVICE_NAME, 1-11
SERVICE_TIME, 1-5
service-level agreements (SLAs), 1-22, 2-8–2-10, 4-21
services
    adding to the Policy Set, 4-14
    and Oracle Clusterware, 2-2
    and work requests, 1-23
    configuring load balancing goals, 2-2
    creating, 3-3
    default database service, 2-2
    defaults
        Performance Classes, 1-11
    discovered automatically, 2-8

services *(continued)*
for database connections, 2-4
load balancing, 2-2
names, 2-4
singleton, 1-23
SINGLETON, 2-2, 5-3
specifying multiple, 4-16
transactional shut down, 1-22
UNIFORM, 2-2
used by workloads, 2-2
session attributes
ACTION, 2-4
and classifiers, 2-4
MODULE, 2-4
program, 2-4
user name, 2-4
Set Policy, 4-24
singleton databases, 2-2
SINGLETON services, 2-2, 5-3
Situation Analysis, 4-8
SLA *See* service-level agreements
See service-level agreements
slice, 1-4
SQL performance issues, 2-6
SRVCTL, 2-2, 4-13, 4-30, 5-2, 5-4
storage interconnect, 2-6
support for Administrator-Managed databases, *xii*, 1-9
supported
connection types, 2-4
Oracle RAC database releases, 2-2
response times, 2-3
workloads, 1-23, 1-28, 2-3
system parameters, 1-26
system performance
evaluating, 1-16, 1-19
system requirements, 2-1, 2-5

T

tagging, 1-12
tags
and classifiers, 2-5
for work requests, 1-12
trace files, 5-4
tracing
enabling, 5-5
troubleshooting
enabling Oracle Database QoS Management for a cluster, 5-1
enabling Oracle Database QoS Management for a database, 5-1

U

UNIFORM services, 2-2
user name, 2-4
USERENV namespace, 2-4
USERNAME, 1-11
users
APPQOSSYS, 5-1
creating administrators, 4-30
default for QoSAdmin, 4-30

V

viewing
current server pool settings, 4-10
latest analysis results, 4-7
performance metrics, 4-26, 4-28
recommendation details, 4-8
recommendations, 4-7
resource wait times, 4-29
violations
and CPU_COUNT, 3-3
configuration, 2-1, 2-2, 4-10
for Performance Objectives, 2-10
logs, 2-10
resource plan not enabled, 5-2
viewing, 4-3

W

wait times
for resources, 4-29
viewing, 4-3
wizards
Create Policy Set, 3-5
Edit Policy Set, 4-9, 4-10, 4-15, 4-17, 4-18
Policy Set, 2-9
work requests
and services, 1-23
and tags, 1-11
creating classifiers, 4-16
definition, 1-22
managing separately, 1-12
metrics, 1-27
workload rank
and database startup order, 1-16
workloads
and creating new Performance Classes, 2-3
and GV$ view requests, 2-4
and parallel queries, 2-4
and server pools, 1-7
calls and Performance Classes, 2-4
consolidating, 1-7
database connections, 1-12
different resource requirements for work subsets, 2-3
managing, 2-4
monitoring, 1-5
OLTP, 2-3
scaling, 1-8
separating, 1-3

workloads *(continued)*
supported response times, *2-3*
supported types, *1-23*, *1-28*, *2-3*
surges, *2-7*, *2-9*, *2-10*
workloads *(continued)*
tuning goals, *1-8*
See also Performance Classes