

*Note: This document is for informational purposes. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.*

# ORACLE REAL APPLICATION TESTING

## REAL APPLICATION TESTING

### KEY FEATURES AND BENEFITS

- 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

*All businesses must deal with IT infrastructure changes. Even though these changes are designed to have a positive impact on the infrastructure, they can, in many instances, lead to unexpected and undesirable behavior. As a result, businesses spend significant amount of time and effort evaluating system changes in test environments to assess their full impact before introducing them in their production systems. However, despite extensive and often expensive testing, many problems are only detected on production deployment resulting in businesses not meeting SLAs, unplanned downtime, and loss of revenue. Oracle Real Application Testing offers extremely cost-effective and easy-to-use solution that enables you to fully assess outcome of a change in a test environment, take any corrective action if necessary, and then to introduce the change safely to production systems, minimizing undesirable impact on them. This allows businesses to continue to benefit from changes but without any associated negative impacts such as performance degradation, outages, etc. Real Application Testing offers two key unique features, Database Replay and SQL Performance Analyzer, that together provide a comprehensive and flexible solution for assessing impact of changes to production system.*

### Real Application Testing

System changes such as hardware/software upgrades, patch application, etc. are essential for businesses 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 executions plans and performance statistics resulting from the change.

### **Database Replay**

Database Replay enables realistic testing of system changes by essentially recreating 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 contentions points or performance regressions. Extensive analysis and reporting is provided to help identify any potential problems, such as new errors encountered and performance divergences. The ability to accurately capture the production workload results in significant cost and timesavings since it completely eliminates the need to develop simulation workloads 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.

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

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

System changes above the database server level such as at the middle-tier, client, and application layers are not supported with Database Replay. Figure 1 illustrates Database Replay overview. As shown in this figure, the replay driver that drives the replay clients is client agnostic and therefore eliminates the need to setup extensive middle tier/client environment to replay the workload. This results in significant savings in time and effort for the businesses.

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

#### **i. 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 values, wall clock time, SCN, etc.

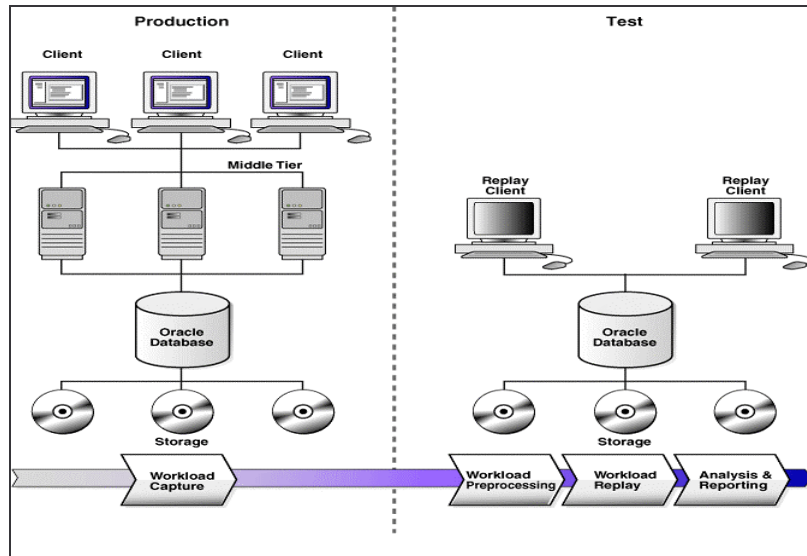


Figure 1: Database Replay Overview

The workload capture start and end time is specified by the user and during this time all information pertaining to external database calls is written to the capture files. The workload capture process has been highly optimized to make sure it incurs negligible overhead even on a busy system. The workload that has been captured on Oracle Database release 10.2.0.4 and higher can also be replayed on Oracle Database 11g release.

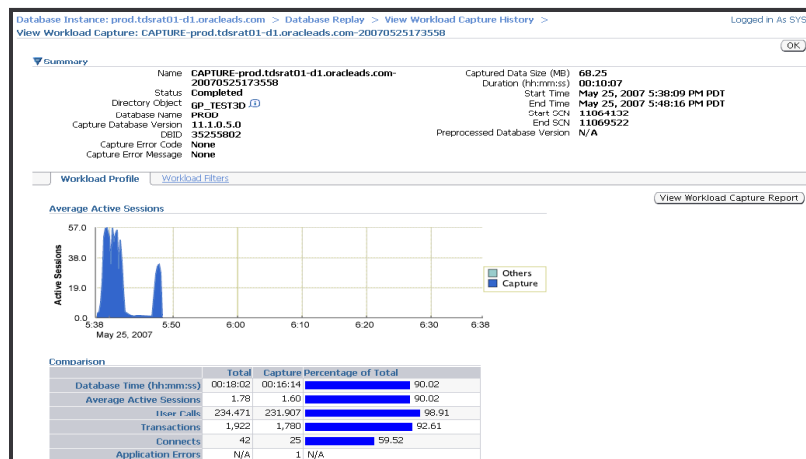


Figure 2: Workload Capture Summary

## ii. 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.

### iii. 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.

The replay driver besides being client-agnostic also uses a scaleable multi-threaded architecture, including using multiple host machines if necessary to drive large workload captures. There are various options that are available to control the behavior of the replay such as to scale up or down the think and login times, and maintain commit synchronization. These options can be used for load/stress testing the system. The replay drivers also can automatically re-map physical locators and preserve sequence numbers or GUIDS during replay.

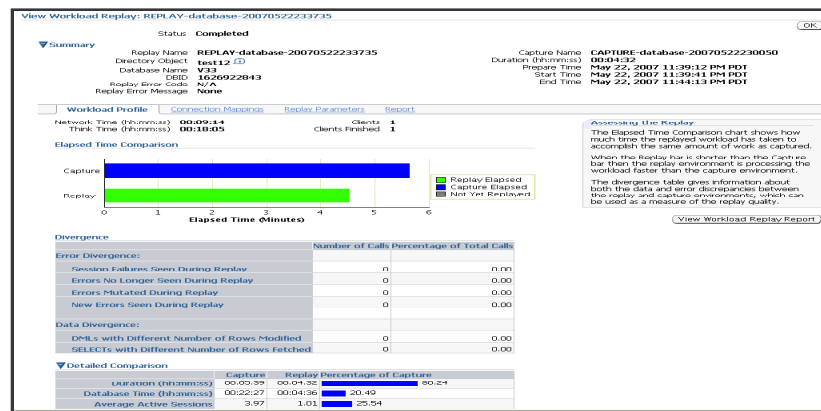


Figure 3: Workload Replay Summary

### iv. Analysis and Reporting

Extensive reports are provided to enable detailed analysis of the capture and replay.

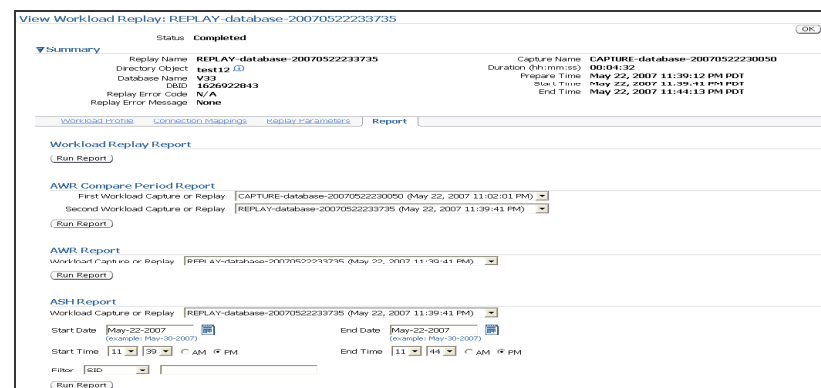


Figure 4: Types of Performance Reports Provided

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.

### 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 Set (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 cost and with a high degree of confidence.

Examples of common system changes for which you can use the SQL Performance Analyzer include:

- Database upgrade, patches, initialization parameter changes
- Configuration changes to the operating system, hardware, or database
- Schema changes such as adding new indexes, partitioning or materialized views
- Gathering optimizer statistics
- SQL tuning actions, for example, creating SQL profiles

The usage model for SQL Performance Analyzer consists of 5 main steps:

- Capture the SQL workload. Oracle Database enables you to capture a SQL workload with negligible performance overhead from several sources into a SQL tuning set (STS) such as cursor cache or Automatic Workload Repository (AWR). Typically, you would capture the SQL workload on a production database and then transport the STS to a test database where the change impact analysis using SQL Performance Analyzer will take place.
- Measure the performance of the workload before a change by executing the SQL tuning set.
- Make a change such as a database upgrade or optimizer statistics refresh
- Measure the performance of the workload after the change by executing the SQL tuning set again.

- Compare performance of the two executions of the SQL tuning set to identify, which SQL statements regressed, improved, or were unchanged.

The SQL workload can also be captured incrementally into STS over a period of time. Additionally, STS's rich filtering and ranking criteria can be used to filter out undesirable SQL. The SQL workload that has been captured into an STS on Oracle Database releases 10.2.0.1 and higher can be used with the SQL Performance Analyzer in Oracle Database 11g. The SQL Tuning Set stores all the information about the workload (such as SQL text, parsing schema, execution plans, statistics, etc.) that is necessary to re-execute the statements in a standalone environment. The seamless integration of SQL Performance Analyzer functionality with STS results in elimination of the time-consuming process of application environment setup to assess the impact of system change.

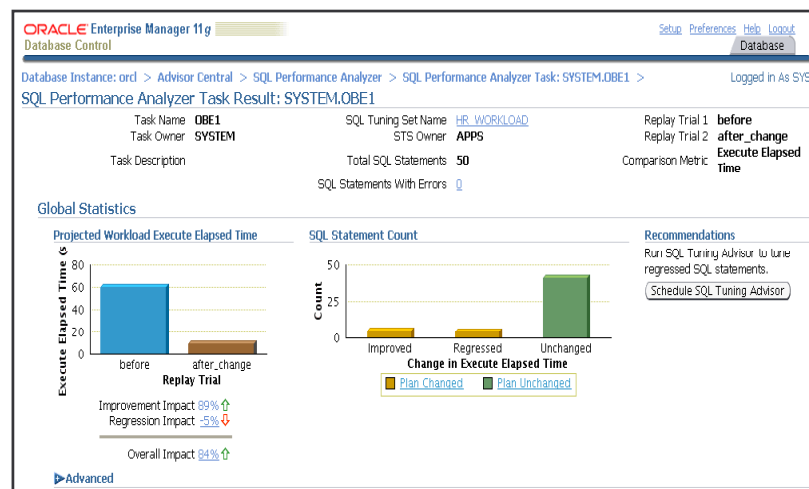


Figure 5: SQL Performance Analyzer Report

The SQL Performance Analyzer report summarizes the change impact on entire workload as well as the net impact on individual SQL statements. The execution frequency of SQL is used in weighting the workload in assessing the net impact. Additionally, improvements, regressions, and SQL with unchanged performance are listed in the report. If there are any regressed SQL statements after the system change, recommendations on how to correct the regressions is also provided with the SQL Tuning Advisor and SQL Plan Baselines features of Oracle Database 11g. Further, the execution plans generated on the test system with system change implemented can be used to seed the SQL Plan Baseline repository to make sure only previously validated execution plans are picked by the optimizer. Any new plans generated by the optimizer after seeding through the SQL Plan Baseline feature can be validated automatically by the database by test-executing them or manually by the DBAs.

## ORACLE REAL APPLICATION TESTING

### RELATED PRODUCTS AND SERVICES:

Oracle Real Application Testing Option delivers maximum benefits when used with the following Oracle Database 11g management products:

### RELATED SERVICES

- Oracle Diagnostics Pack
- Oracle Tuning Pack
- Oracle Configuration Pack
- Oracle Change Management Pack
- Oracle Provisioning Pack

Figure 6 shows an example of regressed SQL on a system change with associated before and after execution plans, statistics, and recommendation.

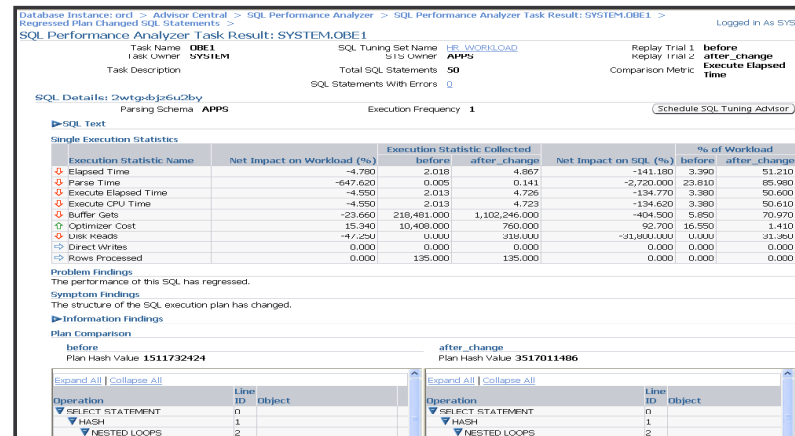


Figure 6: SQL Performance Analyzer Report: Regressed SQL Drill Down

SQL Performance Analyzer helps identify changes in performance of SQL statements for the entire system, resulting from routine administrative activities like optimizer statistics refresh and database upgrades.

## Conclusion

Real Application Testing option provides compelling and easy-to-use solution for evaluating system changes to production systems. Database Replay and SQL Performance Analyzer features of Real Application Testing provide comprehensive and flexible solution for reducing production deployment risk and lowering business costs that that is unmatched by any other product in the market. Real Application Testing adds significant business value by helping DBAs evaluate the changes with high degree of confidence and take corrective actions before business users are negatively impacted by the change.

Copyright 2007, 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 is it 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 is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.