

ORACLE®

Eliminating Guesswork from SQL Tuning



September 18–22, 2016
San Francisco

GP (Gongloor Prabhaker)
Senior Director of Product Management

Dr. Khaled Yagoub
Architect

Systems and DB Manageability Development
Oracle Corporation

Accelerate Your
Digital Transformation
in the Cloud

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.

Program Agenda

- 1 SQL Tuning: Challenges and existing solutions
- 2 Recommended tuning methodology
- 3 Methodology Demo: Cloud Migration
- 4 New Features in Oracle Database 12.2

Please visit us at Demo Grounds:
Moscone South, SDB-043, End to End Performance Management

Program Agenda

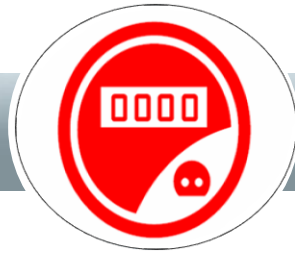
- 1 SQL Tuning: Challenges and existing solutions
- 2 Recommended tuning methodology
- 3 Methodology Demo: Cloud Migration
- 4 New Features in Oracle Database 12.2

SQL Tuning: Why is Guesswork Involved?

Common Causes



Optimizer Related



Resource Contention



Application Specific
Issues



Lack of Automated, E2E
and scalable solution



End-user Expertise



SQL Tuning: Why is Guesswork Involved?

Common Causes - Drilldown

Optimizer Related

- Stale/Missing statistics
- Incomplete statistics
- Improper optimizer configuration
- Upgraded Database: new optimizer
- Rapidly changing data

Resource Related

- Hardware resource crunch
- Contention (row lock/ block contention)
- Not parallelized (no scaling to large data)
- Improperly parallelized (partially parallelized, skews)

Application Issues

- Missing access structures
- Poorly written SQL statements
- Bind-sensitive SQL with bind peeking (Cursor Sharing)
- Literal usage

Lack of SQL & Optimizer Expertise

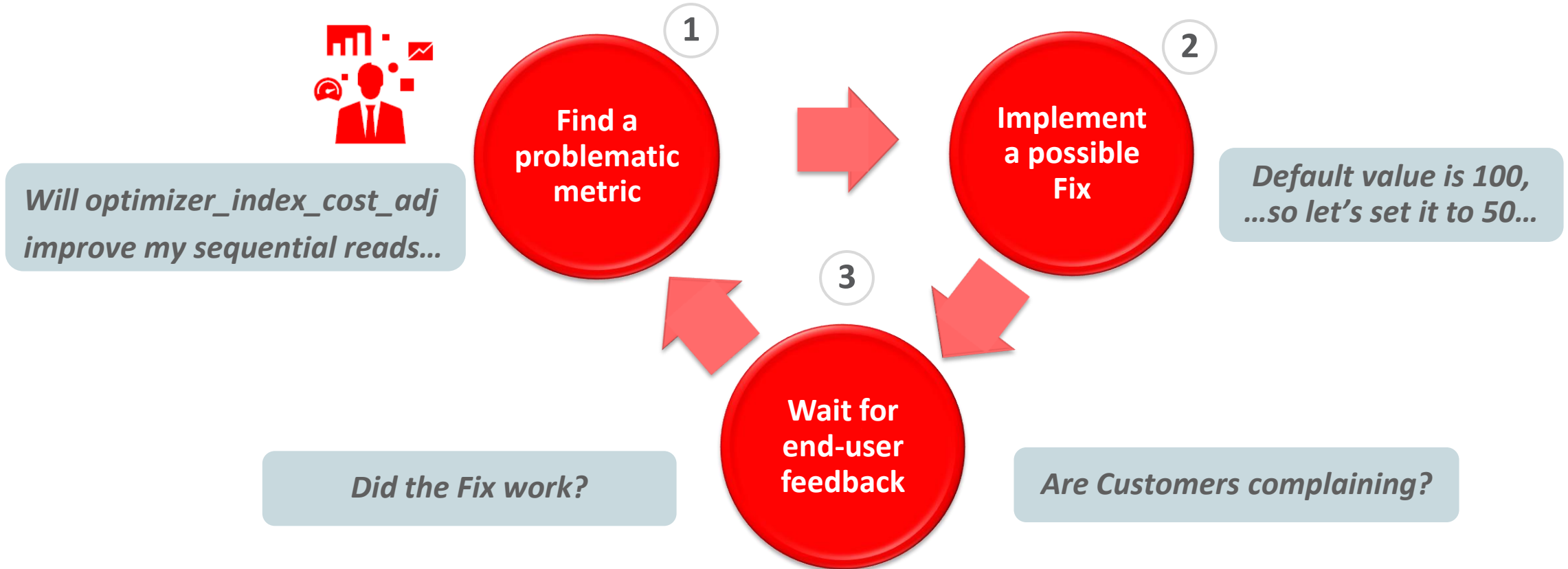
- Limited knowledge of SQL and Optimizer
- Tunes individual or few statements vs. entire workload
- Packaged applications

Lack of Automation

- Manual effort, time consuming
- Lack of scalability with high volume SQL (100K-1.5M)
- Unintuitive techniques and point solutions

Current SQL / DB Tuning Process

Inefficient, Inaccurate, Manual



The “trial and error” method can consume more than 50% of the DBA time

SQL Tuning: Current Solutions

Techniques employed in “Trial and Error” approach

- How DBAs tune SQL today?
 - Use SQL Trace to identify offending SQL solution
 - Prohibitive on production
 - Disable index access by modifying predicates (to_char, to_number)
 - Adding indexes - Most apps are already over-indexed
 - Change optimizer init.ora parameters, or use unsupported underscore parameters
 - Use hints or hand tuning SQL
 - Materialized Views, Logs
 - Partitioning, etc.
- Current solutions have severe limitations, are not scalable, and mostly in reactive nature

Program Agenda

- 1 SQL Tuning: Challenges and existing solutions
- 2 Recommended tuning methodology
- 3 Methodology Demo: Cloud Migration
- 4 New Features in Oracle Database 12.2

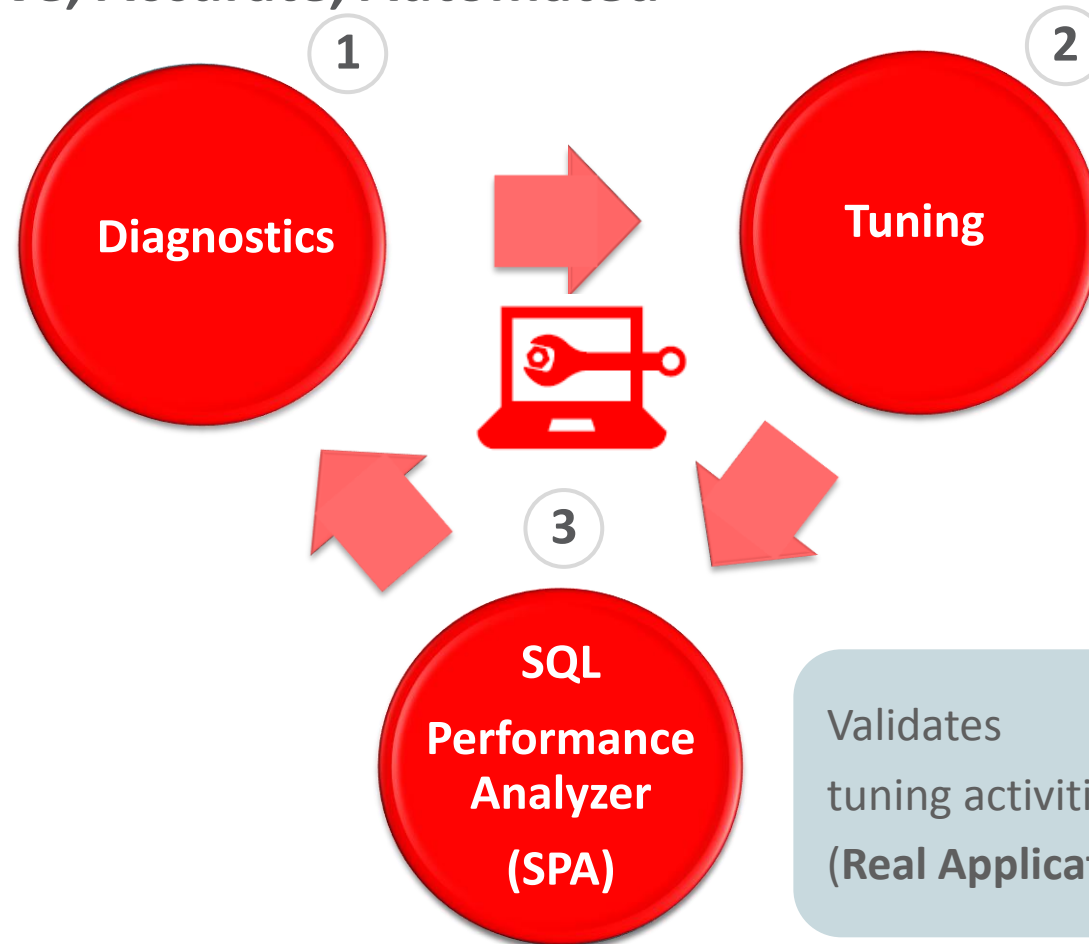


Recommended Tuning Methodology

Find – Fix – Validate

Recommended SQL Tuning Methodology

DB Time Based: Effective, Accurate, Automated

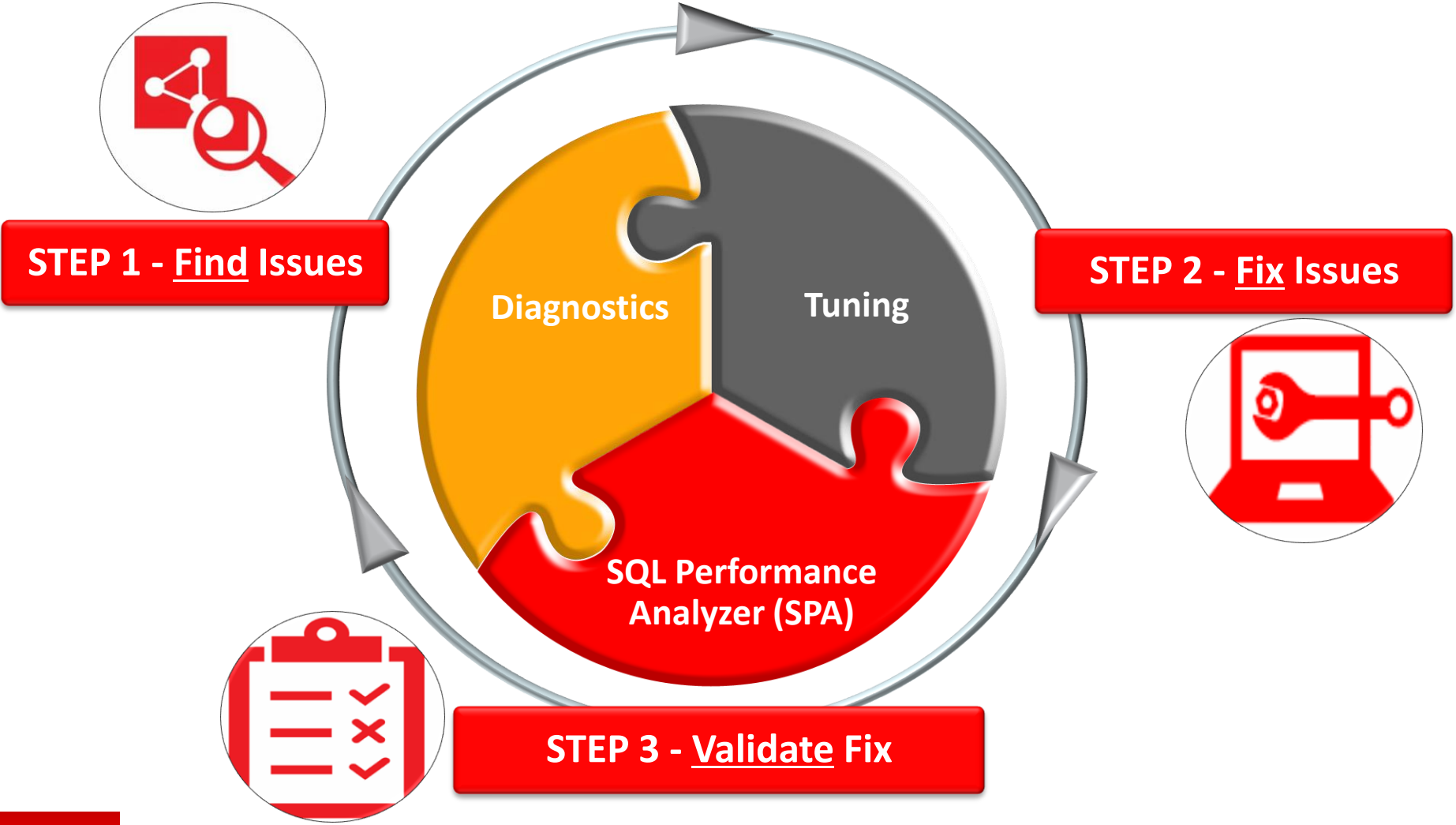


Built-in, self-diagnostics engine: Automatic Database Diagnostics Monitor (ADDM) - (Diagnostics Pack)

Automates complex and time consuming task of application tuning (Tuning Pack)

Validates tuning activities (Real Application Testing)

Find → Fix → Validate Methodology

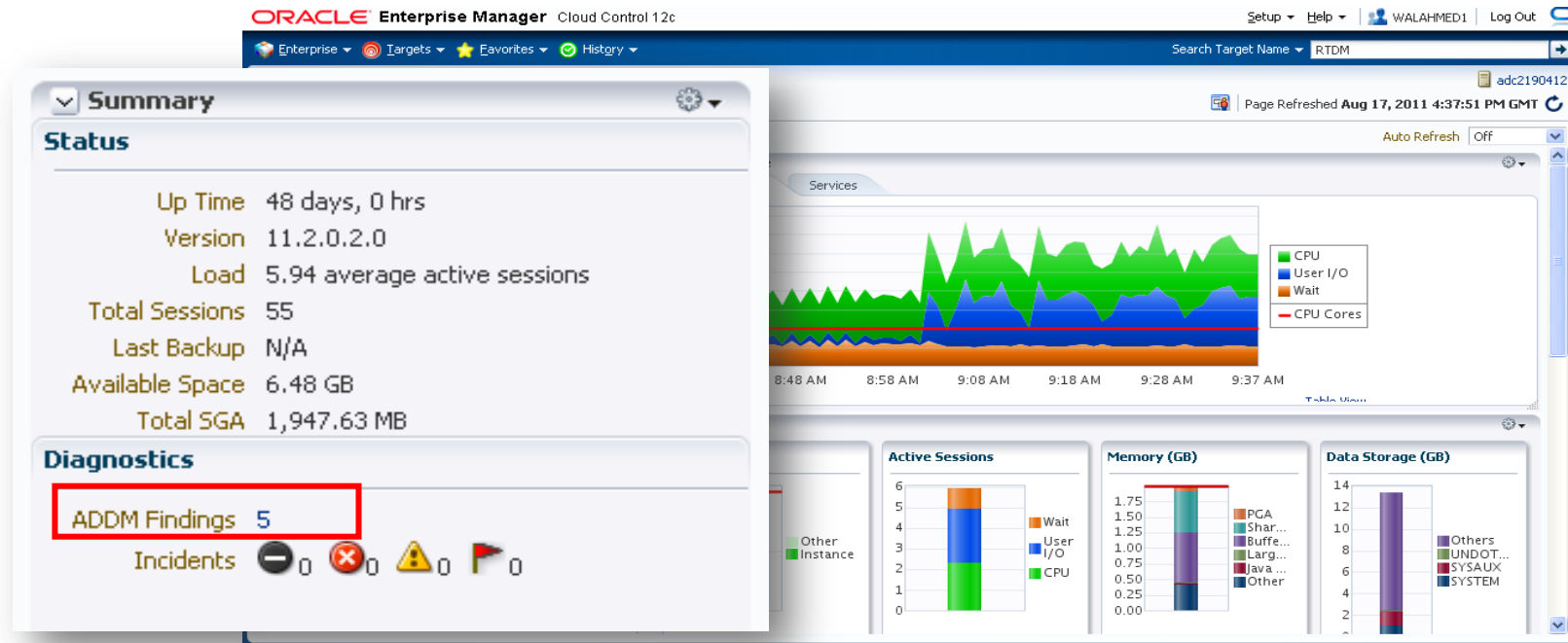


The Find

The Find: Identifying Top SQL

- Key sources for identifying problem SQL
 - Automatic Database Diagnostic Monitor (ADDM)
 - Shows high-load SQL with impact %
 - Based on analysis of SQL, recommends SQL advisors as needed
 - Not all high-load SQL are good candidates for advisors
 - E.g., SQL with HWM enqueue wait problem cannot be tuned by SQL advisors but require space reconfiguration
 - Top Activity viewing and analysis: Enterprise Manager (EM), ASH Analytics, Perfhub
 - Real Time Mode
 - Source: v\$active_session_history (ASH) & Period: Last one hour, customizable time picker
 - Historical Mode
 - Source: Automatic Workload Repository (AWR) & Period: Last 8 days (default)
 - Performance Hub available in EM Express , planned for OEM Cloud Control in future

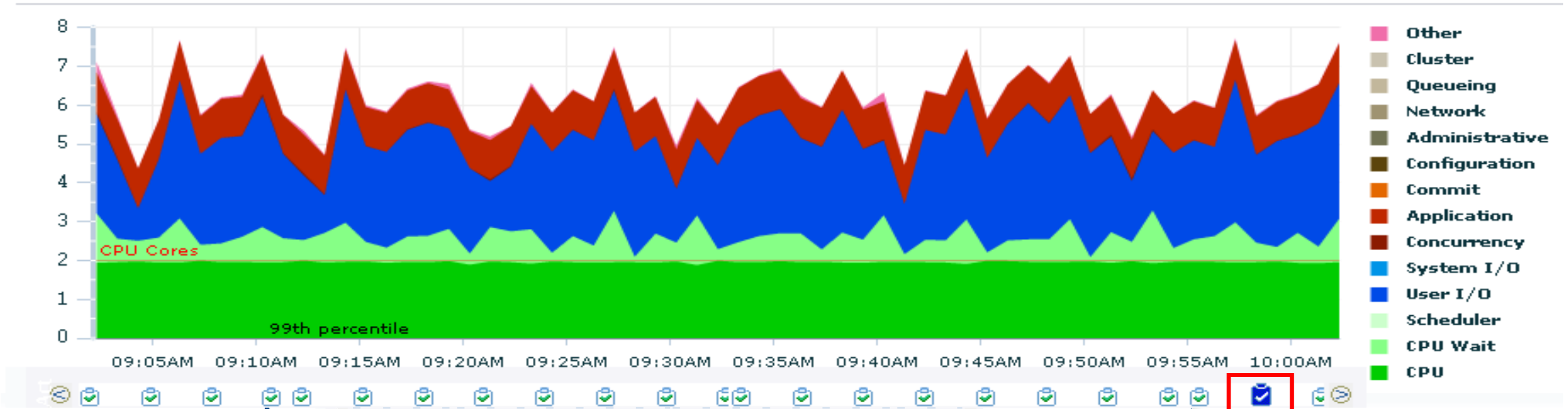
ADDM: Analysis and Recommendations



- Database Home Page gives a quick high level overview of current status
- Database found to be running high load (Active Sessions)
- ADDM has already detected some performance issues (5 findings)

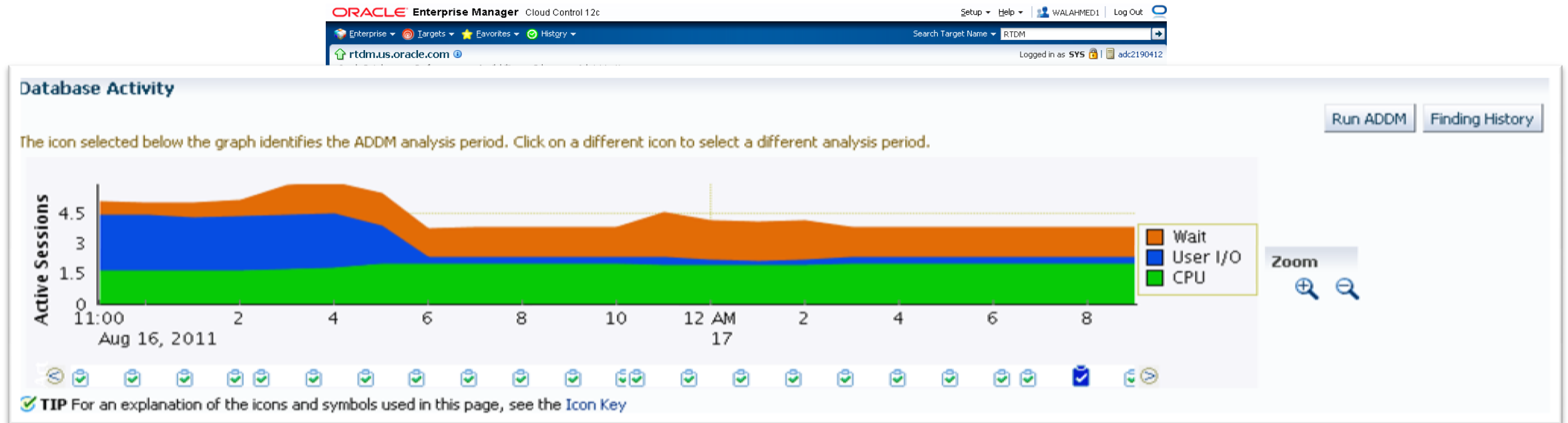
Performance Page

Average Active Sessions Foreground Only Foreground + Background



- For further investigation, look at the DB Perf. Page and “Click on the Big Stuff”
- DB Perf. Page shows significant resource being utilized on CPU and User I/O
- Clicking on the snapshot icon will take us to the ADDM Home Page for that time

ADDM Home Page



- ADDM Home Page: Database activity gives quick performance overview
- Looking at the ADDM performance analysis, SQL statements are impacting the system significantly
- Let's drill down on the Top SQL Findings

Automatic Diagnosis and Recommendations

Advisor Central > Automatic Database Diagnostic Monitor (ADDM):SYS.ADDM:4232357857_1_1738 > Performance Finding Details: Top SQL Statements

Finding SQL statements consuming significant database time were found.

Impact (Active Sessions) 5.63

Percentage of Finding's Impact (%) 91.2

Period Start Time Aug 17, 2011 9:00:49 AM

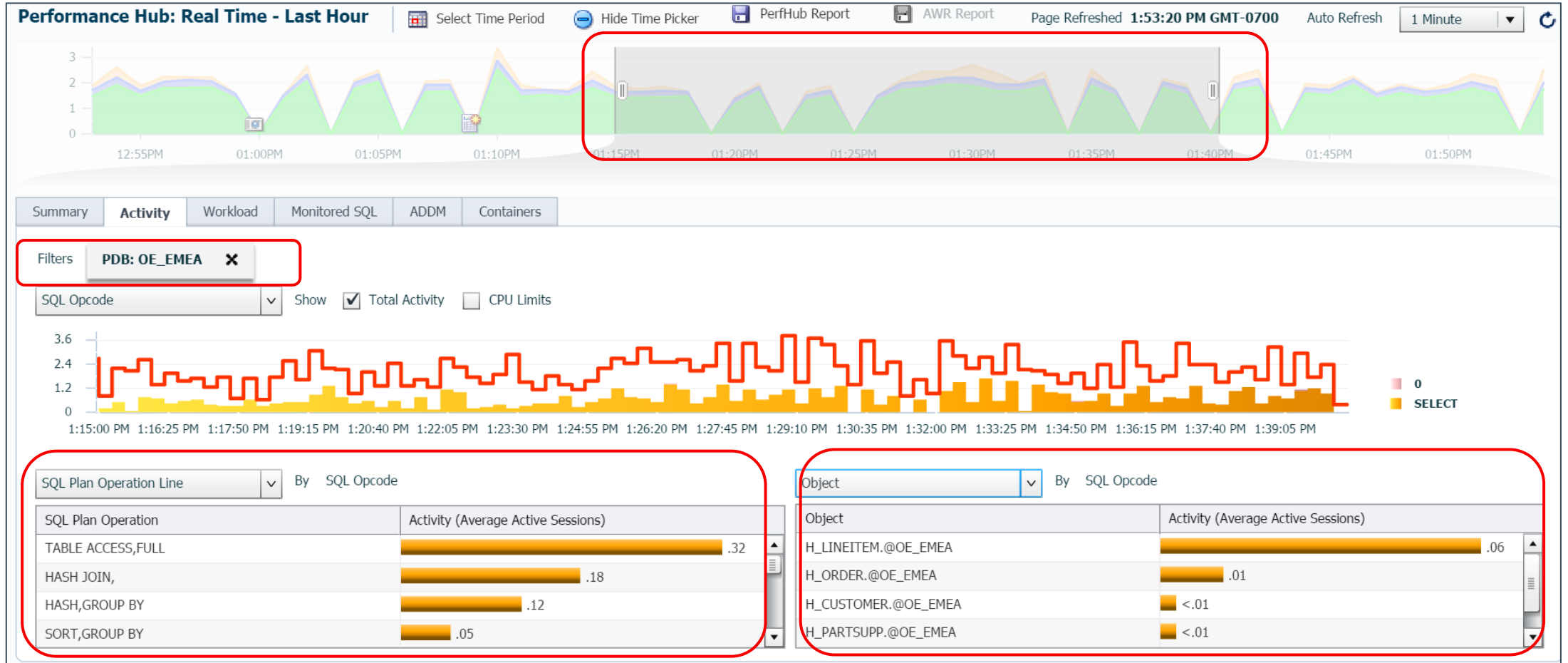
End Time Aug 17, 2011 10:00:00 AM

Filtered No Filters

The screenshot shows a detailed view of a SQL statement identified as a performance finding. The SQL text is: `SELECT MAX(AMOUNT_SOLD) FROM SALES WHERE CUST_ID IN (SELECT CUST_ID FROM CUSTOMER...`. The report includes several rationale sections:
1. The SQL spent 96% of its database time on CPU, I/O and Cluster waits.
2. Database time for this SQL was 100% for SQL execution.
3. The statement was executed 644 times with an average elapsed time of 11 seconds.
4. At least one execution ran in parallel.
5. I/O and Cluster wait for TABLE PARTITION "SH2.SALES.SALES_1995" consumed 8% of the database time.
6. I/O and Cluster wait for TABLE PARTITION "SH2.SALES.SALES_1996" consumed 8% of the database time.
7. I/O and Cluster wait for TABLE PARTITION "SH2.SALES.SALES_H1_1997" consumed 6% of the database time.
Below the rationale, there is a table with two rows, each showing a 'Show SQL Tuning' button and a bar chart with the value 16.5.

- 91.2% of the impact is from the SQL statements in the report
- Performance diagnostics data provided for SQL causing high DB resource usage

ASH Analytics – SQL dimensions



Demo

Using ASH Analytics for identifying SQL performance issues (4 min)

The Fix

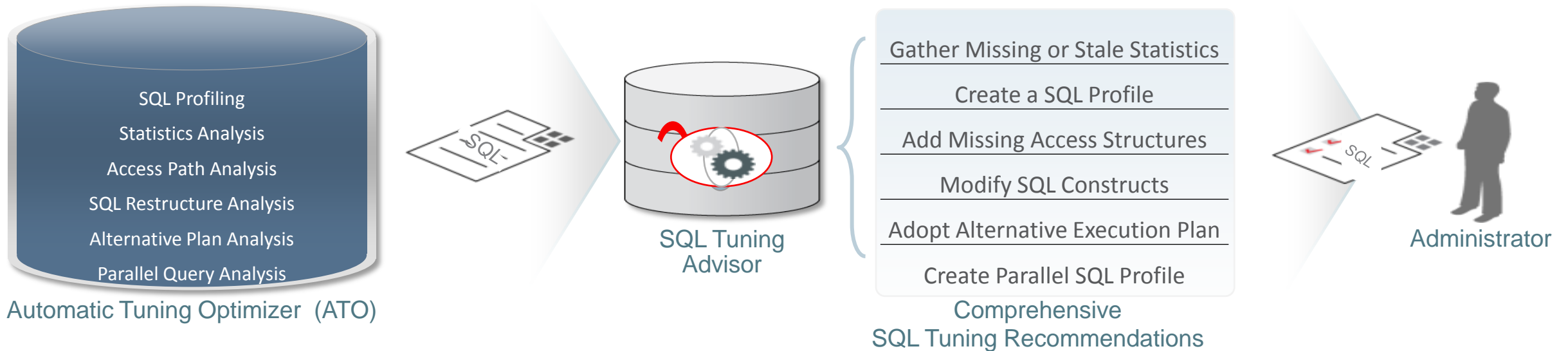
The Fix

Tuning the sub-optimal SQL

- Oracle provides a rich tool set for analyzing and resolving SQL Tuning problems
- Use the right toolset depending on the problem and its scope
 - Reactive tuning: Run SQL Tuning Advisor on high load SQL statements
 - Automatic SQL Advisor: System run, automated, recommendations for Top SQL for the week, day, month, etc.
 - Comprehensive workload tuning: Run SQL Access Advisor on the entire workload taking into account indexes and DML overhead
 - Complex run-time SQL performance analysis: Use Real-Time SQL Monitoring to analyze and understand complex run-time issues such as PQ skews, actual Vs estimated cardinalities, etc.
 - Batch job tuning: Use Database Operations Monitoring for analyzing and understanding batch jobs (similar to SQL Monitoring)

SQL Tuning Advisor

Comprehensive Analysis & Recommendations



- SQL Tuning Advisor

- Gives suggestions on the various problems identified during the diagnosis phase
- Uses the same CBO but has more time budget to perform comprehensive analysis
- Identifies alternate execution plans using real-time and historical performance data
- Recommends parallel profile if it will improve SQL performance significantly (2x or more)



SQL Profiles

- Contains auxiliary information collected by the ATO for a SQL statement
 - Customized optimizer settings
 - Compensates for missing or stale statistics
 - Compensates for errors in optimizer estimates
- Transparent Application Tuning – no change to SQL or application required
- Persistent: Works across shutdowns and upgrades
- Transportable across databases (10.2)
- Supports force matching for literals in SQL text
- Scope testing of SQL profile before making it available to other sessions (sqltune_category session parameter)

Real-Time SQL Monitoring

Looking Inside SQL Execution

Monitored SQL Executions Page Refreshed 8:24:28 AM GMT-0600 Auto Refresh Off

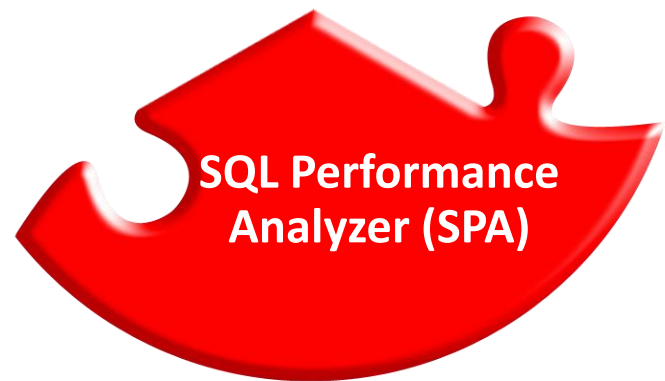
Top 100 By Last Active Time Type All Execution Detail SQL Detail Session Detail Cancel SQL

Status	Duration	Type	Instance ID	ID	SQL Plan Hash	User	Parallel	Database Time	IO Requests	Start	Ended	SQL Text
	9.0s		2	698r2m31u10u	1358400194	HIGH		8.0s		8:24:18 AM		SELECT 'B' tt1.ch_featurevalue_09_id ch_featurevalue_0...
	14.0s		2	698r2m31u10u	1358400194	HIGH		13.4s		8:24:13 AM		SELECT 'B' tt1.ch_featurevalue_09_id ch_featurevalue_0...
	1.2h		2	7waucm1ax9f3	1151357812	CRM		1.2h		7:15:05 AM		SELECT 'B' tt1.ch_featurevalue_09_id ch_featurevalue_0...
	1.2h		2	7waucm1ax9f3	1151357812	CRM		1.2h		7:10:27 AM		SELECT 'B' tt1.ch_featurevalue_09_id ch_featurevalue_0...
	11.0s		2	g4dzf4ak4rus2	2793146607	HIGH		11.7s		8:24:06 AM	8:24:17 AM	SELECT 'B' tt1.ch_featurevalue_09_id ch_featurevalue_0...
	11.0s		2	g4dzf4ak4rus2	2793146607	HIGH		11.4s		8:24:02 AM	8:24:13 AM	SELECT 'B' tt1.ch_featurevalue_09_id ch_featurevalue_0...
	1.7m		2	ay7s449vz34hk	760993341	HIGH		1.7m		8:22:23 AM	8:24:05 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'G' t3.el...
	1.7m		2	ay7s449vz34hk	760993341	HIGH		1.7m		8:22:19 AM	8:24:01 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'G' t3.el...
	1.0m		2	cx163n0q04xz7	3556938729	HIGH		1.0m		8:21:20 AM	8:22:22 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'G' t5.el...
	1.1m		2	cx163n0q04xz7	3556938729	HIGH		1.0m		8:21:15 AM	8:22:18 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'G' t5.el...
	1.6m		2	7j9ytd0f46aw1	3779689249	HIGH		1.6m		8:19:44 AM	8:21:18 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' t2.p...
	1.6m		2	7j9ytd0f46aw1	3779689249	HIGH		1.6m		8:19:38 AM	8:21:13 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' t2.p...
	39.0s		2	cmn5gjo0f0f8	3744261406	HIGH		38.8s		8:19:04 AM	8:19:43 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' t2.p...
	39.0s		2	cmn5gjo0f0f8	3744261406	HIGH		39.0s		8:18:57 AM	8:19:36 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' t2.p...
	1.1m		2	cpm5u0m6b68k5	3744261406	HIGH		1.1m		8:18:00 AM	8:19:04 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'Y' t4.el...
	1.1m		2	cpm5u0m6b68k5	3744261406	HIGH		1.1m		8:17:54 AM	8:18:57 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'Y' t4.el...
	42.0s		2	bxajng3z2vm1	3744261406	HIGH		41.7s		8:17:18 AM	8:18:00 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' t2.p...
	41.0s		2	bxajng3z2vm1	3744261406	HIGH		40.6s		8:17:13 AM	8:17:54 AM	SELECT /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' t2.p...

- Automatically monitors long running SQL
- Enabled out-of-the-box with no performance impact
- Monitors each SQL execution
- Exposes monitoring statistics
 - Global execution level
 - Plan operation level
 - Parallel Execution level
- Guides tuning efforts
- Bind values shown
- SQL level metrics
 - CPU, I/O requests, throughput, PGA, temp space
- Graphical explain plan
- I/O statistics for each operation

Demo

Real-time SQL Monitoring with
Adaptive Plans
Generate script for problem SQL
with binds and execute
(6 min)



VALIDATE

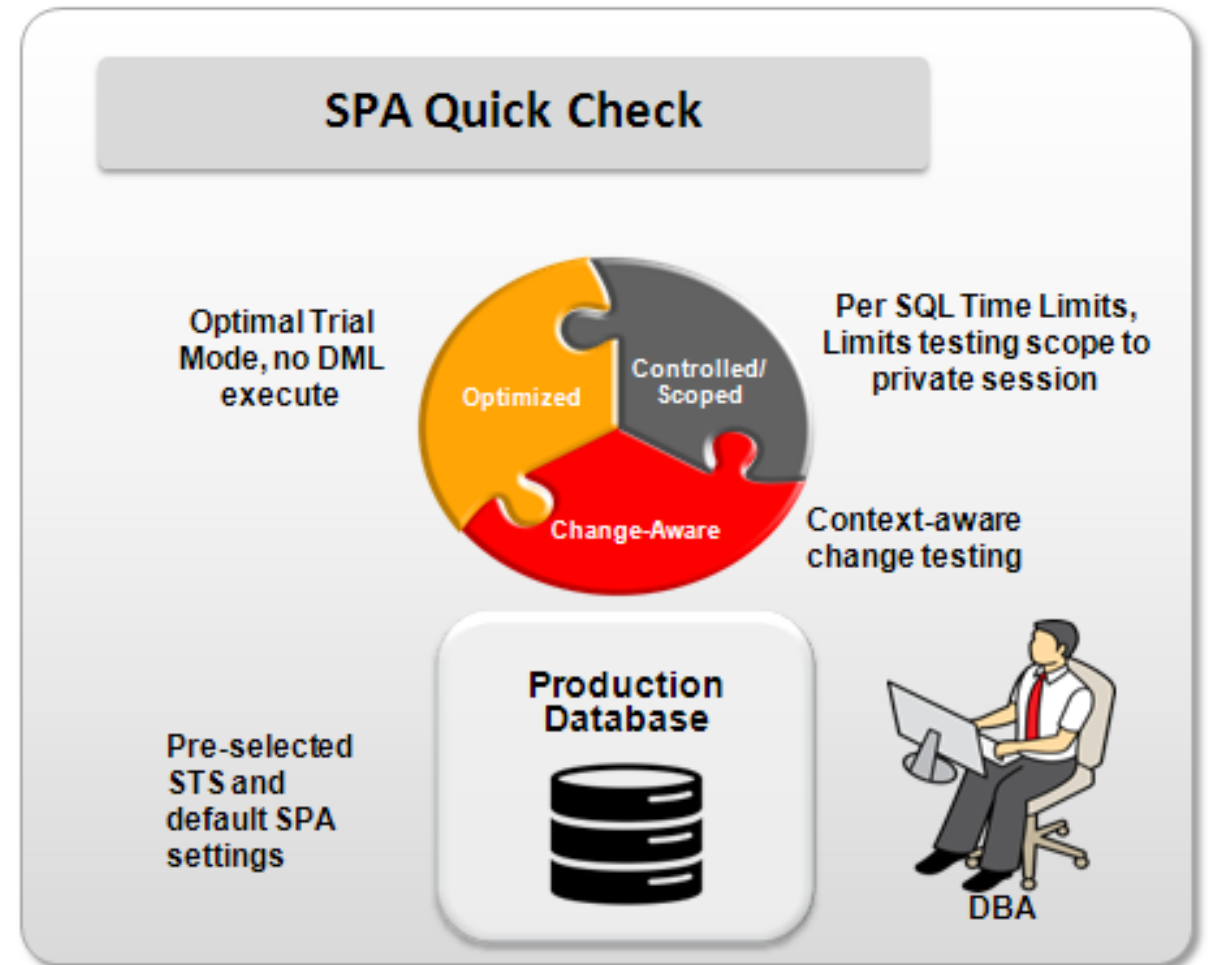
Validate: Predict the impact of the fix or tuning solution

- SQL Tuning Advisor provides comprehensive recommendations
 - Implement SQL Profiles
 - Refresh statistics on table, schema or database level
 - Add missing access structures
- However, before deploying in production it is recommended to validate the change
 - Test system may not be representative of production in terms of hardware and dataset
 - Avoid any production risk with validation through scoped session level testing

SPA Quick Check: Validating routine fixes

Assess routine performance changes

- Helps users quickly predict the impact of system changes on SQL workload on production system
- Designed to be used in production without impacting end-users with no overhead
 - Runs trials in optimal mode that consumes order of magnitude less system resources
- Context aware workflows, controlled and scoped impact assessment
- Useful for routine DBA activities such as SQL Profile validation, statistics refresh, init.ora changes



SPA Quick Check

Optimized

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 – up to 10x reduction

Multiple executions disabled

No full DML (execute Select part of workload)

SPA Quick Check

Controlled


Per-SQL time limit – protects from runaway SQL

Resource throttling - Associate with Resource Consumer Group

Testing scope limited to private session

SQL Performance Analyzer Setup

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


* SQL Tuning Set 

Trial Mode Optimal (Hybrid) Test Execute Explain Plan


Per-SQL Time Limit (Seconds)

Execute Full DML Yes No


Workload Impact Threshold(%) 

SQL Impact Threshold(%) 

Disable Multiple Executions Yes No

Comparison Metric 

Use Resource Consumer Group Yes No

Resource Consumer Group 

Save

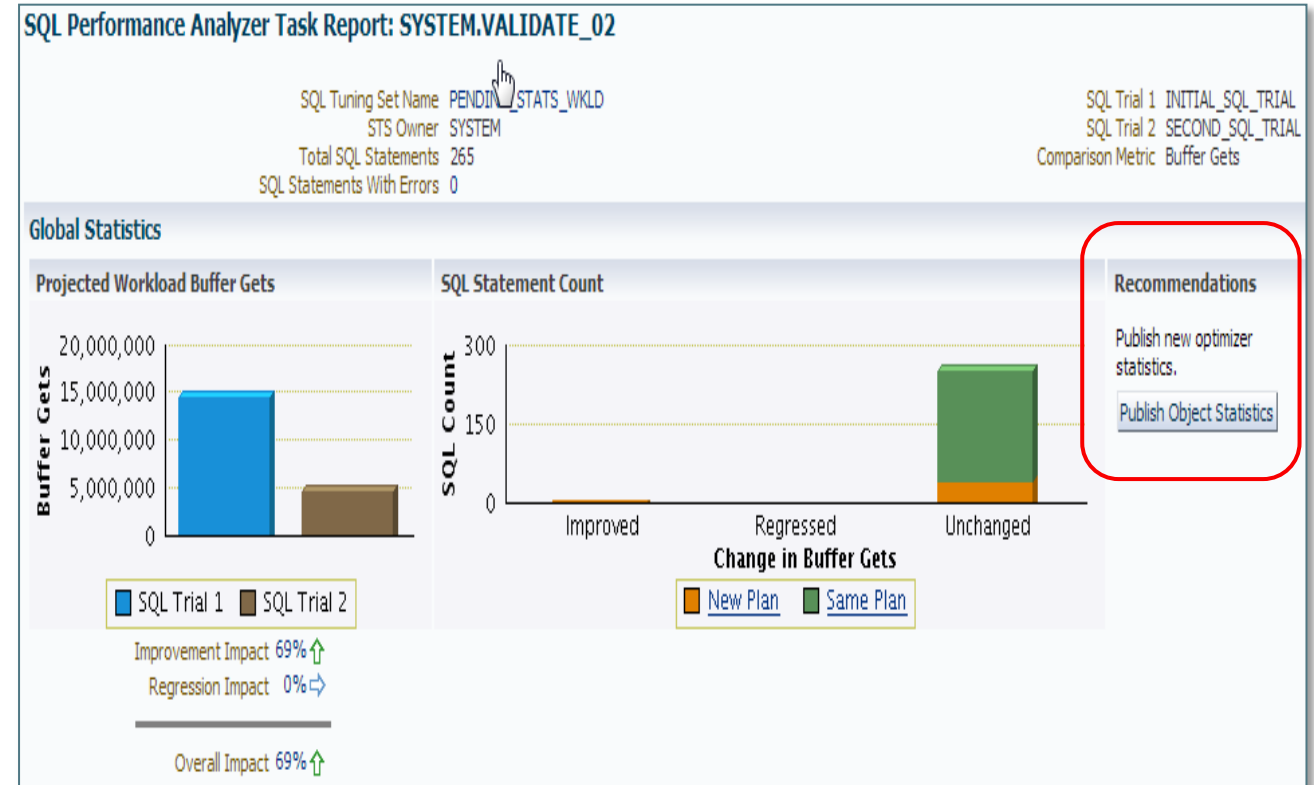
SPA Quick Check

Change-aware

Change-aware: Knows what change is being tested

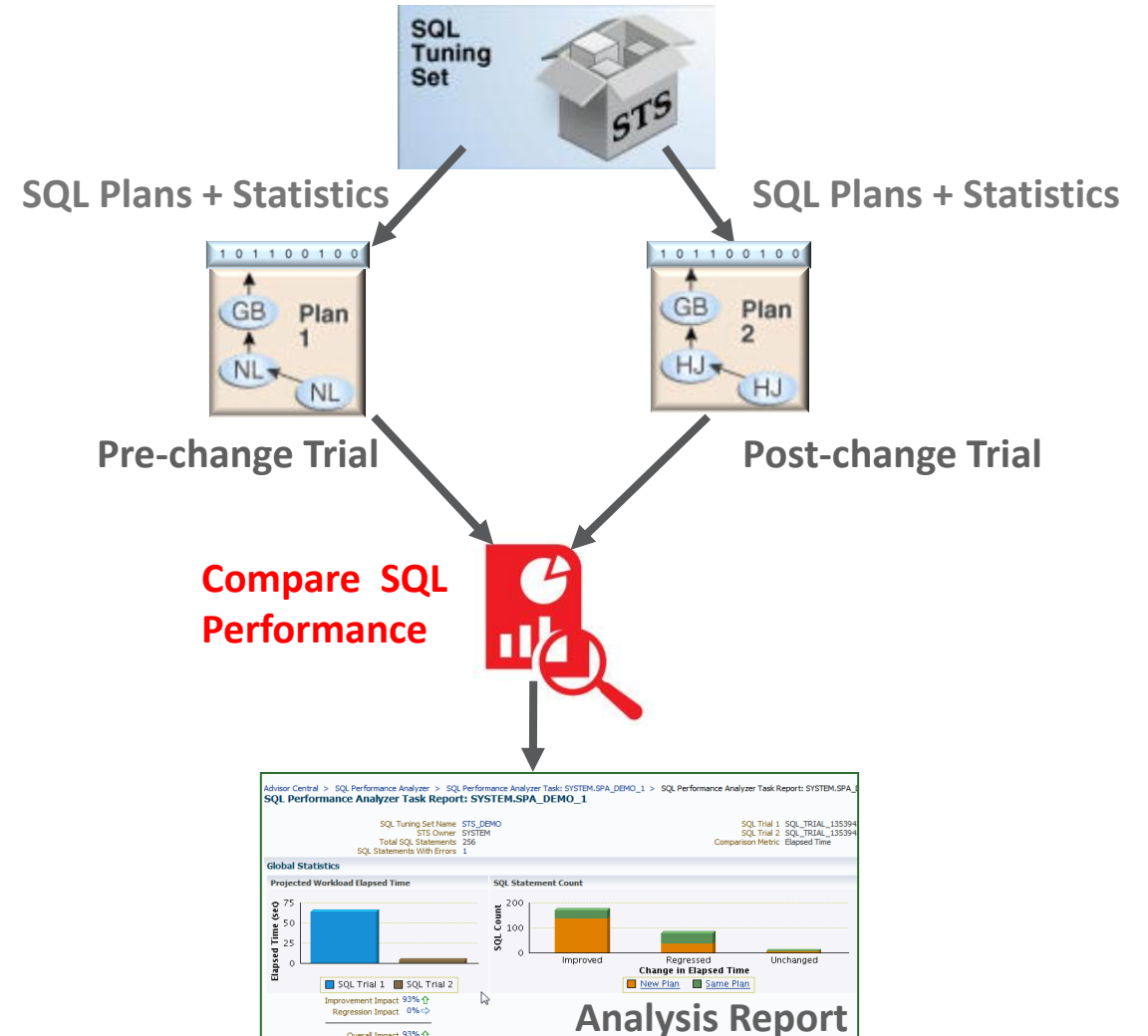
In-line with routine DBA tasks such as statistics gathering, init.ora parameter changes

Intelligently limits impact to private test session



SQL Performance Analyzer

- Helps users predict the impact of system changes on SQL workload
- Low overhead capture of SQL workload to SQL Tuning Set (STS) on production system
- Build different SQL trials (experiments) of SQL statements performance by test execution or explain plan
- Integrated with STS, SQL Plan Baselines, & SQL Tuning Advisor to form an end-to-end solution



Program Agenda

- 1 SQL Tuning: Challenges and existing solutions
- 2 Recommended tuning methodology
- 3 Methodology Demo: Cloud Migration**
- 4 New Features in Oracle Database 12.2

Demo Use Case

- As a DBA, you have been tasked by management to migrate your 11.2 database to the latest Cloud DB release. At the same time, the requirement is to make sure the performance is same or better than before, how can I accomplish this?

Solution: How to Validate Cloud Migration with SPA?

- Step 1: Capture representative workload into SQL Tuning Set (STS) on Production (On-premise)
- Step 2: Clone Database to Cloud using Oracle supported methods
 - For PDB use one-click migration
 - Non-PDB use Transportable Tablespaces or Datapump features
- Step 3: SPA Validation in Cloud
 - Can use EM13 Cloud Control, EM Express or API
 - Trial 1: Build from STS (Convert from STS)
 - Trial 2: Run against Cloud PDB (Test execute or explain plan)
 - Generate Reports to validate plan changes and performance differences
 - Use various metrics such as Buffer Gets, CPU time and Elapsed time to assess performance and fix any issues

Note: None of the other vendors have capability test on-premise and Cloud seamlessly

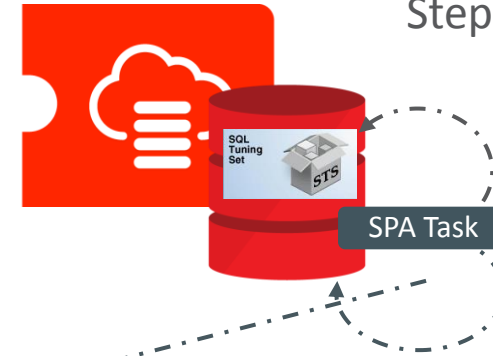
Validate Cloud Migration with SPA

On-Premise



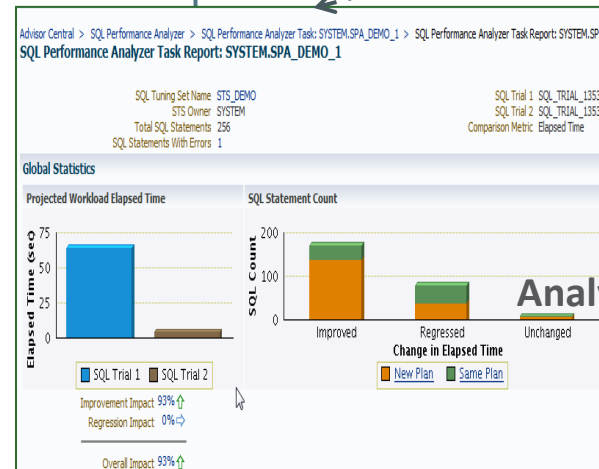
Step 1: Capture representative workload to STS
Step 2: Clone On-premise database to Cloud

Oracle Cloud



Step3a: Conduct SPA trials

Trial 1: Build (Convert) from STS
Trial 2: Test Execute or Explain Plan



Step 3b: Generate SPA Report and fix regressions

Demo

SPA— Cloud Use case (4 min)

Program Agenda

- 1 SQL Tuning: Challenges and existing solutions
- 2 Recommended tuning methodology
- 3 Methodology Demo: Cloud Migration
- 4 New Features in Oracle Database 12.2**

New Features in Oracle Database 12.2

- 1 SQL Tuning Advisor in Active Data Guard
- 2 SQL Performance Analyzer in Enterprise Manager Express
- 3 Database Operations Monitoring enhancement
- 4 Index Usage Statistics

SQL Tuning Advisor Support for Active Data Guard

- Problem

- ADG databases are widely used to offload reporting or ad hoc query-only jobs from primary
- Reporting workload profile is different from primary and often requires tuning

- Solution

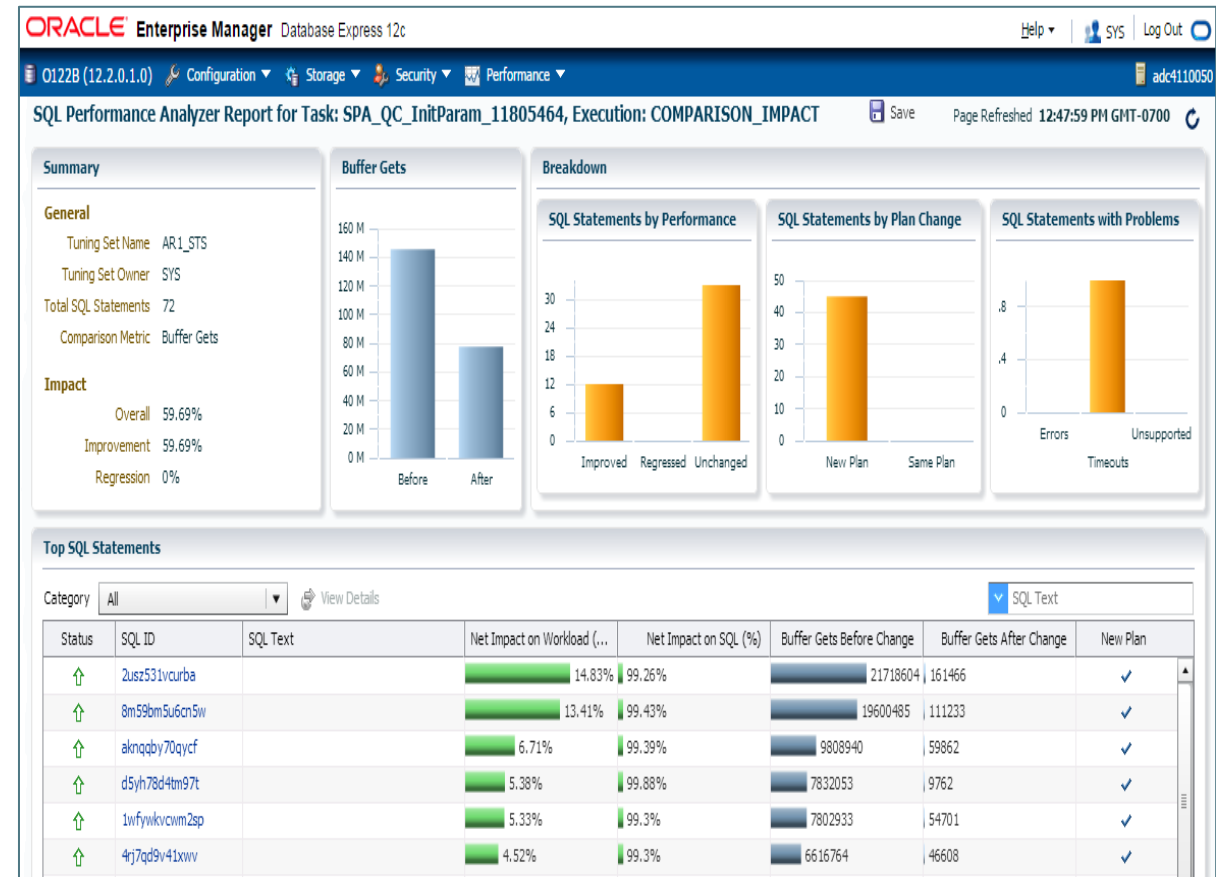
- Oracle Database 12.2 introduces the ability to tune SQLs workloads running on ADG database
- All SQL Tuning Advisor tasks issued at the standby
 - Create tuning task, execute tuning task and implement SQL Tuning Advisor recommendations
 - Test execution (heavy lifting) happens on standby, only minimal write related activity on primary
- The required data for the above tasks are fetched from primary over a database link from standby
- Task details and tuning results are stored at primary and the essential data required to construct the report is accessed remotely from primary
- The report is constructed locally at the standby, with no CPU overhead on primary

Demo

SQL Tuning Advisor Support for
Active Data Guard (5 min)

EM Express (DB12.2)

- SQL Performance Analyzer (SPA) and SPA Quick Check Support
 - SPA Quick Check workflows in context of day to day administrative tasks
 - SPA for traditional testing



Database Operations Monitoring enhancement

- Extended DB Operations Monitoring for external sessions
 - DBA can start/stop a DB Operation in a different arbitrary session, by specifying Session ID and Serial #
 - New parameters added to the function `DBMS_SQL_MONITOR.BEGIN_OPERATION`

- "SESSION_ID"
- "SESSION_SERIAL"

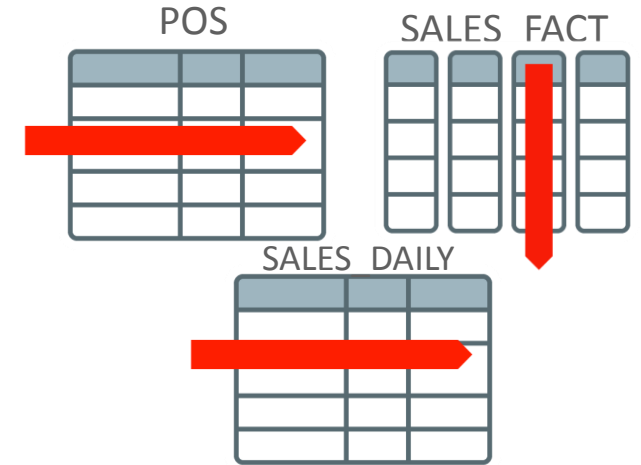
```
EXEC :eid := DBMS_SQL_MONITOR.BEGIN_OPERATION(  
    dbop_name => 'DBOP_EXAMPLE',  
    session_id => 24,  
    session_serial => 2355 );
```

Status	Duration	Type	ID	SQL Plan Hash	User
	16.0s		5gyh8uu8jgbx2	1727794596	DWH_TEST@MFG
	2.1m		DBOP_EXAMPLE_...		DWH_TEST@MFG

- Similar but an easier option of a DBA to enable `SQL_TRACE` on other user's session

Indexes Usage Statistics

- How can I tune my database better?
 - Too many indexes causing slow performance and using space
 - Which indexes should be dropped on Exadata?
 - Which columns are suitable for Database In-memory?
- Currently, limited support in the database to track index usage with low overhead
 - Disabled by default or limited to Top 10-15 objects, no indication how indexes are used
- Oracle Database 12.2 introduces new framework to automatically track index usage over time with no overhead
- New views provide cumulative statistics for each index since database startup
 - {USER_|DBA}_INDEX_USAGE: Track total accesses, executions and usage histogram
 - Columns: Name, total_access_count, total_exec_count, total_rows_returned, bucket_0_access_count, bucket_1_access_count, bucket_2_10_rows_returned,....., bucket_100_plus_rows_returned



Summary: Eliminating Guesswork from SQL Tuning

Recommended Tuning Methodology and Toolset



- Find-Fix-Validate methodology offers a comprehensive and robust mechanism for SQL Tuning, improves DBA productivity by an order of magnitude or more
- Easy to use both on production and test system for routine system changes
- Rich toolset to support every step of the methodology

• Find

- ASH Analytics
- ADDM
- Performance Hub



• Fix

- SQL Tuning Advisor
- Real-Time SQL Monitoring
- Performance Hub



• Validate

- SPA Quick Check
- SPA
- Database Replay



Q & A

Integrated Cloud

Applications & Platform Services

ORACLE®

Real-Time PL/SQL Monitoring



- PL/SQL execution no longer a “black box”
 - Answers questions like “why did my DBMS_STATS job take twice as long this time?”
- Shows global (PL/SQL) and SQL level statistics
- Each SQL called by PL/SQL recursively monitored
- Drill-down to slow SQL for diagnosing unexpected PL/SQL behavior