ORACLE®

ORACLE®

ORACLE OPEN WORLD

Step-by-Step Cookbook for Identifying and Tuning SQL Problems

Ashish Agrawal - Consulting Product Manager, Oracle

Baki Şahin- Database Operation Supervisor, AveA Turkey

HARDWARE AND SOFTWARE ENGINEERED TO WORK TOGETHER

Safe Harbor Statement

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.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle.   This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Program Agenda

- Why SQL statements regress?
- Identifying problematic SQL
- Tuning SQL
- Preventing SQL problems
- Real-World Customer Experiences

    - AveA, Turkey

    - S. Corporation, Korea

Program Agenda

- Why SQL statements regress?
- Identifying problematic SQL
- Tuning SQL
- Preventing SQL problems
- Real-World Customer Experiences

  - AveA, Turkey

  - S. Corporation, Korea

ORACLE

Why SQL statements regress?

3 Broad Categories

Optimizer

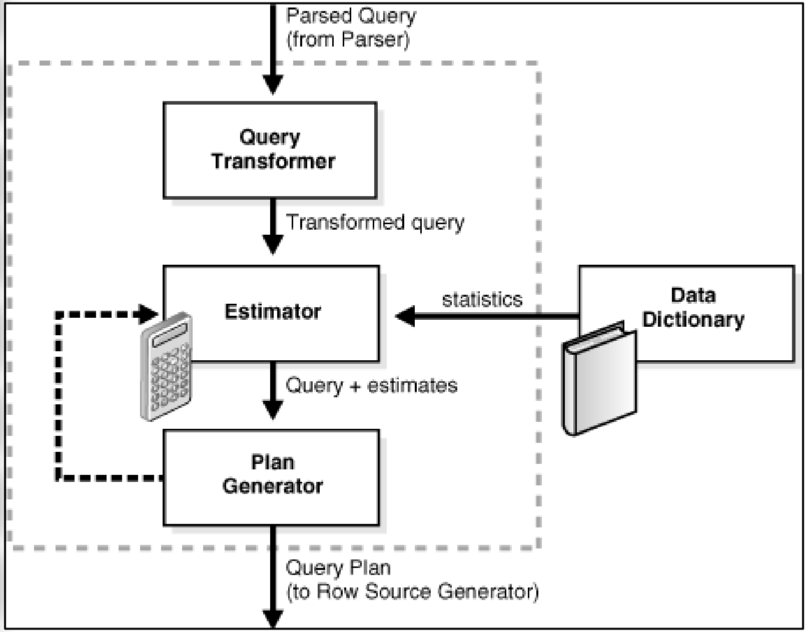
Application

Resource and contention Issues

Why SQL statements regress?

Optimizer-related

- Stale/Missing statistics
- Overly general statistics or incorrect histograms
- Improper optimizer configuration
- Upgraded database: new optimizer
- Changing statistics (refresh)
- Changing data (plans do not scale with data)
- Bind-sensitive SQL with bind peeking
- Not parallelized (no scaling to large data)
- Improperly parallelized (skews, RAC, etc.)



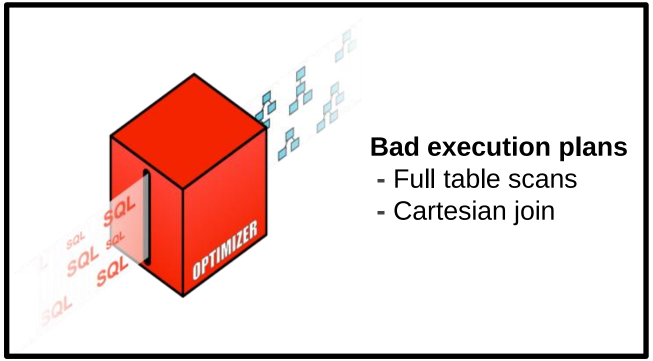
Why SQL statements regress?

Application-related

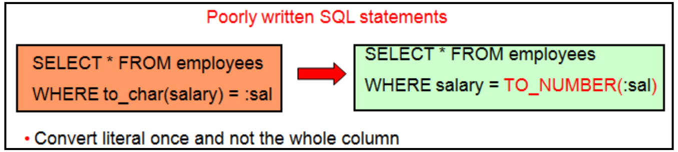
Missing access structures

Poorly written SQL statements

Literal usage



Bad execution plans
- Full table scans
- Cartesian join

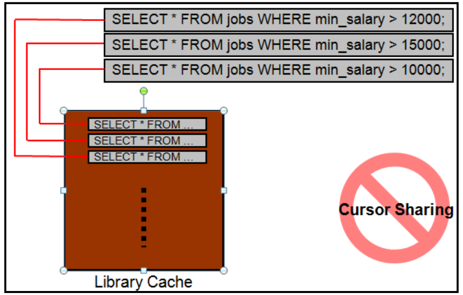


Poorly written SQL statements

```
SELECT * FROM employees
WHERE to_char(salary) = :sal
```

```
SELECT * FROM employees
WHERE salary = TO_NUMBER(:sal)
```

- Convert literal once and not the whole column



```
SELECT * FROM jobs WHERE min_salary > 12000;
SELECT * FROM jobs WHERE min_salary > 15000;
SELECT * FROM jobs WHERE min_salary > 10000;
```

Why SQL statements regress?

Resource and Contention Issues-related

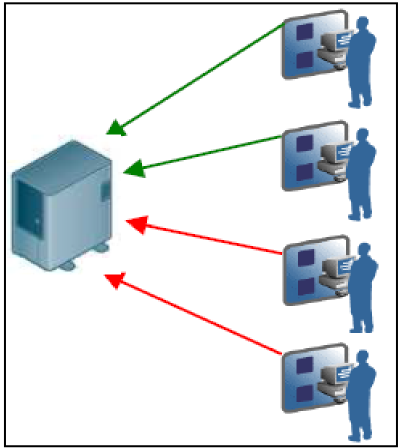
Hardware resource crunch

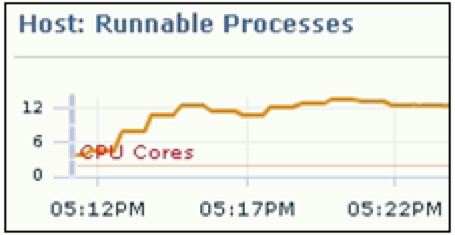
- CPU, Memory, IO, Network

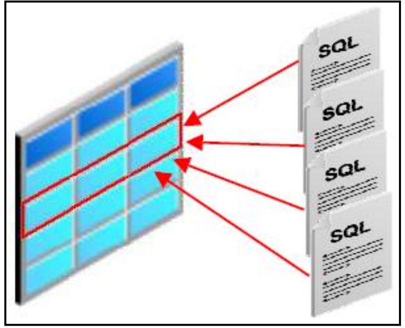
Data fragmentation

Logical Contention

- Row lock contention
- Block update contention







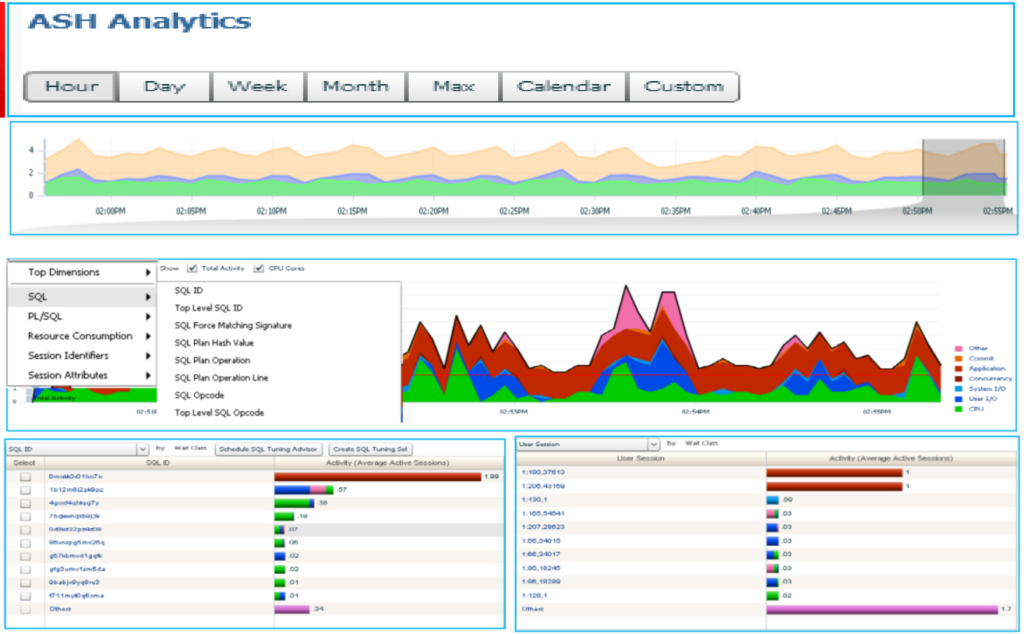
Example:-
enq: TX - allocate ITL entry
enq: TX - contention
enq: TX - index contention
enq: US - contention

Sub-optimal SQL performance: Symptoms

Symptoms

Consumes high CPU, buffer gets, I/O, PGA memory

- Long running SQL or significantly different runtimes
- High I/O, CPU, memory, network waits
- TX Enqueue Waits, Row Lock Contention
- Plan regression
- SQL appear in
  - Top Activity Page
  - ASH Analytics Page
  - ADDM Report, AWR Report, ASH reports



Select	SQL ID	Activity (Average Active Sessions)
☐	8mnkk3r01hn7n	1.99
☐	1b12m8j2pk9pz	.57
☐	4gusf4qhkyg7y	.38
☐	7bqkwnqsb8j3k	.19
☐	0d8kz32pz4z08	.07
☐	86vncpg5mv26q	.06
☐	g57kbmvd1gqfk	.02
☐	gfg3umv1sm5da	.02
☐	9babjv8yq8ru3	.01
☐	f711myt0q6cma	.01
☐	Others	.34

User Session	Activity (Average Active Sessions)
1:190,37613	1
1:206,43169	1
1:130,1	.09
1:165,54641	.03
1:207,28823	.03
1:66,34015	.03
1:68,34017	.03
1:86,18245	.03
1:98,18289	.03
1:126,1	.02
Others	1.7

Program Agenda

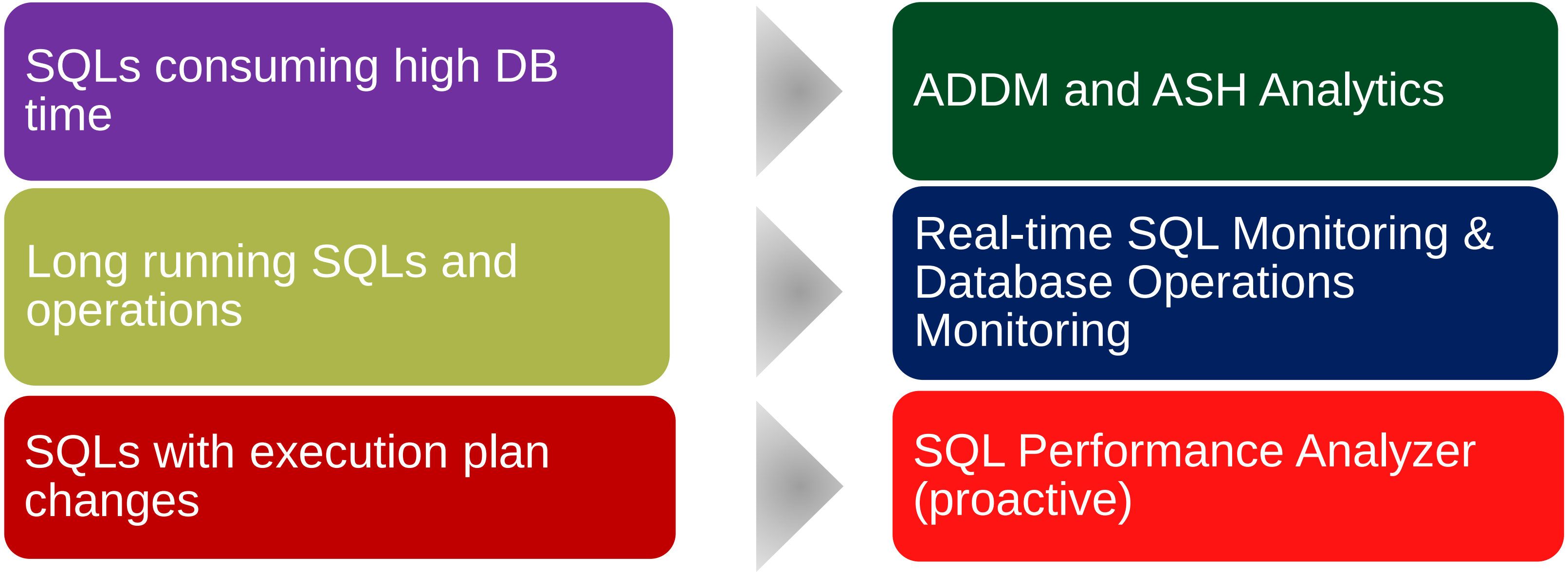
- Why SQL statements regress?
- Identifying problematic SQL
- Tuning SQL
- Preventing SQL problems
- Real-World Customer Experiences

  - AveA, Turkey

  - S. Corporation, Korea

How to identify these SQL performance problems?

Identify SQL performance problems using:



SQLs consuming high DB time

ADDM and ASH Analytics

Long running SQLs and operations

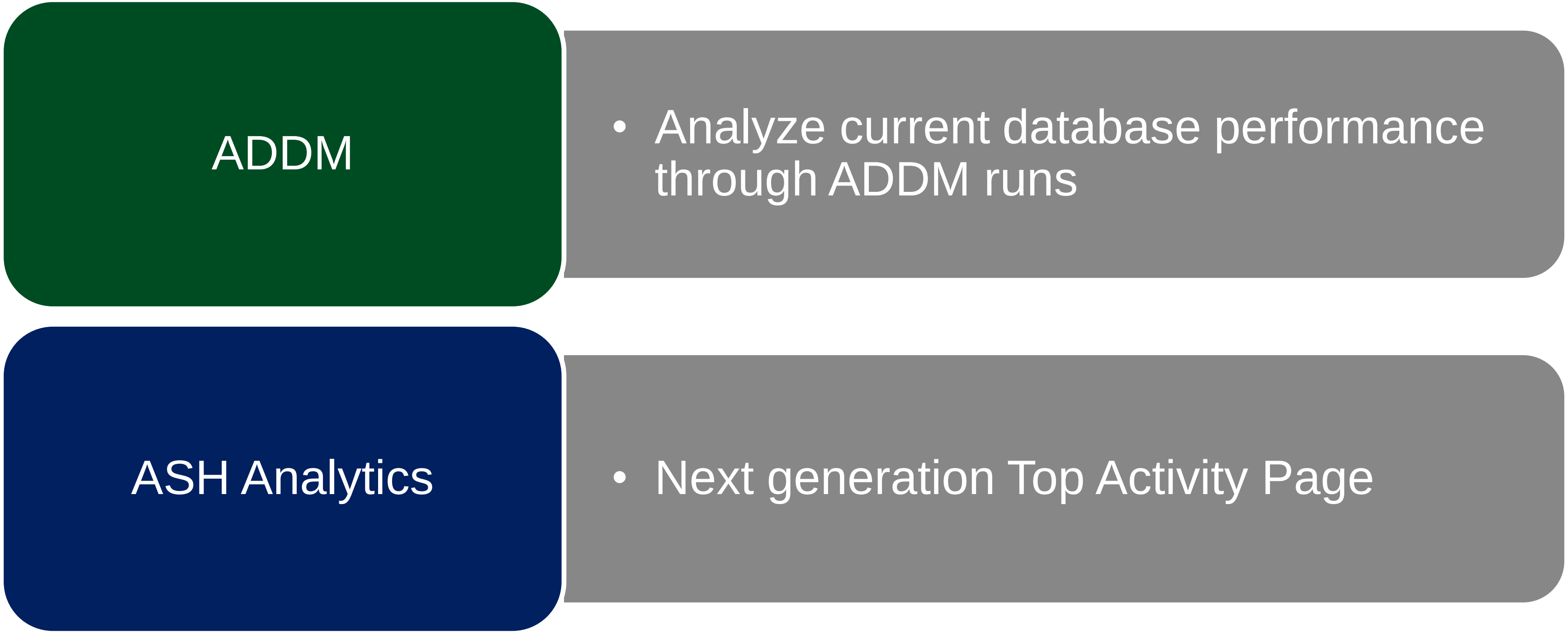
Real-time SQL Monitoring & Database Operations Monitoring

SQLs with execution plan changes

SQL Performance Analyzer (proactive)

Identify expensive SQL (Excessive DB time):

ADDM and ASH Analytics



ADDM

- Analyze current database performance through ADDM runs

ASH Analytics

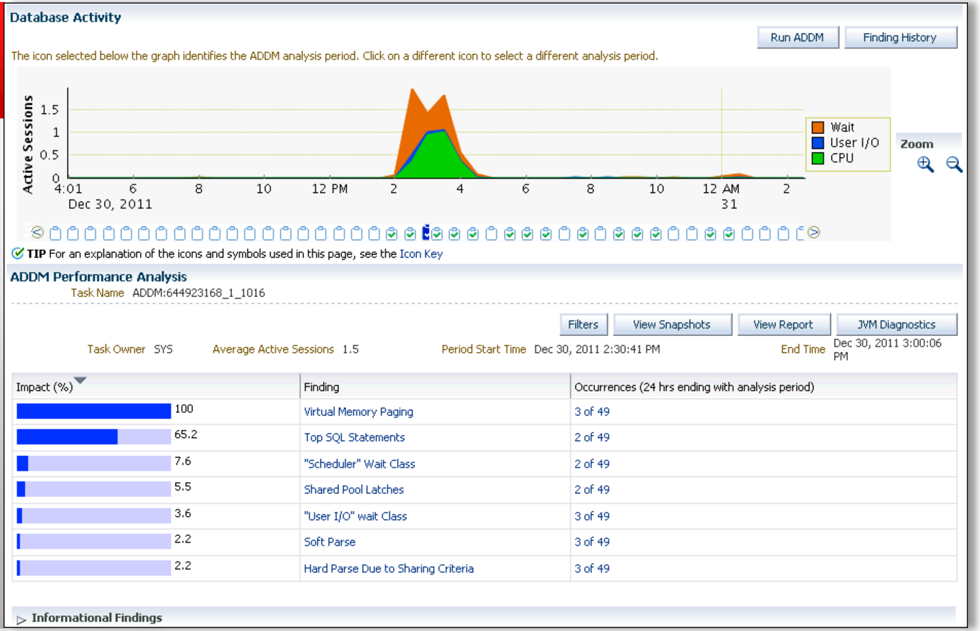
- Next generation Top Activity Page

Identify expensive SQL: ADDM

SQL consuming too much DB time

ADDM

- Throughput centric: Goal is to reduce ‘DB time’
- Identifies top SQL
- Shows SQL impact
- Frequency of occurrence
- Pinpoints root cause



Database Activity

The icon selected below the graph identifies the ADDM analysis period. Click on a different icon to select a different analysis period.

✓ **TIP** For an explanation of the icons and symbols used in this page, see the Icon Key

ADDM Performance Analysis

Task Name ADDM:644923168_1_1016

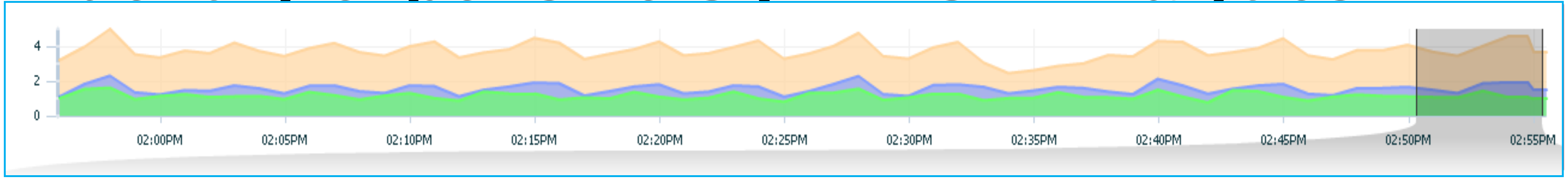
Task Owner SYS Average Active Sessions 1.5 Period Start Time Dec 30, 2011 2:30:41 PM End Time Dec 30, 2011 3:00:06 PM

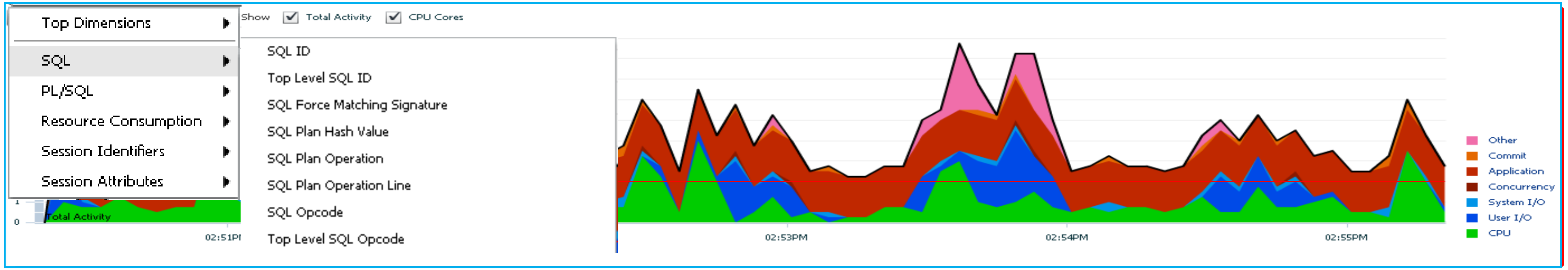
Impact (%)	Finding	Occurrences (24 hrs ending with analysis period)
100	Virtual Memory Paging	3 of 49
65.2	Top SQL Statements	2 of 49
7.6	"Scheduler" Wait Class	2 of 49
5.5	Shared Pool Latches	2 of 49
3.6	"User I/O" wait Class	3 of 49
2.2	Soft Parse	3 of 49
2.2	Hard Parse Due to Sharing Criteria	3 of 49

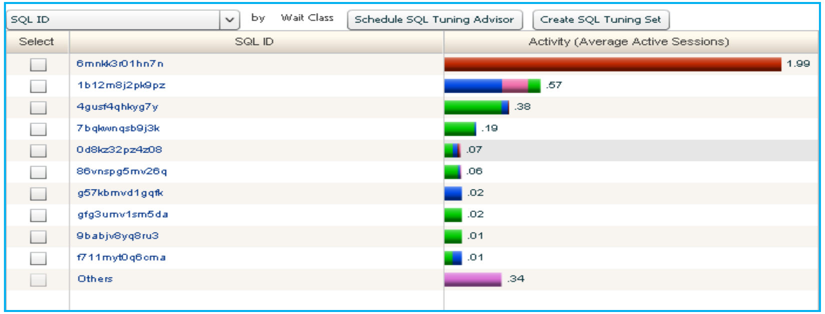
▷ **Informational Findings**

ORACLE®

Identify expensive SQL: ASH Analytics

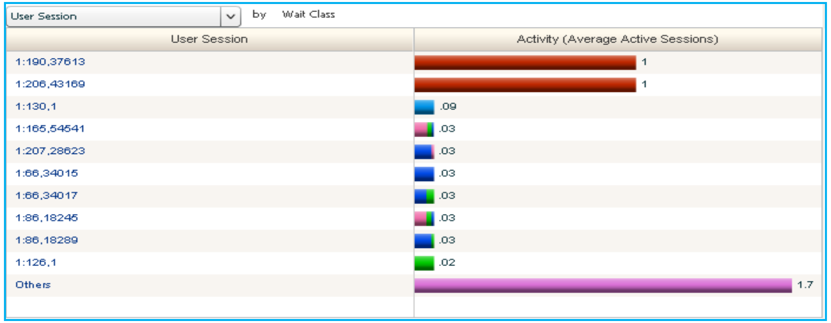






SQL ID by Wait Class | Schedule SQL Tuning Advisor | Create SQL Tuning Set

Select	SQL ID	Activity (Average Active Sessions)
☐	6mnkk3r01hn7n	1.99
☐	1b12m8j2pk9pz	.57
☐	4gusf4qhkyg7y	.38
☐	7bqkwnqsb9j3k	.19
☐	0d8kz32pz4z08	.07
☐	86vnspg5mv26q	.06
☐	g57kbmvd1gqfk	.02
☐	gfg3umv1sm5da	.02
☐	9babjv8yq8ru3	.01
☐	f711myt0q6cma	.01
☐	Others	.34



User Session by Wait Class

User Session	Activity (Average Active Sessions)
1:190,37613	1
1:206,43169	1
1:130,1	.09
1:165,54541	.03
1:207,28623	.03
1:66,34015	.03
1:66,34017	.03
1:86,18245	.03
1:86,18289	.03
1:126,1	.02
Others	1.7

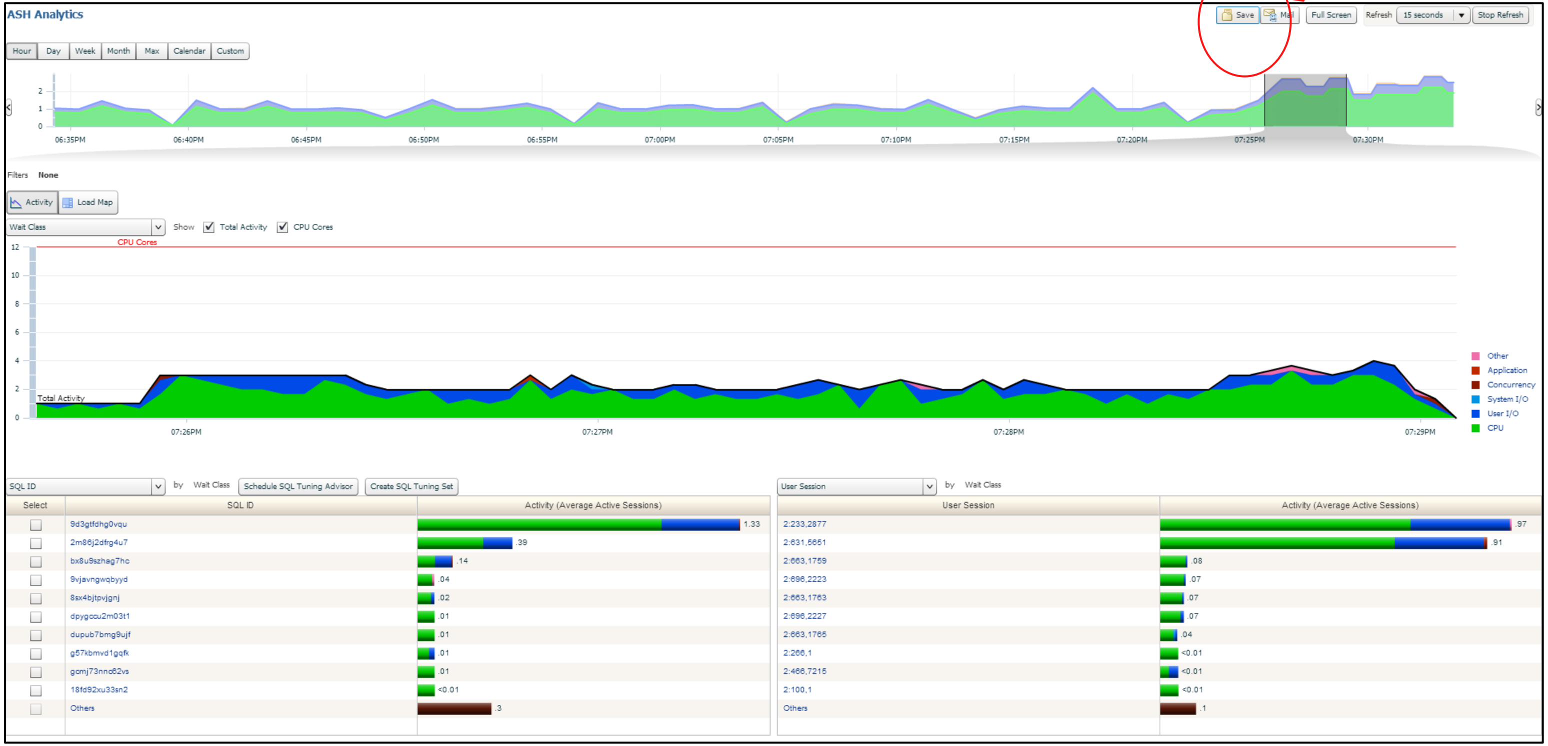
ASH Analytics

Active Reports Demo

Identify expensive SQL: ASH Analytics

Active Reports Demo

Click on Save



Filters None

Wait Class Show ☑ Total Activity ☑ CPU Cores

SQL ID by Wait Class | Schedule SQL Tuning Advisor | Create SQL Tuning Set

Select	SQL ID	Activity (Average Active Sessions)
□	9d3gtfdhg0vqu	1.33
□	2m86j2dfrg4u7	.39
□	bx8u9szhag7hc	.14
□	9vjavngwqbyyd	.04
□	8sx4bjtpvjgnj	.02
□	dpygccu2m03t1	.01
□	dupub7bmg9ujf	.01
□	g57kbmvd1gqfk	.01
□	gcmj73nnc82vs	.01
□	18fd92xu33sn2	<0.01
□	Others	.3

User Session by Wait Class

User Session	Activity (Average Active Sessions)
2:233,2877	.97
2:631,5651	.91
2:663,1759	.08
2:696,2223	.07
2:663,1763	.07
2:696,2227	.07
2:663,1765	.04
2:266,1	<0.01
2:466,7215	<0.01
2:100,1	<0.01
Others	.1

Identify long running SQL:

Real-Time SQL Monitoring

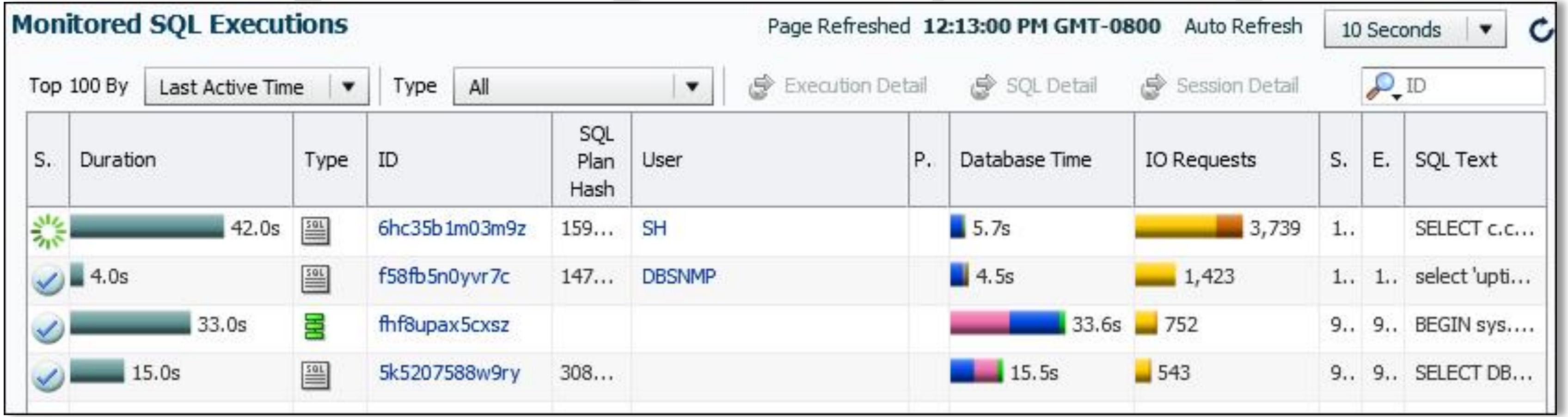
Automatically monitors instances of long running SQL, PL/SQL executions

Enabled out-of-the-box with no performance overhead

Obviates need to trace individual SQL

Shows global PL/SQL and SQL level statistics

Guides tuning efforts



Monitored SQL Executions

Page Refreshed **12:13:00 PM GMT-0800** Auto Refresh 10 Seconds

Top 100 By Last Active Time Type All

S.	Duration	Type	ID	SQL Plan Hash	User	P.	Database Time	IO Requests	S.	E.	SQL Text
	42.0s	SQL	6hc35b1m03m9z	159...	SH		5.7s	3,739	1..		SELECT c.c...
✓	4.0s	SQL	f58fb5n0yvr7c	147...	DBSNMP		4.5s	1,423	1..	1..	select 'upti...
✓	33.0s		fhf8upax5cxsz				33.6s	752	9..	9..	BEGIN sys....
✓	15.0s	SQL	5k5207588w9ry	308...			15.5s	543	9..	9..	SELECT DB...

Identify long running Database operations:

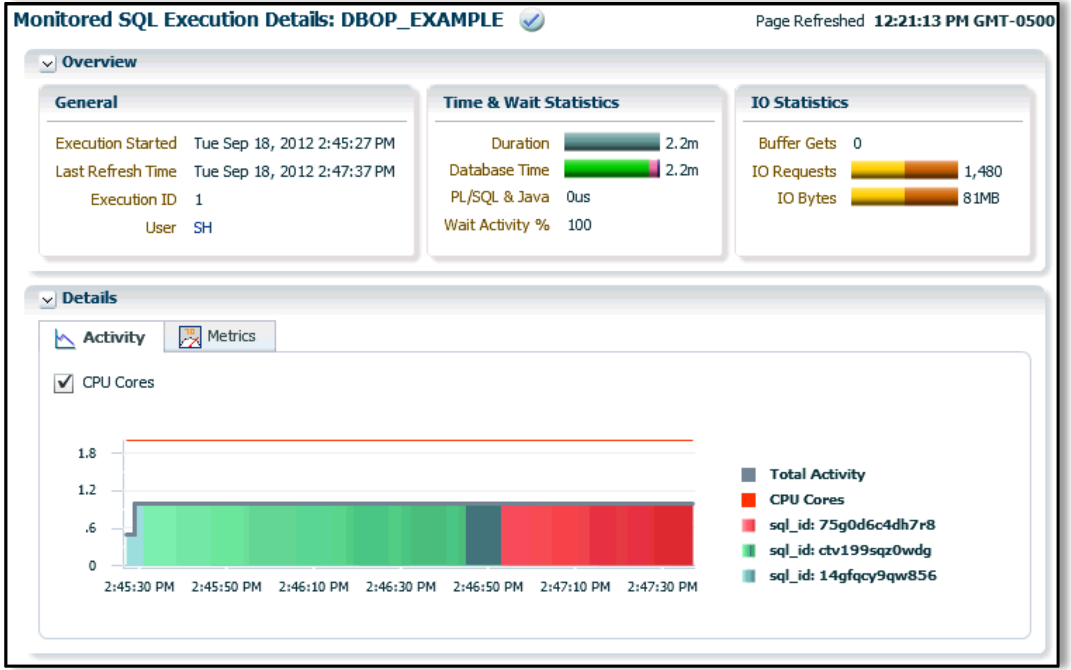
Real-Time Database Operations Monitoring New in ORACLE DATABASE 12c

Database monitoring of application jobs

- Grouping of SQLs, sessions for the application jobs
- Key scenarios: ETL operations, Quarter End Close jobs

Driven by application specified tagging

- Oracle Data Pump jobs automatically monitored
- Tagging ability in PL/SQL, OCI, JDBC



Monitored SQL Execution Details: DBOP_EXAMPLE ✓ Page Refreshed 12:21:13 PM GMT-0500

Overview

General

Execution Started	Tue Sep 18, 2012 2:45:27 PM
Last Refresh Time	Tue Sep 18, 2012 2:47:37 PM
Execution ID	1
User	SH

Time & Wait Statistics

Duration	2.2m
Database Time	2.2m
PL/SQL & Java	0us
Wait Activity %	100

IO Statistics

Buffer Gets	0
IO Requests	1,480
IO Bytes	81MB

Details

Activity Metrics

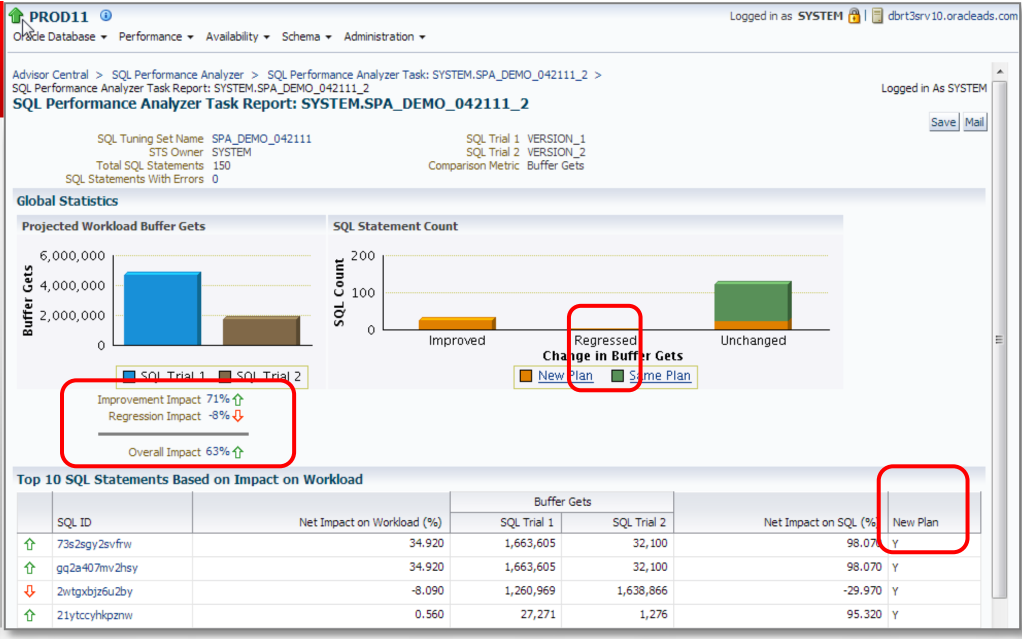
☑ CPU Cores

Visibility into top sql statements

Identify SQL plan changes: SPA

SPA

- Tests and predicts impact of system changes on SQL query performance
- Analyzes overall performance impact including improvements or regressions
- Common plan change scenarios
  - Database parameter, schema changes
  - Statistics gathering refresh
  - I/O subsystem changes, Exadata
  - Database upgrades or patches



PROD11
Logged in as SYSTEM | dbrt3srv10.oracleads.com
Oracle Database ▾ Performance ▾ Availability ▾ Schema ▾ Administration ▾

Advisor Central > SQL Performance Analyzer > SQL Performance Analyzer Task: SYSTEM.SPA_DEMO_042111_2 >
SQL Performance Analyzer Task Report: SYSTEM.SPA_DEMO_042111_2

Logged in As SYSTEM
Save
Mail

SQL Performance Analyzer Task Report: SYSTEM.SPA_DEMO_042111_2

SQL Tuning Set Name	SPA_DEMO_042111	SQL Trial 1	VERSION_1
STS Owner	SYSTEM	SQL Trial 2	VERSION_2
Total SQL Statements	150	Comparison Metric	Buffer Gets
SQL Statements With Errors	0		

Global Statistics

Improvement Impact 71% ⇧
Regression Impact -8% ⇩
Overall Impact 63% ⇧

Top 10 SQL Statements Based on Impact on Workload

	SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	New Plan
			SQL Trial 1	SQL Trial 2		
⇧	73s2sgy2svfrw	34.920	1,663,605	32,100	98.070	Y
⇧	gq2a407mv2hsy	34.920	1,663,605	32,100	98.070	Y
⇩	2wtgxbjz6u2by	-8.090	1,260,969	1,638,866	-29.970	Y
⇧	21ytccyhkpznw	0.560	27,271	1,276	95.320	Y

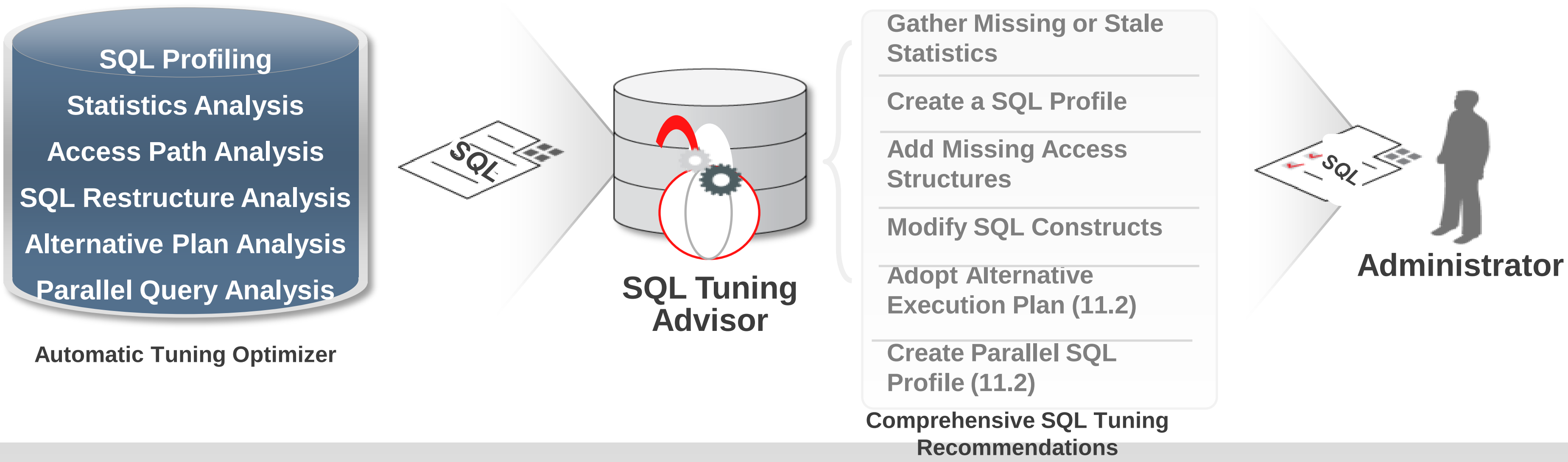
Program Agenda

- Why SQL statements regress?
- Identifying problematic SQL
- Tuning SQL
- Preventing SQL problems
- Real-World Customer Experiences

  - AveA, Turkey

  - S. Corporation, Korea

Tuning SQL: SQL Tuning Advisor



SQL Profiling

Statistics Analysis

Access Path Analysis

SQL Restructure Analysis

Alternative Plan Analysis

Parallel Query Analysis

Automatic Tuning Optimizer

SQL Tuning Advisor

Gather Missing or Stale Statistics

Create a SQL Profile

Add Missing Access Structures

Modify SQL Constructs

Adopt Alternative Execution Plan (11.2)

Create Parallel SQL Profile (11.2)

Comprehensive SQL Tuning Recommendations

Administrator

- Multitenant database-aware
- All SQL’s are tuned across all PDB’s where it has executed before.
- Empowers the CDBA to tune across PDBs in one click

New in ORACLE DATABASE 12c

Tuning SQL: SQL Tuning Advisor

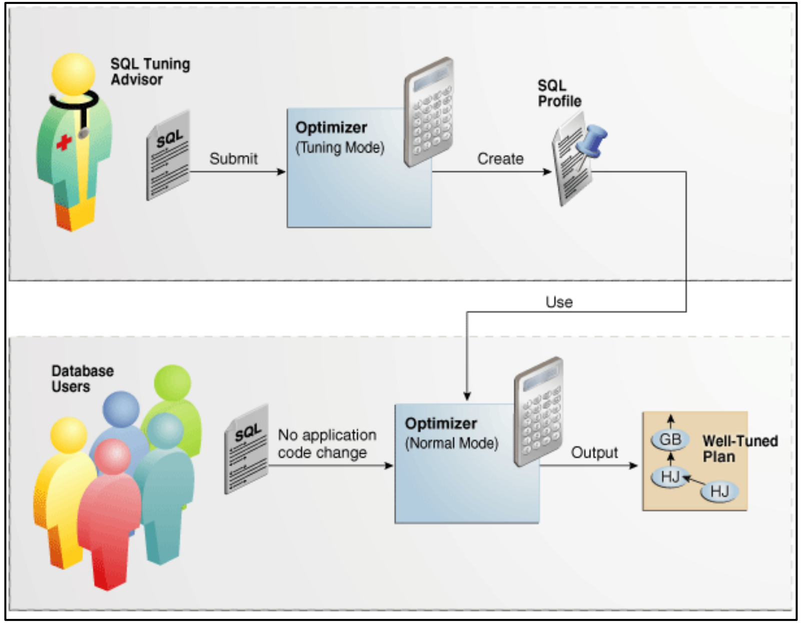
SQL Profiling

Automatic Tuning Optimizer verifies and adjusts its own cardinality estimates

Optimizer provides additional information to generate execution plan

Test executes the recommended execution plan for performance

Reviews execution history



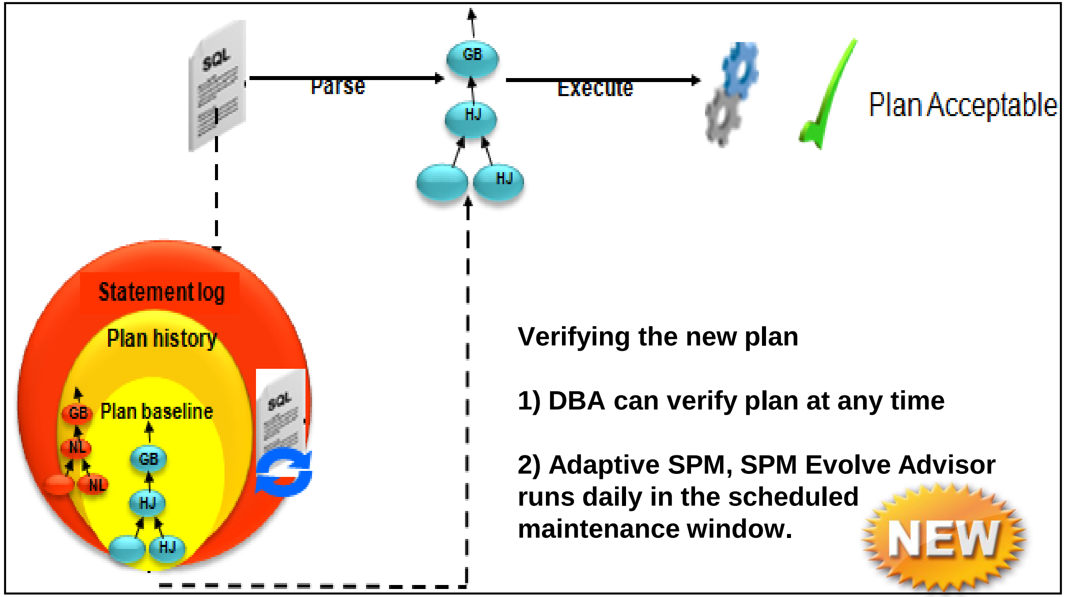
✓Ideal when cardinality estimates are wrong, collection of statistics on objects or creation of new indexes is required

Tuning SQL: SQL Plan Baselines

Some optimizer related change in the environment results in a new plan being generated

New plan is not the same as the baseline – new plan is not executed but marked for verification

Execute known plan baseline - plan performance is “verify by history”



Verifying the new plan

1) DBA can verify plan at any time

2) Adaptive SPM, SPM Evolve Advisor runs daily in the scheduled maintenance window.

✓Ideal when you need to preserve and use good, known and verified execution plans

Tuning SQL: SQL Access Advisor

Recommendations

Indexes

- B-tree indexes
- Bitmap indexes
- Function-based indexes

Materialized views and view logs

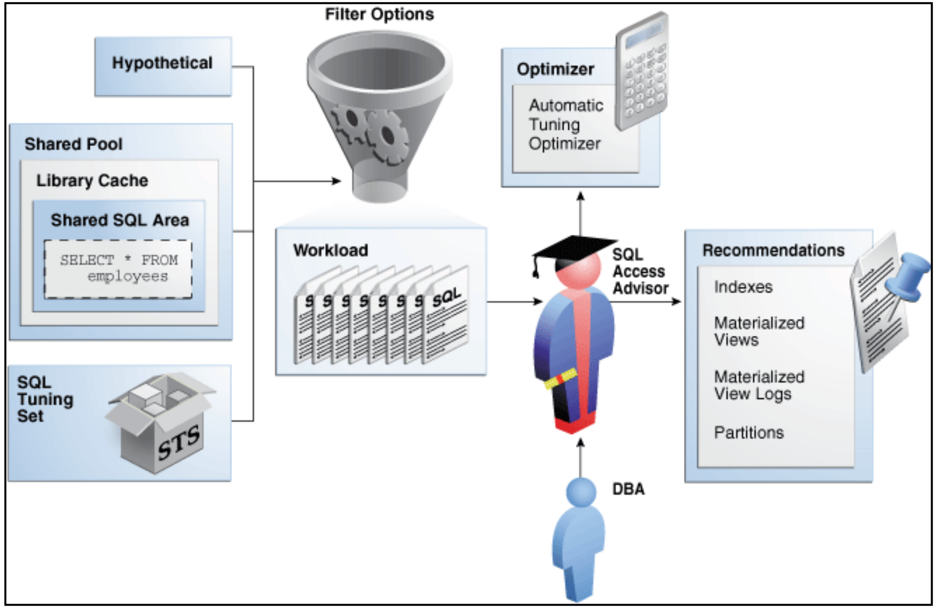
- Fast refreshable
- Full refreshable MVs

Partition table

- Range, Interval, Hash, Range-Hash, Range-List type, List
- New partitioning schemes on already partitioned tables NEW

Partition index

- Local, Range, Hash type



✓Ideal when you need advice for creation of indexes, mviews and partition for your entire workload

Tuning SQL: Real-time SQL Monitoring Case Study

I enabled parallel query, yet this query is taking so long. What’s going on?

Parallel server downgrades?

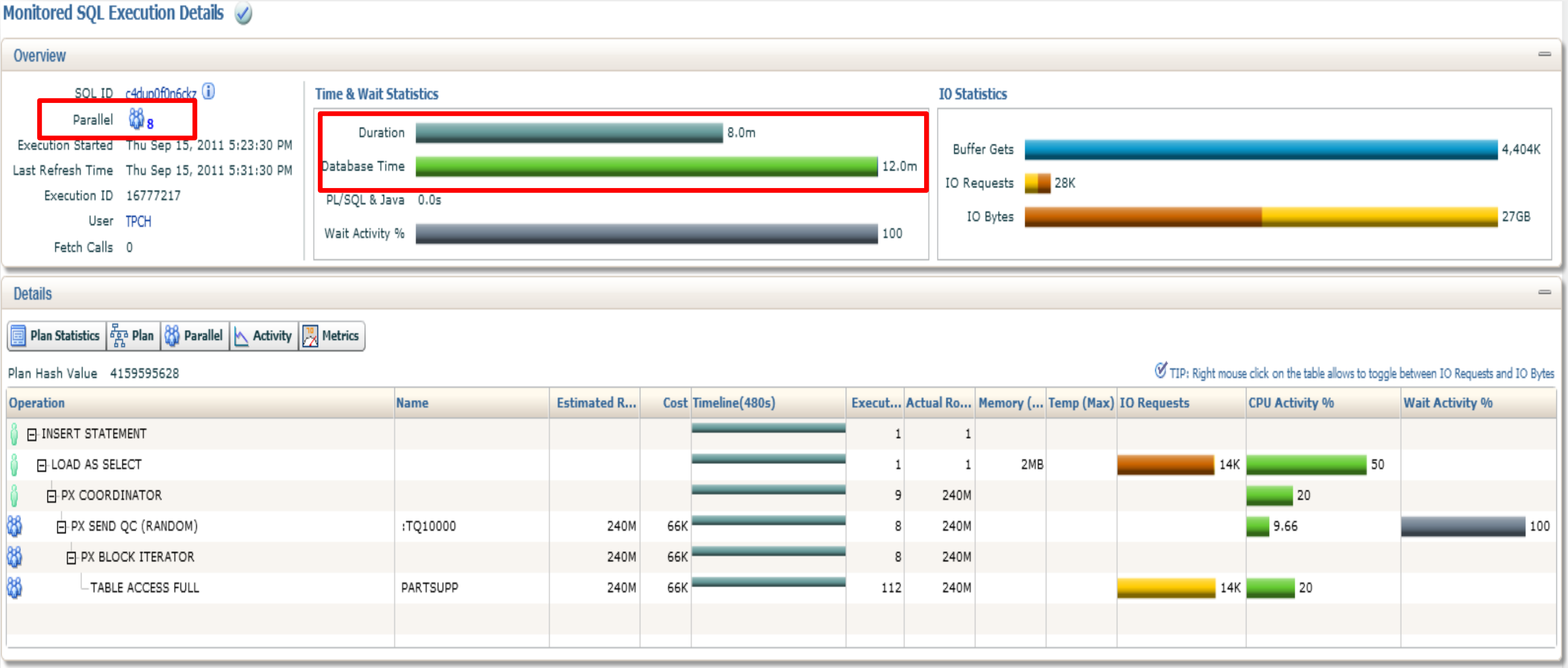
- Uncontrolled parallel execution
- Parallel Server availability
- Object level settings
- Session level settings

✓ Ideal when you need to tune long running SQLs, complex queries with big execution plans, parallel queries, DML and DDL statements, Exadata smart scans, cases of a poor indexing strategies

ORACLE

Tuning SQL: Real-time SQL Monitoring

Insert executed with parallel hint



Monitored SQL Execution Details

Overview

SQL ID c4dup0f0n6ckz

Parallel 8

Execution Started Thu Sep 15, 2011 5:23:30 PM

Last Refresh Time Thu Sep 15, 2011 5:31:30 PM

Execution ID 16777217

User TPCH

Fetch Calls 0

Time & Wait Statistics

Duration 8.0m

Database Time 12.0m

PL/SQL & Java 0.0s

Wait Activity % 100

IO Statistics

Buffer Gets 4,404K

IO Requests 28K

IO Bytes 27GB

Details

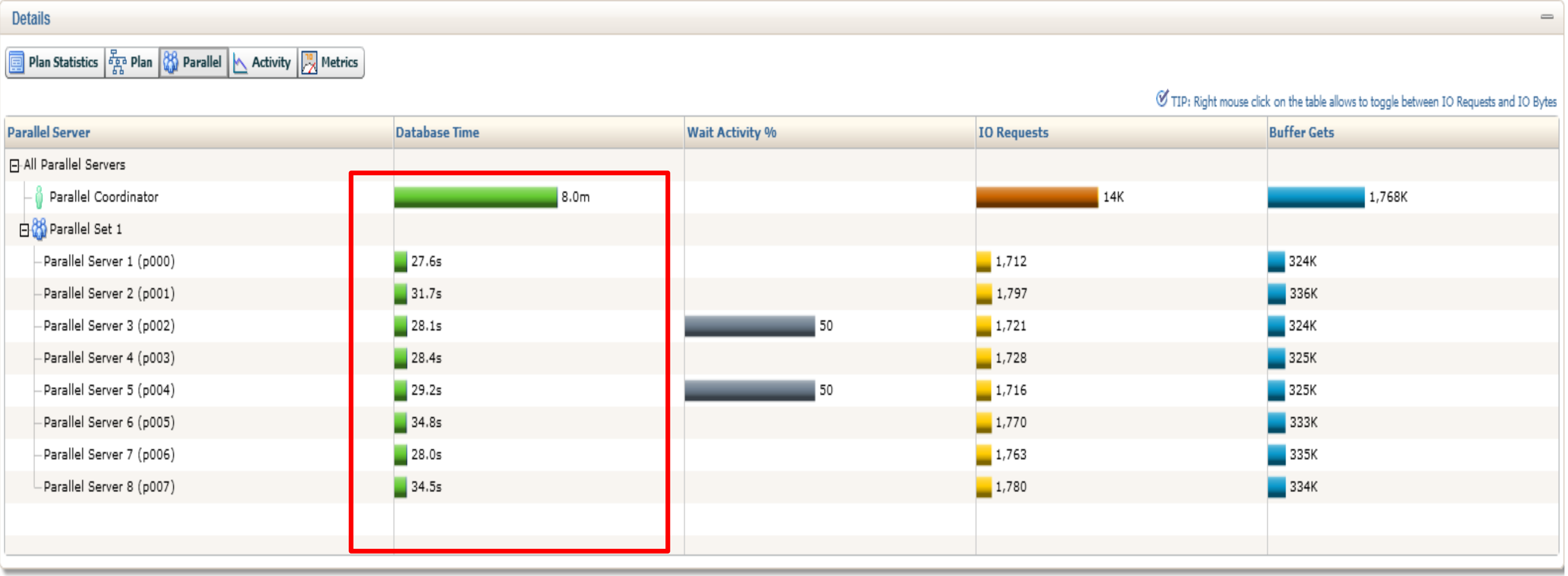
Plan Hash Value 4159595628

TIP: Right mouse click on the table allows to toggle between IO Requests and IO Bytes

Operation	Name	Estimated R...	Cost	Timeline(480s)	Execut...	Actual Ro...	Memory (...	Temp (Max)	IO Requests	CPU Activity %	Wait Activity %
INSERT STATEMENT					1	1					
LOAD AS SELECT					1	1	2MB		14K	50	
PX COORDINATOR					9	240M				20	
PX SEND QC (RANDOM)	:TQ10000	240M	66K		8	240M				9.66	100
PX BLOCK ITERATOR		240M	66K		8	240M					
TABLE ACCESS FULL	PARTSUPP	240M	66K		112	240M			14K	20	

Tuning SQL: Real-time SQL Monitoring

Parallel Tab



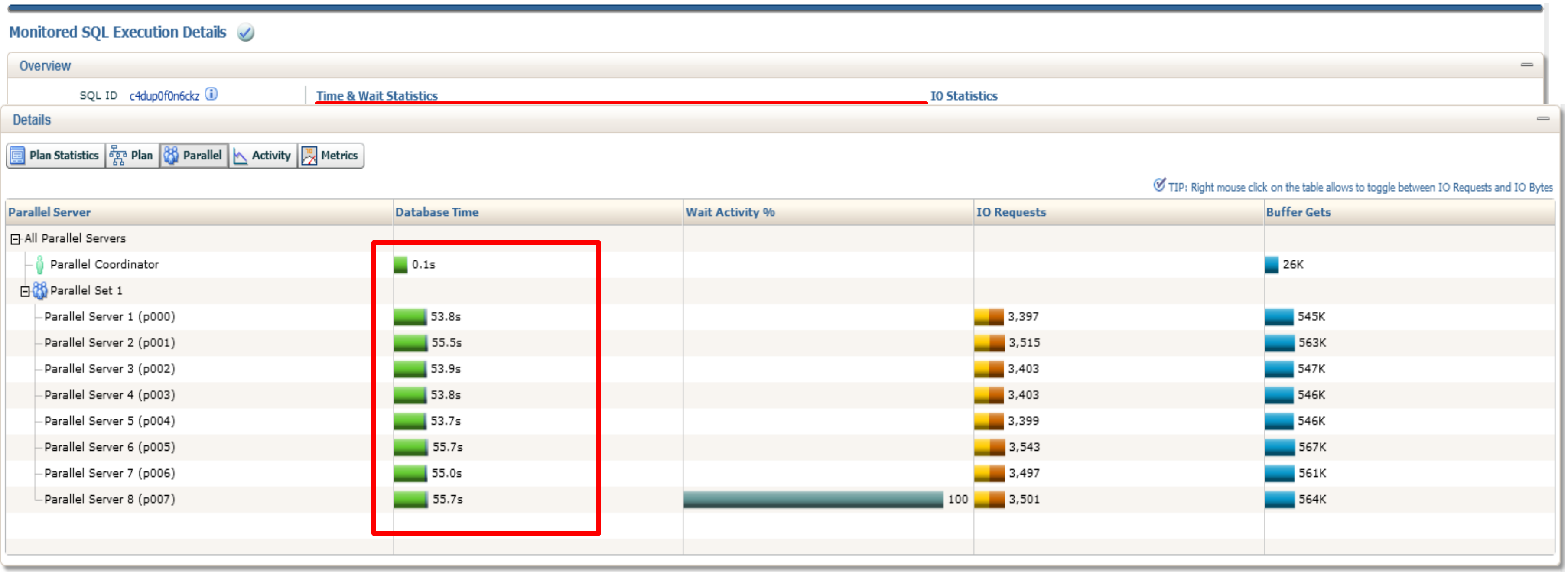
TIP: Right mouse click on the table allows to toggle between IO Requests and IO Bytes

Parallel Server	Database Time	Wait Activity %	IO Requests	Buffer Gets
All Parallel Servers				
Parallel Coordinator	8.0m		14K	1,768K
Parallel Set 1				
Parallel Server 1 (p000)	27.6s		1,712	324K
Parallel Server 2 (p001)	31.7s		1,797	336K
Parallel Server 3 (p002)	28.1s	50	1,721	324K
Parallel Server 4 (p003)	28.4s		1,728	325K
Parallel Server 5 (p004)	29.2s	50	1,716	325K
Parallel Server 6 (p005)	34.8s		1,770	333K
Parallel Server 7 (p006)	28.0s		1,763	335K
Parallel Server 8 (p007)	34.5s		1,780	334K

- Parallel Coordinator busy for the entire duration!!

Tuning SQL: Real-Time SQL Monitoring

Solution: Enabled Parallel DML



SQL ID c4dup0f0n6ckz

Time & Wait Statistics

IO Statistics

TIP: Right mouse click on the table allows to toggle between IO Requests and IO Bytes

Parallel Server	Database Time	Wait Activity %	IO Requests	Buffer Gets
⊟ All Parallel Servers				
Parallel Coordinator	0.1s			26K
⊟ Parallel Set 1				
Parallel Server 1 (p000)	53.8s		3,397	545K
Parallel Server 2 (p001)	55.5s		3,515	563K
Parallel Server 3 (p002)	53.9s		3,403	547K
Parallel Server 4 (p003)	53.8s		3,403	546K
Parallel Server 5 (p004)	53.7s		3,399	546K
Parallel Server 6 (p005)	55.7s		3,543	567K
Parallel Server 7 (p006)	55.0s		3,497	561K
Parallel Server 8 (p007)	55.7s	100	3,501	564K

✓ Parallel Slaves busy for the entire duration!!!

Program Agenda

- Why SQL statements regress?
- Identifying problematic SQL
- Tuning SQL
- Preventing SQL problems
- Real-World Customer Experiences

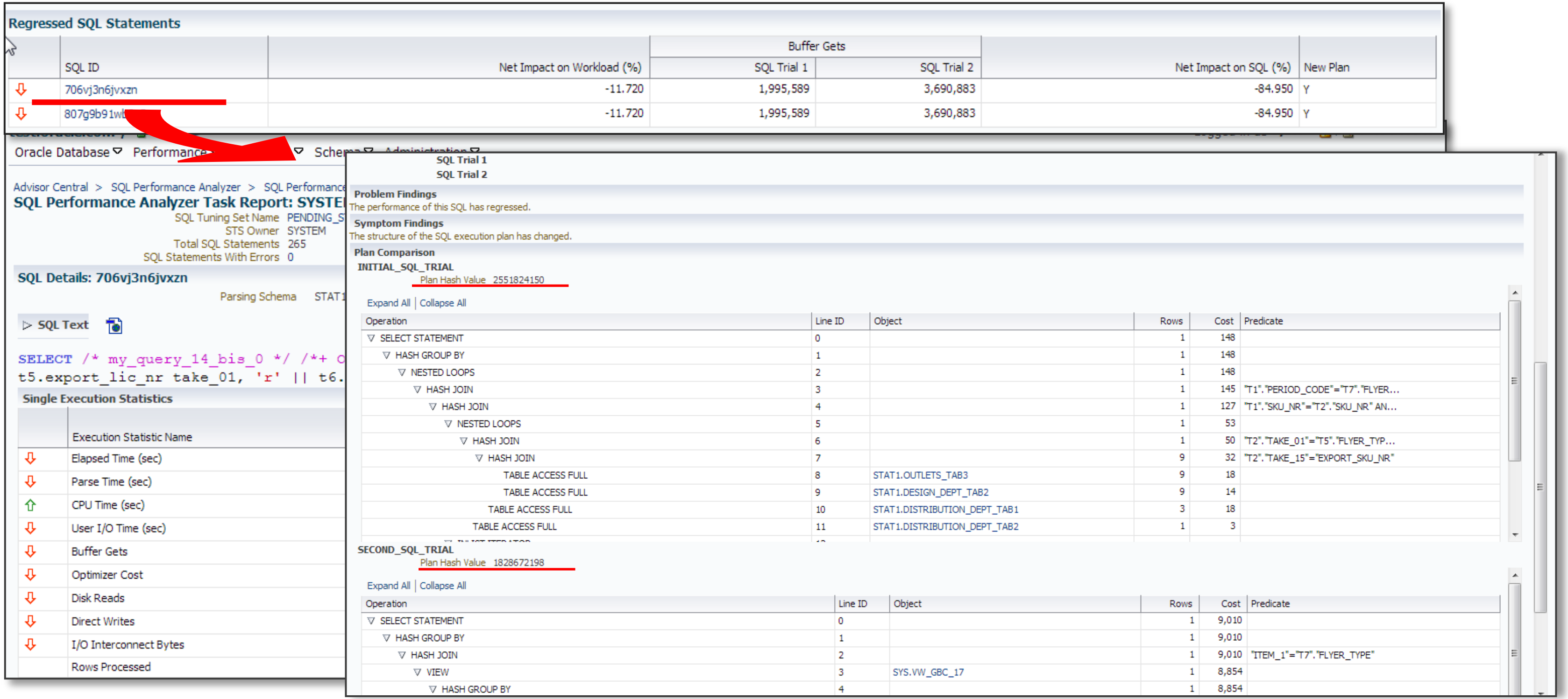
  - AveA, Turkey

  - S. Corporation, Korea

ORACLE

Preventing SQL problems:

Finding Regressed SQL statements proactively using SPA



Regressed SQL Statements

	SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	New Plan
			SQL Trial 1	SQL Trial 2		
⇩	706vj3n6jvxzn	-11.720	1,995,589	3,690,883	-84.950	Y
⇩	807g9b91wb	-11.720	1,995,589	3,690,883	-84.950	Y

Advisor Central > SQL Performance Analyzer > SQL Performance

SQL Performance Analyzer Task Report: SYSTE

SQL Tuning Set Name PENDING_S
STS Owner SYSTEM
Total SQL Statements 265
SQL Statements With Errors 0

SQL Details: 706vj3n6jvxzn

Parsing Schema STAT1

▷ SQL Text

```
SELECT /* my_query_14_bis_0 */ /*+ O
t5.export_lic_nr take_01, 'r' || t6.
```

Single Execution Statistics

	Execution Statistic Name
⇩	Elapsed Time (sec)
⇩	Parse Time (sec)
⇧	CPU Time (sec)
⇩	User I/O Time (sec)
⇩	Buffer Gets
⇩	Optimizer Cost
⇩	Disk Reads
⇩	Direct Writes
⇩	I/O Interconnect Bytes
	Rows Processed

SQL Trial 1
SQL Trial 2

Problem Findings

The performance of this SQL has regressed.

Symptom Findings

The structure of the SQL execution plan has changed.

Plan Comparison

INITIAL_SQL_TRIAL

Plan Hash Value 2551824150

Expand All | Collapse All

Operation	Line ID	Object	Rows	Cost	Predicate
▽ SELECT STATEMENT	0		1	148	
▽ HASH GROUP BY	1		1	148	
▽ NESTED LOOPS	2		1	148	
▽ HASH JOIN	3		1	145	"T1"."PERIOD_CODE"="T7"."FLYER...
▽ HASH JOIN	4		1	127	"T1"."SKU_NR"="T2"."SKU_NR" AN...
▽ NESTED LOOPS	5		1	53	
▽ HASH JOIN	6		1	50	"T2"."TAKE_01"="T5"."FLYER_TYP...
▽ HASH JOIN	7		9	32	"T2"."TAKE_15"="EXPORT_SKU_NR"
TABLE ACCESS FULL	8	STAT1.OUTLETS_TAB3	9	18	
TABLE ACCESS FULL	9	STAT1.DESIGN_DEPT_TAB2	9	14	
TABLE ACCESS FULL	10	STAT1.DISTRIBUTION_DEPT_TAB1	3	18	
TABLE ACCESS FULL	11	STAT1.DISTRIBUTION_DEPT_TAB2	1	3	

SECOND_SQL_TRIAL

Plan Hash Value 1828672198

Expand All | Collapse All

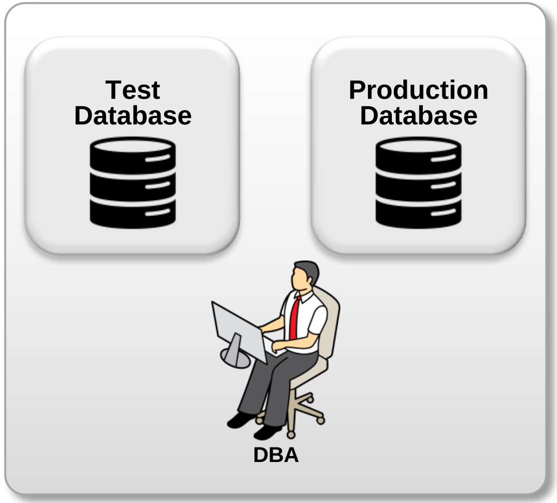
Operation	Line ID	Object	Rows	Cost	Predicate
▽ SELECT STATEMENT	0		1	9,010	
▽ HASH GROUP BY	1		1	9,010	
▽ HASH JOIN	2		1	9,010	"ITEM_1"="T7"."FLYER_TYPE"
▽ VIEW	3	SYS.VW_GBC_17	1	8,854	
▽ HASH GROUP BY	4		1	8,854	

SPA Challenges

Running SPA on:

Test System: Safe But…

- Requires separate HW
- Data in test system should be same as production
- Lengthy, error-prone task



Production System: Easier but…

- Could be resource intensive and impact production performance
- Changes needs to be manually scoped to private session
- Could take a long time to finish
- No resource control by default

SPA Quick Check

New in EM 12c Database Plug-in 12.1.0.5

Supports routine production change use cases

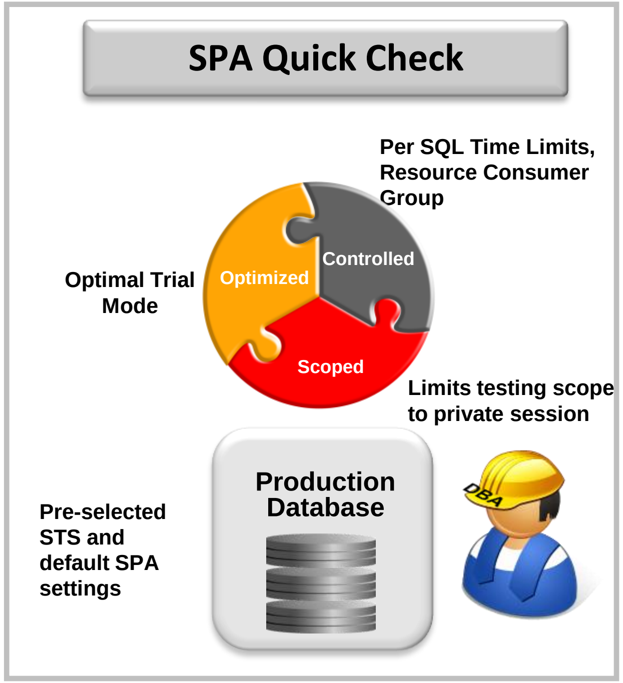
- Optimizer Gather Statistics
- Init.ora parameter changes
- Index creation
- Support for DB Release 11g and above

Simplifies precise identification of plan regressions

- Uses pre-selected STS that represents workload to be tested
- Default SPA settings

Designed and optimized for production use

- Optimal Trial or Explain Plan Mode
- Per SQL Time Limit, Resource Consumer Group
- Limits testing scope to private session



SPA Quick Check

Per SQL Time Limits, Resource Consumer Group

Optimal Trial Mode

Optimized

Controlled

Scoped

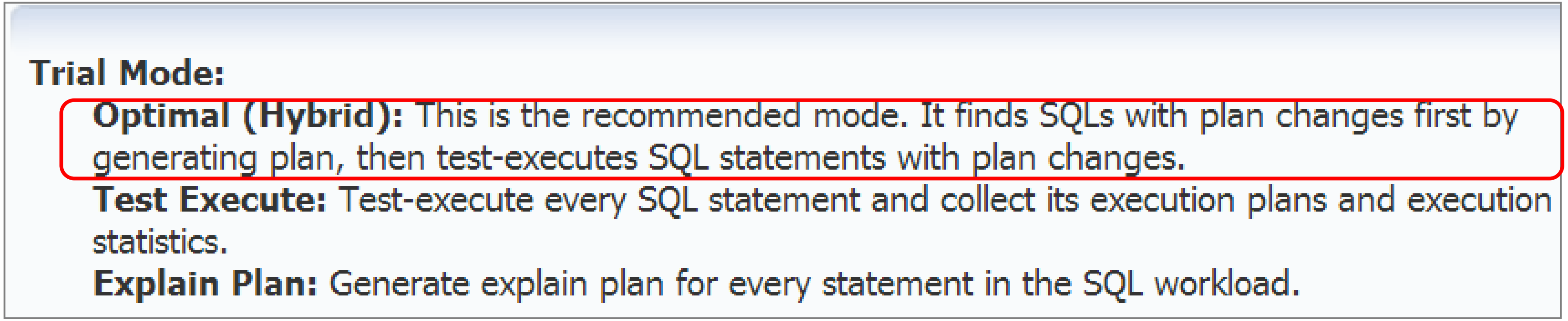
Limits testing scope to private session

Pre-selected STS and default SPA settings

Production Database

SPA Quick Check

Optimal Trial Mode



Trial Mode:

Optimal (Hybrid): This is the recommended mode. It finds SQLs with plan changes first by generating plan, then test-executes SQL statements with plan changes.

Test Execute: Test-execute every SQL statement and collect its execution plans and execution statistics.

Explain Plan: Generate explain plan for every statement in the SQL workload.

Identifies subset SQL workload with plan changes first

Test-executes only SQLs with plan changes

Minimizes use of production resources dramatically

- In general, resource consumption reduced in the range of 10x or more

SPA Quick Check: Setup

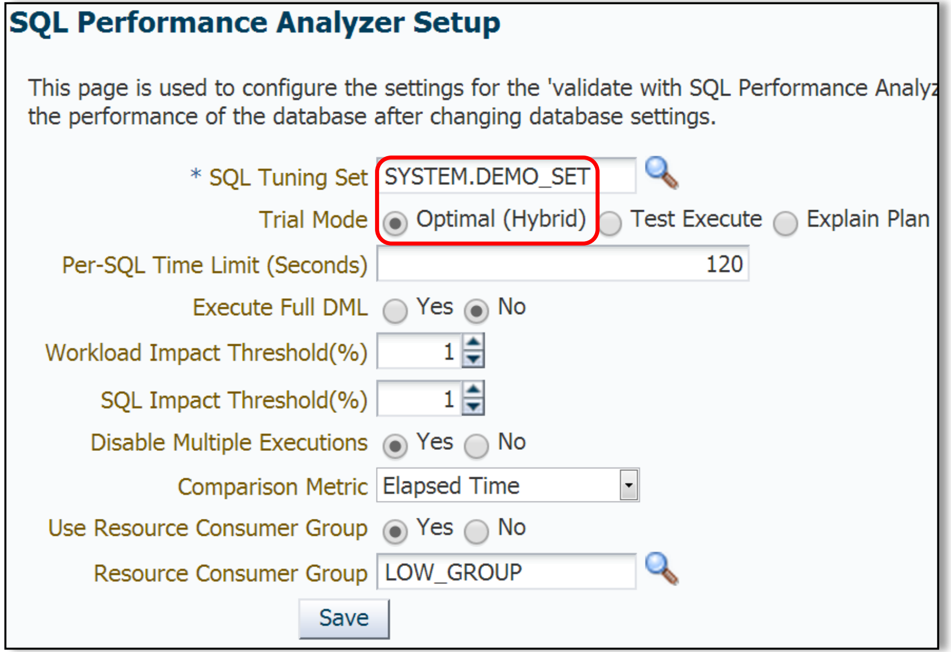
Minimal use of production resources

Per-SQL Time Limit - prevents any runaway SQLs

Disables Multiple Executions

Uses Resource Consumer Group

Executes only the query part of the DML without modifying data



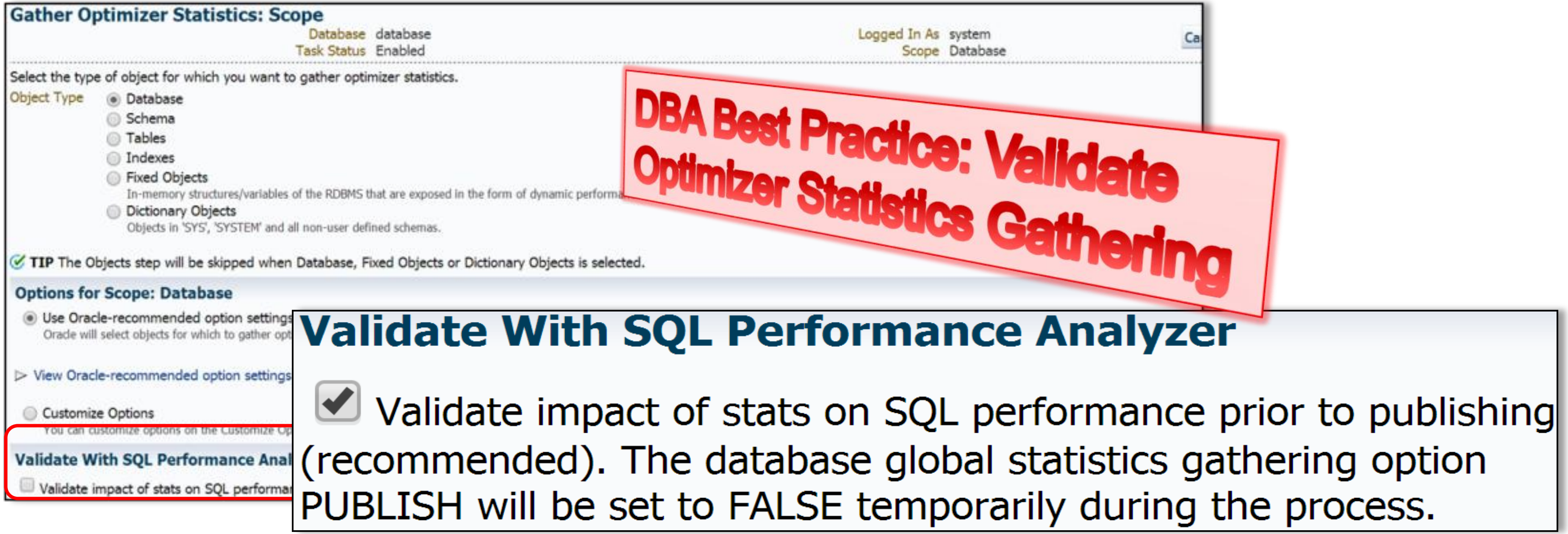
SQL Performance Analyzer Setup

This page is used to configure the settings for the 'validate with SQL Performance Analyz
the performance of the database after changing database settings.

* SQL Tuning Set	SYSTEM.DEMO_SET
Trial Mode	● Optimal (Hybrid) ○ Test Execute ○ Explain Plan
Per-SQL Time Limit (Seconds)	120
Execute Full DML	○ Yes ● No
Workload Impact Threshold(%)	1
SQL Impact Threshold(%)	1
Disable Multiple Executions	● Yes ○ No
Comparison Metric	Elapsed Time
Use Resource Consumer Group	● Yes ○ No
Resource Consumer Group	LOW_GROUP

SPA Quick Check

Launch SPA in-line with Gather Optimizer Statistics workflow



Gather Optimizer Statistics: Scope

Database database
Task Status Enabled
Logged In As system
Scope Database

Select the type of object for which you want to gather optimizer statistics.

Object Type
- ● Database
- ○ Schema
- ○ Tables
- ○ Indexes
- ○ Fixed Objects
  In-memory structures/variables of the RDBMS that are exposed in the form of dynamic performa
- ○ Dictionary Objects
  Objects in 'SYS', 'SYSTEM' and all non-user defined schemas.

✓ **TIP** The Objects step will be skipped when Database, Fixed Objects or Dictionary Objects is selected.

Options for Scope: Database

● Use Oracle-recommended option settings
Oracle will select objects for which to gather opt

▷ View Oracle-recommended option settings

○ Customize Options
You can customize options on the Customize Op

Validate With SQL Performance Anal

□ Validate impact of stats on SQL performan

DBA Best Practice: Validate Optimizer Statistics Gathering

Validate With SQL Performance Analyzer

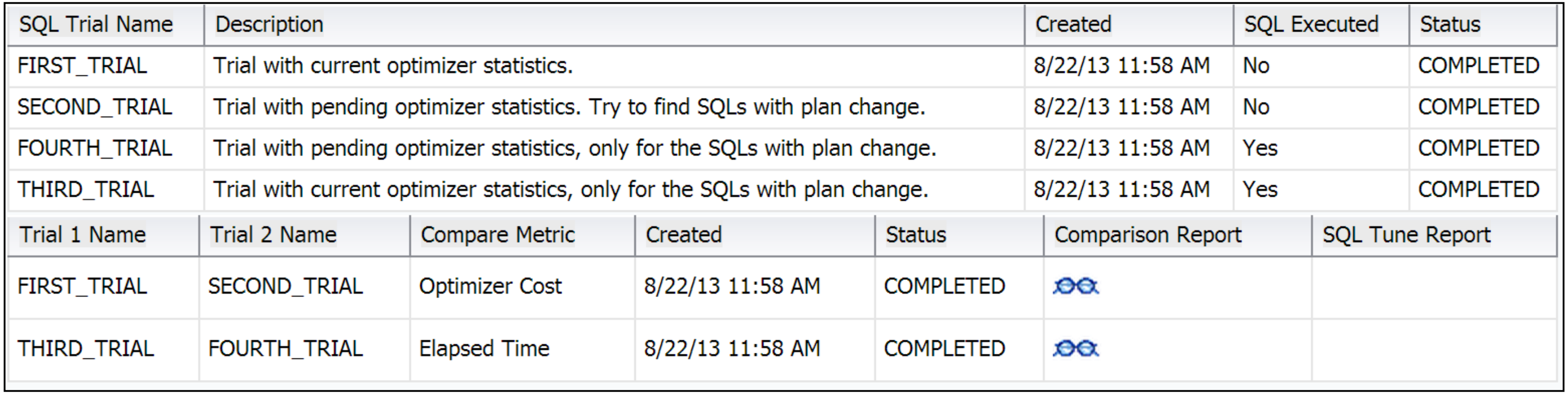
☑ Validate impact of stats on SQL performance prior to publishing (recommended). The database global statistics gathering option PUBLISH will be set to FALSE temporarily during the process.

SPA Quick Check

Workflow for validating Gather Optimizer Statistics

Optimizer statistics gathering option PUBLISH set to FALSE temporarily during the process

Four trials automatically executed, compared and reports generated



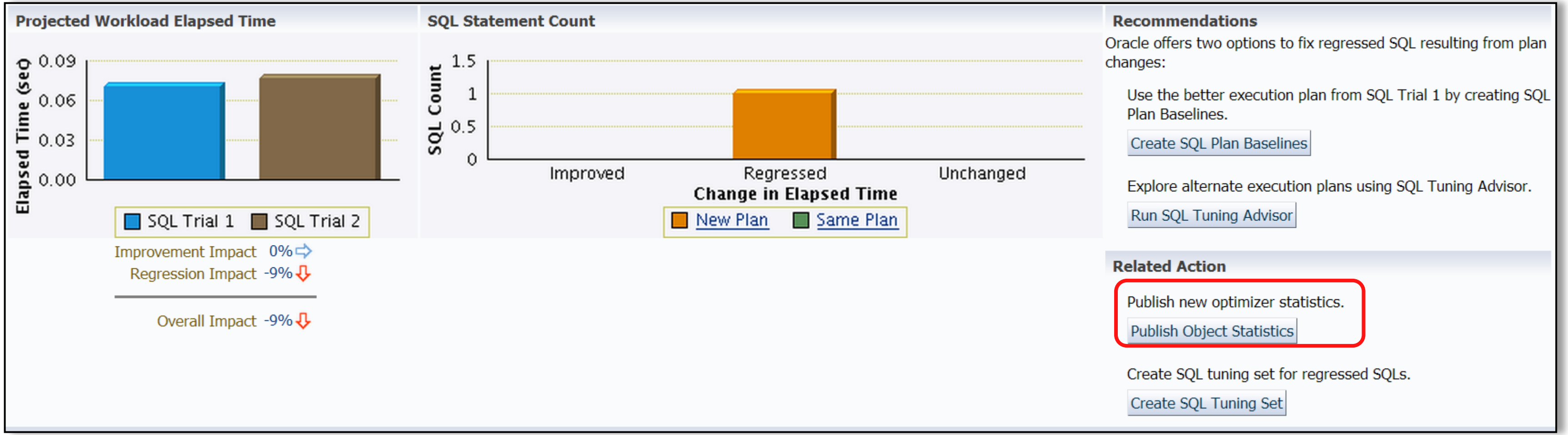
SQL Trial Name	Description	Created	SQL Executed	Status
FIRST_TRIAL	Trial with current optimizer statistics.	8/22/13 11:58 AM	No	COMPLETED
SECOND_TRIAL	Trial with pending optimizer statistics. Try to find SQLs with plan change.	8/22/13 11:58 AM	No	COMPLETED
FOURTH_TRIAL	Trial with pending optimizer statistics, only for the SQLs with plan change.	8/22/13 11:58 AM	Yes	COMPLETED
THIRD_TRIAL	Trial with current optimizer statistics, only for the SQLs with plan change.	8/22/13 11:58 AM	Yes	COMPLETED

Trial 1 Name	Trial 2 Name	Compare Metric	Created	Status	Comparison Report	SQL Tune Report
FIRST_TRIAL	SECOND_TRIAL	Optimizer Cost	8/22/13 11:58 AM	COMPLETED	👓	
THIRD_TRIAL	FOURTH_TRIAL	Elapsed Time	8/22/13 11:58 AM	COMPLETED	👓	

SPA Quick Check

Workflow for Validating Gather Optimizer Statistics

- Provides actionable recommendations on the report
- Publish the pending statistics
- Create SQL Tuning Set for regressed SQLs
- Offers two options to fix regressed SQL resulting from plan changes
- Create SQL Plan Baselines
- Explore alternate execution plans using SQL Tuning Advisor



Improvement Impact 0%

Regression Impact -9%

Overall Impact -9%

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.

Create SQL Plan Baselines

Explore alternate execution plans using SQL Tuning Advisor.

Run SQL Tuning Advisor

Related Action

Publish new optimizer statistics.

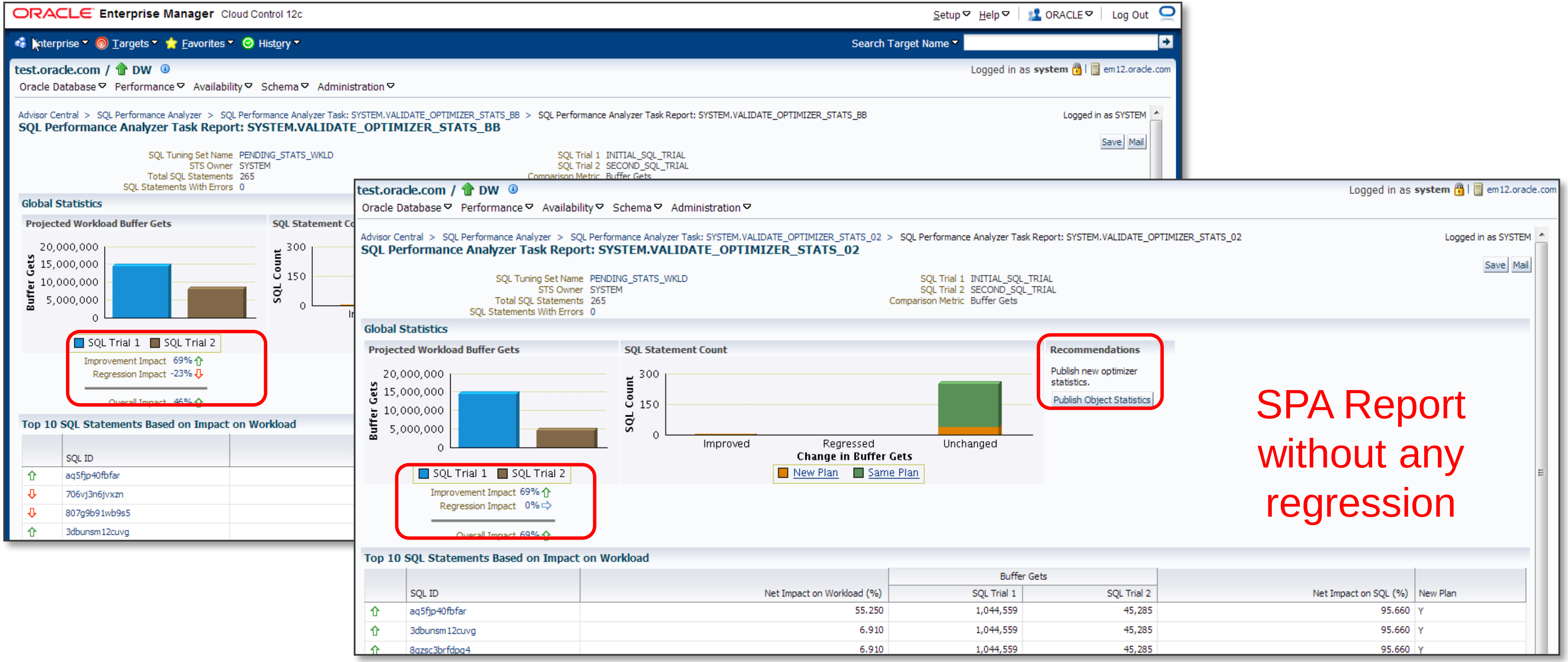
Publish Object Statistics

Create SQL tuning set for regressed SQLs.

Create SQL Tuning Set

SPA Quick Check

Gather Optimizer Statistics – SPA Report



Advisor Central > SQL Performance Analyzer > SQL Performance Analyzer Task: SYSTEM.VALIDATE_OPTIMIZER_STATS_BB > SQL Performance Analyzer Task Report: SYSTEM.VALIDATE_OPTIMIZER_STATS_BB

SQL Performance Analyzer Task Report: SYSTEM.VALIDATE_OPTIMIZER_STATS_BB

SQL Tuning Set Name PENDING_STATS_WKLD
STS Owner SYSTEM
Total SQL Statements 265
SQL Statements With Errors 0

SQL Trial 1 INITIAL_SQL_TRIAL
SQL Trial 2 SECOND_SQL_TRIAL
Comparison Metric Buffer Gets

Global Statistics

Projected Workload Buffer Gets

Improvement Impact 69%
Regression Impact -23%
Overall Impact 46%

Top 10 SQL Statements Based on Impact on Workload

	SQL ID
⇧	aq5fjp40fbfar
⇩	706vj3n6jvxzn
⇩	807g9b91wb9s5
⇧	3dbunsm12cuvg

Advisor Central > SQL Performance Analyzer > SQL Performance Analyzer Task: SYSTEM.VALIDATE_OPTIMIZER_STATS_02 > SQL Performance Analyzer Task Report: SYSTEM.VALIDATE_OPTIMIZER_STATS_02

SQL Performance Analyzer Task Report: SYSTEM.VALIDATE_OPTIMIZER_STATS_02

SQL Tuning Set Name PENDING_STATS_WKLD
STS Owner SYSTEM
Total SQL Statements 265
SQL Statements With Errors 0

SQL Trial 1 INITIAL_SQL_TRIAL
SQL Trial 2 SECOND_SQL_TRIAL
Comparison Metric Buffer Gets

Global Statistics

Projected Workload Buffer Gets

Improvement Impact 69%
Regression Impact 0%
Overall Impact 69%

SQL Statement Count

Recommendations

Publish new optimizer statistics.

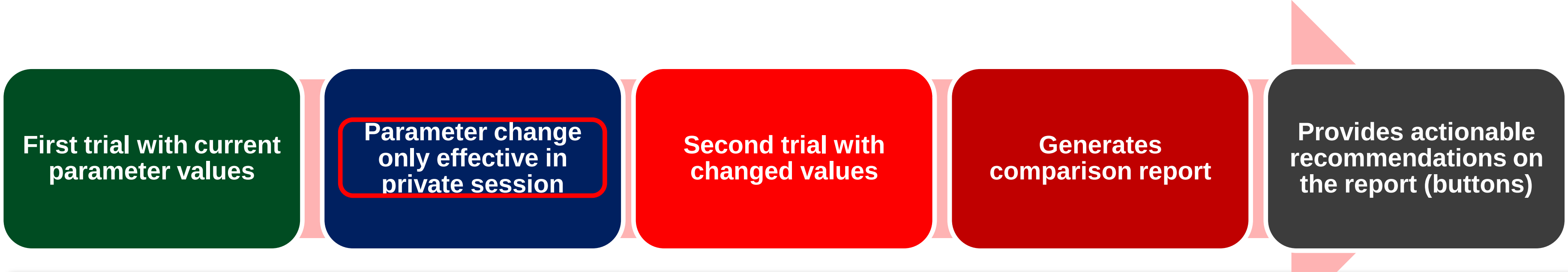
Top 10 SQL Statements Based on Impact on Workload

	SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	New Plan
			SQL Trial 1	SQL Trial 2		
⇧	aq5fjp40fbfar	55.250	1,044,559	45,285	95.660	Y
⇧	3dbunsm12cuvg	6.910	1,044,559	45,285	95.660	Y
⇧	8gzsc3brfdpg4	6.910	1,044,559	45,285	95.660	Y

SPA Report without any regression

Preventing SQL problems: SPA Quick Check

Init.ora changes: Validate optimizer related init parameter change



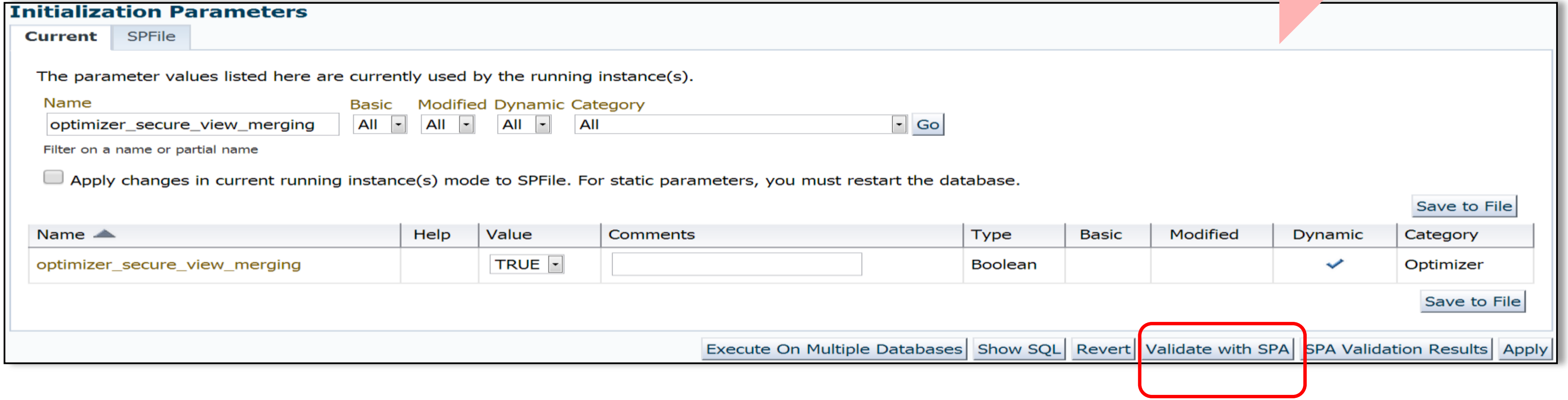
First trial with current parameter values

Parameter change only effective in private session

Second trial with changed values

Generates comparison report

Provides actionable recommendations on the report (buttons)



Initialization Parameters

The parameter values listed here are currently used by the running instance(s).

Name: optimizer_secure_view_merging — Basic: All — Modified: All — Dynamic: All — Category: All

Filter on a name or partial name

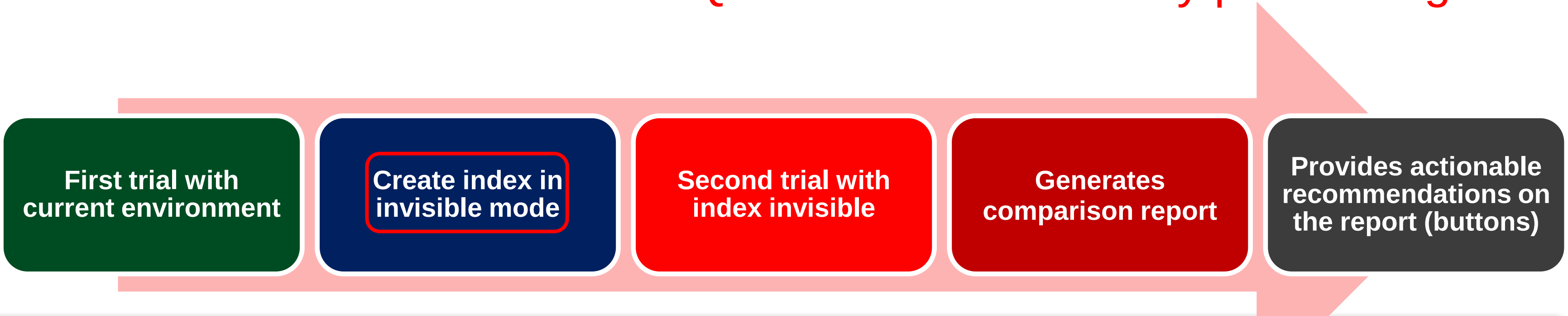
□ Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Name	Help	Value	Comments	Type	Basic	Modified	Dynamic	Category
optimizer_secure_view_merging		TRUE		Boolean			✓	Optimizer

Note: Applies to optimizer related changes and session parameters only

Preventing SQL problems: SPA Quick Check

New index creation: Use SPA Quick Check to find any plan changes



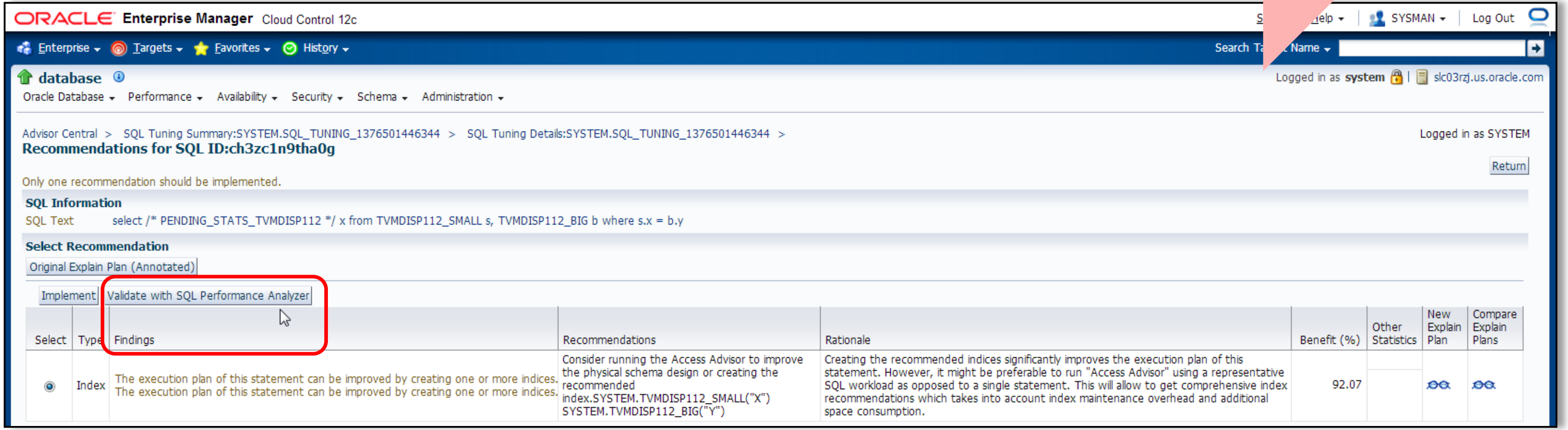
First trial with current environment

Create index in invisible mode

Second trial with index invisible

Generates comparison report

Provides actionable recommendations on the report (buttons)



Logged in as **system** | slc03rzj.us.oracle.com

Advisor Central > SQL Tuning Summary:SYSTEM.SQL_TUNING_1376501446344 > SQL Tuning Details:SYSTEM.SQL_TUNING_1376501446344 >

Logged in as SYSTEM

Recommendations for SQL ID:ch3zc1n9tha0g

Return

Only one recommendation should be implemented.

SQL Information

SQL Text select /* PENDING_STATS_TVMDISP112 */ x from TVMDISP112_SMALL s, TVMDISP112_BIG b where s.x = b.y

Select Recommendation

Original Explain Plan (Annotated)

Implement | Validate with SQL Performance Analyzer

Select	Type	Findings	Recommendations	Rationale	Benefit (%)	Other Statistics	New Explain Plan	Compare Explain Plans
●	Index	The execution plan of this statement can be improved by creating one or more indices. The execution plan of this statement can be improved by creating one or more indices.	Consider running the Access Advisor to improve the physical schema design or creating the recommended index.SYSTEM.TVMDISP112_SMALL("X") SYSTEM.TVMDISP112_BIG("Y")	Creating the recommended indices significantly improves the execution plan of this statement. However, it might be preferable to run "Access Advisor" using a representative SQL workload as opposed to a single statement. This will allow to get comprehensive index recommendations which takes into account index maintenance overhead and additional space consumption.	92.07		👓	👓

Note: Indexes are only visible in a scoped manner to the session, uses optimizer_use_invisible_indexes=true in private session

Program Agenda

- Why SQL statements regress?
- Identifying problematic SQL
- Tuning SQL
- Preventing SQL problems
- Real-World Customer Experiences

  - AveA, Turkey

  - S. Corporation, Korea

İşbu belge sadece Avea İletişim Hizmetleri A.Ş. ("AVEA"), bayileri, çalışanları ve/veya özel olarak yetki verilen diğer kişilerin kullanımına verilmiştir. İşbu belge içeriği gizli olup, belgenin ifşası çoğaltılması, dağıtımı ve/veya içeriğine ilişkin herhangi bir faaliyette bulunulması yasaktır. AVEA bu belgeyi almaya yetkili olmayan üçüncü taraflara belgenin herhangi bir şekilde gönderilmesinden sorumlu değildir.

11g Upgrade Project using Oracle Real Application Testing

Baki Şahin

Database Operation Supervisor

avea

AVEA

Avea is the youngest GSM operator of Turkey with its 13.6 million customers.

- The only GSM 1800 mobile operator of Turkey
- Founded in 2004 merged of 2 GSM Operators (Aria & Aycell)
- Nationwide customer base of 13.6 million
- Provide GSM service 98% of the population in Turkey AVEA
- Around 3000 people work for Avea
- Certified as an R&D company in 2010

Prepaid History Workload Description

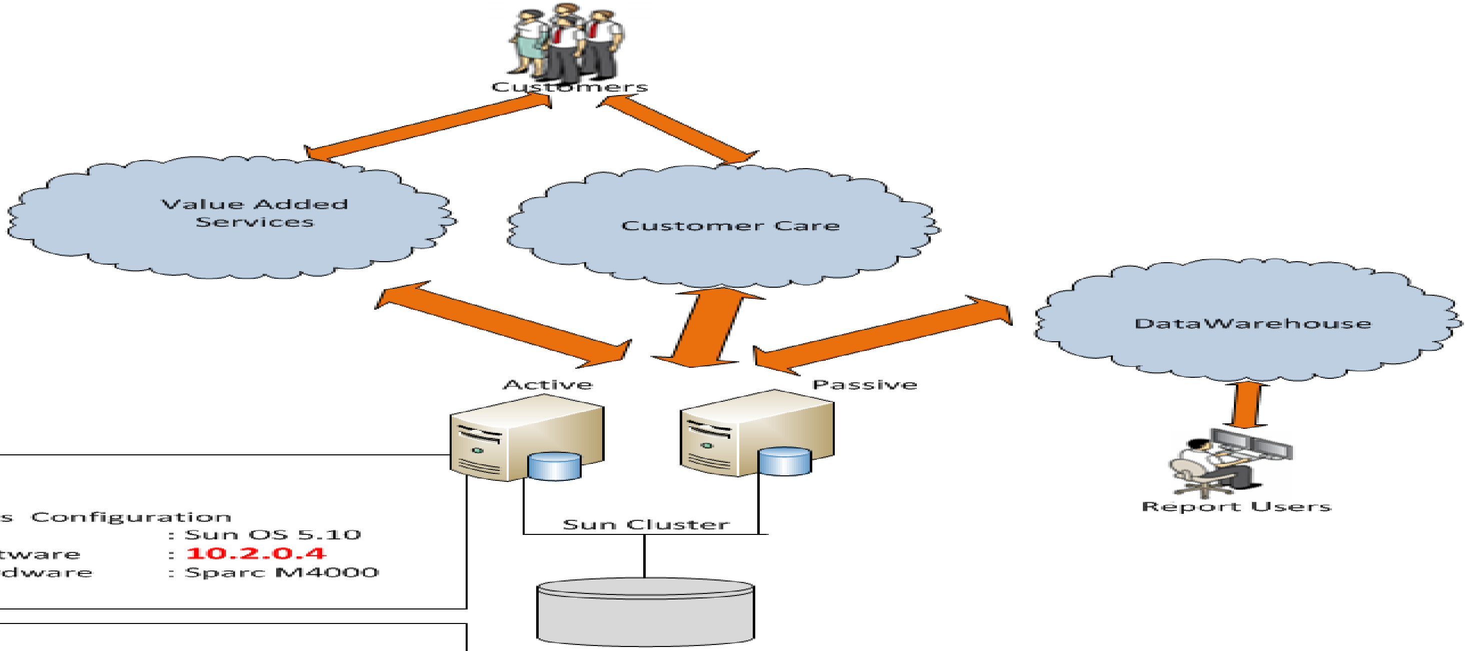
Application

- All transaction related to prepaid customers except bonus
- Store last 6 months traffic
- Business rules for post-sales processes. (Rule engine)
- Configuration of subscriber services

Database

- Database size is ~2.5Tb
- Workload can roughly be classified into 2 parts:
  - ❑ OLTP transactions is done during business hours
  - ❑ Batch processing is done after business hours

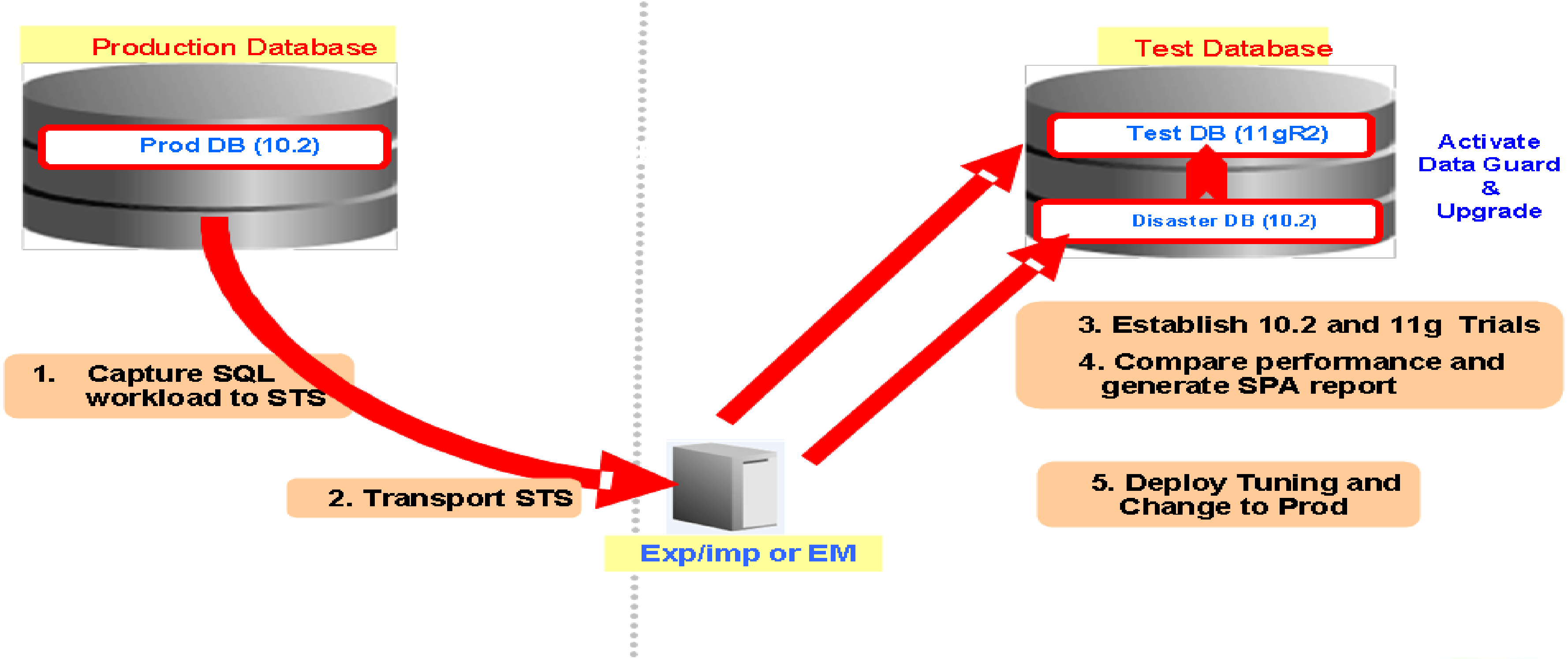
Overview of Configuration



As is Configuration
Os : Sun OS 5.10
Software : **10.2.0.4**
Hardware : Sparc M4000

To be Configuration
Os : Sun OS 5.10
Software : **11.2.0.3**
Hardware : Sparc M4000

10.2 → 11g Database Upgrade



Activate
Data Guard
&
Upgrade

1. Capture SQL workload to STS

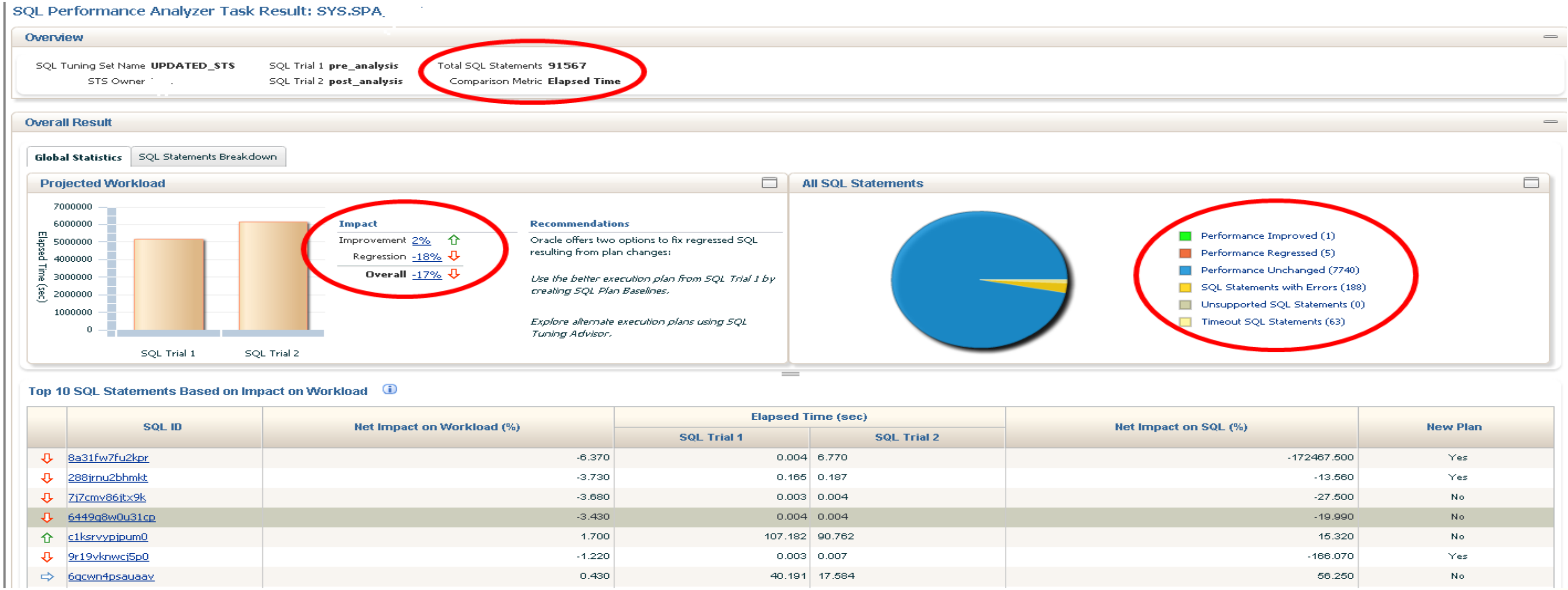
2. Transport STS

3. Establish 10.2 and 11g Trials

4. Compare performance and generate SPA report

5. Deploy Tuning and Change to Prod

Compare Performance



SQL Performance Analyzer Task Result: SYS.SPA_

Overview

SQL Tuning Set Name **UPDATED_STS**
STS Owner

SQL Trial 1 **pre_analysis**
SQL Trial 2 **post_analysis**

Total SQL Statements **91567**
Comparison Metric **Elapsed Time**

Overall Result

Global Statistics | SQL Statements Breakdown

Projected Workload

Impact

Improvement	2% ⇧
Regression	-18% ⇩
Overall	-17% ⇩

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.

All SQL Statements

- 🟩 Performance Improved (1)
- 🟥 Performance Regressed (5)
- 🟦 Performance Unchanged (7740)
- 🟨 SQL Statements with Errors (188)
- ▨ Unsupported SQL Statements (0)
- □ Timeout SQL Statements (63)

Top 10 SQL Statements Based on Impact on Workload

	SQL ID	Net Impact on Workload (%)	Elapsed Time (sec)		Net Impact on SQL (%)	New Plan
			SQL Trial 1	SQL Trial 2		
⇩	8a31fw7fu2kpr	-6.370	0.004	6.770	-172467.500	Yes
⇩	288jrnu2bhmkt	-3.730	0.165	0.187	-13.560	Yes
⇩	7j7cmv86jtx9k	-3.680	0.003	0.004	-27.500	No
⇩	6449q8w0u31cp	-3.430	0.004	0.004	-19.990	No
⇧	c1ksrvypjpum0	1.700	107.182	90.762	15.320	No
⇩	9r19vknwcj5p0	-1.220	0.003	0.007	-166.070	Yes
⇨	6qcwn4psauaav	0.430	40.191	17.584	56.250	No

Regressed SQL Statements

Plans unchanged

- Majority of plans unchanged
- Unlock statistics collections for tables
- Collects optimizer statistics again using 11g database
- Run SPA again

Plans changed – fixing regressions

- Create Baselines
- Work with developers
- Indexes created or dropped
- Run SPA again

Conclusion

Benefits for us

- Risk reduction
  - ❑ Replays All SQL
  - ❑ Real Bind Values
- Test production SQL workload before upgrade
- Tune regressed SQL statements
- No surprises when upgraded to 11.2.0.3
- Report what will happen before upgrade

- To be more safe
- Reusable process
- Reduced resource requirements
  - ❑ 1 team against 4 teams
- Fully proven recommendations
  - ❑ Easier to convince business

ORACLE®

S. Corporation
Large Manufacturing Company in South Korea: Case Study

ORACLE OPEN WORLD

HARDWARE AND SOFTWARE ENGINEERED TO WORK TOGETHER

Agenda

1. Project Overview

2. Oracle Solution for Upgrade

3. Results

Project Overview

● Customer Information

- S. Corporation: Manufacturing company in Korea.

● Target: ERP System

- SAP ERP system.
- ECC : Logistics, Finance, XI and other sub systems with RAC configuration
- Production, Test, DR and etc.

● Necessity of 11g Upgrade

- 11g upgrade for a stable service environment
- 10g version support instability due to End of Support period
- Database environment improvement through 11g new function utilization

● 11g Upgrade Schedule

- Project Term : Feb, 2012 – Dec, 2012
- Test Process: Procedure Test ▶ Stability Test ▶ Application Test (Function, Performance, Compatibility) ▶ Rehearsal ▶ Cut Over

Solution for Upgrade

Oracle Real Application Testing

Factor	Description
Pain Point	▪ Regression had to be checked before the upgrade due to business criticality ▪ Time and cost issue for test and verification of more than one million SQLs within 4 months. ▪ Low efficiency for SQL verification, if done manually.
How	▪ PoC for more than 2 months to validate the real effectiveness of RAT and to assess potential upgrade risks, which can't be found without RAT. ▪ Sort sequence changing target program test through module source check. ▪ Performance test & tuning for individual SQL through RAT SPA. (10g vs. 11g)
Result	▪ Contribution to stable 11g Upgrade through SQL Change Risk exclusion . ▪ 1 million SQLs Performance tuning through SQL Performance Analyzer utilization ▪ Reduced upgrade project time through auto verification.

Results

Factor	Content
Database Performance Improvement	▪ Improved performance through the Optimizer enhancement :About 15~20% improvement ▪ RAC efficiency, CPU usage monitoring improvement through the EM function improvements : About 20% improvement
RAT Solution Utilization	▪ Over 1 million SQLs Performance testing using SPA ▪ Stable 11g upgrade of the most mission critical system through SQL change risk elimination
Collaboration	▪ Verification exercises through the detailed procedure definition and sufficient tests. ▪ Successful collaboration between the customer TFT and Oracle team (Local and Global) through close communication.

Database Manageability

Recommended Sessions

Session	General Session	Day	Time	Location
GEN8792	**General Session: Database Management Innovations - Oracle Database 12c Manageability Highlights**	**Wednesday**	**10:15 AM**	**Moscone South – 103**

Session	Session	Day	Time	Location
CON9582	Oracle Exadata Management Deep Dive with Oracle Enterprise Manager 12c	Monday	12:15 PM	Westin - Metropolitan I
CON9573	Managing the Oracle Identity Management Platform with Oracle Enterprise Manager	Monday	1:45 PM	Moscone South - 130
CON9578	Automatic Workload Repository Soup to Nuts: Fundamentals of Database Performance Analysis	Monday	3:15 PM	Moscone South – 104
CON8788	Maximizing Database Performance with Database Replay	Tuesday	10:30 AM	Moscone South - 308
CON9583	Harness the Power of Oracle Database 12c with Oracle Enterprise Manager Database as a Service	Tuesday	3:45 PM	Moscone South – 305
CON9579	Step-by-Step Cookbook for Identifying and Tuning SQL Problems	Wednesday	1:15 PM	Moscone South – 103
CON4666	Oracle Enterprise Manager 12c Database Lifecycle Management Automatic Provisioning and Patching	Wednesday	3:30 PM	Moscone South – 131
CON8768	DBA Best Practices for Protecting Data Privacy with Oracle's Data Masking	Wednesday	3:30 PM	Moscone West – 2024
CON9577	Active Session History Deep Dive: Advanced Performance Analysis Tips	Wednesday	3:30 PM	Moscone South – 104
CON3255	Being Sure: Confident Consolidations with Oracle Real Application Testing 12c	Wednesday	5:00 PM	Moscone South – 306
CON3103	Real Oracle Real Application Testing: What to Expect and Prepare For	Thursday	12:30 PM	Moscone South - 104

ORACLE®