Oracle Database 10g
Resource Manager

An Oracle White Paper
October 2005
INTRODUCTION

The ability to easily and accurately perform system and resource management is critical to maintaining application and database performance, scalability and availability.

Oracle Database 10g Resource Manager is the database feature that controls how resources are allocated in the database, giving database users more control over resource management decisions. The Resource Manager functionalities enable the prioritization of database operations by aligning the allocation of system resources to the business objectives and priorities of the enterprise. This paper describes these functionalities and gives examples of how real-world applications can benefit from using the Resource Manager feature.

SYSTEM AND RESOURCE MANAGEMENT

Availability of the database encompasses both its functionality and performance. If the database is available but users are not getting the level of performance they need, availability and service level objectives are not being met. Application performance, to a large extent, is impacted by how resources are distributed among various applications accessing the database. Unfortunately, operating systems distribute resources equally among all active user processes and are unable to prioritize one task over another.

With Oracle Database 10g Resource Manager it is possible to balance one user's resource consumption against other users and distribute system resources among tasks of varying importance and priorities to achieve overall enterprise goals. The features of Resource Manager allow you to regulate your database’s query workload so that queries of higher priority can run promptly and your system resources are used efficiently. Using the Resource Manager can reduce the need to increase the server’s processing capacity to maintain acceptable response times for certain categories of users as overall user activity increases over time.

More specifically, using the Oracle Database Resource Manager allows administrators to:

- Specify the distribution of available processing resources by allocating percentages of CPU time to different users and applications, with different levels of priorities. By doing so certain group of users can be guaranteed a
minimum amount of processing resources, regardless of the load on the
system and the number of users, and queries submitted by certain group of
users can be given higher priority for their execution over other database
operations.

- Limit the degree of parallelism for any operation performed by the members
  of a group of users.
- Control the maximum number of concurrent user database calls within a
  group of users.
- Allow automatic switching of users from one group to another group with
different resource requirements, based on administrator defined criteria,
such as the maximum execution time of a query.
- Prevent the execution of operations that the optimizer estimates will run for
  a longer time than a specified limit.
- Specify the automatic cancellation of long-running SQL statements and
termination of long-running sessions so that they do not negatively impact
the overall performance of system.
- Specify the maximum amount of undo space that can be generated by a
  group of users.
- Limit the amount of time that a session can be idle. This can be further
defined to mean only sessions that are idle while blocking other sessions.
- Configure an instance to use a particular method of allocating resources.
  Administrators can dynamically change the method, for example, from a
daytime setup to a nighttime setup, without having to shut down and restart
the instance.
- Control the resources allocated to batch jobs, and specify different
  automatic resource allocations among jobs at various periods of time.

ESTABLISHING RESOURCE PLANS AND POLICIES

Resource Plans and Resource Consumer Groups
Users or applications that share common resource requirements and processing
needs are grouped into resource consumer groups. Administrators can configure the
Resource Manager to automatically assign consumer groups to sessions by
providing mappings between session attributes and the consumer group. Those
mappings can be done by using any of the following attributes:

- Oracle user name,
- Client machine name,
- OS user name,
- Client application name,
• Service name, as used by the client to establish the connection to the database,

• Module or action name, as annotated by the application.

The Resource Manager allocates resources among users or applications based on a resource plan specified by the database administrator. A resource plan specifies the resource consumer groups belonging to the plan and contains resource plan directives specifying how resources should be distributed among the resource consumer groups.

Plans within plans, or sub-plans can be used to further refine the repartition of resources within an application by defining independent entities within which unused resources are recycled. Additionally, available resources can also be allocated at multiple levels within a resource plan to specify hierarchies of resource consumer group allocation limits and how leftover resources at a given level should be further distributed within lower levels.

In a Real Application Clusters (RAC) environment, member instances can be assigned different resource plans to tailor them to support different workloads. For example, in a two-node cluster, one instance can use a resource plan that locates most of its resources to online users while the other instance can use a different plan allocating most of its resources to batch operations. This allows different applications or users to use the same database without impacting each other.

These capabilities of the Database Resource Manager provide database administrators unprecedented control over the distribution of database resources and enable them to implement very simple, yet powerful resource allocation policies.

Resource Allocation Policies

Using resource plan directives, an administrator can define how resources should be distributed among various resource consumer groups. The resource available to a user or an application depends on resources allocated to the resource consumer group it belongs to. The grouping of various database users or applications among different resource consumer groups is done when the resource plan is created but the Resource Manager can also dynamically change the allocation of resources based on criteria defined by the administrator.

The Resource Manager provides several policies for allocating resources among resource consumer groups. These capabilities allow administrators to proactively limit the resources consumed by long running operations and hence prevent them from negatively impacting other users.
**CPU Method**

This method enables administrators to specify how CPU resources are to be allocated among consumer groups or sub-plans when the system is at 100% CPU utilization. By allocating a minimum amount of CPU to sessions in each consumer group, the Resource Manager can lower the CPU consumption of low-priority sessions and thus give more resources to sessions of users or applications of higher priority. By default, the resource allocation method uses percentages and can be used with multi-level plans. An alternative method, only available for single-level plans, uses ratios.

The multiple levels of CPU resource allocation (up to eight levels) provide a means of further prioritizing CPU usage within a plan schema. Level 2 gets resources only after level 1 is unable to use all of its resources. Multiple levels not only provide a way of prioritizing, but they provide a way of explicitly specifying how all primary and leftover resources are to be used.

**Active Session Pool with Queuing**

You can control the maximum number of concurrent database calls allowed at any time within a consumer group. This maximum designates the active session pool. When a call cannot be initiated because the pool is full, the session is placed into a queue. When an active call completes, the first session in the queue can then be scheduled for execution. It is also possible to specify a timeout period after which a session waiting for execution in the queue will timeout, causing it to terminate with an error. This method can be used in particular to restrict the number of batch jobs running during peak hours.

**Degree of Parallelism Limit**

Specifying the maximum degree of parallelism enables administrators to control the maximum number of parallel execution servers associated with any single operation within a consumer group. This method is particularly useful in data warehousing environments where it ensures that no single operation ran in parallel will hog all the processing resources of the system.

**Automatic Consumer Group Switching**

This method enables administrators to control resources by specifying criteria that, if met, cause the automatic switching of sessions to another consumer group (switch group), usually with lower priority. The criteria that can be used to determine the switching of the session are:

- The length of time that the session can execute before it is switched to the switch group. The Resource Manager can be directed to switch an active session to the switch group permanently (switch time) or temporarily for the

---

1 A session is active if it is running and consuming resources, not waiting idly for user input or waiting for CPU cycles.
duration of the top call \((\text{switch time in call})\), the session being restored to its original consumer group once the top call finishes.

- The estimated time of how long the operation will take to execute to its completion \((\text{switch estimate})\). If the database estimate is longer than the value specified as the switch time, then the database switches the session before execution starts.

**Canceling SQL and Terminating Sessions**

You can also use the consumer group switching directives to cancel long-running SQL queries or to terminate long-running sessions.

**Execution Time Limit**

You can specify a maximum execution time allowed for an operation. If the database estimates that the operation will run longer than the specified maximum execution time, the operation is not started. Database administrators can use this method to prevent unacceptably long resource intensive operations from starting.

**Undo Pool**

Administrators can specify an undo pool for each consumer group. An undo pool controls the total amount of undo that can be generated by the sessions of a consumer group. When the total undo generated by a consumer group exceeds its undo limit, the current SQL statement generating the redo is terminated. No other members of the consumer group can perform further data manipulation until undo space is freed from the pool. This method prevents run-away operations from consuming excessive undo space.

**Idle Time Limit**

You can specify an amount of time that a session can be idle, after which it will be terminated. You can further restrict such termination to only sessions that are idle and blocking other sessions from acquiring resources.

**USING THE DATABASE RESOURCE MANAGER IN THE REAL WORLD**

**OLTP Applications**

To illustrate how the database resource manager can be used in the real world, let’s examine how this might be used with an ATM banking application. The primary goal of such an application is to ensure a quick and predictable response time to ATM users. At the same time the database that supports this application can also be used to perform other batch processing such as summarizing monthly and/or yearly activities. In order to ensure that these operations do not have any undesirable impact on the response time for ATM users, a resource plan such as the following can be created:
The plan shown above ensures that at least 90% of system CPU resources are made available to service ATM users by capping the resource utilization of batch processing operations to 10%. However if there are no other activities going on the system (i.e. batch operations are not consuming any CPU resources), 100% of CPU resources will be used to service ATM users.

Specifying how resources are allocated among different classes of ATM users can further refine the previous plan. Let us assume that the bank categorizes its customers into two groups i.e. premium and standard based on the amount of their assets deposited at the bank. In order to ensure better service to premium customers, an “ATM Users” sub-plan is created. This sub-plan governs how resources allocated to ATM users will be sub-divided. In this case, premium users are allocated 80% of ATM users resources while standard users get the remaining 20%.
Using a sub-plan is different than creating two different ATM consumer groups in the primary plan. If we had done that, any resources allocated but not used by premium customers would have been distributed across all other resource consumer groups including “Batch Users”. ATM users sub-plan ensures that any resources unused by premium users resource consumer group are first offered to standard ATM users before being passed on to other resource consumer groups.

The resource plan discussed so far allocates resources at a single level i.e. any allocation not used by a sub-plan or resource consumer group is proportionally distributed among other sub-plans or resource consumer groups. Let us suppose that the bank introduces a third resource consumer group to this plan to ensure availability of at least 5% of CPU resources for printing statements for its customers. It is required that any resources not used by ATM users should be first offered for printing statements before being made available for batch processing. This can be accomplished by creating a multilevel plan such as the one shown below:

<table>
<thead>
<tr>
<th>Resource Consumer Group</th>
<th>Level 1 CPU Resource Allocation</th>
<th>Level 2 CPU Resource Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM Users Sub-plan</td>
<td>90%</td>
<td>0%</td>
</tr>
<tr>
<td>Printing Customer Statements</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Batch Users</td>
<td>5%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 2: Multilevel Banking Application Plan

The revised plan ensures that at least 5% of CPU resources are made available for printing customer statements by limiting the resource allocation at level 1 to 95%. Any resources that are either not allocated or not consumed at level 1 are made available to level 2 resource consumer groups/sub plans. Any unused resources from the ATM users quota is, therefore, first made available for printing statements before batch operations can use it. If we had created a single level plan with CPU resources divided in 90, 5, 5 ratio, unused resources allocated to ATM users would have been equally divided between printing statements and back office users.

Packaged Applications

For a database supporting a packaged ERP or CRM application, the requirements could be very different than those of the banking database. Unlike the database supporting ATM application where transactions tend to be short and predictable in nature, the workload of an ERP database can be highly varied. There may be a mixture of short transactions and long running batch jobs such as large parallel queries. Also, unlike the ATM database, it may not be possible to classify transactions beforehand. In such a system, the goal is to provide quick response
time to OLTP users by limiting the resources consumed by batch jobs. In order to meet this goal, the following plan may be created:

<table>
<thead>
<tr>
<th>Initial Resource Consumer Group</th>
<th>CPU %</th>
<th>Maximum Number of Concurrently Active Operations</th>
<th>Switch Resource Consumer Group</th>
<th>Maximum Estimated Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLTP</td>
<td>70%</td>
<td></td>
<td>Switch Group = BATCH</td>
<td>3 minutes</td>
</tr>
<tr>
<td>BATCH</td>
<td>30%</td>
<td>5</td>
<td>Switch Time = 3 minutes</td>
<td>12 hours</td>
</tr>
</tbody>
</table>

Table 3: Packaged Application Plan

This plan classifies the workload into two distinct groups, OLTP and BATCH. The high priority OLTP group is allocated 70% of CPU resources while the BATCH group gets the remaining 30% as well as the unused OLTP allocation. An operation is allowed to run in the OLTP group for a maximum of 3 minutes. If the operation is not complete in this time interval, it is automatically switched to the BATCH group thereby reducing its allocation of CPU resources. This automatic resource consumer group switching feature enables the Oracle Database Resource Manager to detect long running operations and limit their impact on the overall system performance.

With a small portion of CPU resources available to them, batch jobs may not get completed in time if too many of them are being executed concurrently. In order to prevent this, the plan restricts the maximum number of concurrently active batch jobs to 5. Once this maximum is reached, all other batch jobs will be queued and will be allowed to run only after one of the five running jobs gets completed.

Finally, any batch job which is likely to run for more than 12 hours is not allowed to begin execution allowing the database to block potentially rogue operations proactively.

This example demonstrates the immense power the Database Resource Manager provides to database administrators in maximizing the return on their hardware investment by letting them take control of how system resources are deployed. In addition, it automatically and proactively governs resource utilization to achieve service level agreements with minimal human intervention.

**Data Warehouse**

In a data warehouse, the primary goal is to maximize throughput, and responsiveness may not be as critical as in an OLTP or ERP application.
environment. A data warehouse might support several short, critical operations, but most of its workload is made up of long running queries.

<table>
<thead>
<tr>
<th>Initial Resource Consumer Group</th>
<th>CPU%</th>
<th>Maximum Number of Concurrently Active Operations</th>
<th>Switch Resource Consumer Group</th>
<th>Maximum Estimated Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Priority</td>
<td>60%</td>
<td>20</td>
<td>Switch Group = Low Priority</td>
<td>15 hours</td>
</tr>
<tr>
<td>Medium Priority</td>
<td>30%</td>
<td></td>
<td>Switch Time = 60 minutes</td>
<td></td>
</tr>
<tr>
<td>Low Priority</td>
<td>10%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Data Warehouse Plan

Using the plan shown above, critical operations are executed as a part of the “High Priority” resource consumer group and are allocated 60% of CPU resources. Since it may not always be possible for administrators to know beforehand whether a job should be run at medium or low priority, all “non-critical” operations start in the “Medium Priority” group. If they are not completed within an hour or if they are estimated to run for more than an hour, they will be switched to the “Low Priority” group. Any non-critical job which is estimated to run for more than 15 hours will not be allowed to start at all.

FLEXIBLE AND ADAPTABLE RESOURCE MANAGEMENT

Resource plans and policies that achieve the desired results in the morning may not do so in the evening. Therefore it is important that all aspects of the Resource Manager configuration may be changed while the database is running. Plans, sub-plans, levels, membership in resource consumer groups and resource allocation directives can all be changed dynamically without restarting the database instance.

A database administrator can create many different resource plans and resource consumer groups in a database. Any one of these plans can be made the default that gets activated automatically at the time the database is started. Other plans can act as alternate plans for different times of the day, month, quarter or other times that require a different resource allocation scheme. Using this ability, the jobs blocked from running in the previous examples due to the maximum estimated execution time specification may be allowed to run at a later time.

The resource allocation mechanism provided by the Database Resource Manager is significantly superior to a conventional priority based scheme. Using percentages to allocate CPU ensures that all resource consumer groups receive a certain minimum resource and hence cannot be starved by a high demand from other groups. In
addition, resource allocation policies remain constant across hardware changes. The distribution of resources among users or applications, therefore, remains the same no matter which hardware or operating system the database operates on.

Database Integration
The Resource Manager is fully integrated with other components of Oracle Database 10g.

Oracle Scheduler
Oracle Scheduler enables database administrators and application developers to schedule the execution of a task, or job, at a particular date and time or when a particular event occurs. The Scheduler and the Resource Manager are tightly integrated. Jobs that share common characteristic and behavior can be grouped into larger entities called job classes. Each job class can specify a resource consumer group as an attribute, to which all of its member jobs then belong. Administrators create windows to change resource allocation among jobs during various time periods. Each window specifies the resource plan to activate when the window becomes active and each job class specifies a resource consumer group or specifies a database service, which can map to a consumer group. A job that runs within a window therefore has resources allocated to it according to the consumer group of its job class and the resource plan of the window.

Security
The Database Resource Manager is fully integrated into the database security system. Administrators can create, update or delete resource plans and resource consumer groups using either a PL/SQL interface or the Oracle Database 10g Enterprise Manager console. The administrator assigns a user a default consumer group and required privileges. A user can switch his or her session’s resource consumer group to change the resources available to it if the user has been granted the necessary privileges. In addition, the administrator can change a user’s default consumer group or move any active session from one group to another dynamically.

Adaptive Degree of Parallelism
Finally, the Adaptive Degree of Parallelism (ADOP) feature takes the Database Resource Manager allocations into account while choosing the optimal degree of parallelism for a parallel operation. ADOP attempts to optimize system utilization by automatically adjusting the degree of parallelism for parallel queries and parallel DML operations.

Profiles
Oracle Database 10g continues to support the use of user profiles to implement hard resource limits but the Resource Manager provides a more sophisticated way of managing database resources since it can balance different requests for service
against each other within the defined resource allocation plan and proactively control the resource consumption of long running resource intensive processes.

**Administer the Resource Manager**

**Access through Enterprise Manager**

The easiest way to manage Oracle Database 10g Resource Manager is with the graphical interface of the Enterprise Manager.

![Oracle Enterprise Manager interface for managing the Database Resource Manager](image)

The Resource Manager interface can be reached through the Administrator page of the Enterprise Manager DB console. It provides access to pages used for viewing and creating consumer groups, maintaining resource group mappings, creating and editing resource plans and plan directives.

The Monitors page is used to select and activate plans, and to display statistics and charts that depict the current state of the active resource plan.
Oracle-supplied PL/SQL packages

Alternatively administrators can use the DBMS_RESOURCE_MANAGER PL/SQL package to maintain plans, consumer groups, and plan directives, and the DBMS_RESOURCE_MANAGER_PRIVS PL/SQL package to maintain privileges associated with the Resource Manager.

Oracle Database 10g also provides views to help administrators gather information and monitor the results of their Resource Manager settings.
CONCLUSION

Traditionally, all users and applications have been provided equal access to database resources. While this approach works in many cases, it fails to recognize a fundamental business need i.e. some activities in the enterprise are more important than others. With the Database Resource Manager in Oracle Database 10g, database administrators are able to implement resource allocation policies to guarantee measured database service to applications and users. Oracle Database 10g Resource Manager makes it extremely easy to deliver predictable service level with minimal human intervention and facilitates almost unlimited system scalability without compromising performance.