Migrating to the Oracle Database Cloud with SQL Performance Analyzer

ORACLE WHITE PAPER | DECEMBER 2016

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. 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 for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Introduction 1

Using SQL Performance Analyzer to Validate Migration to Cloud 2

1. Create a SQL Tuning Set 2

2. Transport the SQL Tuning Set to the Oracle Cloud Database 3

3. Create a SPA Task 5

4. Convert the SQL Tuning Set to SPA Trial 1 5

5. Execute SPA Trial 2 5

6. Produce the SPA Report 6

7. Review SPA Report Results 7

8. Remediate Any Regressions 7

Licensing 7

Conclusion 8

References 9

Introduction

When migrating your database to the Oracle Cloud, SQL Performance Analyzer (SPA) is ideal for validating that your database will perform as expected. A typical on-premise database to Cloud migration involves significant and multiple changes in the stack such as database upgrade, platform changes (both HW/OS), etc. These changes can potentially impact performance SLAs and availability of the applications running on the database. SPA helps you to identify and remediate any changes in performance of SQL statements in your database after your Cloud migration and before production go-live.

SQL Performance Analyzer (SPA), a key feature of Real Application Testing in the Oracle Database, can predict and prevent SQL execution performance problems caused by system changes. SQL Performance Analyzer provides a granular view of the impact of changes on SQL execution plans and execution statistics by comparing the performance of individual SQL statements before and after a change. SQL Performance Analyzer compares the SQL execution result, before and after the change, and generates a report outlining the net benefit on the workload due to the changes as well as the set of regressed SQL statements, if any. For regressed SQL statements, appropriate execution plan details along with recommendations to remedy them are provided.

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). DBAs can use SQL plan baselines and SQL Tuning Advisor to remediate the regressed SQL statements and generate optimally performing execution plans. In a nutshell, SQL Performance Analyzer is the ideal tool for Cloud migration because:

- it is simple to use
- the time to test is short - a few hours or days - providing good time to value
- tests can be repeated multiple times without restoring the database
- it addresses SQL related performance issues which form a significant part of overall database performance issues
- it works with most workloads (packaged, in-house) out of the box

Thus, using SQL Performance Analyzer, DBAs can validate with a high degree of confidence that a system change, such as migration to the Oracle Cloud, in fact results in net positive improvement at a significantly lower cost. Note that while we focus on Cloud migration use case in this paper, the same methodology works for Cloud to on-premise database migration such as with UAT/dev instances on Cloud and on-premise production databases.

Using SQL Performance Analyzer to Validate Migration to Cloud

SQL Performance Analyzer (SPA) helps you validate that your database will perform as necessary when migrating to the Oracle Cloud.

The input to SPA is a SQL Tuning Set (STS). A SQL Tuning Set is a database object that includes one or more SQL statements, along with their execution statistics and execution context. SQL statements can be loaded into a SQL Tuning Set from different sources, including the cursor cache, Automatic Workload Repository (AWR), and existing SQL Tuning Sets.

SPA compares the performance of each individual SQL in the STS from the source environment to the performance in the target environment. In this case, the source environment will be your on-premise database. The target environment will be your Oracle Cloud database.

The most common method of using SPA in on-premise environments is to use SPA to first execute all the SQL in the unchanged database as trial 1, then make the system change that will impact SQL workload performance, and again execute all the SQL as trial 2. Finally, SPA produces a report highlighting the performance impact of the change – both for the workload as a whole and also the impact of each individual SQL.

When migrating to the Oracle Cloud, the performance information stored in the SQL Tuning Set can be used to create trial 1. SPA can then execute all the SQL in the STS on the Oracle Cloud database as trial 2. After the two trials have been created, the SPA report will show the difference in performance between the on-premise and Cloud databases.

You can use SPA from Oracle Enterprise Manager interface or PL/SQL API. Both methods are documented in the Database Testing Guide available in the Oracle database documentation library. The use of SPA from Oracle Enterprise Manager interface is fairly simple and is not the focus of this document. The only nuance when using Oracle Enterprise Manager interface for SPA Cloud migration use case is to use “Build from SQL Tuning” trial method option for Trial 1. The example below demonstrates the use of SPA with PL/SQL API.

1. Create a SQL Tuning Set

The first step is to create the SQL Tuning Set (STS) that contains the set of SQL to be tested from your on-premise database. An STS can be created by polling the cursor cache periodically over a time duration (recommended best practice), from existing AWR snapshots, or as a subset of an existing STS.

To create an STS from the cursor cache using PL/SQL, you use the DBMS_SQLTUNE. CAPTURE_CURSOR_CACHE_SQLSET procedure. You can specify a SQL Tuning Set name, the duration of time in seconds for the capture to run and the polling interval. For example:

```
EXEC DBMS_SQLTUNE.CAPTURE_CURSOR_CACHE_SQLSET( -
                                sqlset_name     => 'CLOUD_STS', -
                                time_limit      =>  3600, -
                                repeat_interval =>  60);
```

This would create an STS named CLOUD_STS by polling the cursor cache for SQL every 60 seconds over the next 3600 seconds (one hour).

You can also create an STS from SQL captured in Automatic Workload Repository (AWR) snapshots. AWR automatically captures high load SQL at regular intervals in your database. You can create an STS with the SQL from a range of snapshots in the AWR. For example, the following code snippet creates an STS named CLOUD_STS. It then loads that STS with SQL from AWR snapshots 1000 through 1010.

```
EXEC DBMS_SQLTUNE.CREATE_SQLSET (sqlset_name  => 'CLOUD_STS');
DECLARE
  cur DBMS_SQLTUNE.SQLSET_CURSOR;
BEGIN
  OPEN cur FOR
    SELECT VALUE(P)
      FROM table(
        DBMS_SQLTUNE.SELECT_WORKLOAD_REPOSITORY(1000,1010) ) P;

  DBMS_SQLTUNE.LOAD_SQLSET(sqlset_name => 'CLOUD_STS',
                           populate_cursor => cur);
END;
/
```

2. Transport the SQL Tuning Set to the Oracle Cloud Database

Once you've created your STS, you need to copy it to your Oracle Cloud database. The steps do to so are:

a) Create a staging table to hold the SQL Tuning Set
b) Copy the SQL Tuning Set to that table
c) Export the table holding the SQL Tuning Set with datapump
d) Copy the exported SQL Tuning Set to the Cloud using operating system commands.
e) Using datapump, import the SQL Tuning Set to a staging table
f) Load the SQL Tuning Set from that table

Here are specific details on how to perform those steps.

a) Create a staging table to hold the SQL Tuning Set. To create a staging table, you specify a schema and a table name to hold the STS. For example:

```
BEGIN
       DBMS_SQLTUNE.CREATE_STGTAB_SQLSET (
              table_name => 'STS_STAGING_TABLE'
              , schema_name => 'SCOTT');
END;
/
```

b) Next, copy the STS to that table. For example:

```
BEGIN
       DBMS_SQLTUNE.PACK_STGTAB_SQLSET (
              sqlset_name => 'CLOUD_STS'
              , sqlset_owner => 'SYS'
              , staging_table_name => 'STS_STAGING_TABLE'
              , staging_schema_owner => 'SCOTT');
END;
/
```

c) Export that table to the filesystem using Oracle Datapump. For example (*the following should be on a single line or an appropriate line continuation character should be supplied at shell prompt*):

```
expdp system/pwd DIRECTORY=mydir DUMPFILE=cloud_sts.dmp
TABLES=scott.sts_staging_table
```

d) Copy the datapump file to the Oracle Cloud host. For example:

scp -i *mykeyfile cloud*_sts.dmp oracle@*mycloud_host*:

e) Import the table containing the SQL Tuning Set into the Oracle Cloud Database.

impdp system/*pwd* DIRECTORY=*mydir* DUMPFILE=cloud_sts.dmp TABLES=scott.sts_staging_table

f) Load the SQL Tuning Set from that table:

```
BEGIN
       DBMS_SQLTUNE.UNPACK_STGTAB_SQLSET (
              sqlset_name => 'CLOUD_STS'
              , replace => true
              , staging_table_name => 'STS_STAGING_TABLE'
              , staging_schema_owner => 'SCOTT');
END;
/
```

3. Create a SPA Task

A container for SPA trials (or experiments) is called a SPA 'task'. A SPA task consists of related SPA trials, along with their reports. The first step for SPA is to create a SPA task on the Oracle Cloud Database. For example:

```
VARIABLE task_name VARCHAR2(100);
EXEC :task_name := DBMS_SQLPA.CREATE_ANALYSIS_TASK( -
         sqlset_name => 'CLOUD_STS' -
         , task_name => 'CLOUD_TASK');
```

4. Convert the SQL Tuning Set to SPA Trial 1

The SQL Tuning Set loaded into the Cloud database above contains execution statistics for each of the SQL statements in the STS. These execution statistics can be used as trial 1 for SPA. We can use the ‘convert_sqlset’ trial method option (equivalent to Oracle Enterprise Manager interface “Build from STS” trial method) since we may not have access to the on-premise database for creating the trials or Oracle Enterprise Manager Hybrid Cloud Management has not been setup for whatever reasons. For example:

```
BEGIN
       DBMS_SQLPA.EXECUTE_ANALYSIS_TASK (
              task_name => 'CLOUD_TASK'
              ,execution_type => 'convert sqlset'
              ,execution_name => 'CLOUD_TRIAL_1');
  END;
/
```

5. Execute SPA Trial 2

The next step is to execute all of the SQL in the STS on the Cloud database to measure any changes in performance. For example:

```
BEGIN
       DBMS_SQLPA.EXECUTE_ANALYSIS_TASK (
              task_name => 'CLOUD_TASK'
              ,execution_type => 'TEST EXECUTE'
              ,execution_name => 'CLOUD_TRIAL_2');
  END;
/
```

6. Produce the SPA Report

After the two trials have been completed, SPA can be used to compare the results. To create a report showing the results of the two SPA trials previously executed, you generate the comparison and then produce a report. The code snippet below tells SPA to create a comparison between trial 1 and trial 2, using 'buffer gets' as the comparison metric.

```
BEGIN
       DBMS_SQLPA.EXECUTE_ANALYSIS_TASK(
              task_name => 'CLOUD_TASK'
              ,execution_type => 'compare performance'
              ,execution_params => dbms_advisor.arglist('execution_name1'
                                                  ,'CLOUD_TRIAL_1'
                                                  ,'execution_name2'
                                                  ,'CLOUD_TRIAL_2'
                                                  ,'comparison_metric'
                                                  ,'buffer_gets'));
END;
/
```

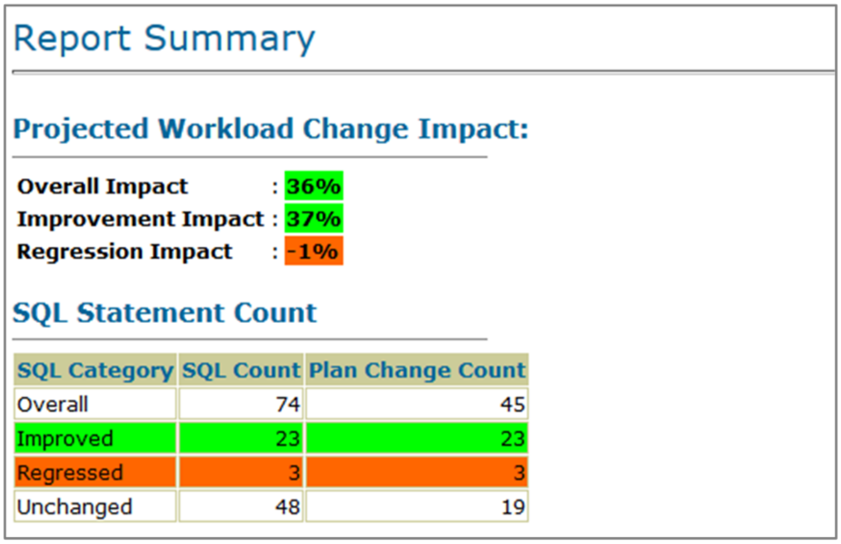
You can produce a report of the results in text, html or 'active html', which is an html report with interactive feaures. For example, to produce an html report from SQLPLUS:

```
SET LONG 100000 LONGCHUNKSIZE 100000 LINESIZE 130 PAGESIZE 0 ECHO OFF
spool cloud_task_report.html

SELECT DBMS_SQLPA.REPORT_ANALYSIS_TASK('CLOUD_TASK','html')
FROM DUAL;

spool off
```

7. Review SPA Report Results

After generating the SPA report, review the results.

Here is an example of a SPA report summary. It shows an overall 36% performance improvement, after the migration.   The summary shows that some SQL statements improved overall performance by 37%, but a few statements regressed for a negative 1% impact, resulting in a 36% improvement overall.



Report Summary

Projected Workload Change Impact:

Overall Impact	:	36%
Improvement Impact	:	37%
Regression Impact	:	-1%

SQL Statement Count

SQL Category	SQL Count	Plan Change Count
Overall	74	45
Improved	23	23
Regressed	3	3
Unchanged	48	19

Below this section of the report, all the individual SQL statements are listed, showing the metrics for each statement, along with percentage improvements for each SQL and for the workload as a whole.

8. Remediate Any Regressions

If there are any SQL regressions, SQL Tuning Advisor and SQL Plan Baselines can be used to remediate them. SQL Tuning Advisor can further improve performance by exploring alternate execution plans. Or, SQL Plan Baselines functionality can be used to revert to previously known plans.   For further details on using the SQL Tuning Advisor, or creating SQL Plan Baselines, see section 6.2.4 in the Database 12.2 Testing Guide.

Licensing

STS capture for on-premise databases is free and not part of the Database Tuning Pack or Real Application Testing. Please refer to the Oracle Database Licensing documentation for complete details. Licensing for other components of Real Application Testing is unchanged for on-premise databases.

Real Application Testing is available on Oracle Database Cloud with Enterprise Edition High Performance Extreme Performance. Please refer to the licensing documentation for more details.

Conclusion

When migrating your on-premise Oracle database to the Oracle Cloud, SQL Performance Analyzer can help assure that your database performs the same, or better in the Oracle Cloud as it did in your on-premise database. SQL Performance Analyzer can help you perform your database performance validation with significantly lower effort and cost assuring a smooth migration to the Oracle Cloud.

References

Oracle Database Testing Guide

Oracle Database Licensing Guide

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

- blogs.oracle.com/oracle
- facebook.com/oracle
- twitter.com/oracle
- oracle.com

Integrated Cloud Applications & Platform Services

White Paper Title
December 2016
Author: [OPTIONAL]
Contributing Authors: [OPTIONAL]

Integrated Cloud Applications & Platform Services