<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Overview of Oracle Database QoS Management</td>
<td>1</td>
</tr>
<tr>
<td>Benefits of Using Oracle Database QoS Management</td>
<td>2</td>
</tr>
<tr>
<td>Operational Overview of Oracle Database QoS Management</td>
<td>3</td>
</tr>
<tr>
<td>The QoS Management Policy Set</td>
<td>4</td>
</tr>
<tr>
<td>QoS Management and Server Pools</td>
<td>5</td>
</tr>
<tr>
<td>QoS Management and Performance Classes</td>
<td>6</td>
</tr>
<tr>
<td>QoS Management’s Performance Policies</td>
<td>6</td>
</tr>
<tr>
<td>Collecting and Analyzing Performance Metrics</td>
<td>8</td>
</tr>
<tr>
<td>QoS Management’s Resource Recommendations</td>
<td>9</td>
</tr>
<tr>
<td>Implementing a QoS Management Recommendation</td>
<td>12</td>
</tr>
<tr>
<td>Protecting Workloads with QoS Management’s Memory Guard*</td>
<td>13</td>
</tr>
<tr>
<td>Measuring and Monitoring Performance with QoS Management</td>
<td>14</td>
</tr>
<tr>
<td>High Availability Management and QoS Management</td>
<td>14</td>
</tr>
<tr>
<td>Conclusion</td>
<td>15</td>
</tr>
</tbody>
</table>
Introduction

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. With an open workload comes a new type of application failure that is caused by demand surges that cannot be fully anticipated or planned for. To keep applications available and performing within their target service levels in this type of environment, one must pool resources, have management tools that detect performance bottlenecks in real time, and reallocate resources to meet the change in demand.

Oracle Database Quality of Service Management allows system administrators to directly manage application service levels hosted of Oracle Real Application Cluster databases. Using a policy-based architecture, QoS Management correlates accurate run-time performance and resource metrics, analyzes this data with its expert system to identify bottlenecks, and produces recommended resource adjustments to meet and maintain performance objectives under dynamic load conditions. Should sufficient resources not be available QoS will preserve the more business critical objectives at the expense of the less critical ones.

In conjunction with Cluster Health Monitor, QoS Management’s Memory Guard* detects nodes that are at risk of failure due to memory over-commitment. It responds by automatically preventing new connections to the stressed node thus preserving existing workloads and restores connectivity once sufficient memory is again available.

Overview of Oracle Database QoS Management

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

*NOTE: starting in Oracle Grid Infrastructure Release 12.1.0.2 Memory Guard is integrated into Oracle Clusterware and no longer requires QoS Management to be enabled.)
Oracle Database QoS Management monitors the performance of each work request on a target system. It starts to track a work request from the time it 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 - the time spent using resources and the time spent waiting to use resources - QoS Management can quickly detect bottlenecks in the system. QoS Management then makes suggestions to reallocate resources to relieve a bottleneck, thus preserving or restoring service levels.

Oracle Database QoS Management manages the system resources such 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, more critical business performance requirements are met at the expense of less critical ones;

- When load conditions severely exceed capacity, resources remain available.

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-related 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
Meeting your Performance Objectives with Quality of Service Management

- 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
- Protects existing workloads from over committed memory-induced server failures

Oracle Database QoS Management helps manage the resources that are shared by applications in a cluster. It 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. QoS Management does this by managing the time a workload is waiting for resources based upon its following relationship to performance.

\[
\text{RESOURCE USE TIME} + \text{RESOURCE WAIT TIME} = \text{APPLICATION PERFORMANCE}
\]

A systems or database administrator has no control over a workload’s use of resources in a production system. This is determined during development and test cycles. However, QoS Management provides resource re-allocation recommendations and dynamic runtime control over resource wait times. Therefore, administrators can manage to SLA’s by trading off resources between applications with the full knowledge of the impact of such a re-allocation.

**Operational Overview of Oracle Database QoS Management**

In previous database releases, you could use services to manage the workload on your system by starting services on groups of servers that were dedicated to particular workloads. At the database tier, for example, a group of servers might be dedicated to online transaction processing (OLTP), while another group of servers is dedicated to application testing, and a third group is used 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. The workloads are isolated from each other to prevent demand surges, failures, and other problems in one workload from affecting the other workloads. In this type of deployment, each workload must be separately provisioned for peak demand and failover because resources are not shared.

Starting with Oracle Database 11g release 2, you can create logical pools of servers within a cluster to provide workload isolation. Oracle Database QoS Management can recommend/implement moving a server from one server pool to another based on the measured and projected demand, to restore the Performance Objectives currently in effect. QoS Management can also recommend/implement promotion of a service to a different consumer group in the Oracle Database Resource Manager plan to give the workload using
that service more frequent access to the CPU. Finally when more than one database shares a set of servers, QoS Management can recommend/implement moving CPUs between databases to support running more sessions in parallel.

Workload is monitored for clients and applications that connect to the database using database services that are managed by Oracle Clusterware.

The QoS Management Policy Set

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 and Rank for each Performance Class. A Policy Set can also specify constraints for resource availability. The Performance Policies used by QoS Management are implemented system-wide. These policies are used to manage the availability of resources for each Performance Class so that the Performance Objectives specified in the Performance Policy are satisfied.

Oracle Database QoS Management provides default classification rules and associated Performance Class names when a new Default Policy Set is created using Oracle Enterprise Manager. For example, all database services in a cluster are discovered when an initial Policy Set is created. A Performance Class for each of these services is created. The Performance Class is named by appending “_pc” to the service name, for example, sales_pc.

![Figure 1: QoS Management Policy Set](image)

Only one Performance Policy in the Policy Set can be active at any time. Performance policies can be activated based upon a calendar schedule, maintenance windows, events, and so on.

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, 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 might 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. These workload priorities can be expressed in the policy by ranking Performance Classes relative to each other.

QoS Management and 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.

Figure 2: Database Applications deployed in server pools

As the administrator, you can define the workloads that can run in various server pools, as shown in Figure 2. 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, considering the example illustrated in Figure 2, connections and applications that use the ERP service access only the servers in the BackOffice server pool, so that work done by those connections does not interfere with the applications using the services running in different server pools such as Online and HR. QoS Management also assists in managing the resource allocations within each of the server pools, for example between AR, AP and ERP in the BackOffice server pool,
to meet your service levels and in automatically redistributing resources to meet changes in your business requirements.

QoS Management and Performance Classes

When you create a Policy Set, you define the Performance Objectives for various Performance Classes, or workloads, that run on your cluster. In order to determine which Performance Class a work request belongs to, a set of classification rules is evaluated against the work requests when they establish a database session. These rules enable value matching against session attributes of the work request; when there is a match the work request is classified into its associated Performance Class. The default classifier used to assign work requests to Performance Classes is the name of the service that is used to connect to the database.

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 objective. In effect, the tag connects the work request to the Performance Objective for the associated Performance Class. Tags are permanently assigned to each work request so that every component of the system can take measurements and provide data to QoS Management for evaluation against the applicable Performance Objectives.

To manage the workload for an application, the application code is configured to connect to the database 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 CHKOUT belong to SALESCART_pc Performance Class.

QoS Management’s Performance Policies

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 might define a Performance Policy for normal business hours, another for weekday nonbusiness hours, one for weekend operations, one to be used during processing for the quarter-end financial closing and one for maintenance periods. At any given time, a single Performance Policy is in effect as specified by the 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 meeting the performance objectives of certain workloads during specific time periods.
A Performance Policy has a collection of Performance Objectives in effect at the same time—one or more for each Performance Class identified workload that runs on the cluster. Some workloads and their Performance Objectives are more critical to the business than others. Some Performance Objectives might be more critical at certain times and less critical at other times. Therefore, Performance Policies allow you to rank each Performance Class from lowest to highest.

Performance Policies also include a set of Server Pool Directive Overrides that sets the availability properties of Min, Max, and Importance for a server pool when the Performance Policy is in effect. Server Pool Directive Overrides serve as constraints on the allocation changes that QoS Management recommends, because the server pool directive overrides are honored during the activation period of the Performance Policy. For example, 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.

Performance Policies can be created for your system to manage workload based on the time of year or time of day, as shown in Figure 1. Under normal conditions, these Performance Policies keep your database workload running at a steady rate. If the workload requests for a database daily surge for a period causing a performance objective violation, then the hosting server pool might require additional resources beyond what is specified by the current Performance Policy. This may indicate the need for an additional policy with different directives.

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 as in Figure 3, and specifying that the Back Office server pool can have a minimum of one server and a maximum of two servers, enabling Oracle Database QoS Management to move servers to the Online server pool, as needed.
Meeting your Performance Objectives with Quality of Service Management

After the telephone sales department closes, the workload for the sales applications decreases; however your back office workloads have been increasing. To accommodate this new demand you create the After Hours Performance Policy and specify that the Back Office server pool can have a minimum of two servers and 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.

Collecting and Analyzing Performance Metrics

The QoS Management Server retrieves metrics data from each database instance running in managed server pools. The data is correlated by Performance Class every five seconds. This data includes many metrics such as request average arrival rate and response time, CPU use and wait times, I/O use and wait times, Global Cache use and 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 QoS Management (illustrated in Figure 4) analyzes the data to determine the overall performance profile of the system with regard to 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 objective, the recommendation specifies which resource is the bottleneck. A specific resource corrective action is included in the recommendation if QoS Management can reallocate the bottlenecked resource. The recommendation also includes a listing of the projected performance impact on all Performance Classes in the system if you decide to implement the recommended action.

Figure 4 diagrams the collection of data from various data sources and shows how the QoS Management Server implemented as a set of J2EE MBeans and Oracle Enterprise Manager interact with that data. In this figure, CHM refers to Clusterware Health Monitor and Server Manager (SRVM) is a component of Oracle Clusterware.

QoS Management’s Resource Recommendations

If your business experiences periodic demand surges or must support open workloads, then to retain performance levels at all times for your applications you must design your system to support the peak workload and failover. Creating a system capable of handling the peak workload and failover typically means acquiring additional hardware to be available when needed and sit idle when not needed or run all servers at lower utilization. Instead of having servers remain idle except when a demand surge occurs, you might decide to use those servers to run other application workloads. However, if the servers are busy running other applications when a demand surge hits, your system might not be able to satisfy the peak workload and your main business applications do not perform as expected. QoS Management
enables you to both leverage and manage shared capacity to meet specific performance goals through its fine-grained monitoring, runtime analysis and recommendations.

When you use 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 is sent to Oracle Enterprise Manager for display in the 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, 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 QoS Management signals "No action required: all Performance Objectives are being met", as in Figure 5.

If Performance Objectives are not being met for a Performance Class, then QoS Management issues recommendations to rebalance the use of resources to alleviate bottlenecks. QoS Management evaluates several possible solutions and chooses the one that offers the best overall system improvement.

Figure 5: QoS Management Dashboard

If Performance Objective is not being met for a Performance Class, and the Performance Class accesses the same database as other Performance Classes, then QoS Management can recommend consumer group mapping changes as in Figure 6. Changing the consumer group mappings can give more access to the CPU resource to the Performance Class that is most critical to your business. It can also give less CPU access to competing Performance Classes. QoS Management issues consumer group mapping recommendations only for Performance Classes that are in the same database and server pool.
Another type of recommended action that QoS Management supports is changing the number of CPUs each database in a server pool can make use of. This occurs if there is more than one RAC database running on the servers in a server pool and you have allocated CPUs to each by using the “instance caging” feature of Oracle Database Resource Manager by setting each database’s CPU_COUNT value. Allocating another CPU to a database causes Resource Manager’s scheduler to increase the number of sessions that run concurrently, thereby reducing the wait time for each transaction.

The final type of recommended action that QoS Management can display is to move a server from one server pool to another to provide additional resources to meet the Performance Objective for a Performance Class. This will help all Performance Classes in the receiving server pool while reducing the performance of Performance Classes in the donor pool. A server pool will not be considered a donor if it is already at its minimum or has higher ranked Performance Classes that would violate their objectives should a server be taken. If there are no donor servers available then the displayed action will be “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, 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 as needed.
If you set the minimum size of a server pool to zero, you are allowing all servers to be removed if the workloads in the other server pools require the servers. This results in no further work appearing in Performance Classes hosted by database instances in that server pool. 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.

Implementing a QoS Management Recommendation

When trying to improve the performance of a particular Performance Class, QoS Management recommends adding more of the bottlenecked resource (such as CPU time) for that Performance Class or making the bottlenecked resource available more quickly to work requests in the Performance Class. The recommendations take the form of moving servers between server pools, moving CPUs between databases running in the same server pool, and promoting the target Performance Class to a higher Consumer Group, or demoting competing Performance Classes within the resource plan.

Implementing a recommendation makes the resource less available to other Performance Classes. When generating recommendations, 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 QoS Management reports that the performance gain is too small, and the change is not recommended.

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, or

- 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 QoS Management recommends an action that results in the highest overall performance gain with the least settling time or disruption. You can also view the alternative recommendations generated by QoS Management and see whether a specific action was recommended for implementation and what were its projected results. For example, one possible solution to resolving a bottleneck on the CPU resource is to demote the Performance Class that is using the CPU the most to a consumer group with fewer CPU shares. By limiting access to the CPU for the work requests in this Performance Class, the work requests in the other Performance Classes for that database get a larger share of the CPU time. However, QoS Management might have designated it an Alternate (Alt) action, not recommending it because the gain in response time
for the target Performance Class is too small. Figure 7 shows an example of such a list of actions.

![Figure 7: List of Recommended Actions](image)

**Protecting Workloads with QoS Management’s Memory Guard**

Enterprise database servers can use all available memory due to too many open sessions or runaway workloads. Running out of memory can result in failed transactions or, in extreme cases, a reboot of the server and the loss of a valuable resource for your applications. QoS Management detects memory over-commitment on a server in real time and its Memory Guard action automatically redirects new sessions to other servers to prevent using all available memory on the stressed server and to protect the existing workloads. Once the memory use is reduced, Memory Guard automatically restores the server’s availability to new workload connections.

When QoS Management is enabled and managing an Oracle Clusterware server pool, Cluster Health Monitor sends a metrics stream that provides real-time information about memory resources for the cluster servers to QoS Management. This information includes the amount of available physical memory and the amount of memory currently in use.

This memory data allows QoS Management to determine whether a server is at risk of overcommitting memory. If this condition is detected then Memory Guard will transactionally stop all the database services managed by Oracle Clusterware on that node, thus preventing new

*NOTE: starting in Oracle Grid Infrastructure Release 12.1.0.2 Memory Guard is integrated into Oracle Clusterware and no longer requires QoS Management to be enabled.
connections from being created. Once the existing workload completes sufficiently to reduce memory to safe levels, Memory Guard will automatically restart the services on that node and the listener will start sending it new connections.

Rerouting new sessions to different servers protects the existing workloads on the memory-stressed server and enables the server to remain available. QoS Management’s Memory Guard adds a new resource protection capability in managing service levels for applications hosted on Oracle RAC databases.

Measuring and Monitoring Performance with QoS Management

Up to this point the focus has been on how QoS Management supports runtime managing of a consolidated database, private database cloud or DBaaS environment. This requires a specific policy managed deployment type that supports server pools. This deployment type is not always possible or needs to be phased in. For these deployments the QoS Management Dashboard provides an excellent view of the overall database cluster performance from an application or workload perspective without the need to decipher internal database reports.

A Policy Set can be created where all performance classes are marked Measure-Only in the policy and optional performance objectives can be set. In this mode QoS Management will measure and monitor performance but not produce resource re-allocation recommendations. Alerts can still be generated should performance objectives be exceeded via the Enterprise Manager Cloud Control’s notification framework.

This allows QoS Management to be deployed in any RAC or RAC One Node cluster regardless of whether the databases are administrator or policy managed and provide through its dashboard a key performance indicator of the overall performance health of the cluster databases in a single view. This view can then be provided to application DBAs or middle tier administrators to easily understand the actual performance being delivered in real time.

High Availability Management and 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 QoS Management and its policy-based Performance Objectives to specify and maintain acceptable performance levels.

QoS Management is a runtime performance management tool that trades off resource allocations to help your system meet service-level agreements under dynamic workload conditions. QoS Management provides recommendations to help the work that is most critical to your business get the necessary resources. It 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.
QoS Management is not a feature to use for improving performance; the goal of QoS Management is to maintain optimal performance levels and resource allocation. QoS Management assumes that system parameters that affect both performance and availability have been set appropriately, and that they are constant.

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. However, to the extent that availability is also defined by meeting service level agreements, QoS Management is a valuable and essential part of the solution.

Conclusion

Consolidating applications on shared resources or implementing Cloud technologies such as Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS) requires service level agreements and flexible architectures. Managing such a dynamic and distributed environment can be complex especially as more applications are exposed to the Internet and unbounded demand. Oracle QoS Management’s highly integrated design and implementation in Oracle Real Application Clusters, Oracle’s consolidation platform for databases provides the task-based, policy-managed, just-in-time resource management capability to deliver a complete solution.