Oracle Enterprise Manager
Oracle Database and Application Testing

Real Application Testing Lab

Session S318966
Oracle Enterprise Manager 11g
Real Application Testing
Hands-on Lab

Introduction to Enterprise Manager 11g

Oracle Enterprise Manager 11g is the centerpiece of Oracle's integrated IT management strategy, which rejects the notion of management as an after-thought. At Oracle, we design manageability into each product from the start, enabling Oracle Enterprise Manager to then serve as the integrator of manageability across the entire stack encompassing Oracle and non-Oracle technologies. Fueled by this unique vision, Oracle Enterprise Manager 11g has introduced business-driven IT management to help IT deliver greater business value through three highly differentiated capabilities:

- **Business-driven application management**, which combines industry-leading capabilities in real user experience management, business transaction management and business service management to improve application users’ productivity while enhancing business transaction availability
- **Integrated application-to-disk management**, which provides deep management across the entire Oracle stack to reduce IT management complexity and eliminate disparate point tools
- **Integrated systems management and support**, which utilizes industry-first technology bring support services into the IT management console; enabling proactive IT administration, increased application and system availability, and improved customer satisfaction

Lab Configuration

Overview

For these lab exercises, you will need the following infrastructure components to be running and available.

- **Database: DB02: 11.1.0.7 Database**
- **Database: DB03: 10.2.0.4 Database**
- **Enterprise Manager Grid Control**

We will step through the simplified version of starting the necessary infrastructure components using the desktop.

Setup & Preparation

- You will be given a virtual machine address to use for this lab. For ease of reference, you may want to write this below:

  **Virtual Machine Address: ________________________________**

  Log into the VNC for the virtual machine you will be working with. VNC password is g0Oracle12#.
• **Host and Database Credentials**
  - Host userid/password: `oracle/g0Oracle12#`
  - Enterprise Manager Grid Control: `sysman/oracle1`
  - All databases: `system/oracle1`

• Once you log into your virtual machine, double click on “demos” folder All scripts used in this lab exercise can be found in the directory

• Open “**Enterprise Management**” folder by double-clicking the icon
• Open “Real Application Testing” folder by double-clicking the icon

This takes about **7-10 minutes** to complete, starts up DB2, DB03, GC and all their components.

**LAB EXERCISE 00 – REAL APPLICATION TESTING OVERVIEW**

**Introduction**

Reduce Testing Efforts by Up to 80 Percent

Agile businesses want to be able to quickly adopt new technologies, whether it's operating systems, servers, or software, to help them stay ahead of the competition. However, change often introduces a period of instability into mission-critical IT systems. Oracle Real Application Testing—with Oracle Database 11g Enterprise Edition—allows businesses to quickly adopt new technologies while eliminating the risks associated with change. Oracle Real Application Testing combines a workload capture and replay feature with an SQL performance analyzer to help you test changes against real-life workloads, and then helps you fine-tune the changes before putting them into production. Oracle Real Application Testing supports older versions of Oracle Database, so customers running Oracle Database 9i and Oracle Database 10g can use it to accelerate their database upgrades.

• Real Application Testing, a new database option of the Oracle 11g Enterprise Edition, reduces the task of assessing the impact of system changes for even complex production environment from months to days
• Database Replay, a feature of Real Application Testing is the only technology in the market that makes it possible to capture actual production workload including timing, concurrency and dependency information with negligible performance overhead and with minimal time and effort
• Database Replay allows captured workload to be replayed on a test system to assess the impact of change
• SQL Performance Analyzer, a feature of Real Application Testing, allows fine-grain impact analysis of database environment change on SQL execution plan changes and performance
• Existing self-managing database infrastructure leveraged to analyze and fix regressions caused by changes to provide a comprehensive solution
• Enhances system performance, reliability, and quality of service by accurately validating change before production deployment
A. Lab Scenarios and Objectives

In our fictitious company, CashBankTrust is currently evaluating technologies that will help maintain optimal performance within their database environment in the easiest possible manner.

- **Uses real workloads**—Replays real workloads, not synthetic or artificial loads
- **Comprehensive**—100 percent coverage for entire change lifecycle
- **Scalable**—Requires similar effort for small and large changes
- **Predictable**—Transfers exact resolution of changes from testing to production
- **Cost-efficient**—Reduces testing effort by up to 80 percent

The Real Application Testing labs that you will complete will demonstrate solutions specifically to the identified challenges below.

<table>
<thead>
<tr>
<th>Product</th>
<th>Identified Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Application Testing &amp; SQL Performance Analyzer</td>
<td>Today, enterprises have to make sizeable investments in hardware and software to roll out infrastructure changes. For example, a data center may have an initiative to move databases to a low cost computing platform, such as Oracle Enterprise Linux. This would, traditionally, require the enterprise to invest in duplicate hardware for the entire application stack, including web server, application server and database, to test their production applications. Organizations therefore find it very expensive to evaluate and implement changes to their data center infrastructure. In spite of the extensive testing performed, unexpected problems are frequently encountered when a change is finally made in the production system. This is because test workloads are typically simulated and are not accurate or complete representations of true production workloads. Data center managers are therefore reluctant to adopt new technologies and adapt their businesses to the rapidly changing competitive pressures. Changes that affect SQL execution plans can severely impact application performance and availability. As a result, DBAs spend enormous amounts of time identifying and fixing SQL statements that have regressed due to the system changes. SQL Performance Analyzer (SPA) can predict and prevent SQL execution performance problems caused by environment changes. For database administrators and application developers, application tuning is a critically important area and a considerable amount of their time is spent performing this very important function. A poorly tuned business application can potentially affect not just a few users but an entire business operation and for this reason companies invest significant resources to ensure smooth running of applications vital for their businesses.</td>
</tr>
</tbody>
</table>
## Identified Challenge –

Changes that affect SQL execution plans can severely impact application performance and availability. As a result, DBAs spend enormous amounts of time identifying and fixing SQL statements that have regressed due to the system changes. SQL Performance Analyzer (SPA) can predict and prevent SQL execution performance problems caused by environment changes. SPA provides a simple and easy to use solution for testing SQL response time deviations or “unit” testing of SQL workload.

## Introduction

**SQL Performance Analyzer**

Changes that affect SQL execution plans can severely impact system performance and availability. As a result, DBAs spend an enormous time in identifying and fixing SQL statements that have regressed due to the system change. SQL Performance Analyzer feature provides functionality similar to Database Replay, but is focused on predicting and preventing performance problems for any environment change that affects the SQL execution performance. SQL Performance Analyzer provides fine-grain assessment of environment change on SQL execution plan and statistics by running the SQL statements serially in before-change and after-change environments. A report that comprises of the net benefit on the workload due to the system change and regressed SQL statements is provided. For regressed SQL statements, appropriate executions plan details along with recommendations to tune them are provided. SQL Performance Analyzer functionality is well integrated with existing SQL Tuning Sets (STS) and SQL Tuning Advisor functionality. As a result, SQL Performance Analyzer completely automates and simplifies the manual and time consuming process of assessing the impact of change on extremely large SQL workloads (thousands of SQL statements). Thus, using SQL Performance Analyzer businesses can validate that a system change to production environment in fact results in net positive improvement at a significantly lower costs and with a high degree of confidence.
**Overview**

One of the biggest fears while managing a production database environment is whenever there are changes implemented into production, there are unknown and potential regression issues that could have a huge impact. The change could be as simple as introducing an index to deploying a Critical Patch Update. We will walk through this exercise with the objective in mind of identifying regressions, if any, that are introduced when we execute our 50 important SQL statements to the changed environment.

In this lab exercise, the SQL workload has already been captured into a STS named APPS.HR_WORKLOAD you will accomplish the following:

1. Create SPA Task – container for holding our SPA experiment results
2. Create SQL Tuning Trials (experiments)
3. Identify regression of SQL with the introduction of change
4. Make Change: Running the make change script effectively performs an equivalent of database patch application
5. Identifying and fixing SQL regressions
6. Validating Tuning thru another SPA trial and making sure no regressions exists before change is deployed in production

**Setup & Preparation**

- Go to the Linux Desktop, double click on “demos” folder. All scripts used in this lab exercise can be found in this directory

- Open “Enterprise Management” folder by double-clicking the icon
• Open "Real Application Testing" folder by double-clicking the icon
• Double click on **Step3a - Undo_Changes.sh** – this is to set/reset the demo.

![Real Application Testing window showing steps 1 to 5]

• Click on **Run in Terminal** button and wait until the script completes.

![Dialog box asking to run Step 3a]

Then proceed with the Lab exercise.
Demonstrating SPA using Enterprise Manager Grid Control Interface

1. Open a Firefox browser and navigate to Grid Control. Log in as sysman/oracle1

2. Click on Targets.
3. Click on **Databases**.

![Oracle Enterprise Manager 11g Grid Control UI](image)

4. Click on db02.oracle.com

Since the system has just been brought up and no workload running, the databases are mostly idle, the Load Map tells us the databases have low Average Active Session. Let’s use db02.oracle.com for these labs.

![Oracle Enterprise Manager 11g Load Map](image)
DB credentials: **system/oracle1**
Connect as Normal and Click on Login

5. At the home page, click on the **Software and Support** sub-tab
6. Click on the SQL Performance Analyzer link under Real Application Testing.

7. And here we are at the home page of SQL Performance Analyzer. For the purpose of this Lab exercise, we will go through the Guided Workflow, which is a super set of the above workflows and allows you to introduce change(s) of your choice. Click on Guided Workflow.
So, the first thing we will do here is **Create a SQL Performance Analyzer Task based on a SQL Tuning Set.**

Click on the **Execute icon** for Step 1.

**A SQL Tuning Set is a database object that includes one or more SQL statements and their execution statistics and execution context. You can use the set as an input source for various advisors, such as SQL Tuning Advisor, SQL Access Advisor, and SQL Performance Analyzer. You can load SQL statements into a SQL Tuning Set from different SQL sources, such as AWR, the cursor cache, or high-load SQL statements that you identified.**

A SQL Tuning Set includes the following:

- A set of SQL statements
- Associated execution context, such as user schema, application module name and action, list of bind values, and the cursor compilation environment
- Associated basic execution statistics
- Associated execution plans and row source statistics for each SQL statement (optional)

SQL statements can be filtered using the application module name and action, or any of the execution statistics. In addition, SQL statements can be ranked based on any combination of execution statistics.

SQL Tuning Sets are transportable across databases and can be exported from one system to another, allowing SQL workloads to be transferred between databases.

The SQL Tuning Set in this exercise has been prepared by collecting a sample from the 10g database cursor cache and transporting it into the 11g database. For detailed instructions please refer to the Performance Tuning Guide.

Click on the **Create** button after selecting the correct SQL Tuning Set.
10. Let’s run the 50 SQL in the SQL Tuning Set by clicking on the **Execute icon** in Step 2, **Create SQL Trial in Initial Environment**.
11. Enter a **SQL Trial Name** such as **WXA_SQL_TRIAL_PRE_CHG**. Check-off the “Trial environment established” check box and click on the **Submit** button.

12. Wait until a check mark appears in the **Status** column for Step 2. This means that SQL Performance Analyzer has completed the before change trial.
13. Minimize your browser and let’s deploy some changes into the database.

Go to the following folder from your desktop:

Demos ➔ Enterprise Management ➔ Real Application Testing ➔ RAT Step 3b – SPA Make Changes.sh, double click on this icon.

14. Click on Run in Terminal and wait until the Terminal closes. The changes have been made! It runs a script that does equivalent of applying a database patchset.

Let’s see how they affect our 50 very important SQL
15. Return to your browser window and click on the **Execute** icon of **Step 3, Create SQL Trial in Changed Environment**.

16. Provide a name for SQL Trial Name, such as **WXA_SQL_TRIAL_POST_CHG**, signifying my trial, after the changes have been implemented.

Check the box that the **“Trial environment established”** and click on the **Submit** button.
17. Wait until the check mark appears beside Step 3 in the Status column. If so, let’s move on to the next step, the comparison between Step 2 and Step 3. This will provide information regarding performance before and after the changes were made. Click on the Execute icon for Step 4, Compare Step 2 and Step 3.
18. For **Trial 1 Name**, select the name you gave to the trial before introducing the change—in this case, **WXA_SQL_TRIAL_PRE_CHG**.

For **Trial 2 Name**, select the name you gave to the trial after introducing the change—in this case, **WXA_SQL_TRIAL_POST_CHG**.

For the **Comparison Metric**, choose **Buffer Gets**, and then click on the **Submit** button. Wait until the check mark appears beside Step 4 in the **Status** column. **Why Buffer Gets is chosen?** For the purpose of this demo — repeatability, we choose this metric. Our recommendation in general is to use multiple metrics that encompass overall performance characteristics, Elapsed Time, CPU Time, Disk Reads, and Buffer Gets.

![Run SQL Trial Comparison](image)

19. Flawless, we have just analyzed in depth, how our key SQL are affected by the change.

Let’s review the results, click on the **Execute** icon for Step 5, **View Trial Comparison Report**.

![Guided Workflow](image)

20. Let’s look at the results highlighted.
For my 50 SQL statements, given the change, there was a 24% improvement. However, what we are concerned with is the -32% Regression Impact.

As you can view from the table, Top 10 SQL Statements Based on Impact on Workload, there are 3 SQL statements out of the 50 that performed worse with the introduction of the change. If we had deployed this change into production, without the use of SQL Performance Analyzer, these 3 SQL statements would have woken us up in the middle of the night.

Let us dive deeper into the Regressions. Click on the first SQL ID link with the red down arrow.
21. Here you can see all the metrics of the query and also see the query text itself.

Scroll down and see that SQL Performance Analyzer has provided a plan comparison of all queries in our Trial.

Click on the “SQL Performance Analyzer Task Report: …” breadcrumb at the header of the page
22. **Back to the SPA Task page**: With SQL Performance Analyzer we were able to identify from 50 queries, which ones were **Unchanged**, which ones **Improved** and most importantly, which ones **Regressed**

We also were able to dive deep into the individual queries and see how their performance was affected because of the change.

![SQL Performance Analyzer Task Report](image)

23. Now that we have found the queries that regressed, how do we fix them? This step shows you how to remediate regressed SQL statements through SQL Tuning Advisor. Click on the button, **Run SQL Tuning Advisor**

When we do so, SPA is smart enough to send only the SQL that have Regressed. In our case, the three SQL with the red down arrows…
24. Enter a name for the Tuning Task Name, such as WX_A_SPA_TUNE and click on the OK button.

25. Let’s go to our Advisor Central by clicking on the breadcrumb link and see the status of the tuning efforts of the Tuning Pack. When the Status is changed to COMPLETED, let’s review the details. Select the radio button to choose the Advisory Type of SQL Tuning Advisor and click on the View Result button.
26. Notice that SQL Tuning Advisor has identified three **SQL Profile(s)** that will fix the respective Regressions. Let’s implement these. Click on **Implement All Profiles** button.

Click on the **Yes** button to confirm. **Leave the check box alone in default state.** Forced matching means that similar literal SQL can also use the same SQL Profile (analogous to cursor sharing=force).
27. Then click on the **Advisor Central** breadcrumb link after the SQL Profile has been created successfully.

28. Great! The Regressions have been fixed. Let us just verify, remember we do not like surprises, right… so click on the **SQL Performance Analyzer** link and verify.

29. Click on our **SQL Performance Analyzer Task**.
Let’s create a new Trial as we have done above.

We have, in essence, made another change in our environment by implementing three SQL Profiles, recommended by the SQL Tuning Advisor.

Click on **Create SQL Trial** button…

Provide the Trial a meaningful Name—in this case **WXA_SQL_TRIAL_POST_TUNE**. Check the **Trial environment established** and Click on the **Submit** button.

Wait until the Status shows **COMPLETED** for the trial you just created. Hit refresh button to check for the new status. Click on Run SQL Trial Comparison button.
33. Here, our comparison is going to be between the trials after the change, where we identified the Regression to after the tuning cycle.

Trial 1 Name: WXA_SQL_TRIAL_POST_CHG
Trial 2 Name: WXA_SQL_TRIAL_POST_TUNE

Choose the same **Comparison Metric** of **Buffer Gets** for consistency. **Why Buffer Gets is chosen?** For the purpose of this demo – repeatability, we choose this metric. Our recommendation in general is to use multiple metrics that encompass overall performance characteristics, Elapsed Time, CPU Time, Disk Reads, and Buffer Gets.

Click on the **Submit** button

34. Notice that new line in the **SQL Trial Comparisons**. Click on the “goggles” for this trial under Comparison Report. Hit refresh Button.
If you do not see the “goggles” click on the Refresh button, top right of the page.

Voila…. Regression Impact = 0%
So, we can deploy the change to production, provided we also deploy the three SQL Profiles that SQL Tuning Advisor recommended—all with confidence.
Summary

In this lab exercise, you will accomplish the following:

1. Create SPA Task – container for holding our SPA experiment results
2. Create SQL Tuning Trials (experiments)
3. Identify regression of SQL with the introduction of change
4. Identifying and fixing SQL regressions
5. Validating Tuning thru another SPA trial and making sure no regressions exists before change is deployed in production

References: For more information, see:

2. OTN Page for Real Application Testing and Manageability
3. Real Application Testing Datasheet
LAB EXERCISE 02 – REAL APPLICATION TESTING – DATABASE REPLAY

Identified Challenge

Today, enterprises have to make sizeable investments in hardware and software to roll out infrastructure changes. For example, a data center may have an initiative to move databases to a low cost computing platform, such as Oracle Enterprise Linux. This would, traditionally, require the enterprise to invest in duplicate hardware for the entire application stack, including web server, application server and database, to test their production applications. Organizations therefore find it very expensive to evaluate and implement changes to their data center infrastructure. In spite of the extensive testing performed, unexpected problems are frequently encountered when a change is finally made in the production system. This is because test workloads are typically simulated and are not accurate or complete representations of true production workloads. Data center managers are therefore reluctant to adopt new technologies and adapt their businesses to the rapidly changing competitive pressures.

Introduction

System changes such as hardware/software upgrades, patch application, etc. are essential for business to maintain competitive edge or for compliance/security purposes. As a result, businesses conduct extensive testing and validation to assess the impact of a change before going live. Despite such testing using various simulation tools, many issues often go undetected until production deployment and negatively impact system performance and availability. The main reason for low success rate of testing is the inability of existing tools to test using real production workloads. Similarly, changes in query execution environment can often impact system performance and availability, therefore, the ability to do fine-grain SQL response time assessment and fix any regression due to the change are important to the smooth functioning of any business.

Oracle Real Application Testing option includes two solutions to test the effect of system changes on real-world applications, Database Replay and SQL Performance Analyzer. Database Replay enables you to effectively test system changes in test environments by replaying a full production workload on the test system to help determine the overall impact of the change. The SQL Performance Analyzer enables you to assess the impact of system changes on SQL performance by identifying any variation in SQL execution plans and performance statistics resulting from the change.

Database Replay

Database Replay enables realistic testing of system changes by essentially recreating the production environment on a test system. It does this by capturing a workload on the production system with negligible performance overhead and replaying it on a test system with the exact timing, concurrency, and transaction characteristics of the original workload. This makes possible complete assessment of the impact of the change including undesired results; new contention points or performance regressions. Extensive analysis and reporting is provided to help identify potential problems, such as new errors.
encountered and performance divergences. The ability to accurately capture the production workload results in significant cost and time savings since it completely eliminates the need to develop simulated/synthetic workload or scripts. As a result, realistic testing of even complex applications using load simulation tools/scripts that previously took several months now can be accomplished at most in a few days with Database Replay and with minimal effort. Thus using Database Replay, businesses can incur much lower costs and yet have a high degree of confidence in the overall success of the system change and significantly reduce production deployment risk.

Database Replay workload capture of external clients is performed at the database server level. Therefore, Database replay can be used to access the impact of any system changes below the database tier level such as:

1. Database upgrades, patches, parameter, schema changes, etc.
2. Configuration changes such as conversion from a single instance to RAC, ASM, etc.
3. Storage, network, interconnect changes
4. Operating system, hardware migrations, patches, upgrades, parameter changes

The Database Replay process can be broken down to 4 main steps:

1. **Workload Capture** – When workload capture is enabled, all external client requests directed to the Oracle Database are tracked and stored in binary files, called capture files, on the file system. These files contain all relevant information about the call needed for replay such as SQL text, bind variables, wall clock time, SCN, etc.

2. **Workload Processing** – Once the workload has been captured, the information in the capture files has to be processed, preferably on the test system. This processing transforms the captured data and creates all necessary metadata needed for replaying the workload.

3. **Workload Replay** – Once captured workload has been processed, it is now ready for replay on the test system. The test system has the system change applied and database restored to the point in time before the capture started using Oracle Database 11g Snapshot Standby or other equivalent mechanism. The replay can be configured appropriately to re-map connection strings, database links and directory objects to that of the test system. Once replay is initiated, a special client program called the “replay client” replays the workload from the processed files. It submits calls to the database with the exact same timing and concurrency as in the capture system and puts the exact same load on the system as seen in the production environment. This allows the identification of any instability caused by the change and their subsequent remediation in test environment before the introduction of the change in production.

4. **Analysis and Reporting** – Extensive reports are provided to enable detailed analysis of the capture and replay. Any errors encountered or divergences in data returned are reported. Basic performance comparison between capture and replay is provided, and for advanced analysis, AWR, ASH, and Compare Period reports are available.

A. **Overview**

In this lab exercise, you will accomplish the following:

1. Capture an Oracle 10g R2 Database Workload: **In interest of time, we will use a pre-captured workload. SKIP these steps [1 thru 32 below], but you can come back to this part if you completed rest of the lab. You can read thru the lab instructions for workload capture to understand the workflow.**

2. Replay the Oracle 10G R2 Database Workload on an 11G Database, use “SAMPLE_REPLAY1” directory object where the workload that was captured has been moved…
3. Generate AWR Comparison Report for Advanced Analysis

B. Setup & Preparation
   1. Already done by Infrastructure team, students need not worry about this step!

C. Demonstrating Enterprise Manager – Real Application Testing - Database Replay
   This lab sets the stage of upgrading an Oracle 10g database to an Oracle 11g database. We will capture a workload on our 10g database and seamlessly replay that in our 11g database all using Oracle Enterprise Manager 10 R5.

   So let’s get started…

   Since we will be capturing on a 10G database and Real Application Testing is an 11G feature that has been back ported, we need to, for 10.2.0.4 database, enable a parameter by calling a script. Follow the steps below…

   1. [Students - Skip performing this, but just read through it…] Once db03.oracle.com is up
      i) Open a terminal window
      ii) Source db03 environment by executing `db03` at the command prompt: $ db03
      iii) Start a SQL*Plus session and log in to db03 as sysdba
      iv) Execute the following command in SQL*Plus
          (1) `SQL>@/rdbms/admin/wrrenbl.sql`
          (2) At the SQL prompt, execute the below and verify that the parameter shows a boolean True values
              (a) `SQL> show parameter pre_11g_enable_capture`
      v) Proceed to the next step. We are ready to capture…
2. [Students - Skip performing this, but just read through it…] Open up a browser window and navigate to Oracle Enterprise Manager (OEM) Grid Control by selecting the icon on the tool bar. Log in as `sysman/oracle1`

![Oracle Enterprise Manager](image)

3. [Students - Skip performing this, but just read through it…] We will be capturing workload from our 10G database, so let’s navigate there, Click on Targets tab.

![Oracle Enterprise Manager - Targets Tab](image)
4. [Students - Skip performing this, but just read through it…] Click the Database sub-tab and then click on db03.oracle.com

DB credentials: system/oracle1
Connect as Normal and Click on Login
5. **[Students - Skip performing this, but just read through it…]** Once at the 10G home page, select the **Software and Support** sub-tab.

6. **[Students - Skip this, but just read through it…]** Click on **Database Replay** in the **Real Application Testing** section.

7. **[Students - Skip performing this, but just read through it…]** You will now arrive at the home page of **Database Replay**. An intuitive workflow with the ability to capture workloads, provision test databases and replay all from within Oracle Enterprise Manager. So let’s proceed. Click on the task for **Capture Workload** in “Capture Production Workload”
8. [Students - Skip performing this, but just read through it…] Review the Prerequisites and Acknowledge by checking the check boxes in the Prerequisite section and click on the Next button. We have already taken care of this in your environment.

9. [Students - Skip performing this, but just read through it…] On the Capture Workload: Options page, leave everything default on this page and click on the Next button. Just bring your attention to the radio-button “Do not restart the database prior to capture”; this should be checked but just make sure.
10. [Students - Skip performing this, but just read through it...] On the Capture Workload: Parameters page and under the section Workload Capture Parameters, choose the option in the drop down list to select the RAT_CAP10G Directory Object and click on the Next button.

This has already been created and points to the directory /u01/oracle/rat/capture, which is empty right now, but will soon be filled with many capture files.

11. [Students - Skip performing this, but just read through it...] On the Capture Workload: Schedule screen, leave everything default—but carefully review the Capture Duration section. We can schedule a start of capture and auto-stop based on duration. This is a great feature if you wanted to capture a batch load that runs in the middle of the night and did not want to stay up for!

Scroll down and enter the Host Credentials using the oracle user. and click on the Next button.
12. [Students - Skip performing this, but just read through it...] In the Capture Workload: Review screen, review your choices and click on the **Submit** button.
13. [Students - Skip performing this, but just read through it…] That’s it! You are now capturing on the 10G database. Because this is a training database, let’s start giving it work to do. When the below screens returns jump to the next step….

![Database capture screen](image)

14. [Students - Skip performing this, but just read through it…] Go to the folder RAT Step 2 – Lab Exercise 01 DB Replay navigating from the Labs folder on your desktop Labs ➔ EM - Real Application Testing ➔ RAT Step 2 – Lab Exercise 01 DB Replay

15. [Students - Skip performing this, but just read through it…] Double-click on the icon, Run_Load.sh

![Icon with Run_Load.sh highlighted](image)

16. [Students - Skip performing this, but just read through it…] Click on Run in Terminal. Wait until the script completes running

![Terminal window with Run_Load.sh option](image)
17. [Students - Skip performing this, but just read through it…] Switch back over to your Oracle Enterprise Manager Grid Control screen and notice how the capture processes has started capturing the activity. You may have to select the **Refresh** button. Now when the script in the above step completes, it will auto-close. When it does move to the next step…
18. [Students - Skip performing this, but just read through it…] The load script has completed so no more activity. Let’s stop capture by clicking on the Stop Capture button.

19. [Students - Skip performing this, but just read through it…] On the Confirmation screen, click on Yes

Processing….
20. [Students - Skip performing this, but just read through it…] Whenever a capture is stopped, Real Application Testing option takes an AWR snap for the duration of the capture. Let’s go ahead and export the AWR data by selecting the Yes button. We will review this data at the end of this lab.

21. [Students - Skip performing this, but just read through it…] Right-click the View Job and select the Open in New Tab option in the browser menu.
22. **[Students - Skip performing this, but just read through it…]** There’s our AWR Export Job. We should wait until this job completes. Click on Refresh.

23. **[Students - Skip performing this, but just read through it…]** When complete, you will be able to see the job in the **History** sub-tab with a tick beside it. Close this tab.

24. **[Students - Skip performing this, but just read through it…]** We are done on the 10G database.

We have captured a workload and exported the statistics of the database during the capture period….

*Let’s replay this 10G workload on our 11G database….*
25. [Students - Skip performing this, but just read through it…] Click on Targets tab and then the Database sub tab. Click on db02.oracle.com link to navigate to the home page of this database.

![Database screenshot]

DB credentials: system/oracle1
Connect as Normal and Click on Login

![Database login screenshot]

26. [Students - Skip performing this, but just read through it…] An Oracle 11.1.0.7.0 database has been already setup for you to run your replay on. Click on Software and Support sub-tab
27. [Students - Skip performing this, but just read through it…] Scroll down to the Real Application Testing section and click on Database Replay link. Just like we did when on the 10G database to capture.
28. [Students - Skip performing this, but just read through it...] Excellent! Here is the Database Replay page that we are already familiar with. If we did not have a test database to replay our capture, we would very easily and conveniently select the “Prepare Test Database” link and continued with the logical workflow. However, to save some time, we already have a Test Database setup. Proceed to the Prepare for Replay section and click on the Preprocess Workload task in the Go to Task column.

29. [Students - Skip performing this, but just read through it...] We used a database link on the 10G database that pointed to the capture directory `/u01/oracle/rat/capture`. We have created a database link on our 11G database that points to the replay directory `/u01/oracle/rat/replay` Select the radio-button to “Copy the workload directory...” and click on Next button.
30. [Students - Skip performing this, but just read through it...] In the Copy Workload Step page, select the flash light icon under Target Name to select the database that we just captured on.

31. [Students - Skip performing this, but just read through it...] Click on the Go button and you should get the list of available databases. Since we captured on db03.oracle.com (10g), select the db03.oracle.com database and click on the Select button...
32. [Students - Skip performing this, but just read through it…] In the Target Name, click the flashlight and select the db03.oracle.com database. This is our 10G database that we captured the workload from.

As soon as we select it, and return to the below screen, all the captures performed on that database appears below. Since we only have one, scroll down and enter the information as shown below:

Current Location of the Workload Directory
- Host → dbsecurity.oracle.com
- Directory → /u01/oracle/rat/capture
- Host User Name → oracle
- Host Password → g0Oracle12#

New Location of the Workload Directory
- Host → dbsecurity.oracle.com
- Directory → /u01/oracle/rat/replay
- Host User Name → oracle
- Host Password → g0Oracle12#

Click the Next button when the above information is filled out.
33. Using EM Grid Control, go to db02 (11.1.0.7) database, Software and Support Tab → Database Replay link → Database Replay Home Page → Expand “Prepare for Replay Step”. Click on go to task “Preprocess workload”: Select Directory screen. Select the SAMPLE_REPLAY1 Directory Object from the drop down.

34. Select “Use an existing Workload directory on this host”  
   Click “Next”
As soon as you select the directory object, database replay will read the directory and pull the information of the capture that we just copied over from the 10G database.

Click on the Next button

35. Enter in the Host Credentials using the oracle/g00Oracle12# user &password and click on the Next button.
36. Review your choices and click on the Submit button.

37. Now, wait until the preprocessing of the capture file is complete. How do we know that it is complete? Follow the next steps to find out.
38. To get details of the job, right click the **View Job** link and in the browser menu choose **Open Link in New Tab**

39. There is our Preprocess Job; Status should read **Succeeded**. If yes, go ahead and close this tab and return to your original OEM tab.
40. So let’s start replaying!

Click on the Go to Task icon for “Replay Workload” under “Replay Workload on Test Database”

41. Since we have already copied over the capture files and preprocessed them, select the radio button for “Use existing workload directory on this host” and click on the Next button.
42. Select the SAMPLE_REPLAY1 from the Directory Object and click on the Next button.

![Select Directory Image]

43. Click on the Next button.

![Next Button Image]
44. Choose Use the default replay options. Click on the Next button.

45. Leave the default Connection Mapping and click on the Replay Parameters sub-tab.
46. We will leave the default Replay Parameters and click on the Next button. But please feel free to see what wonderful things they do!

47. Now we need to start the Replay Clients. These are the multi-threaded programs that replay the workload into our test database hence eliminating the need to setup the application tier. We just need a test database to replay on!

First Click on the “Estimate” button to estimate how many wrc replay clients are required to drive the workload. For larger workloads, you may need multiple wrc replay clients. Estimate uses the calibration mode of wrc to give a conservative number of clients needed of this workload based on the CPUs and concurrency noted in the workload capture files.

Click on Add Replay Client Hosts button in the Replay Workload, Add as many clients are recommended by Estimate (in this case 1 wrc client) Prepare Replay Clients screen.
48. Uncheck the check box “Include only targets with detected Replay Client installation” and click on the Go button.

We should see our Target in the table below.

Check the check-box for dbsecurity.oracle.com and click on the Select button.

![Image of target selection](image)

49. Now that we have selected the host that we are going to run the replay clients from, we will configure them.

So either click on the Configure button or the “No” link to proceed.

![Image of configuration](image)
50. Make sure your pop-up blocker is turned off. Enter some information (most of the fields will already be entered so just confirm that they are as below).

| Host User Name               | oracle          |
| Host Password                | g0Oracle12#     |
| Database User Name           | system          |
| Database Password            | oracle1         |
| Server Connection Identifier | dbsecurity.oracle.com:1521/db02.oracle.com |
| Number of Replay Clients     | 1               |
| Client Oracle Home           | /u01/oracle/product/11.1.0/db_1 |
| Client Replay Directory      | /u01/oracle/rat/prev_replay/sample_replay1 |

Once entered, click on the Apply ** button first and then the Close button.

** Make sure you click on the Apply button, make sure to type this in correctly, otherwise you’ll have to cancel the replay and start from the first step of replay!
51. Perfect! The Replay client is all configured, click on the Next button

Wait… While we start the replay client that we configured in the previous step…

52. There it is, one client configured and one client connected. Click the Next button.
53. Review and click on the **Submit** button

54. **Processing…**

55. **Congratulation! The Replay has started!** Go ahead and change the refresh rate to something smaller than 60 seconds if you like to watch the replay progress…
56. Replay done. A 10G database workload, replayed on the 11G database in just over half the time here!

Time to Upgrade!

Click on the Reports sub-tab, to look at some reports.…

57. Click on “Run Replay Compare Period Report” for performing replay analysis for the system change in question (database upgrade).

Since this was a small duration test run, we cannot perform meaningful replay analysis, in fact the DB Time will be less than < 5min for Capture/Replay and the Replay Compare Period Report will not be generated in this case. So skip following steps and proceed to step 64 to look at some real-world reports from customers.

The steps are documented here to help you understand the work flow.

Click on Run Replay Compare Period Report button…
58. [Students - Skip performing this, but just read through it…] Remember that we exported the AWR report when our Capture completed on the 10G database. Let us now proceed to import it. Click on the Yes button to continue.

59. [Students - Skip performing this, but just read through it…] Click on the Import AWR Data button.

60. [Students - Skip performing this, but just read through it…] Right click on the View Jobs link and select the Open in a new tab from the browser menu.
61. [Students - Skip performing this, but just read through it...] Wait until the Import completes successfully. Once complete, close the browser tab and return to your original Oracle Enterprise Manager tab. Wait for 2 minutes

62. [Students - Skip performing this, but just read through it...] Go to Database Replay → Analyze Results → Select SAMPLE_REPLAY1 dir object (This step is required to avoid a browser issue for the lab, the issue is addressed in next release, see Step 63 also) Now that the import has completed successfully, let click on the Run Replay Compare Period Report...
63. [Students - Skip performing this, but just read through it…] Since DB Time was less than < 5min for Capture/Replay, the Replay Compare Period Report was not generated, this was left this way due to 45 min time limit on the lab, however, here is an actual report for illustrative purposes. Click on OK to continue.

![Compare Period Report](image)

64. Go to the Linux Desktop, double click on “demos”

![Linux Desktop](image)

65. Open “Enterprise Management” folder by double-clicking the icon.

![Enterprise Management](image)
66. Open “**Real Application Testing**” folder by double-clicking the icon

![Real Application Testing folder](image)

67. Double click on one of following reports (cp1.html, cp2.html or telco_discsubsystemchange.html)

![Real Application Testing report](image)

68. I did not print out the entire report to conserve paper so just tore out the first page.

Go ahead and scroll down and read until you heart is merry. With this report you can do incredible diagnostics….
Compare Period Report: Capture vs. Replay

General Information

This report compares the performance of a workload replay against the performance of the original captured system. Throughout the report, "Capture" refers to the original captured system, while "Replay" refers to the replayed workload. The most reliable experiment will compare two replays. The first replay will try to mimic the captured system as much as possible without any system changes. The second replay would be similar in that it will apply the same change as the first variable. (Test systems are almost always an approximation of production.) The idea is comparing two replays is to isolate the change we want to apply and thus assess the effect of such a change on a system similar to production.

Data Sources

This section describes the experimental setup. Check it carefully to verify that the intended experiment was performed.

Information About Databases

<table>
<thead>
<tr>
<th>Capture</th>
<th>Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>DNAME</td>
</tr>
<tr>
<td>Original Database ID</td>
<td>660578865</td>
</tr>
<tr>
<td>Version</td>
<td>11.1.0.7.0</td>
</tr>
<tr>
<td>Stats</td>
<td></td>
</tr>
</tbody>
</table>
D. Additional Steps
   1. Generate other reports, ASH Report for example, for Replay Compare Period Report, note that you need to have at least 5 MIN of DB Time captured in the capture and replay periods, otherwise due to the small duration of db activity, the analysis might be meaningless.
   2. Modify an Initialization Parameter and run through the replay again
   3. If time permits, do a new capture on 10.2.0.4 and transfer the workload and repeat the steps

E. Summary
   You accomplished the following in this lab exercise:
   1. Captured workload from a 10G database (Optional due to time constraints)
   2. Replayed on an 11G database
   3. Created an AWR comparison report for advanced analysis

F. For additional information, visit

Oracle Real Application Testing
http://www.oracle.com/database/real-application-testing.html?SC=NAMK08051382MPP001.GCM.8328.110.real+application+testing.br

Oracle Enterprise Manager
http://www.oracle.com/enterprise_manager/index.html