

# Managing Workloads in an Oracle RAC Environment using Resource Manager and Database Services – Siebel CRM as an example

*An Oracle White Paper  
August 2010*

# Managing Workloads in an Oracle RAC Environment using Resource Manager and Database Services – Siebel CRM as an example

## TABLE OF CONTENTS

Introduction.....	3
Scenario 1 .....	3
Scenario 2 .....	4
Database resource manager and database services .....	4
Type of Resource Manager Plans.....	4
Database Resource Manager Concepts .....	5
Steps to Create a Complex Resource Manager .....	6
Scenario 1: Detailed Steps on Using the Resource Manager to Prioritize Siebel OLTP over Siebel Batch Jobs .....	6
Scenario 2: Case Study of Telecom Personal, S.A. Use of DBRM to Manage Long Running Queries.....	8
Database Servers System Information .....	9
Resource Plan Implementation Details .....	9
Telecom Personal: Future Plans .....	10
Conclusion .....	11
References.....	12
Appendix A: Steps on creating a Complex Plan (Scenario 1) for giving higher priority to OLTP over batch work on Oracle 10g Database.....	13
Appendix B: Workload Management Test Scenario .....	15
Appendix C: Telecom Personal Implementation Scripts (Scenario 2) ....	16
Appendix D: Telecom Personal Implementation Scripts for canceling of Long Running Query .....	18

# Managing Workloads in an Oracle RAC Environment using Resource Manager and Database Services – Siebel CRM as an example

## INTRODUCTION

Customer Relationship Management applications, such as Siebel CRM, are generally one of the most critical applications in any corporation. Since they are used directly by the sales force, service staff and call centers in their interactions with customers, a consistent response time is critical, and any downtime or diminishing of services because of overrunning batch jobs, resource intensive reports, or long running queries could result in dissatisfied customers and in turn loss of revenue. Therefore, it is important for companies to ensure this application is not only always available, but also that these customer-serving employees are getting good, consistent response time from the application. With the combination of Database Resource Manager (Resource Manager) and Database Services (services) in an Oracle Real Application Clusters (RAC) environment, it is now possible to give these customer-facing application users higher priority over backend work by guaranteeing them more CPU resources. Conversely, if the users send a misbehaving request that consumes excessive resources and slows down the application, such as a long running query, that request can be deprioritized by moving it to a lower priority group with less resource allocation to reduce the impact to other users.

This technical white paper focuses on managing resource distribution on the database tier and provides information on how customers running Siebel on Oracle RDBMS can use the Resource Manager and services to better manage their workload. For example, if customers have OLTP (online users) and Batch (EIM) jobs running, the Resource Manager can be used to give more resources to response time critical OLTP workload, and the services can be used to segregate workload among the Oracle RAC nodes. This paper focuses on the following two scenarios to show the usage of the Resource Manager and services by Siebel Applications:

### Scenario 1

This scenario illustrates the use of services and the Resource Manager to segregate the workloads and to manage the CPU resources. It provides detailed steps on how to assign more resources to OLTP work over batch jobs.

## Scenario 2

This scenario is a real life example of how a Siebel customer uses the Resource Manager to control long running queries. It demonstrates how the Resource Manager can be used to prevent long queries from excessive CPU consumption.

### **DATABASE RESOURCE MANAGER AND DATABASE SERVICES**

Traditionally, operating systems (OS) regulate resource management for various applications, including Oracle databases that run on a system. However, the OS does not have the capability to prioritize database operations among sessions within a single database instance. The Database Resource Manager (Resource Manager) and Database Service (services) fill this gap by providing database administrators (DBAs) more control over database resource management. The Resource Manager and services can help in an environment with multiple concurrent database sessions by enabling DBAs to define the priority of one service relative to others. Essentially, high priority sessions, such as online users are allocated more resources to minimize response time, while low priority sessions such as batch jobs or reports, which are less sensitive to response time, could take longer. This gives the DBAs more granular control over resources and the ability to prioritize depending on the criticality of each task. The priority is defined in terms of either ratios or percentage of CPU consumption. The Resource Manager enables DBAs to classify sessions into groups based on session attributes, and then allocate CPU resources to those groups in a way that optimizes hardware utilization and also ensures higher priority work gets more system resources.

Services enable DBAs to configure a workload, administer it, enable and disable it, and measure the workload as a single entity. Services abstract the workloads in Oracle database for easier manageability. Services divide workloads into mutually disjoint groupings. Each service represents a workload with common attributes, service-level thresholds, and priorities. In Oracle Real Application Clusters (RAC), a service can span one or more instances and facilitate workload balancing based on transaction performance. Services not only enable DBAs to segregate the workloads, but also enable DBAs to get measurements for the different services. For Siebel, it is recommended to create separate services for batch, OLTP and reporting.

### **TYPES OF RESOURCE MANAGER PLANS**

With the Oracle Database Resource Manager, the DBA can now:

- Distribute available processing resources by allocating CPU resources in percentages. This scenario will be covered in detail in this paper.
- Allow automatic switching of sessions from one group to another group based on administrator-defined criteria. For example if a session runs for longer than a specified amount of time, then this session can be automatically switched to another consumer group with different resource

allocation. This is the second scenario that will be covered in detail in this paper.

- Prevent the execution of operations that are estimated to run for a longer time than a predefined limit.
- Configure an instance to use a particular resource allocation plan. The DBA can now dynamically change the resource allocation plan, for example, from a daytime plan to a nighttime plan, without having to shut down and restart the instance.

Note that there can only be one resource plan active at a time. In addition, resource manager will only kick-in if there's a resource constraint on the database server. It does not change the process OS priority, and controls CPU usage solely by limiting the number of running Oracle foreground processes and selecting who next gets scheduled.

For an up-to-date and complete list of DBRM features and more detailed information on database services, please refer to the Oracle RDBMS Administrator Guide available at <http://otn.oracle.com>.

## **DATABASE RESOURCE MANAGER CONCEPTS**

This paper will be using the following terms for creation of the Resource Manager plan:

- **Resource Plan.** A container for directives that specifies how the resources are to be distributed among various users (resource consumer groups).
- **Resource consumer groups.** A group of sessions that are grouped together based on resource requirements. The Resource Manager allocates resources to resource consumer group, not to individual sessions. One database user can have different sessions assigned to different resource consumer groups.
- **Resource plan directive.** A resource plan directive associates a resource consumer group with a particular plan and specifies how resources are to be allocated to that resource consumer group.

There are three methods to create a resource manager plan. The easiest method is to use Oracle Enterprise Manager. The other simple method is using the *CREATE\_SIMPLE\_PLAN* procedure. This procedure enables you to both create consumer groups and allocate resources to them by executing a single procedure call. The last method is the complex resource plan. It requires creating the plan, the directive and consumer group using separate procedure calls. In this paper, we will be using the complex method to create the Resource Manager plan.

## **STEPS TO CREATE A COMPLEX RESOURCE MANAGER PLAN**

There are two major parts in creating a complex resource plan. The first part creates the consumer groups and defines the resource allocation for each consumer group. The second part creates the mappings from sessions to consumer groups based on the session attributes.

Below are the steps for the first part:

1. Create a pending area
2. Create, modify, or delete the consumer group
3. Create a resource plan
4. Create resource plan directives
5. Validate the pending area
6. Submit the pending area

Below are the steps for second part:

1. Create a pending area
2. Grant permission to switch to a particular consumer group
3. Define the mappings from sessions to consumer groups
4. Validate the pending area
5. Submit the pending area

The procedures in the DBMS\_RESOURCE\_MANAGER PL/SQL package are used to complete the above steps. Please refer to Appendix A for more details on creating a resource plan.

## **SCENARIO 1: DETAILED STEPS ON USING THE RESOURCE MANAGER TO PRIORITIZE SIEBEL OLTP OVER SIEBEL BATCH JOBS**

Siebel Business Applications is an OLTP application. But it is commonly used to handle batch and OLTP requests. An example of the batch task is using the Enterprise Integration Manager (EIM) component to import or export data into or out of the Siebel database. An OLTP task is a call center agent or a bank teller working with a customer such as taking orders, working on service request or helping customer with their bank accounts. As response time is very important to the agent servicing a customer, companies do not want the batch job to interfere with the customer service task. Without the Resource Manager, DBAs will have to leave it to the OS to manage the CPU allocation. Unfortunately, the OS is unable to allocate resources based on the different types of database work. Given that most Siebel customers have a mixed workload, we have performed testing internally to determine if the Resource Manager and services can be used to help manage Siebel workload. The testing is performed based on the following test scenario to determine how well the two different types of work can be distributed

in a Siebel on Oracle RAC environment. Each test scenario is executed with both the Resource Manager enabled and disabled. The results are extrapolated based on the test completion time. We will present the test results and suggest how the different workloads, both OLTP and batch job, should be run in the Siebel on Oracle RAC environment. Please refer to Appendix A for implementation details of creating a resource plan used in Scenario 1.

Database service is used to segregate the workloads into OLTP and batch services and the Resource Manager is used to give higher priority to the OLTP service.

Once the resource plan has been configured and executed successfully, your Siebel application is ready to be managed by the Resource Manager. The following test scenarios are performed to determine if the Resource Manager is suitable to be used to manage Siebel OLTP and Batch workloads:

Scenarios		
Resource Manager Enabled	1	Run EIM on one node, OLTP across all nodes
	2	Run EIM and OLTP across all nodes
	3	Run EIM and OLTP on separate nodes
Resource Manager Disabled	4	Run EIM on one node, OLTP across all nodes
	5	Run EIM and OLTP across all nodes
	6	Run EIM and OLTP on separate nodes

For detailed test results of each scenario, please refer to Appendix B. Below is the graphical diagram of the time taken to complete each test scenario. Figure 1 shows the elapsed time for OLTP workloads. As shown in the figure, OLTP jobs running on both nodes with Resource Manager enabled has the best elapsed time of 96 minutes. This is a large improvement from running OLTP jobs only on a single node, which takes more than 200 minutes. In the uniform scenario, the overall response of batch and OLTP is degraded from the inefficiencies of running shared data batch from a single node. This degradation is not due to Resource Manager.

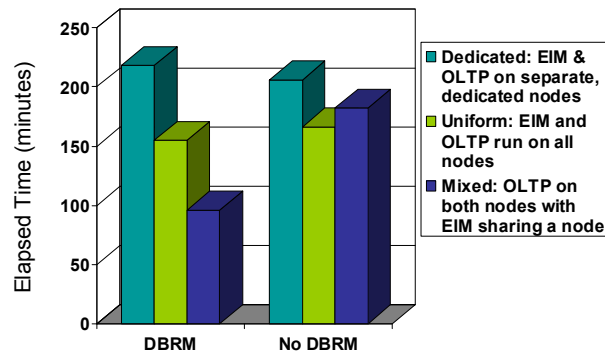


Figure 1: OLTP Result

Figure 2 shows the elapsed time to complete 3 batch jobs, importing account, contact and opportunity, for each test. The EIM batch job that is running on dedicated Oracle RAC node has the best elapsed time of less than 200 minutes to perform 3 imports. It takes approximately 470 minutes when running EIM across the two Oracle RAC nodes. Since the different EIM processes are doing large updates to related data, they run best when they run on a single node. Some expected degradation in EIM is seen when it no longer has a dedicated node.

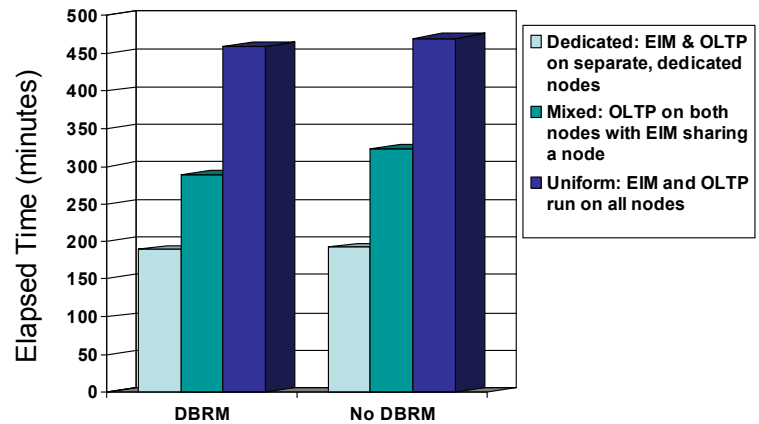


Figure 2: EIM Result

Based on the test results, OLTP workload performs most optimally when running across the Oracle RAC nodes with the Resource Manager enabled. The test shows that EIM batch job performs best when running on dedicated Oracle RAC node.

While the Resource Manager ensures the optimal distribution of resources, it can starve lower priority work. Thus when running both batch and OLTP jobs on the same node, it is important to monitor system resources, such as the CPU, memory and I/O, to ensure that the server has sufficient capacity to handle the load. If the server resources are maxed out, it is important that the workload is re-scheduled or additional capacity added to ensure the system performs optimally.

## SCENARIO 2: CASE STUDY ON TELEKOM PERSONAL, S.A. USE OF DBRM TO MANAGE LONG RUNNING QUERIES

Telecom Personal S.A., owned by Telecom Argentina, is one of the four mobile phone companies in Argentina. Personal provides a Global System for Mobile Communication (GSM) based mobile telephony and communication services. It was founded in 1996 and is based in Buenos Aires, Argentina.

Personal is one of the first Siebel customers in Latin American who have implemented Siebel Business Applications (CRM) on Oracle Real Application Clusters (RAC). All their call center agents use the CRM application to support their large customer base throughout Argentina. Time is of the essence, and customers expect prompt response to their calls. In order to guarantee application

performance, the Resource Manager is used to manage backend CPU resource allocations. The following are some key reasons why Telecom Personal uses Oracle Database Resource Manager:

- They want to ensure long running queries generated either by users or configuration changes do not impact their overall service level agreements. Using the Resource Manager and moving long running queries to lower priority consumer group helps ensure the queries do not consume the system resources and affect other online users.
- By moving the long running queries to a lower priority group and tagging them, it enables DBAs to investigate those queries and do fine-tuning accordingly.

Some information on their environment is provided in the following sections.

### Database Servers System Information

Telecom Personal's environment is running Siebel on a two-node Oracle RAC cluster. The hardware and software details are as follows:

System Information:

Item	Value
Machine & Version	SunOS Solaris 10
Platform	64 Bit Sun
Processor(s)	16 Processor(s) Installed
Total Physical Memory	64 GB
Swap	71 GB

Cluster/Database Information:

Item	Value
Cluster	Sun Cluster 3.2
Storage	ASM
RDBMS	Real Application Cluster 10.2.0.3

### Resource Plan Implementation Details

Planning: The first step is to plan and determine the business goals. This is followed by deciding on the number of resource consumer group needed to accomplish the goal.

Business Objective	Resource Manager Group	Explanation
Separate main workload, guaranteeing successful short query execution.	<b>HIGH_PRIORITY</b>	All the queries that complete within 3 minute execution will fall into this category
Long running queries should not impact the on-line performance	<b>LOW_PRIORITY</b>	All the queries that exceed 3 minute execution time
Reporting	<b>BATCH</b>	To be implemented.

Implementation: The Resource Manager plan could be implemented using scripts within SQL\*Plus. A much easier way to administer the Resource Manager is with the graphical user interface of Oracle Enterprise Manager. The sample source code for using SQL\*Plus is provided in Appendix C. To administer using Enterprise Manager, refer to the “Oracle Database 2 Day DBA” guide under the “Resource Manager” section for details. We strongly recommend that you test the plan before putting it into your production environment.

The main advantage of Telecom Personal’s implementation is to guarantee CPU resources to the Siebel users in the system. The consumer group for each session can be changed dynamically. Therefore, the Resource Manager can be used to prevent long running SQL from hogging the system by assigning those sessions to a “low priority” resource consumer group. The sessions can then be switched back to the “high priority” consumer group once they are done with the long running queries. Below is a snapshot of the resource utilization of each consumer group via the Enterprise Manager. It shows the ‘Consumed CPU Time’ by each consumer group. Noticed that HIGH\_PRIORITY consumer group has the highest consumed CPU time and it is exactly what Telecom Personal wishes to achieve, giving users whose SQL takes less than 3 minutes more resources.

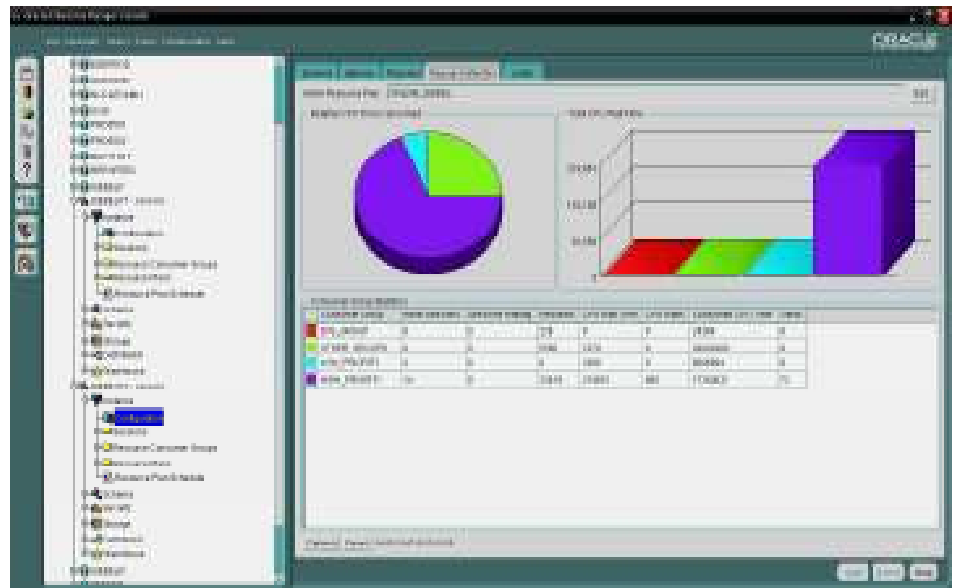


Figure 3: Resource Manager Administration with EM

### TELECOM PERSONAL: FUTURE PLANS

To further streamline the resource utilization and operation process, Telecom Personal is in the process of enhancing their current Resource Manager implementation. This involves updating the current plan to cancel queries that

would take ten minutes in LOWER\_PRIORITY group. This plan will help to save resources further on the database servers. Please refer to Appendix D for example on how to update the Resource Manager plan to cancel long running SQL.

## **CONCLUSION**

In summary, the Oracle Database Resource Manager is an effective tool for managing priorities and ensuring service levels agreement for applications such as Siebel CRM, whereby response time is critical. The Resource Manager should be seriously considered particularly when running mixed workload or when a task is of higher importance over another to avoid diminished service levels for application users. Siebel workloads currently run best when OLTP runs on all nodes with a high priority and when Batch runs on a single node with a lower priority. Database services are a critical tool to control workload placement and performance monitoring, and for Siebel it is recommended to create separate services for batch, OLTP and reporting.

## REFERENCES

- Oracle Database Concepts 10g Release 2 (10.2)
- Oracle Database Administrator's Guide 10g Release 2 (10.2)
- Oracle Database 10g Resource Manager white paper (Oct 2005)
- Oracle9i Database Resource Manager white paper (Jun 2001)
- Oracle Database Concepts 11g Release 1 (11.1)
- Oracle Database Administrator's Guide 11g Release 1 (11.1)

## APPENDIX A: STEPS ON CREATING A COMPLEX PLAN (SCENARIO 1) FOR GIVING HIGHER PRIORITY TO OLTP OVER BATCH WORK ON ORACLE 10G DATABASE.

### Part 1: Create the consumer groups and define the resource plan.

Create the resource consumer groups for OLTP and Batch work.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
(CONSUMER_GROUP =>'sebl_oltp', COMMENT => 'oltp group');
EXEC DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
(CONSUMER_GROUP => 'sebl_batch', COMMENT => 'batch group');
```

Create a resource plan.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN(PLAN => 'sebl_plan',
COMMENT => 'Siebel Plan');
```

Create resource plan directives. The plan directives request RM to give priorities to sys\_group, oltp, batch and the least priority to groups that fall into neither of the mentioned consumer group.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'sebl_plan', GROUP_OR_SUBPLAN => 'SYS_GROUP', COMMENT =>
'sys sessions', CPU_P1 => 100);
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'sebl_plan', GROUP_OR_SUBPLAN => 'sebl_oltp', COMMENT =>
'oltp sessions', CPU_P2 => 100);
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'sebl_plan', GROUP_OR_SUBPLAN => 'sebl_batch', COMMENT =>
'batch jobs', CPU_P3 => 100);
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'sebl_plan', GROUP_OR_SUBPLAN => 'OTHER_GROUPS', COMMENT =>
'mandatory', CPU_P4 => 100);
```

Validate and submit the pending area.

```
EXEC DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

### Part 2: Create the session to consumer group mappings and grant permission to the consumer groups.

Create a pending area using the Create\_Pending\_Area command.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

Grant the Switch Privilege to role PUBLIC to 'sebl\_oltp' and 'sebl\_batch' consumer group.

```
EXEC
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP
('PUBLIC', 'sebl_oltp', FALSE);
EXEC
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP
('PUBLIC', 'sebl_batch', FALSE);
```

Assign the services used by OLTP and Batch jobs to a consumer group.

```
EXEC DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
(DBMS_RESOURCE_MANAGER.SERVICE_NAME, 'siebel',
'sebl_oltp');
EXEC DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
(DBMS_RESOURCE_MANAGER.SERVICE_NAME, 'sebl_eim',
'sebl_batch');
```

\* siebel is the service name used by the OLTP users. sebl\_eim is the service name used by EIM job.

Validate and submit the pending area.

```
EXEC DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

Once the resource manager plan has been created and setup successfully, the next step is to enable resource manager on the database.

Enable Oracle Resource Manager via SQL\*Plus.

```
To enable RM, execute:
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'SEBL PLAN';
To disable RM, execute:
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = '';
```

To verify which consumer group the users are assigned or get an understanding of the workload, issue the following query on the v\$session or gv\$session.

```
Select username, resource_consumer_group from gv$session;
```

## APPENDIX B: WORKLOAD MANAGEMENT TEST SCENARIO

Workload Management Test	RM Enabled Elapsed Time	RM Disabled Elapsed Time
<b>EIM On One Node, OLTP Across both Oracle RAC nodes</b>		
a) OLTP (100 Users x 50 Iterations)	96	182
b) EIM Job: Import Account	109	115
c) EIM Job: Import Contact	99	104
d) EIM Job: Import Opportunity	81	103
Total EIM Time:	281	322
<b>EIM and OLTP run across all Oracle RAC nodes</b>		
a) OLTP (100 Users x 50 Iterations)	155	166
b) EIM Job: Import Account	115	143
c) EIM Job: Import Contact	126	169
d) EIM Job: Import Opportunity	103	158
Total EIM Time:	344	470
<b>EIM On One Node, OLTP on a separate Oracle RAC nodes</b>		
a) OLTP (100 Users x 50 Iterations)	218	206
b) EIM - Import Account	70	71
c) EIM - Import Contact	65	67
d) EIM - Import Opportunity	54	54
<b>Total EIM Time:</b>	<b>189</b>	<b>192</b>

Note that in scenario 1 testing, the batch job may or may not take longer than OLTP work. It was not our intent to simulate batch job overrun scenarios, but instead to show how resource manager manage workload by giving higher priority to OLTP using more CPU resources.

## APPENDIX C: TELECOM PERSONAL IMPLEMENTATION SCRIPTS (SCENARIO 2)

### Part 1: Create the consumer groups and define the resource plan.

Create the resource consumer groups for OLTP and Batch work.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
(CONSUMER_GROUP => 'high_priority', COMMENT => 'siebel
online high priority');
EXEC DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP
(CONSUMER_GROUP => 'low_priority', COMMENT => 'siebel
online low priority');
```

Create a resource plan.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN(PLAN =>
'online_siebel', COMMENT => 'siebel plan');
```

Create resource plan directives. The plan directives request the Resource Manager to give more resources to sys\_group and high\_priority consumer group, lesser resource to low\_priority consumer group and the least priority on resources to other\_groups that fall into neither of the mentioned consumer group.

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'online_siebel', GROUP_OR_SUBPLAN => 'SYS_GROUP', COMMENT
=> 'sys sessions', CPU_P1 => 100);

EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'online_siebel', GROUP_OR_SUBPLAN => 'high_priority',
COMMENT => 'switch to low priority after 3 minutes',
CPU_P2 => 80, SWITCH_GROUP => 'low_priority',
SWITCH_TIME_IN_CALL => 180);

EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'online_siebel', GROUP_OR_SUBPLAN => 'low_priority',
COMMENT => 'siebel online low priority', CPU_P2 => 20);

EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN =>
'online_siebel', GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
COMMENT => 'mandatory', CPU_P3 => 100);
```

Validate and submit the pending area.

```
EXEC DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

## Part 2: Create the session to consumer group mappings and grant permission to the consumer groups.

Create a pending area using the Create\_Pending\_Area command:

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

Grant the Switch Privilege to role PUBLIC to 'high\_priority' and 'low\_priority' consumer group.

```
EXEC
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP
('PUBLIC', 'high_priority', FALSE);
EXEC
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP
('PUBLIC', 'low_priority', FALSE);
```

Assign the services used by OLTP and Batch jobs to a consumer group.

```
EXEC DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
(ATTRIBUTE => DBMS_RESOURCE_MANAGER.SERVICE_NAME, VALUE =>
'siebel_atcli', CONSUMER_GROUP => 'high_priority');
EXEC DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
(ATTRIBUTE => DBMS_RESOURCE_MANAGER.SERVICE_NAME, VALUE =>
'siebel_pc', CONSUMER_GROUP => 'high_priority');
EXEC DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
(ATTRIBUTE => DBMS_RESOURCE_MANAGER.SERVICE_NAME, VALUE =>
'siebel_batch', CONSUMER_GROUP => 'low_priority');
```

Validate and submit the pending area.

```
EXEC DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

Once the resource manager plan has been created and setup successfully, the next steps is to enable resource manager on the database. It can be enabled using Alter System or Switch\_Plan command. In this example, we will activate it using switch\_plan.

Enable Oracle Resource Manager using switch\_plan.

```
EXEC DBMS_RESOURCE_MANAGER.SWITCH_PLAN(PLAN_NAME =>
'online_siebel', SID => 'siebelp2');
EXEC DBMS_RESOURCE_MANAGER.SWITCH_PLAN(PLAN_NAME =>
'online_siebel', SID => 'siebelp1');
```

## APPENDIX D: TELECOM PERSONAL IMPLEMENTATION SCRIPTS FOR CANCELING OF LONG RUNNING QUERY

```
EXEC
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE(PLAN =>
'online_siebel', GROUP_OR_SUBPLAN => 'low_priority',
NEW_CPU_P2=>20, NEW_SWITCH_GROUP => 'CANCEL SQL',
NEW_SWITCH_TIME => 600);
EXEC DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
EXEC DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```



Managing Workloads in an Oracle RAC Environment using Resource Manager and Database Services – Siebel CRM as an example  
August 2010

Author: Mark MacDonald

Contributing Authors: Kelly Tan, Erik Peterson

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

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

[oracle.com](http://oracle.com)

Copyright © 2010, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.