Real Application Testing in Oracle Database

Ulrike Schwinn

Distinguished Data Management Expert

Ulrike.Schwinn@oracle.com

https://www.linkedin.com/in/ulrikeschwinn/
https://blogs.oracle.com/coretec/

Safe harbor statement

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions, and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of February 2021 and Oracle undertakes no duty to update any statement in light of new information or future events.

Database Management

Top challenges

Environments - where Oracle Databases are deployed - change regularly

- **upgrade/release update**
- platform changes
- storage, network, and interconnect changes
- Operating System and hardware migrations
- database migrations
- architectural changes
- new feature implementations
- workload and capacity changes

Acceleration factor: Companies currently switch their infrastructure to cloud environments

➢ But Businesses need to adapt to changes to stay competitive, compliant and evolve AND save time and money!

Solution: Real Application Testing (RAT)

What?

- 2 components: **Database Replay (DB Replay)** and **SQL Performance Analyzer (SPA)**

Why?

- guarantees smooth risk free transition to the new environment
- is THE unique technology to address challenges/changes for Oracle Databases
- is available out-of-the-box
- works with different setups/environments (i.e. analytics, OLTP or mixed workloads, SAP, legacy applications, cloud, on-premises, Exadata etc.)

Business benefit

- Cost reduction e.g. 224% ROI over 3 years 5.9 months payback period (*)

Availability

- Oracle Database Option, but free in Oracle Cloud

(*) Source: The total Economic Impact of Real Application Testing (Forrester)

The scopes

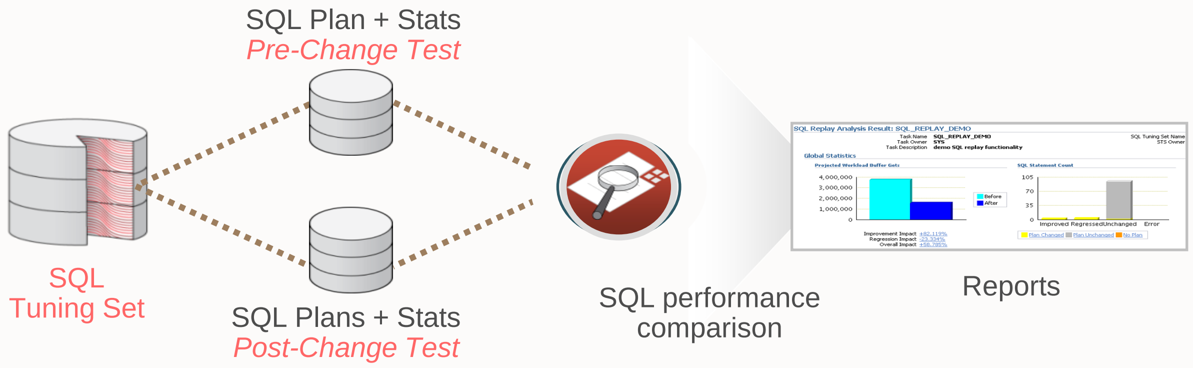
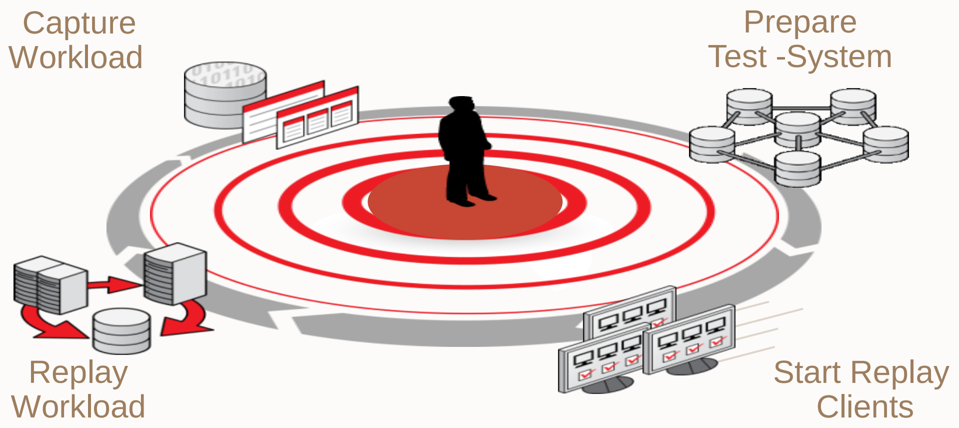
Preparation	Setup	Tests	Analysis/follow-up
• Scope • Skills • Test environment	Test environment: • Cloning • RMAN • Data Pump • Data: Masking/Subsetting	• Scripts written in-house • Special testing tools • Swingbench • Oracle Tools	• Documentation • Evaluation • Tuning/Remediation of bad SQL execution

Real Application Testing

Real Application Testing

- Included in the Oracle Database Software
- First release with Oracle Database 11*g* Release 1
- Pre-Installed and configured
- No limitations, restrictions, no changes
- Usage via Linemode Scripting and/or graphical interface
  - Rich GUI through Enterprise Manager Cloud Control
  - PL/SQL packages
  - Can also be used in online mode
- Provides detailed reporting
- Components
  - SQL Performance Analyzer (SPA) – to assess the impact of change on SQL response time
  - Database Replay (DB Replay) – to assess the impact of change on workload throughput

Oracle Real Application Testing – at a glance

SQL Performance Analyzer (SPA)	Database Replay (DB Replay)
for detailed analysis of an SQL workload	for performance tests of the entire workload
SQL Response Time	Throughput Testing/ Concurrency
 SQL Plan + Stats *Pre-Change Test* SQL Tuning Set SQL Plans + Stats *Post-Change Test* SQL performance comparison SQL Replay Analysis Result: SQL_REPLAY_DEMO Task Name SQL_REPLAY_DEMO Task Owner SYS Task Description demo SQL replay functionality SQL Tuning Set Name STS Owner Global Statistics Projected Workload Buffer Gets 4,000,000 3,000,000 2,000,000 1,000,000 0 Before After Improvement Impact +82.119% Regression Impact -23.334% Overall Impact +58.785% SQL Statement Count 105 70 35 0 Improved Regressed Unchanged Error Plan Changed Plan Unchanged No Plan Reports	 Capture Workload Prepare Test -System Replay Workload Start Replay Clients

Why use SPA?

Ensure applications consistently perform well

- SQL performance regressions: #1 cause of poor system performance

Applications are complex and consistently maintaining **high QoS** is difficult

- Large workloads (high number of SQL statements)
- It is time consuming to check every single SQL and tune it manually
- Partial workload with non production optimizer information
- Validation in production is not too uncommon

SQL Performance Analyzer (SPA)

➢ Proactively detects ALL SQL regressions, BEFORE actual change is deployed

➢ Integrated comprehensive solution for end-to-end SQL workload testing

Real Application Testing

SQL Performance Analyzer (SPA)

Why?

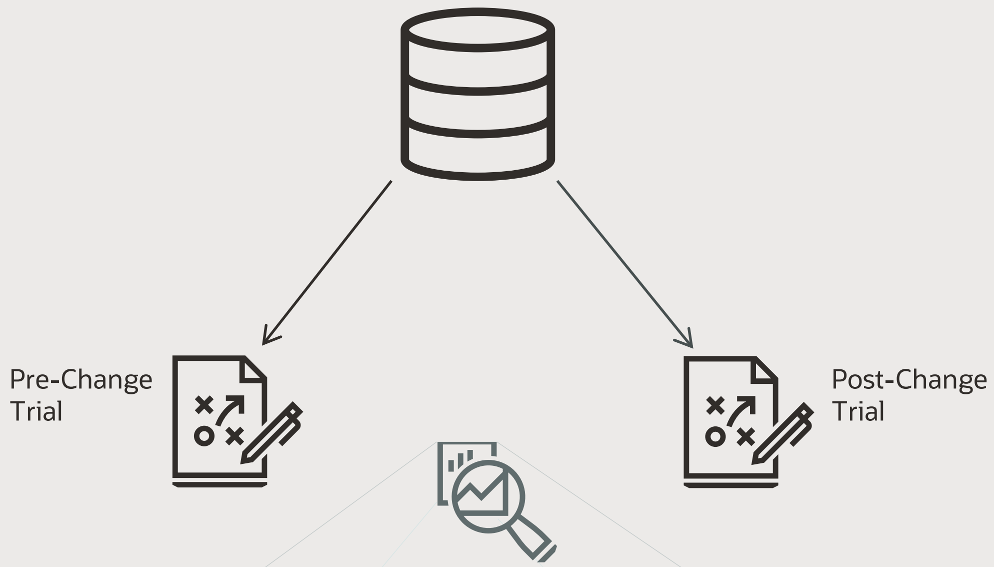
- SQL performance regressions: #1 cause of poor system performance
- Applications are complex and consistently maintaining high QoS is difficult
  - Large workloads (high number of SQL statements)
  - It is time consuming to check every single SQL and tune it manually
  - Partial workload with non production optimizer information
  - Validation in production is not too uncommon

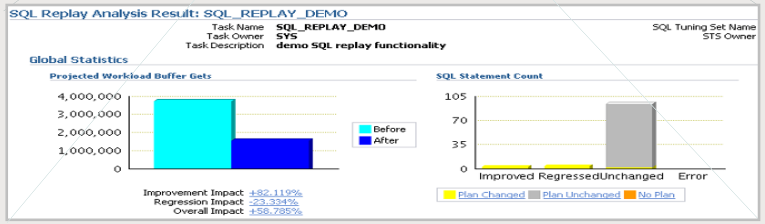
SQL Performance Analyzer

➢ Proactively detects ALL SQL regressions, BEFORE actual change is deployed

➢ Integrated comprehensive solution for end-to-end SQL workload testing such as STS, SQL Plan Baselines, & SQL Tuning Advisor

(*) Blog Posting:
[SQL Performance Analyzer in Autonomous Database (English)]





SQL Replay Analysis Result: SQL_REPLAY_DEMO

Task Name SQL_REPLAY_DEMO
Task Owner SYS
Task Description demo SQL replay functionality

SQL Tuning Set Name
STS Owner

Global Statistics

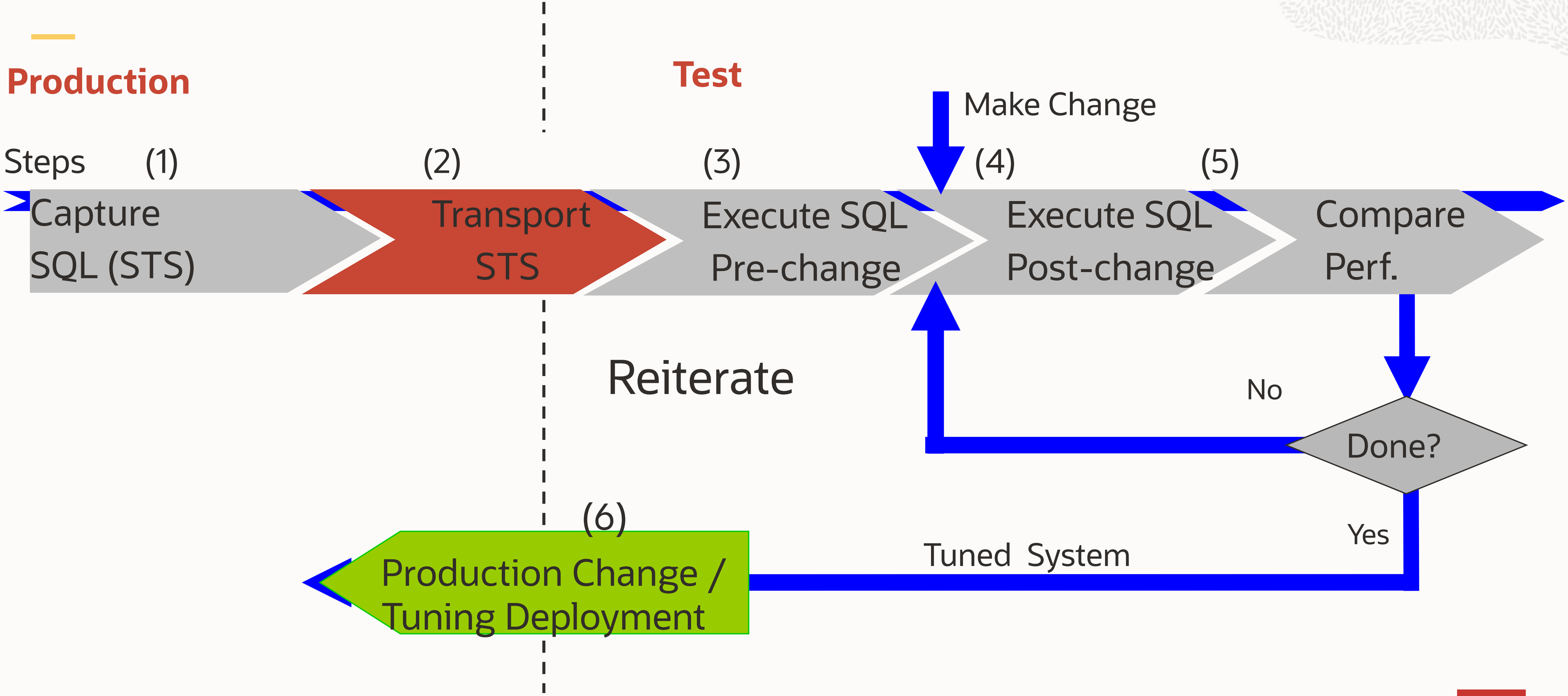
Projected Workload Buffer Gets

Improvement Impact +82.119%
Regression Impact -23.334%
Overall Impact +58.785%

SQL Statement Count

Analysis report

Workflow



Production

Test

Make Change

Steps (1) (2) (3) (4) (5)

Capture SQL (STS)

Transport STS

Execute SQL Pre-change

Execute SQL Post-change

Compare Perf.

Reiterate

No

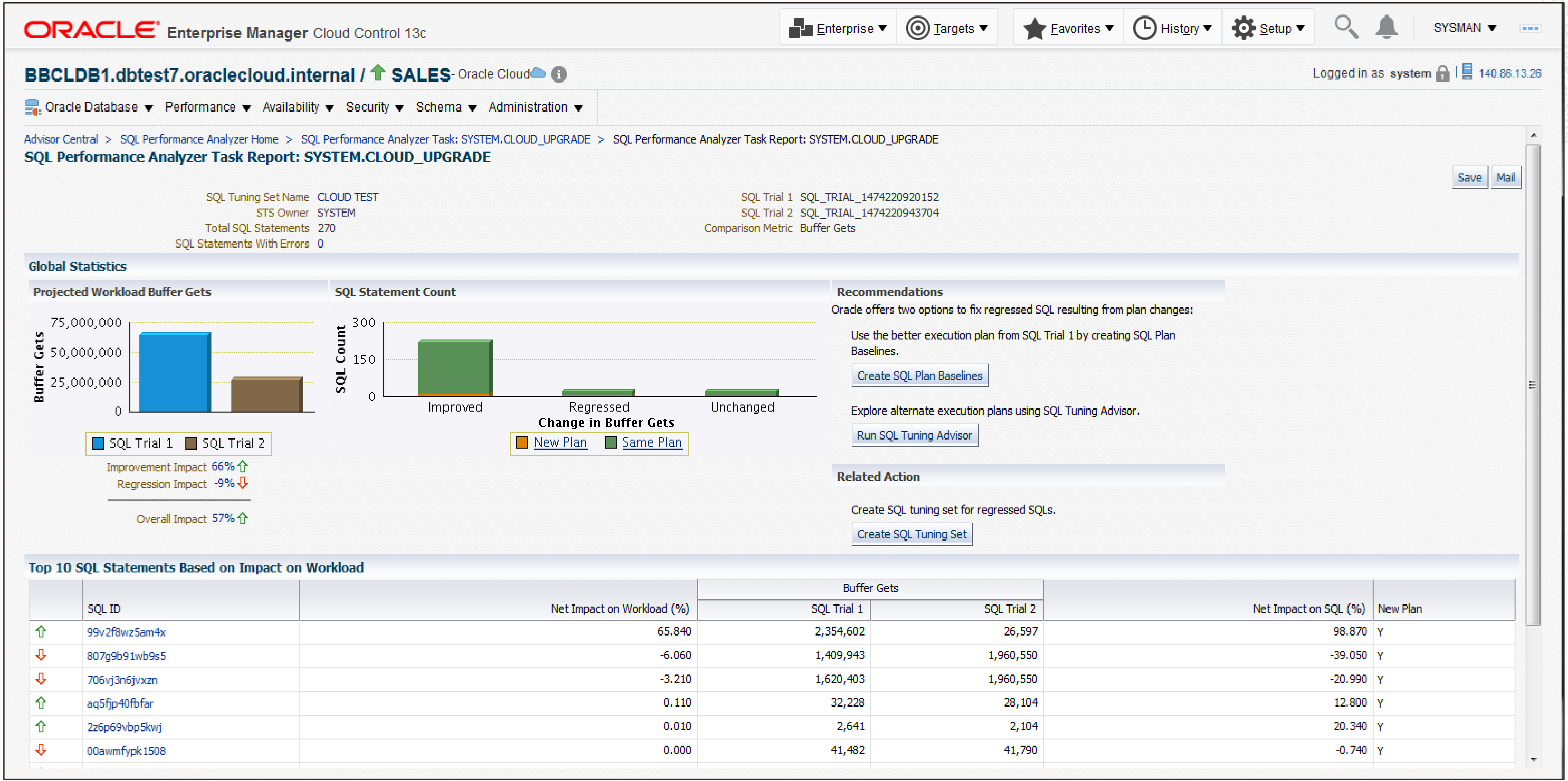
Done?

Yes

Tuned System

(6)

Production Change / Tuning Deployment



BBCLDB1.dbtest7.oraclecloud.internal / SALES - Oracle Cloud

Logged in as system | 140.86.13.26

Advisor Central > SQL Performance Analyzer Home > SQL Performance Analyzer Task: SYSTEM.CLOUD_UPGRADE > SQL Performance Analyzer Task Report: SYSTEM.CLOUD_UPGRADE

SQL Performance Analyzer Task Report: SYSTEM.CLOUD_UPGRADE

SQL Tuning Set Name	CLOUD TEST	SQL Trial 1	SQL_TRIAL_1474220920152
STS Owner	SYSTEM	SQL Trial 2	SQL_TRIAL_1474220943704
Total SQL Statements	270	Comparison Metric	Buffer Gets
SQL Statements With Errors	0		

Global Statistics

Projected Workload Buffer Gets

Improvement Impact 66% ⇧
Regression Impact -9% ⇩
Overall Impact 57% ⇧

SQL Statement Count

Recommendations

Oracle offers two options to fix regressed SQL resulting from plan changes:

Use the better execution plan from SQL Trial 1 by creating SQL Plan Baselines.

Explore alternate execution plans using SQL Tuning Advisor.

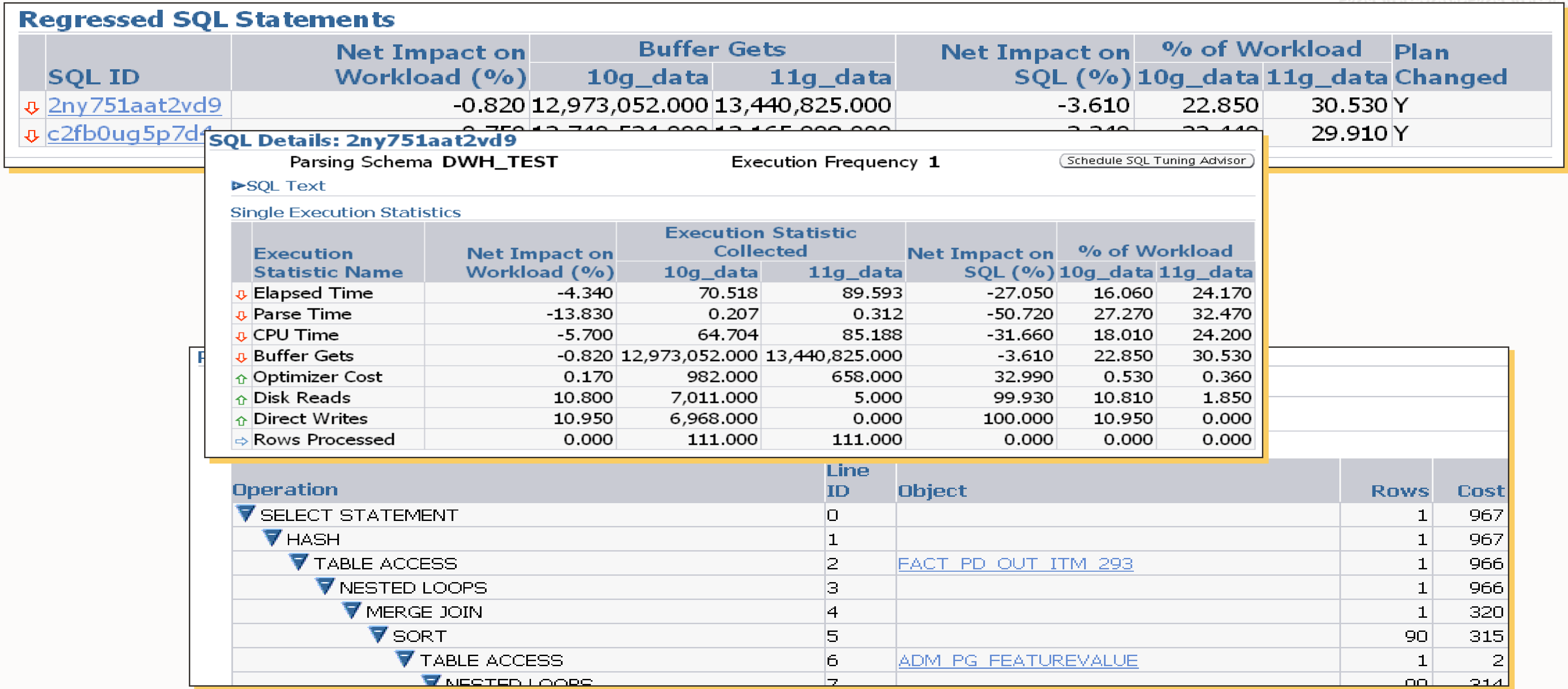
Related Action

Create SQL tuning set for regressed SQLs.

Top 10 SQL Statements Based on Impact on Workload

	SQL ID	Net Impact on Workload (%)	Buffer Gets SQL Trial 1	 SQL Trial 2	Net Impact on SQL (%)	New Plan
⇧	99v2f8wz5am4x	65.840	2,354,602	26,597	98.870	Y
⇩	807g9b91wb9s5	-6.060	1,409,943	1,960,550	-39.050	Y
⇩	706vj3n6jvxzn	-3.210	1,620,403	1,960,550	-20.990	Y
⇧	aq5fjp40fbfar	0.110	32,228	28,104	12.800	Y
⇧	2z6p69vbp5kwj	0.010	2,641	2,104	20.340	Y
⇩	00awmfypk1508	0.000	41,482	41,790	-0.740	Y

SPA Report



Regressed SQL Statements

	SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	% of Workload		Plan Changed
			10g_data	11g_data		10g_data	11g_data	
⇩	2ny751aat2vd9	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530	Y
⇩	c2fb0ug5p7d4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	29.910	Y

SQL Details: 2ny751aat2vd9

Parsing Schema DWH_TEST

Execution Frequency 1

▶SQL Text

Single Execution Statistics

	Execution Statistic Name	Net Impact on Workload (%)	Execution Statistic Collected		Net Impact on SQL (%)	% of Workload	
			10g_data	11g_data		10g_data	11g_data
⇩	Elapsed Time	-4.340	70.518	89.593	-27.050	16.060	24.170
⇩	Parse Time	-13.830	0.207	0.312	-50.720	27.270	32.470
⇩	CPU Time	-5.700	64.704	85.188	-31.660	18.010	24.200
⇩	Buffer Gets	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530
⇧	Optimizer Cost	0.170	982.000	658.000	32.990	0.530	0.360
⇧	Disk Reads	10.800	7,011.000	5.000	99.930	10.810	1.850
⇧	Direct Writes	10.950	6,968.000	0.000	100.000	10.950	0.000
⇨	Rows Processed	0.000	111.000	111.000	0.000	0.000	0.000

Operation	Line ID	Object	Rows	Cost
▼ SELECT STATEMENT	0		1	967
▼ HASH	1		1	967
▼ TABLE ACCESS	2	FACT_PD_OUT_ITM_293	1	966
▼ NESTED LOOPS	3		1	966
▼ MERGE JOIN	4		1	320
▼ SORT	5		90	315
▼ TABLE ACCESS	6	ADM_PG_FEATUREVALUE	1	2
▼ NESTED LOOPS	7		90	314

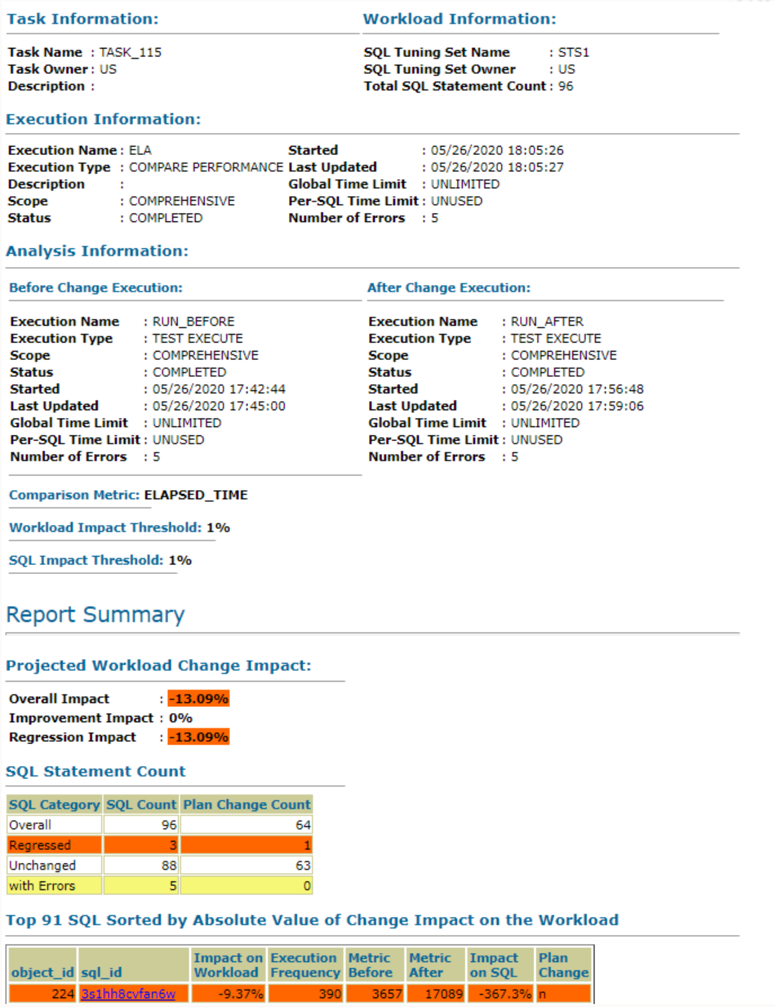
Linemode usage

```
-- create task
execute  :tname := DBMS_SQLPA.CREATE_ANALYSIS_TASK(sqlset_name => 'STS_HINTS');

-- first run
execute DBMS_SQLPA.EXECUTE_ANALYSIS_TASK(task_name  => 'TASK_115', -
                                         execution_type  => 'test execute', -
                                         execution_name  => 'RUN_BEFORE’);
-- change: Ignore Optimizer Hints
alter session set optimizer_ignore_hints=true;
Session altered.

-- second run
execute DBMS_SQLPA.EXECUTE_ANALYSIS_TASK(task_name  => 'TASK_115', -
                                         execution_type  => 'test execute', -
                                         execution_name  => 'RUN_AFTER');
```

SPA reports

```
set long 1000000 longchunksize 1000000 linesize 200
set head off feedback off echo off
spool detail.html
select dbms_sqlpa.report_analysis_task('&Taskname','HTML', 'REGRESSED',
         'ALL', null, 100, '&executionname') detailed_rep from dual;
spool off
```

- Different reports are available:
  `REGRESSED, IMPROVED, CHANGED_PLANS, ALL etc.`



Task Information:

Task Name : TASK_115
Task Owner : US
Description :

Workload Information:

SQL Tuning Set Name : STS1
SQL Tuning Set Owner : US
Total SQL Statement Count : 96

Execution Information:

Execution Name : ELA
Execution Type : COMPARE PERFORMANCE
Description :
Scope : COMPREHENSIVE
Status : COMPLETED
Started : 05/26/2020 18:05:26
Last Updated : 05/26/2020 18:05:27
Global Time Limit : UNLIMITED
Per-SQL Time Limit : UNUSED
Number of Errors : 5

Analysis Information:

Before Change Execution:

Execution Name : RUN_BEFORE
Execution Type : TEST EXECUTE
Scope : COMPREHENSIVE
Status : COMPLETED
Started : 05/26/2020 17:42:44
Last Updated : 05/26/2020 17:45:00
Global Time Limit : UNLIMITED
Per-SQL Time Limit : UNUSED
Number of Errors : 5

After Change Execution:

Execution Name : RUN_AFTER
Execution Type : TEST EXECUTE
Scope : COMPREHENSIVE
Status : COMPLETED
Started : 05/26/2020 17:56:48
Last Updated : 05/26/2020 17:59:06
Global Time Limit : UNLIMITED
Per-SQL Time Limit : UNUSED
Number of Errors : 5

Comparison Metric: **ELAPSED_TIME**

Workload Impact Threshold: **1%**

SQL Impact Threshold: **1%**

Report Summary

Projected Workload Change Impact:

Overall Impact : -13.09%
Improvement Impact : 0%
Regression Impact : -13.09%

SQL Statement Count

SQL Category	SQL Count	Plan Change Count
Overall	96	64
Regressed	3	1
Unchanged	88	63
with Errors	5	0

Top 91 SQL Sorted by Absolute Value of Change Impact on the Workload

object_id	sql_id	Impact on Workload	Execution Frequency	Metric Before	Metric After	Impact on SQL	Plan Change
224	3s1hh8cvfan6w	-9.37%	390	3657	17089	-367.3%	n

Real Application Testing

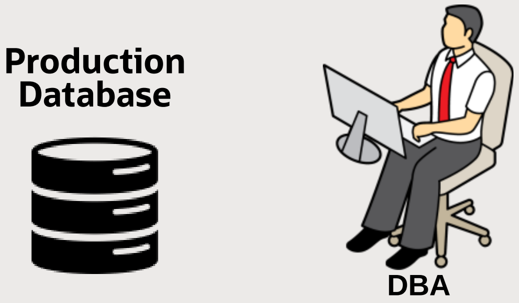
SPA Quick Check

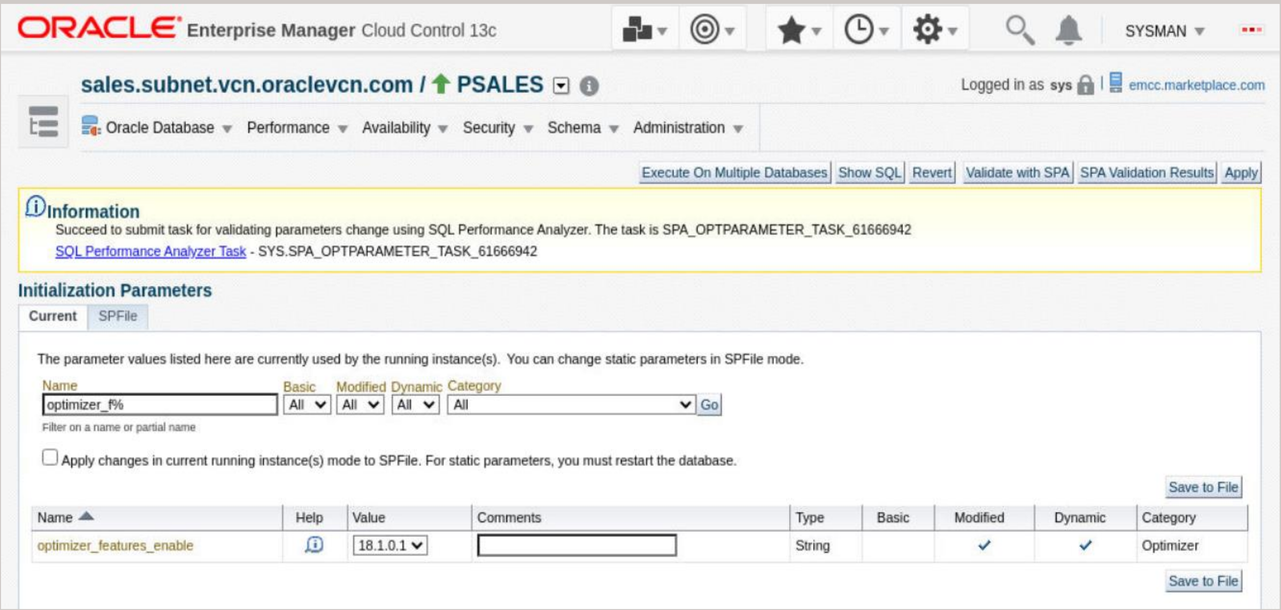
Why?

- Testing requires adequate separate hardware
- Testing can be resource intensive, lengthy, and error prone

SPA Quick Check

- ➢ Included in Enterprise Manager Cloud Control
- ➢ Optimized for use on production environment
- ➢ Context-aware change testing





sales.subnet.vcn.oraclevcn.com / PSALES

Logged in as sys | emcc.marketplace.com

Information

Succeed to submit task for validating parameters change using SQL Performance Analyzer. The task is SPA_OPTPARAMETER_TASK_61666942

SQL Performance Analyzer Task - SYS.SPA_OPTPARAMETER_TASK_61666942

Initialization Parameters

Current | SPFile

The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFile mode.

Name	Basic	Modified	Dynamic	Category	
optimizer_f%	All	All	All	All	Go

Filter on a name or partial name

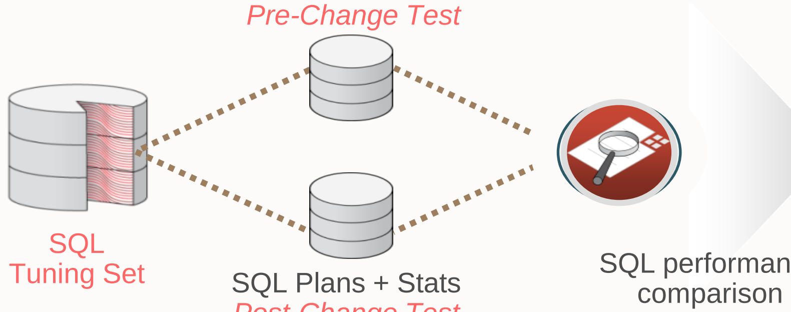
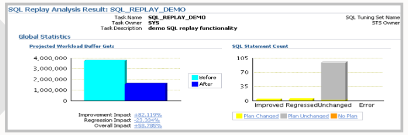
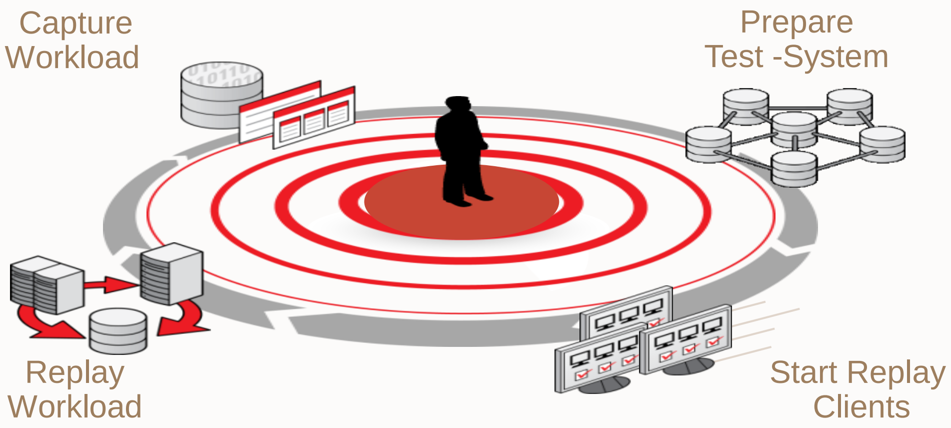
□ Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Save to File

Name ▲	Help	Value	Comments	Type	Basic	Modified	Dynamic	Category
optimizer_features_enable	ⓘ	18.1.0.1		String		✓	✓	Optimizer

Save to File

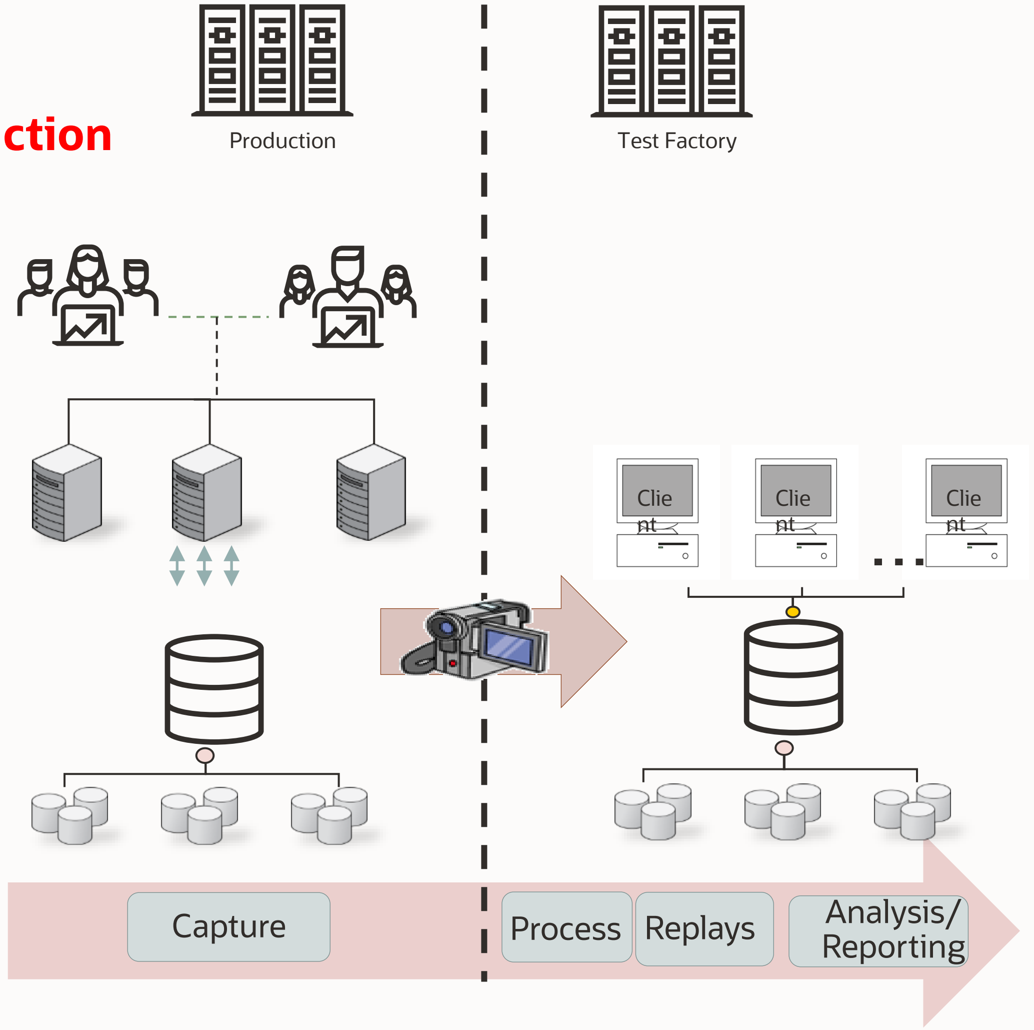
Oracle Real Application Testing – at a glance

SQL Performance Analyzer (SPA)	Database Replay (DB Replay)
for detailed analysis of an SQL workload	for performance tests of the entire workload
SQL Response Time	Throughput Testing/ Concurrency
 SQL Plan + Stats *Pre-Change Test* SQL Tuning Set SQL Plans + Stats *Post-Change Test* SQL performance comparison  SQL Replay Analysis Result: SQL_REPLAY_DEMO Task Name **SQL_REPLAY_DEMO** Task Owner **SYS** Task Description **demo SQL replay functionality** SQL Tuning Set Name STS Owner Global Statistics Improvement Impact +82.119% Regression Impact -23.334% Overall Impact +58.785% Projected Workload Buffer Gets 4,000,000 3,000,000 2,000,000 1,000,000 0 Before After SQL Statement Count 105 70 35 0 Improved Regressed Unchanged Error Plan Changed Plan Unchanged No Plan Reports	 Capture Workload Prepare Test -System Replay Workload Start Replay Clients

Database Replay

- **Database load and performance testing with real production workloads**
  - Production workload characteristics such as timing, transaction dependency, think time etc. are fully maintained

- **Remediate issues pre-production for risk-free migration**
  - Test and measure transaction throughput
  - Identify application scalability and concurrency problems
  - Ability to detect potential issues

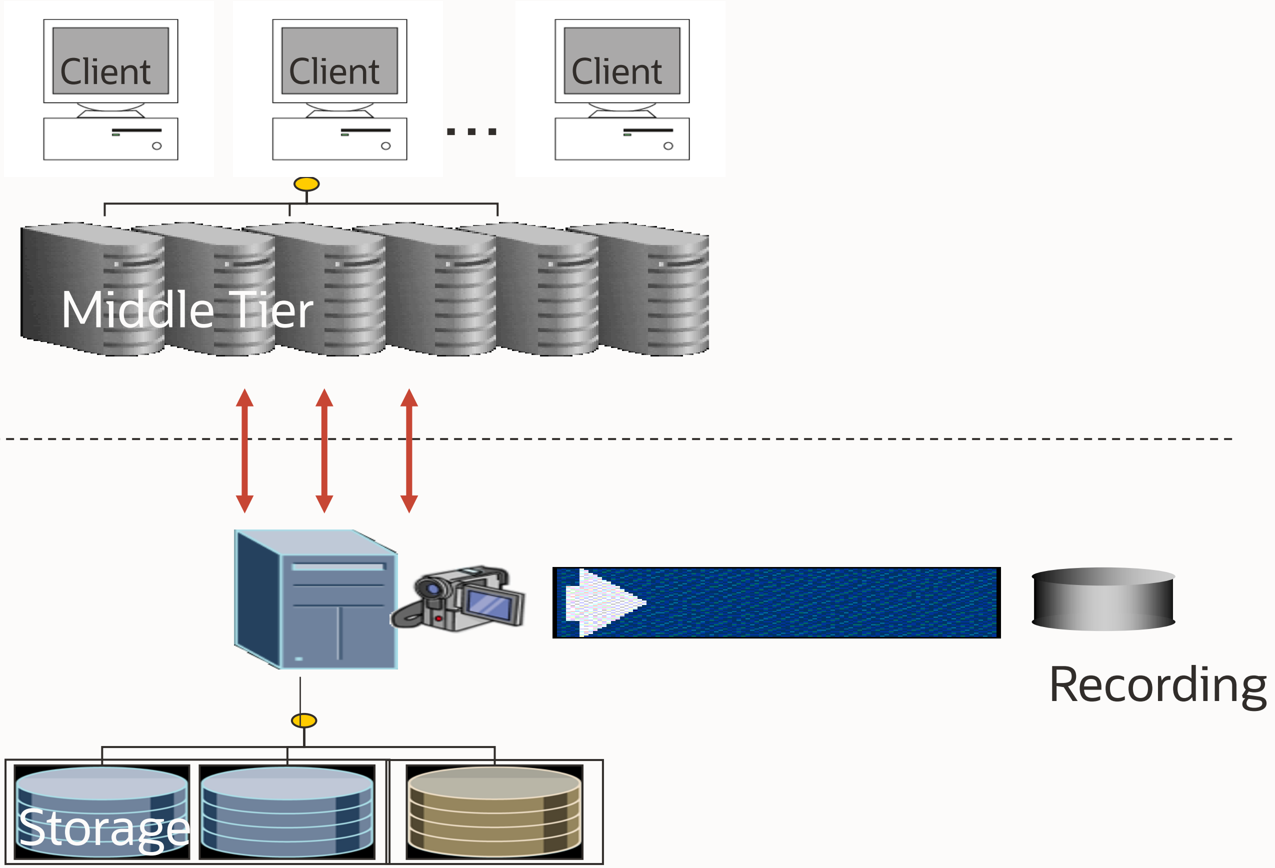


Supported Changes

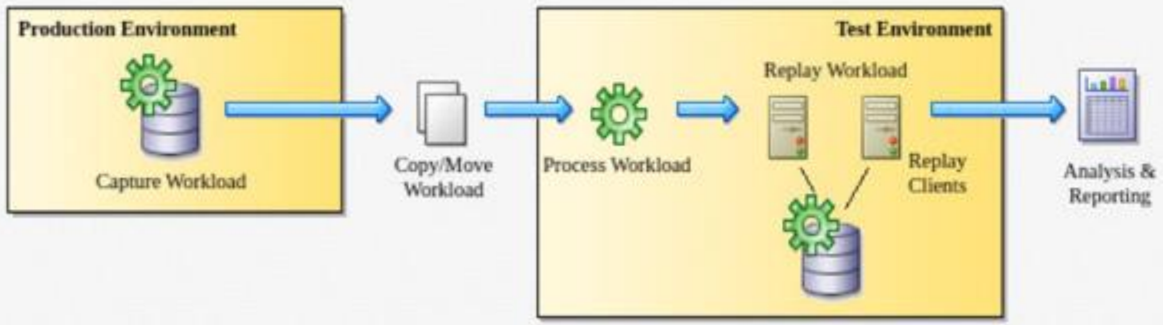
Changes Unsupported

Changes Supported

- Database Upgrades, Patches
- Schema, Parameters
- RAC nodes, Interconnect
- OS Platforms, OS Upgrades
- CPU, Memory
- Storage …

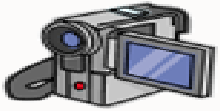


Database Replay - Setup



Production System:

1. Workload Capture



Preparing the test system (backup, reseting, isolation etc)

Copy the workload capture

Test System:

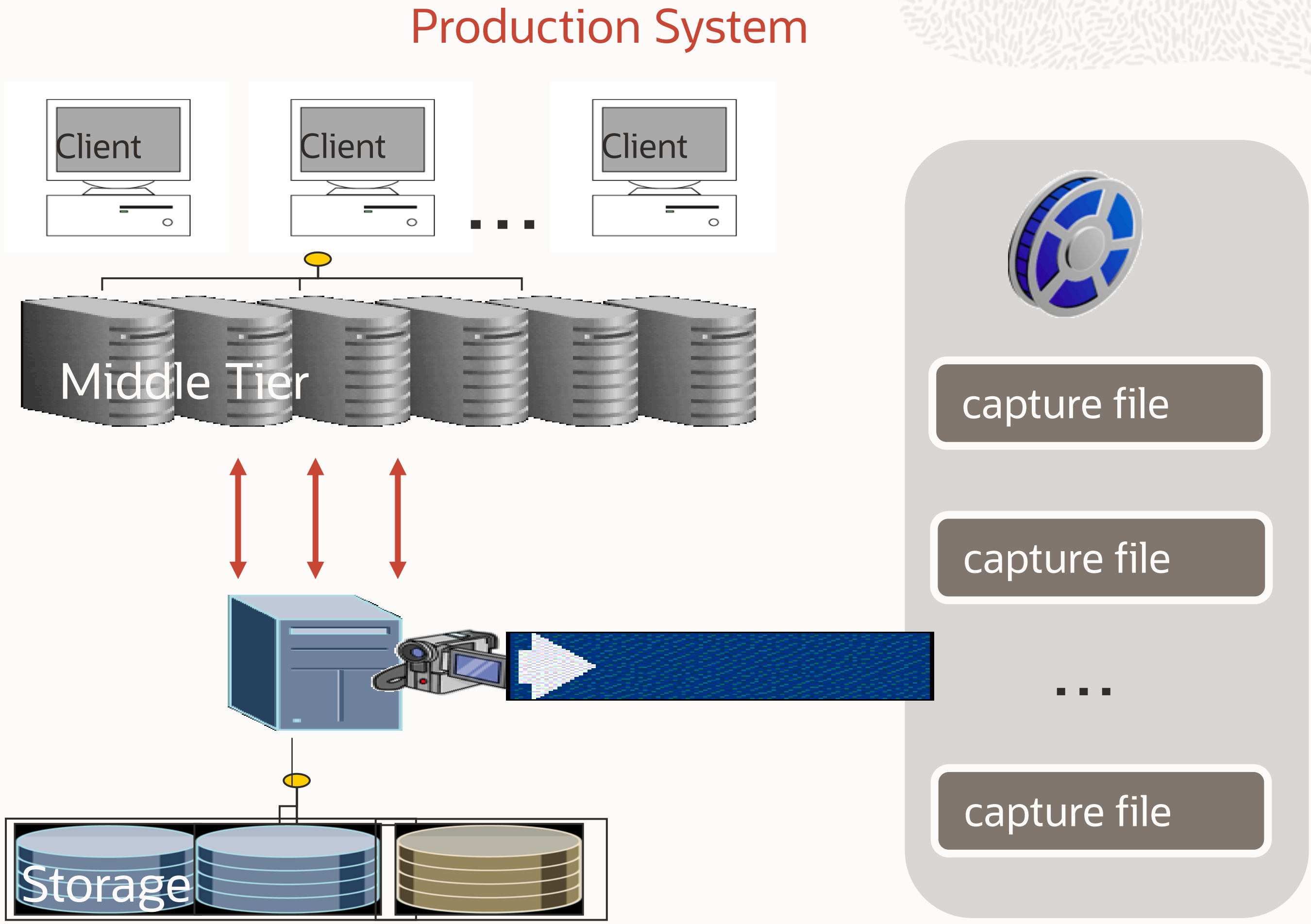
2. Processing
3. Replay Workload mit Workload Replay Clients
4. Analysis

Challenges

- Set expectations
  - Provide a clear objective for your testing
  - Provide success criteria
  - Provide a test plan
  - Decide on the capture duration/workload
  - Check Capture Restrictions
- Setup the test environment
  - Should be as close as possible to the start capture **SCN**
  - Size database replay servers similar to production (CPU, Memory, Network bandwith)
  - Methods: Clone, Standby, Export/Import, Backup, Flashback
  - Resolve references to **external systems** such as database links, web services, flat files access, but also email notifications etc. => Isolation!

Step 1: Workload Capture

- Determine capture interval e.g. peak workload, month-end processing, etc.
- All external client requests are captured in binary files
- Minimal performance overhead
- Consider using filters such as INSTANCE_NUMBER, USER, MODULE, ACTION, PROGRAM, SERVICE, PDB_NAME
- Size of Capture Data Calculation
  - Proportional to the data sent by the clients
  - AWR: 2* (bytes received via SQL*Net from client or bytes received by DB links)
- Examples
  - TPC-C 20min: 100 users 10 DWHs: 1.2G
  - Cebit 9 hours: 40 GB
  - SAP system : 2 hours almost 300 GB
  - OLTP System: 50 GB but 250 000 files

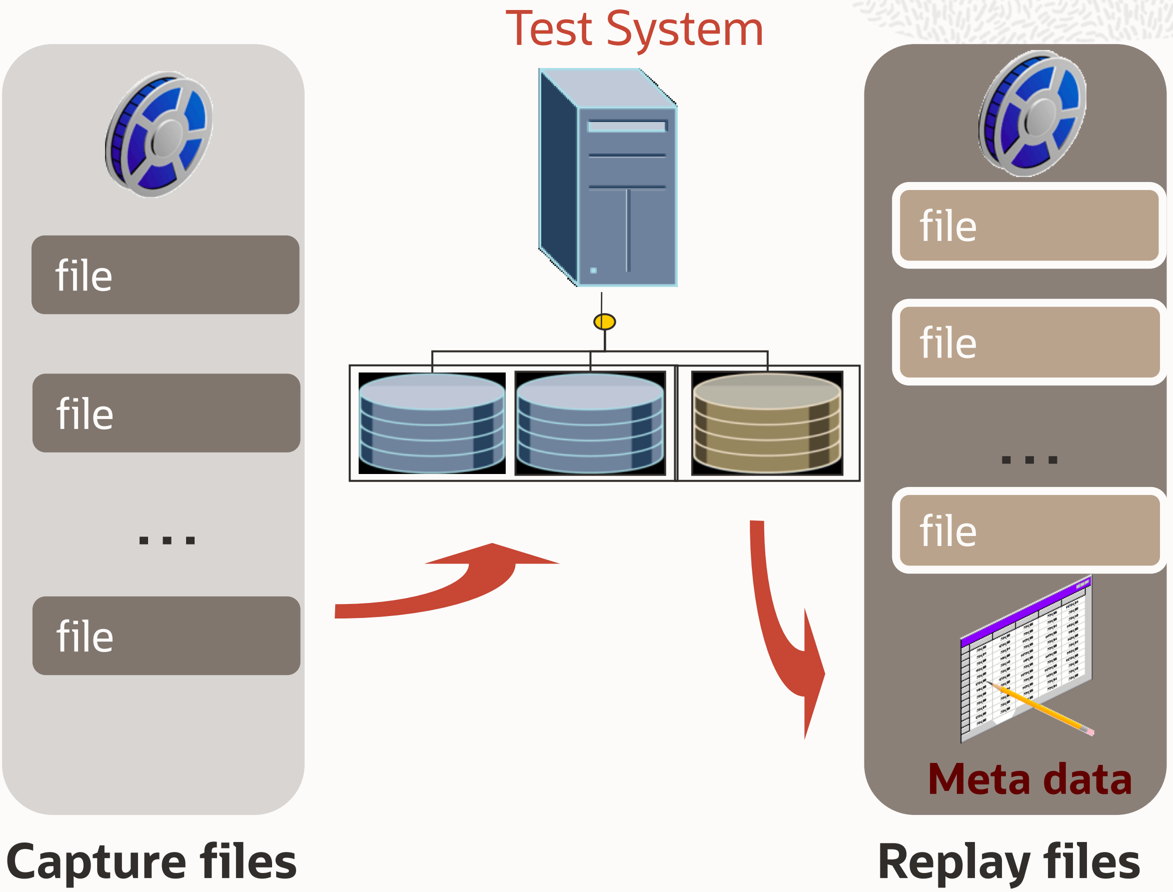


Linemode usage : Capture

```
-- ADD filters (optional)
exec dbms_workload_capture.add_filter('M_OMS', 'Program', 'OMS');
exec dbms_workload_capture.add_filter('M_OMAGENT', 'Program', 'emagent%');
exec dbms_workload_capture.add_filter('M_DBSNMP', 'User', 'DBSNMP');
exec dbms_workload_capture.add_filter('M_RMAN', 'Module', 'rman%');


-- CAPTURE start with default INCLUDE without duration
execute dbms_workload_capture.start_capture (name=>'&capturename',
                                             dir=>'&dir',
                                             plsql_mode => 'extended');
-- CAPTURE stop
execute dbms_workload_capture.finish_capture();
```

Step 2: Setup the test system and process capture files

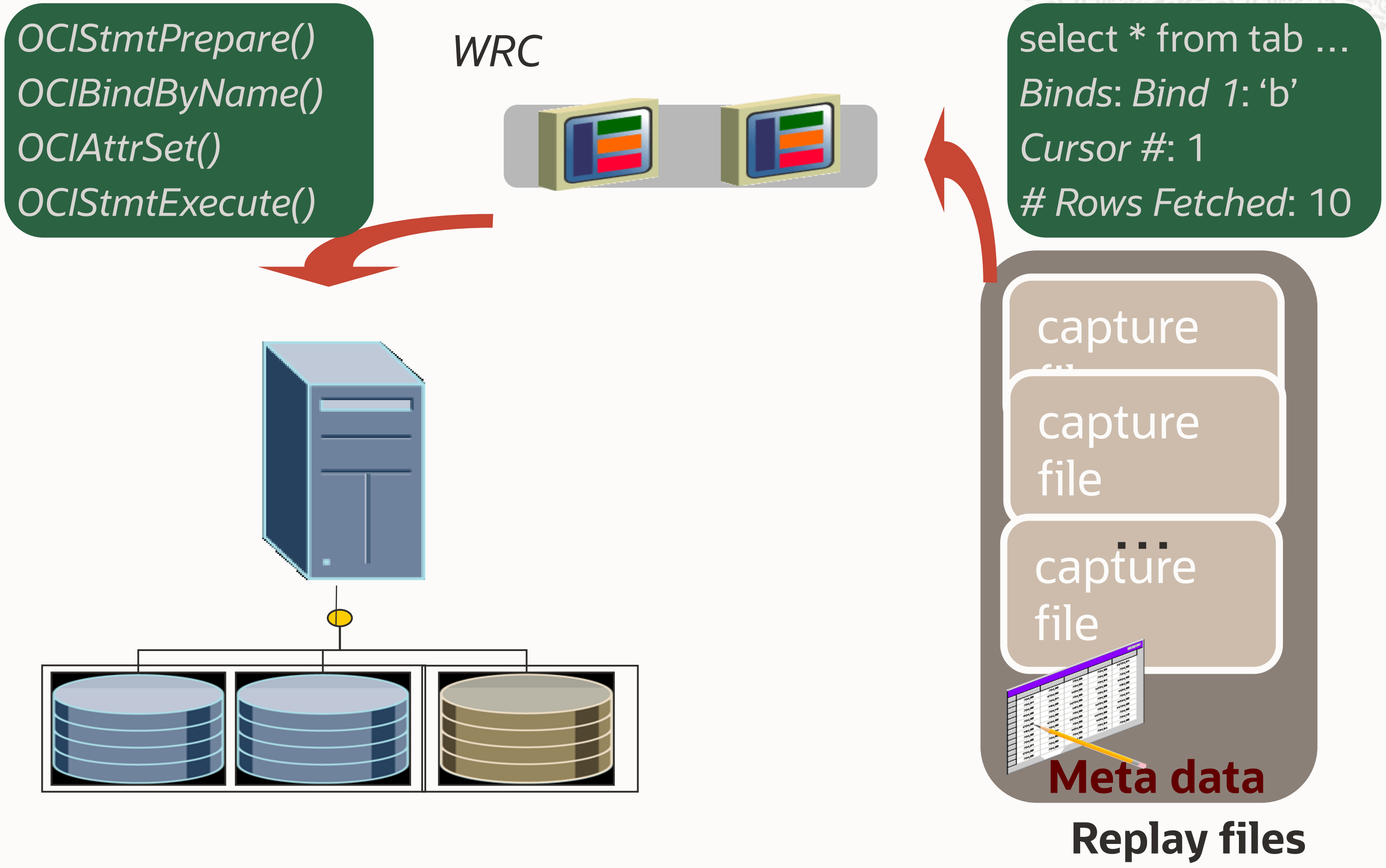
- First setup the test system
  - Use RMAN, Snapshot Standby, imp/exp, Data Pump, etc. to create test system
  - Make change: upgrade db and/or OS, change storage, migrate platforms, etc.
  - Resolve references to external system e.g. db links, utl_http, utl_ftp
- Processing transforms captured data into replayable format
  - Once processed, workload can be replayed many times



Step 3: Replay Workload

- Replay captured workload
  - Replayed operations see the same data and perform the same work
  - Preserve timing and concurrency characteristics
  - Same number of user connections
  - Filtering is possible
- Different replay modes/options
- Use Workload Replay Clients (WRC) for reading & replaying the files
  - Multithreaded OCI Clients that drives multiple captured processes

Test System



OCIStmtPrepare()
OCIBindByName()
OCIAttrSet()
OCIStmtExecute()

WRC

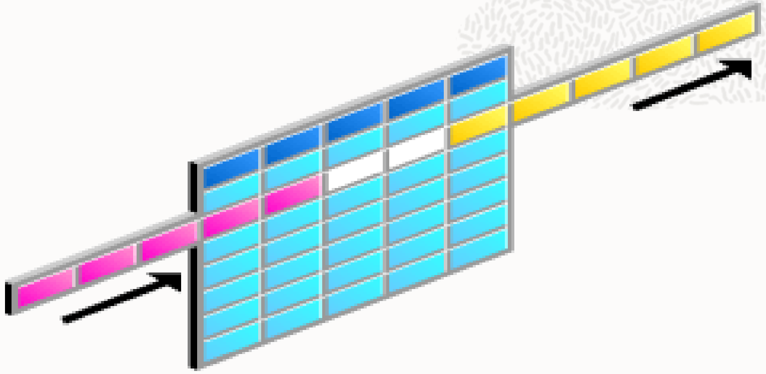
select * from tab …
Binds: *Bind 1*: ‘b’
Cursor #: 1
# Rows Fetched: 10

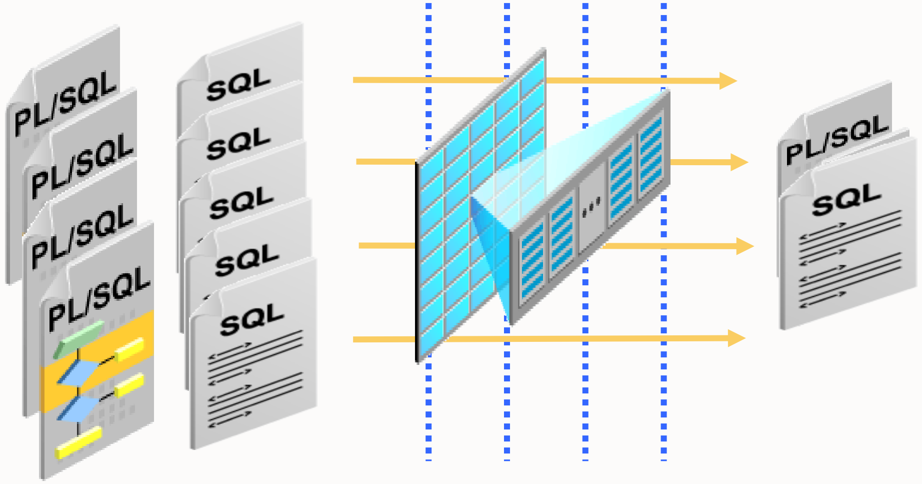
capture
capture file
…
capture file
Meta data

Replay files

Replay Setup

- Replay Parameters:
  - Commit order synchronization: SYNCHRONIZATION (TIME, SCN(default))
  - Think time synchronisation: THINK_TIME_SCALE
  - Connect (Logon) time synchronisation: CONNECT_TIME_SCALE
- Connection Remapping
- Number of WRCs
- Replay filters
  - Helps target workload for replay
  - Workload that is filtered is skipped

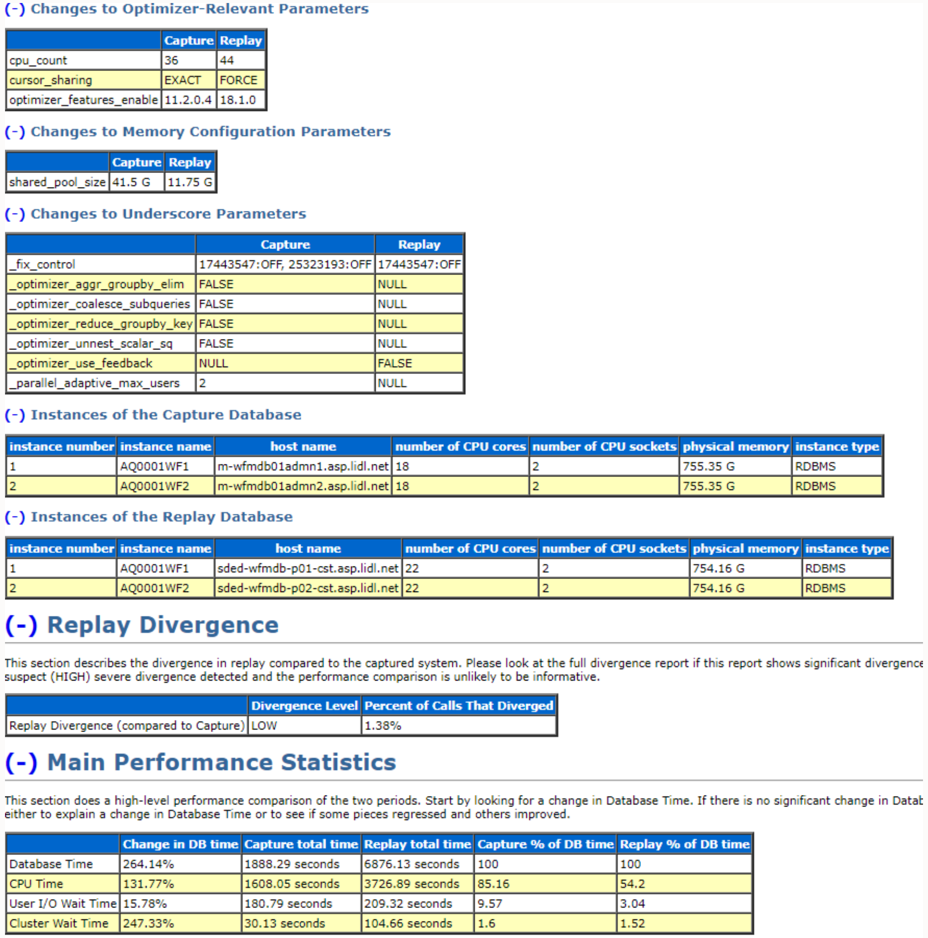




Replay Reports

3 Types of divergences are reported:

- Error Divergence
  For each call error divergence is reported:
  New, Mutated, Not Found
- **Performance Divergence**
  Replay report, Compare period report, AWR, ASH, ADDM
- Data Divergence
  #of rows returned by each call are compared



(-) Changes to Optimizer-Relevant Parameters

	Capture	Replay
cpu_count	36	44
cursor_sharing	EXACT	FORCE
optimizer_features_enable	11.2.0.4	18.1.0

(-) Changes to Memory Configuration Parameters

	Capture	Replay
shared_pool_size	41.5 G	11.75 G

(-) Changes to Underscore Parameters

	Capture	Replay
_fix_control	17443547:OFF, 25323193:OFF	17443547:OFF
_optimizer_aggr_groupby_elim	FALSE	NULL
_optimizer_coalesce_subqueries	FALSE	NULL
_optimizer_reduce_groupby_key	FALSE	NULL
_optimizer_unnest_scalar_sq	FALSE	NULL
_optimizer_use_feedback	NULL	FALSE
_parallel_adaptive_max_users	2	NULL

(-) Instances of the Capture Database

instance number	instance name	host name	number of CPU cores	number of CPU sockets	physical memory	instance type
1	AQ0001WF1	m-wfmdb01admn1.asp.lidl.net	18	2	755.35 G	RDBMS
2	AQ0001WF2	m-wfmdb01admn2.asp.lidl.net	18	2	755.35 G	RDBMS

(-) Instances of the Replay Database

instance number	instance name	host name	number of CPU cores	number of CPU sockets	physical memory	instance type
1	AQ0001WF1	sded-wfmdb-p01-cst.asp.lidl.net	22	2	754.16 G	RDBMS
2	AQ0001WF2	sded-wfmdb-p02-cst.asp.lidl.net	22	2	754.16 G	RDBMS

(-) Replay Divergence

This section describes the divergence in replay compared to the captured system. Please look at the full divergence report if this report shows significant divergence
suspect (HIGH) severe divergence detected and the performance comparison is unlikely to be informative.

	Divergence Level	Percent of Calls That Diverged
Replay Divergence (compared to Capture)	LOW	1.38%

(-) Main Performance Statistics

This section does a high-level performance comparison of the two periods. Start by looking for a change in Database Time. If there is no significant change in Datab
either to explain a change in Database Time or to see if some pieces regressed and others improved.

	Change in DB time	Capture total time	Replay total time	Capture % of DB time	Replay % of DB time
Database Time	264.14%	1888.29 seconds	6876.13 seconds	100	100
CPU Time	131.77%	1608.05 seconds	3726.89 seconds	85.16	54.2
User I/O Wait Time	15.78%	180.79 seconds	209.32 seconds	9.57	3.04
Cluster Wait Time	247.33%	30.13 seconds	104.66 seconds	1.6	1.52

Challenges

- Set expectations
  - Provide a clear objective for your testing
  - Provide success criteria
  - Provide a test plan
  - Decide on the capture duration with or without database start
- Setup the test environment
  - Should be as close as possible to the start capture SCN
  - Size database replay servers similar to production (CPU, Memory, Network bandwith)
  - Methods: Clone, Standby, Export/Import, Backup, Flashback
  - Resolve references to external systems such as database links, web services, flat files access, but also email notifications etc. => Isolation!

Challenges

- Who will carry out the tests?
  - Available resources for the tests
  - Required knowledge/skills: real application testing, application knowledge, environment setup
  - Involve additional experts (consulting, ASC, partners ...)
- Provide a time plan
  - Include learning curve
  - Start learning on a small database/workload
  - Get to know the application (peak time, batch jobs, distinct workloads etc)
  - Setup test environment
  - Reserve time for several replays, the analysis AND for the tuning tasks
  - Test one change at a time

Test governance

Documentation of tests

- Write storyboard for each test conditions such as configuration, changes, lessons learned
- Document your results e.g. performance from AWR, alert files, log files etc
- High level documentation/summary
- Proven changes => to do list for going live
- Each test cycle: export results e.g. baselines, SQL profiles, AWR reports

Prepare for repeating replays: flashback database, backups

- Sufficient space for flashback archive logs
- Check retention time for backups

Real Application Testing customer projects feedback - quotes

Automotive Company

“Error-free Database Replay became **customer major acceptance criterion** for Oracle upgrades !

Telecommunication

“**Exadata** was sized, ordered, configured based on the test results and the migration was completed successfully.”

Trading

“We **look positively at the upgrade**”
“Thanks to the team – great team engagement”

Entertainment

“Important to get confidence in **migration to Autonomous Database**”
“Well suited as an argument for our management”

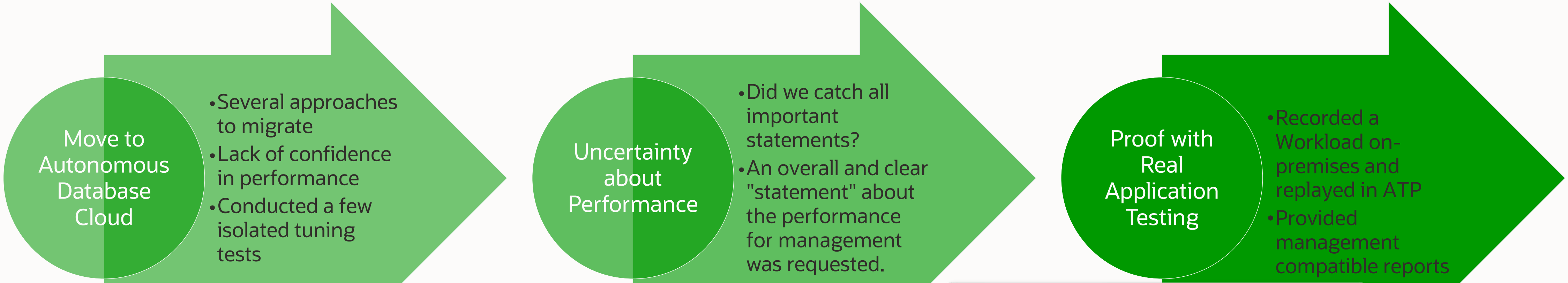
Smooth transition to Autonomous Database using SPA

Ulrike Schwinn | 13 minute read

Further reading

- Documentation Testing Guide: https://docs.oracle.com/en/database/oracle/oracle-database/19/ratug/index.html
- My Oracle Support (MOS) notes
  - Primary Note for Real Application Testing Option (Doc ID 1464274.1)
  - Using Workload Capture and Replay (Doc ID 445116.1)
  - Mandatory Patches for Database Testing Functionality for Current and Earlier Releases (Doc ID 560977.1)
  - Database Testing: Best Practices (Doc ID 1535885.1)
- Postings
  - SQL Tuning Sets:
    Oracle SQL Tuning Sets - Grundlage zum SQL Tuning
  - SQL Performance Analyzer
    SPA mal anders - SQL-Tuning mit SQL Performance Analyzer (SPA)
    Smooth transition to Autonomous Database using SPA
  - Database Replay
    Testing with Oracle Database Replay
    Real Application Testing Database Replay Demo
- Upgrade your Database Now! Mike Dietrich: https://mikedietrichde.com/

Customer case



Move to Autonomous Database Cloud

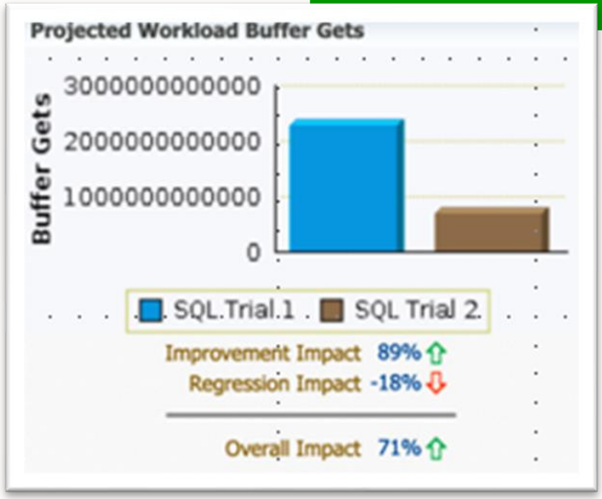
- Several approaches to migrate
- Lack of confidence in performance
- Conducted a few isolated tuning tests

Uncertainty about Performance

- Did we catch all important statements?
- An overall and clear "statement" about the performance for management was requested.

Proof with Real Application Testing

- Recorded a Workload on-premises and replayed in ATP
- Provided management compatible reports



https://blogs.oracle.com/coretec/post/spa-und-autonomousdatabase

Customer case

Reasons for use

- RAT for testing an application/Complex legacy architecture with operational and cost challenges
- DB parameter changes
- OS parameter changes
- Oracle One Off patches
- Table architecture changes (e.g. partitioning)

Implementation

- Capture: 1 hour (approx. 8 million user calls)
- Replay: 2 hours - due to the high I/O times
- Statement analysis and use of SQL Performance Analyzer

Results

- Reduction of the runtime of SQL baseline statements by approx. 90% !

German chemical company with SAP environment

Reasons for use

- POC for platform change to Exadata
- Load test

Challenges

- Complexity: SAP system, Exadata and RAT
- No shutdown or filtering possible
- Long duration of replay to be expected

Key data for implementation

- Capture duration: 2 hours with approx. 2700 sessions/600 million calls
- Capture size: approx. 300 GB
- Replay: 2,5 hours

Customer Challenge (Logistics Company)

- *“It is not possible to simulate the production workload in our test database.*
  - *Even when using synthetic workloads, it fails to mimic the production workload in terms of concurrency of the workload mix.”*

Our answer: DB Replay

- Capture the workload from production and Replay the same in the test environment.

- *“The inability to test the impact of some changes (e.g. Optimizer/Parameter changes) without actually making the changes. It is also not possible to find all or most SQL statements that have regressed due to the changes.”*

Our answer: SPA

- Validate system changes even on production databases without actually impacting the end users.
- Automate and simplify the manual and time-consuming assessment of extremely large SQL workloads.
- Automate the remediation of any SQL regressions as a result of the system change.

Further Readings

Documentation: Testing Guide
Blogs:

- **Testing with Oracle Database Replay**
- **Real Application Testing Database Replay Demo**
- **Autonomous Database Replay**
- **Smooth transition to Autonomous Database using SPA**

MOS Notes

- Primary Note for Real Application Testing Option (Doc ID 1464274.1)
- FAQ: Database Upgrade Using Real Application Testing (Doc ID 1600574.1)
- Database Testing: Best Practices (Doc ID 1535885.1)
- Real Application Testing: Database Capture FAQ (Doc ID 1920275.1)