Oracle Enterprise Manager

Oracle Database Performance Diagnostics and Tuning Lab

Session S318965
Oracle Enterprise Manager 11g
Oracle Database Diagnostics and Tuning
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

Get Maximum Performance With ROI of 100%

Oracle provides an integrated management solution for managing Oracle database with a unique top-down application management approach. With new self-managing capabilities, Oracle eliminates time-consuming, error-prone administrative tasks, so database administrators can focus on strategic business objectives instead of performance and availability fire drills.

Oracle Management Packs for Oracle Database provide significant cost and time-saving capabilities for managing Oracle Databases. Independent studies demonstrate that Oracle Database is 40 percent easier to manage over DB2 and 38 percent over SQL Server.

Diagnostics Pack

Oracle Diagnostics Pack, a part of the Oracle Database 11g product set, offers a comprehensive set of automatic performance diagnostics and monitoring functionality built into core database engine and Oracle Enterprise Manager. Whether you are managing one or more databases, Oracle Diagnostics Pack offers a complete, cost effective, and easy to use solution for managing the performance your Oracle Database environment. When used as part of Enterprise Manager Grid Control, Diagnostics Pack additionally provides enterprise-wide performance and available reporting, a centralized performance repository, and valuable cross-system performance aggregation, significantly simplifying the task of managing large sets of databases.
Tuning Pack

Oracle Tuning Pack, a part of Oracle Database 11g product set, offers an extremely cost effective and easy-to-use solution that automates the entire application tuning process. Enhancement of SQL performance is achieved through SQL Advisors that are seamlessly integrated with the Enterprise Manager Database Control and Grid Control, and together provide a comprehensive solution for automating the complex and time-consuming task of application tuning.

This lab will demonstrate:
- Performance Diagnostics
- Application (SQL) Tuning

for Oracle Database 11g.

Please feel free to seek assistance from the instructor or Oracle Demo staff at any point in time.

Before we start taking you through the demonstration, please note the following:
- 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: __________________________________________________________

- You will connect to that system using a simple browser connection using Firefox. The username and password are:

- You will connect to that system using VNC. VNC password is g0Oracle12#
- Operating System Accounts: oracle/g0Oracle12# and root/g0Oracle12#
- Database(db02) Accounts: system/oracle1
- Grid Control Accounts: sysman/oracle1
LAB CONFIGURATION – ENTERPRISE MANAGER – TUNING AND DIAGNOSTICS

OVERVIEW

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

- Database: DB02
- Database: EMREP
- Enterprise Manager Grid Control

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

1. On the desktop, navigate to the Oracle_Open_world_2010 folder, double-click and open the contents.
2. Select the Enterprise_Management Folder
3. Select the folder **Tuning & Diagnostics** folder.

4. Next, we'll start our workload running. First, click on the icon 'Step1a – Start Access Load.sh.' You can move on to the next step while the terminal window displays, this script runs for a very long time in a loop. You may want to minimize the window for the script.
5. For the second part of the workload, click on the icon 'Step1b – Start Tuning Load.sh.' You can move on to the next step while the terminal window displays, this script runs for a very long time in a loop. You may want to minimize the window in which the script is running.

6. At this point, all of the necessary infrastructure is ready for you to move forward with the EM – Diagnostics & Tuning labs. After completion of the labs, return to this folder and to stop the load on the database, reset the environment and stop the infrastructure. The complete instructions for doing so are at the end of the tuning lab.
Diagnostics Pack

Identified Challenge

Diagnosing a slowly performing system is a time consuming task often surrounded by myths and legends, few of them based on fact. A number of third party tuning tools are available today but few of them are geared towards answering common questions such as, ‘How can I make the biggest improvements in the system?’ or ‘Why is the system slower today than it was last week?’ Most products simply provide a graphical display of raw database statistics, leaving users to determine the root cause on their own by drilling through large amounts of raw data. Even with the best of tools, this can be a complex and tedious task.

Introduction

Automatic Performance Diagnostic

Diagnosing a slowly performing system is a time consuming task. Oracle Diagnostics Pack 11g includes a self-diagnostic engine built right into the Oracle Database 11g kernel, called the Automatic Database Diagnostic Monitor (ADDM). This is a revolutionary, first of its kind performance self-diagnostic solution that enables the Oracle Database 11g to automatically diagnose its performance problems, thereby completely liberating administrators from this complex and arduous task.

ADDM periodically samples the state of the database, automatically identifies potential database performance bottlenecks, and recommends corrective actions. This is all done within a few seconds with negligible impact on overall system performance. Oracle Enterprise Manager presents ADDM’s findings and recommendations in a convenient and intuitive fashion, and guides administrators step-by-step to quickly resolve performance problems by implementing ADDM’s recommendations.
A. Overview

In this lab exercise, you will accomplish the following:

1. Top-Down Automatic Diagnostic
2. Reviewed ADDM Findings
3. Reviewed Performance Analysis
4. Deep Dive Diagnostic into SQL Performance

B. Setup & Preparation

1. Open up a browser and Log into Grid Control using system as the username and password oracle1

![Login to Oracle Enterprise Manager](image)

Navigate to the database home page by clicking on Targets tab

![Oracle Enterprise Manager](image)

and then click on Databases.

If you see a Load Map view of the Oracle databases, select the Search List option to view the list of databases and then select db02.oracle.com from the list of databases.
2. With the Diagnostics Pack there are many new capabilities including drill-down capabilities from the Host CPU and Active Sessions graph.

Click on the ADDM Findings value under Diagnostic Summary in the Database home page. When prompted for the Database Login, use system/oracle1 and click on the Login button.
3. If you have ADDM findings showing skip to Step 5. If for some reason there were no ADDM Findings, click on the Performance tab.

4. Scroll down the page to the Average Active Sessions section. Click on the icon highlighted

5. You will be taken to the Automatic Database Diagnostic Monitor (ADDM) Advisor Central screen. On this screen, click on the icon. Click on the Icon Key to view the ADDM Task and Snapshot icons.

6. By clicking on the numeric link or through the alternative path, you can now drill down into ADDM Findings. The database, by default, starts taking scheduled snapshots of the database and stores them in a repository, Automatic Workload Repository (AWR).
**Automatic Workload Repository (AWR)** is a built-in repository in every Oracle Database. At regular intervals, the database makes a snapshot of all its vital statistics and workload information and stores them in AWR. The **Automatic Database Diagnostic Monitor (ADDM)** analyzes the AWR data on a regular basis, then locates the root causes of performance problems, provides recommendations for correcting any problems, and identifies non-problem areas of the system. Because AWR is a repository of historical performance data, ADDM can be used to analyze performance issues after the event, often saving time and resources reproducing a problem.

On this page you can review historical database activity. By default, the current ADDM performance analysis findings are presented. If you want to look at a different ADDM period, click on the appropriate icon below the chart. The Performance Analysis section displays the findings as well as the potential impact for resolving the identified issues.
7. Go ahead and click on one of the Finding description links, e.g. CPU Usage, in the ADDM Performance Analysis section to review details (shown below). Notice the Benefit (%) column if the action is taken to fix the issue. But don’t start fixing anything... I know you want to, but we will tune in the next lab!

![Performance Finding Details: CPU Usage](image)

An alternative way to diagnose performance issues is to review the database performance over time and drill into any abnormalities indicated on the performance charts.
8. Navigate back to the database homepage by clicking on the database sub-tab and then click on the Performance sub-tab link.

This page provides comprehensive information needed to investigate any performance issues, including correlation with host performance.
9. By default, the performance page is set to real-time with automatic refresh. This option is on the upper right corner of every page. To look back in time, change View Data drop down list to Historical.

10. After reviewing the available data, change the View Data selector back to the Real Time: 15 Seconds Refresh.
11. Click on CPU Used on the legend or on the performance graph itself in the Average Active Sessions sections.

12. On the Active Sessions Working chart (shown below), the shaded time period can be dragged to different time periods. Click and drag the shaded box on the Active Session chart to focus on a different period of time. Notice how the supporting details change.

This page shows the waits causing CPU issues as well as Top SQL and Top Sessions contributing to the CPU issues for that period.

13. Click on one of the identifiers located in the SQL ID column in the Top Working SQL section.
The SQL Statement is displayed in addition to all the associated activity linked to this SQL. This is how you can spot and drill down on offending SQL. Similarly to the previous performance charts, the shaded block can be dragged backward in time to review past activity.

Remember to drag the shaded block to the right to display the offending SQL producing the high load as a hyperlink.

14. Click on the Statistics tab in the Details section
Here you can review activity by time, by waits, Execution Statistics that tabulate critical information that would have been laborious to attain yet so critical.

15. Click on the Plan tab in the Details section

16. Under the Details section, choose the Table radio-button highlighted below as the view option.
17. Review the Plan of the SQL that is causing performance issues. Notice the significant amount of CPU being used to perform this SQL statement. In addition to the breakdown and details of the SQL itself, other important statistics, explain plans and other information is available for review and analysis.

**TIME CHECK:**

If you have less than 20 minutes left before the end of the lab, **SKIP** the optional SQL Monitoring task below and go on to the next section, Tuning Pack.
SQL Monitoring (Optional)

18. As an alternative method to view data being generated in real-time, navigate back and click on the Performance Tab.

19. Scroll down to the section for Additional Monitoring Links and click on the SQL Monitoring link.

You will be brought to the Monitored SQL Executions interactive screen. This screen provides real-time information on the queries that are currently being executed.

20. Right-click on the SQL ID to show the option available. You have already viewed some of the screens in this exercise for the options: SQL Text, SQL Details, and Session Details. This provides an additional alternative path to reach this information. Click on the Monitored SQL Execution Details from the menu.
21. Hover, highlight and view the links and information available on this screen to drill into.

C. Summary

We looked at how the Diagnostics Pack, with the used of Automatic Database Diagnostic Monitor (ADDM) automatically gets enabled when a database is created and started. It will perform top-down analysis of database activity using the automatic snapshots crated and stored in the Automatic Workload Repository.

ADDM identifies resource intensive operations and determines whether or not these components or operations are acting as performance bottlenecks. ADDM then diagnoses these potential problems and provides advice.

We also looked into the Performance Analysis and Top Activity feature and were able to go back in time to view what was occurring in the database. We then drilled down into individual SQL’s and looked at their performance metrics.

1. You accomplished the following in this lab exercise:
   i) Reviewed Top- Down Automatic Diagnostic
   ii) Reviewed ADDM Findings
   iii) Reviewed Performance Analysis
   iv) Reviewed Deep Dive Diagnostic into SQL Performance

2. Additional Information
   i) For more information, see:
Tuning Pack

<table>
<thead>
<tr>
<th>Identified Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>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>

Introduction

Oracle Tuning Pack, a part of Oracle Database 11g product set, offers an extremely cost effective and easy-to-use solution that automates the entire application tuning process. Enhancement of SQL performance is achieved through SQL Advisors that are seamlessly integrated with the Enterprise Manager Database Control and Grid Control, and together provide a comprehensive solution for automating the complex and time-consuming task of application tuning.

SQL Tuning Advisor

Manual SQL tuning is a complex process that presents many challenges. It requires expertise in several areas, is very time consuming, and requires an intimate knowledge of the schema structures and the data usage model of the application. All these factors make manual SQL tuning a challenging and resource intensive task that is ultimately very expensive for businesses.

SQL Tuning Advisor is Oracle’s answer to all the pitfalls and challenges of manual SQL tuning. It automates the SQL tuning process by comprehensively exploring all the possible ways of tuning a SQL statement. The analysis and tuning is performed by the database engine’s significantly enhanced query optimizer. Four types of analysis are performed by the SQL Tuning Advisor:

- **Statistics Analysis**: The query optimizer needs up-to-date object statistics to generate good execution plans. In this analysis objects with stale or missing statistics are identified and appropriate recommendations are made to remedy the problem.

- **SQL Profiling**: This feature, introduced in Oracle Database 10g, revolutionizes the approach to SQL tuning. Traditional SQL tuning involves manual manipulation of application code using optimizer hints. SQL Profiling eliminates the need for this manual process and tunes the SQL statements without requiring any change to the application code. This ability to tune SQL without changing the application code also helps solve the problem of tuning packaged applications. Packaged application users now no longer need to log a bug with the application vendor and wait for several weeks or months to obtain a code fix for tuning the statement. With SQL profiling the tuning process is automatic and immediate.

- **Access Path Analysis**: Indexes can tremendously enhance performance of a SQL statement by reducing the need for full table scans. Effective indexing is, therefore, a common tuning technique. In this analysis new indexes that can be significantly enhance query performance are identified and recommended.
• SQL Structure Analysis: Problems with the structure of SQL statements can lead to poor performance. These could be syntactic, semantic, or design problems with the statement. In this analysis relevant suggestions are made to the restructure selected SQL statements for improved performance.

The output of this analysis is in the form of recommendations, along with a rationale for each recommendation and its expected performance benefit. The recommendation relates to collection of statistics on objects, creation of new indexes, restructuring of the SQL statement, or creation of a SQL Profile. A user can choose to accept the recommendation to complete the tuning of the SQL statements.

The SQL Tuning Advisor offers a powerful, intuitive, and user-friendly way for performing SQL tuning. Tuning of SQL statements no longer has to be the domain of experts. Oracle has built a tuning expert inside the database engine to perform this very important function for the database administrators in a fraction of the time and cost needed to carry out the same task manually.

SQL Access Advisor

The design of the database schema can have a big impact on the overall application performance. SQL Access Advisor, provides comprehensive advice on how to optimize schema design in order to maximize application performance. SQL Access and SQL Tuning Advisors, together, provide a complete solution for tuning database applications. These two advisors automate all manual-tuning techniques currently practiced and form the core of Oracle’s automatic SQL tuning solution.

The SQL Access Advisor accepts input from all possible sources of interest, such as the cursor cache, the Automatic Workload Repository (AWR) any user-defined workload, and will even generate a hypothetical workload if a schema contains dimensions or primary/foreign key relationships. It comprehensively analyzes the entire workload and provides recommendations to create new partitions or indexes if required, drop any unused indexes, create new materialized views and materialized view logs. Determining the optimal partitioning or indexing strategy for a particular workload is a complicated process that requires expertise and time. SQL Access Advisor considers the cost of insert/update/delete operations in addition to the queries on the workload and makes appropriate recommendations, accompanied by a quantifiable measure of expected performance gain as well as scripts needed to implement the recommendations.

The SQL Access Advisor takes the mystery out of access design process. It tells the user exactly what the type of indexes, partitions, and materialized views are required to maximize application performance. By automating this very critical function, SQL Access Advisor obviates the need for the error-prone, lengthy, and expensive manual tuning process. It is fast, precise, easy-to-use and, together with the SQL Tuning Advisor, offers the most accurate and cost-effective solution for application performance tuning.
A. Overview

In this lab exercise, you will accomplish the following:

1. Created SQL Tuning Sets
2. Used SQL Tuning Advisor to recommend improvements on SQL’s
3. Used SQL Access Advisor to recommend improvements on schema structures

1. Open up a browser and Log into Grid Control using system as the username and password oracle1

2. If you are continuing on from the Diagnostics Lab, we have seen how quickly we are able to triage the performance bottleneck by the one-of-its kind, Automatic Database Diagnostic Monitoring. We were also able to look deep into the culprit using the Performance Analysis page and look at key performance metrics.

   So let us continue and fix the issue.

3. Navigate to your database db02.oracle.com home page and click on the numeric count link (in this case 2) of ADDM Findings.
4. Click on SQL Statements consuming significant database time were found.
5. We see below that ADDM has already identified the SQL’s that are causing the performance bottleneck and from this screen we can directly navigate to the features of the Tuning Pack here, SQL Tuning Advisor.

However, wait, we will do one better, we will use SQL Tuning Sets. A container that allows us to accumulate multiple DML’s with context directly through cursor cache. No need to SQL Trace, let’s tune!

![Performance Finding Details](image)

**Recommendations**

- **Action**: Investigate the SQL statement with SQL ID "387Igqspwth" for possible performance improvements.
- **SQL Text**: `SELECT * FROM sh1.sales s WHERE s.quantity = 2 AND s.price`.

**Reason**:

- SQL statement with SQL ID "387Igqspwth" was executed 1373 times and took an average elapsed time of 0.32 seconds.
- Average CPU used per execution was 0.3 seconds.
6. So Tune it is, navigate to the Performance sub-tab. Scroll to the bottom and select Top Activity.
7. In the Top SQL section, change the Actions drop down to Create SQL Tuning Set. Click the checkbox next to the SQL ID’s (two or three of them) and click on the Go button.

Note: Before creating a SQL tuning set, you may need to move the historical slide (grey area) back to a time when the workload is peak. If the current time happens to be in a "valley", you may only get one SQL statement in the Tuning set.

8. Give a meaningful, (and unique if you are running through this lab again), Name to provide easy identification and click the OK button.

A SQL Tuning Set (STS) is a database object that includes one or more SQL statements along with their execution statistics and execution context, and could include a user priority ranking. The SQL statements can be loaded into a SQL Tuning Set from different SQL sources, such as the Automatic Workload Repository, the cursor cache, or custom SQL provided by the user. An STS includes:

- 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, such as elapsed time, CPU time, buffer gets, disk reads, rows processed, cursor fetches, the number of executions, the number of complete executions, optimizer cost, and the command type

SQL statements can be filtered using the application module name and action, or any of the execution statistics. In addition, the SQL statements can be ranked based on any combination of execution statistics.
9. You should arrive in the SQL Tuning Sets page when step 8 completes. Select the SQL Tuning Set you just created and click on the Schedule SQL Tuning Advisor tab.

10. Make sure the SQL Tuning Set is entered, if not, select the flashlight icon and select your tuning set. Also, if you decide to change the Name, make sure it is unique. Leave everything else default and select the Submit button.
11. View the recommendations on the SQL (select /*+ ORDERED USE_NL(c) FULL (c)*/count(*) from sh.sales s, sh.customers c where c.cust_id = s...*) by selecting the appropriate radio button and clicking on the View button.

12. Notice that there are recommendations for collecting statistics and implementing a SQL Profile. Click on the Compare Explain Plans, the before and after plans of the SQL if we implemented the SQL Profile. Let’s implement this recommendation, one by one selecting the radio button and selecting the Implement button on the Confirmation screen.

(Depending on how long you have been running the load, you may not see the Statistics. You may only get one row for SQL Profile).
and for the second...

13. Once the SQL Profile has been implemented, click on the Advisor Central breadcrumb to return to the Advisor Central home page.
14. Click on the SQL Advisor link in the Advisor Central

TIME CHECK:
If you have less than 5 minutes left before the end of the lab, SKIP the optional SQL Access Advisor section below and go to the Summary section in the last page.
15. We looked first at the SQL Tuning Advisor; we will now work with SQL Access Advisor, another feature of Tuning Pack. Click on the SQL Access Advisor.
16. Make sure Recommend new access structures is selected and click the Continue button

![Image of SQL Access Advisor: Initial Options]

17. Select the Use an existing SQL Tuning Set radio button and click on the Flashlight to select the SQL Tuning Set you created earlier. Leave the Filter setting to defaults and click the Next button

![Image of SQL Access Advisor: Workload Source]
18. For the purpose of this lab, let us just select the Indexes check box, knowing that we could also get SQL Access Advisor to recommend on Materialized Views and Partitioning.

19. Select the Comprehensive radio button then select the Next button. Leave the Advanced button to defaults.
20. Time to schedule the job. Leaving all to defaults will run the SQL Access Advisor immediately, so let do that. Click the Next button.

![SQL Access Advisor: Schedule](image)

21. Click on Show All Options button and have a look. Ignore the No Filter Options. Click the Submit button.

![SQL Access Advisor: Review](image)
22. Let’s have a look at what SQL Access Advisor has recommended. Click on View Results button once the Status shows completed. Make sure you have the radio button selected for your Name column.
23. Wow, look at the cost savings and the Query Execution Time Improvement. Click on the Recommendations sub-tab.
24. There are two Recommendations for Implementation at the bottom of the page identified by the ID column. Let us take a look at ID 1 first. Click on the link associated with ID 1.

<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>Cost Improvement</th>
<th>Cost Improvement (%)</th>
<th>Estimated Space Used (MB)</th>
<th>Affected SQL Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Insert</td>
<td>3185984</td>
<td>0.99</td>
<td>2.122</td>
<td>1</td>
</tr>
</tbody>
</table>

25. Review the Recommendation of creating an index in the SH schema that will help the SQL identified below. Implementing the Recommendation will result in an improvement.... How about the second recommendation?
26. Go back (just click the back browser button) and select ID 2

![Diagram showing SQL Access Advisor recommendations]

27. Review Recommendation: 2. Another index creation in SH1 schema with a cost improvement of about 18%
28. So let go back to the Recommendation page and implement the two index creations by selecting the check boxes and clicking on the Schedule Implementation button.

29. You should see the Schedule Implementation page. Make sure the Stop on Error is checked and click on Submit button.
30. Excellent job!

31. Let’s see how the Job is progressing as we left the default start time as Immediate while scheduling. Go to the database Server sub-tab and click on the Jobs link. You may need to click on the Refresh button in the Scheduler Jobs screen.

32. SUCCEEDED…. And that’s it…
We went through two key features of the Tuning Pack, SQL Tuning Advisor and SQL Access Advisor after identifying the queries that were the majority cause of CPU bottleneck on the host. We encapsulated with context, these queries into unique SQL Tuning Set (STS).

Running the SQL Tuning Advisor, on the reference STS, we discovered that there were stale statistics, which we fixed by gathering statistics on the necessary objects. Then we discovered a sub-optimum plan of a SQL, which we fixed by implementing a SQL Profile.

We then ran SQL Access Advisor on the reference STS, which recommended implementing two indexes in two schema, which we implemented, and voila, improvements already… New sessions will now take advantage of our tuning efforts.

B. Summary

1. You accomplished the following in this lab exercise:
   i) Created SQL Tuning Sets
   ii) Used SQL Tuning Advisor to recommend improvements on SQL’s
   iii) Used SQL Access Advisor to recommend improvements on the Schema

2. Additional Information
   i) For more information, see:

This concludes the Oracle Enterprise Manager Hands-on Lab.

Additional information can be found at:
Diagnostics and Tuning Packs demo booths located at Oracle Moscone West
For additional information, visit:

Oracle Database Manageability:

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