

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.



ORACLE

Find Fix & Validate

Fool-proof Approach for Managing Database Performance

Björn Bolltoft

Consulting Product Manager
Oracle Product Development

Kurt Engeleiter

Principal Product Manager
Oracle Product Development

Daniel Suherman

Principal Product Manager
Oracle Product Development



Top Challenges

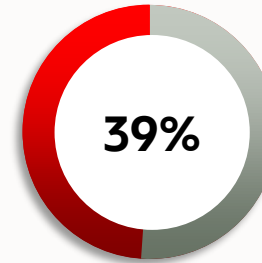
Database Management



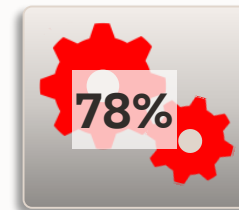
For the Complete Technology & Database Professional



Key Takeaway:
Improve & Ensure Higher Quality of Service



Handle more than
50 DBs each



Downtime resulting
from untested changes

Top Challenges

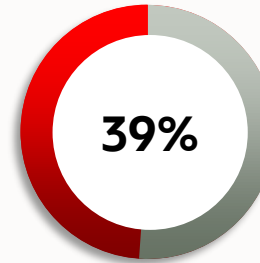
Database Management



For the Complete Technology & Database Professional

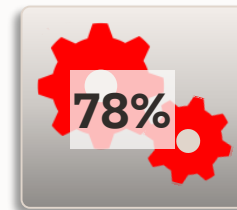


Key Takeaway:
Improve & Ensure Higher Quality of Service



Handle more than
50 DBs each

Automate

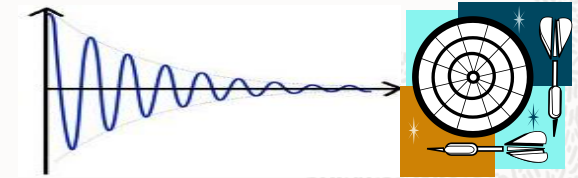


Downtime resulting
from untested changes

Validate

Past DB Tuning Process

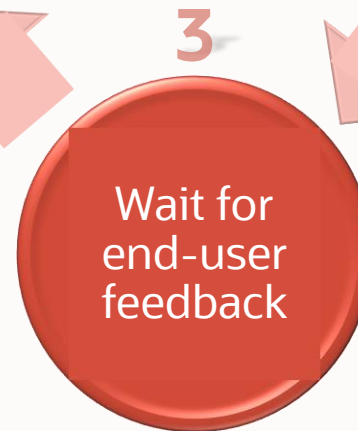
Inefficient, Inaccurate, Manual



Will optimizer_index_cost_adj
improving my sequential reads...



Default value is 100,
...so let's set it to 50...



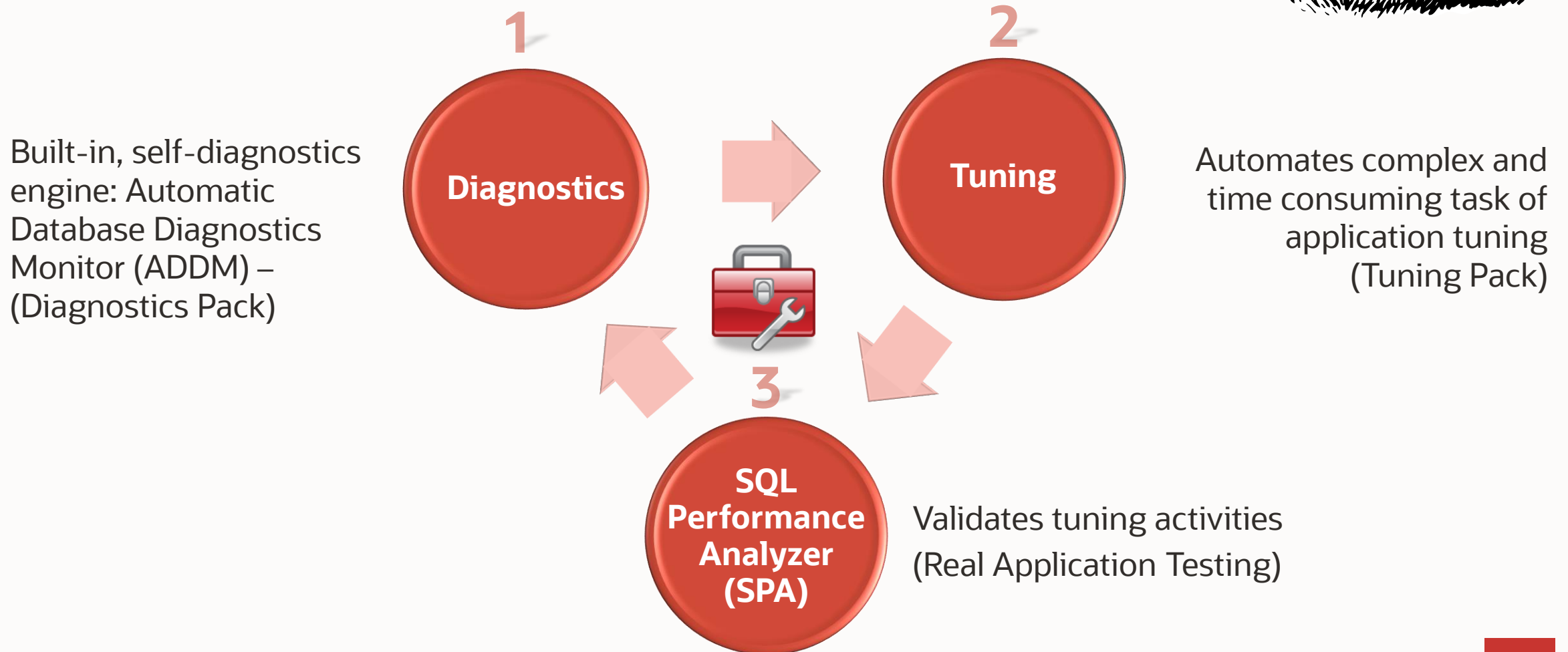
Did it work?

Are customers complaining?

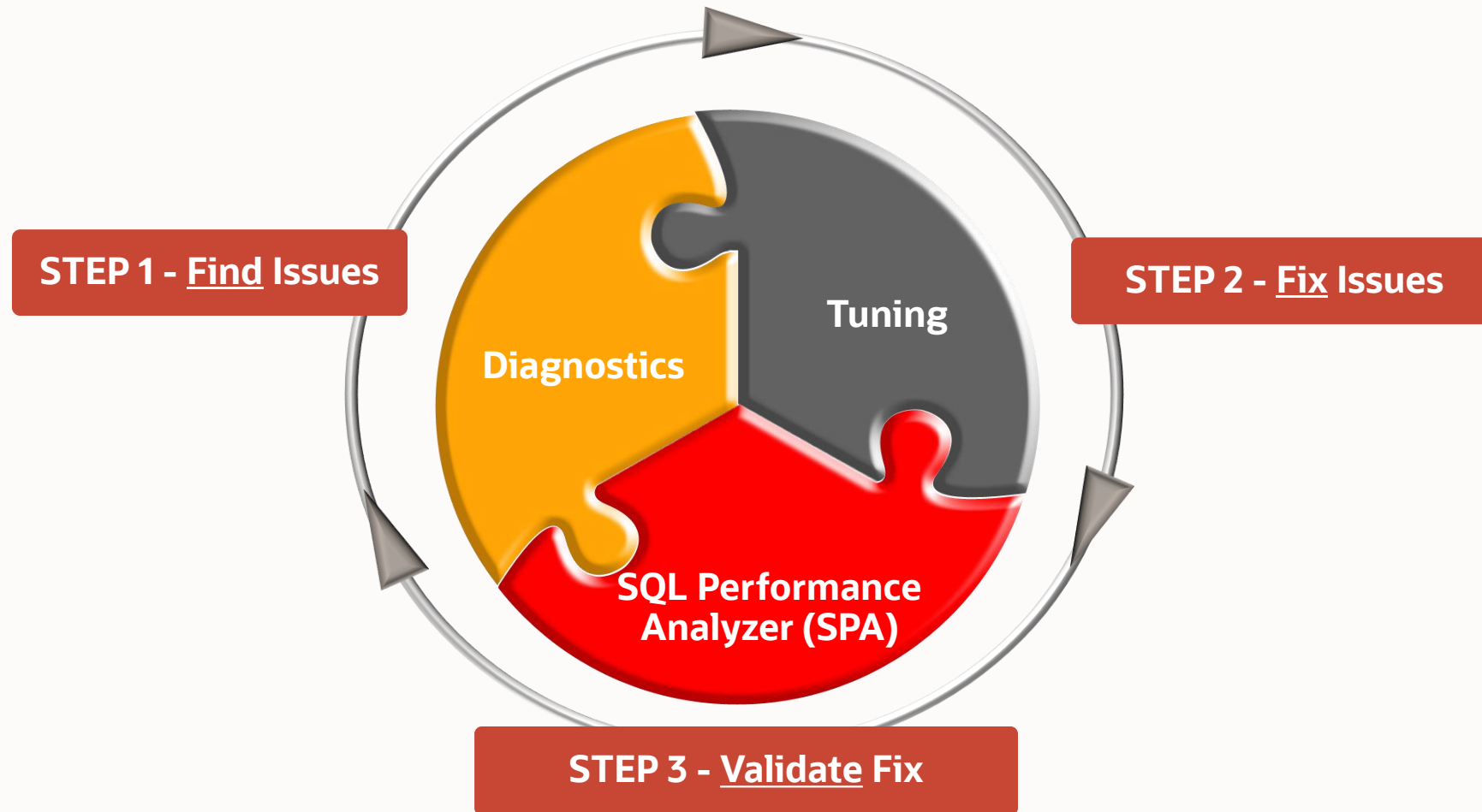
The “trial and error” method can consume more than 50% of the DBA time

Future Generation DB Tuning Process

Effective, Accurate, Automated



Find → Fix → Validate



Database Time (DB Time)

- Total time in database calls by foreground sessions
- Includes CPU time, IO time and non-idle wait time
- DB Time <> response time
- New metric for Oracle performance analysis

Database time is total time spent by user processes either actively working or actively waiting in a database call.

Fundamental Concepts

Active Session =

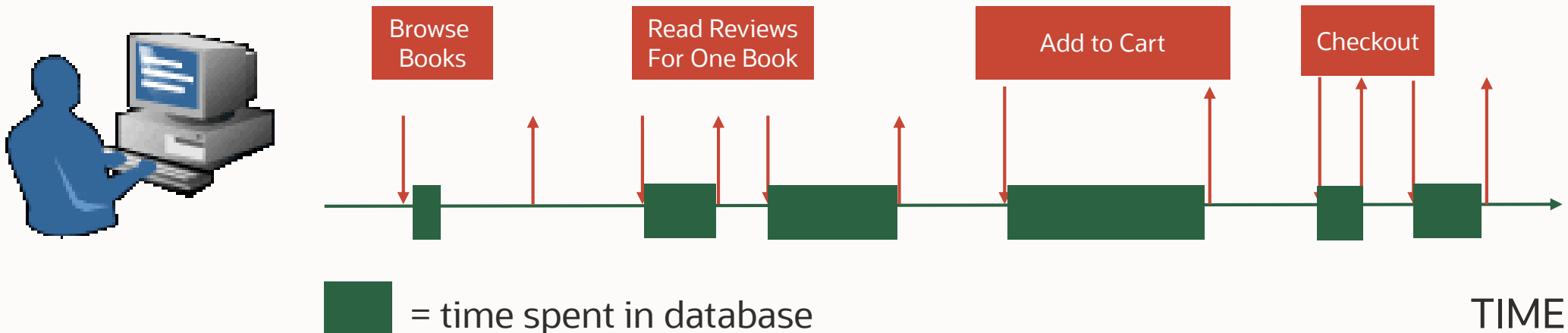
Session currently spending time in a database call

Database Time (DB Time) =

Total time session spent in all database calls

Average Activity of the Session (% Activity) =

The ratio of time active to total wall-clock time





THE FIND

Automatic Workload Repository

Challenge

- How does a DBA monitor and be aware of problems as they arise
- Capture associated database statistics for problem detection and tuning



Solution

- AWR
- **Benefit:**
By automating the gathering of database statistics for problem detection and tuning, AWR serves as the foundation for database self-management.

AWR Warehouse

Challenge

- Default retention period of 8 days prevents diagnosis of long term performance problems
- Compare performance during this quarter's books close with last quarter's
- Increasing AWR retention period increases storage overhead and cost in critical production environments



Solution

- AWR Warehouse
- **Benefit:**
The AWR Warehouse is set up with the same objects as in a standard AWR schema of any Oracle database. The enhancement lies in the partitioning, that allows for quick loads, efficient querying and when requested, effective purging of unwanted data.

Active Session History

Challenge

- DBA on a production system and get an emergency call like “The Database is dead slow!”
- Identify the root cause of performance issues at a precise point in the past—even when the sessions have disconnected.
- Increasing AWR retention period increases storage overhead and cost in critical production environments



Solution

- Active Session History
- **Benefit:** Enables targeted performance analysis of transient problems
- Built into the Oracle kernel and highly optimized

Active Session History (ASH)

- Samples active sessions every one second into memory (v\$active_session_history)
- Helps answers questions like “has this SQL gotten slower over time?”
- Direct access to kernel structures
- One of ten samples flushed to AWR at every snapshot
- Data captured includes:
 - SID
 - SQL ID
 - Program, Module, Action
 - Wait event#
 - Object, File, Block
 - actual wait time (if captured while waiting)



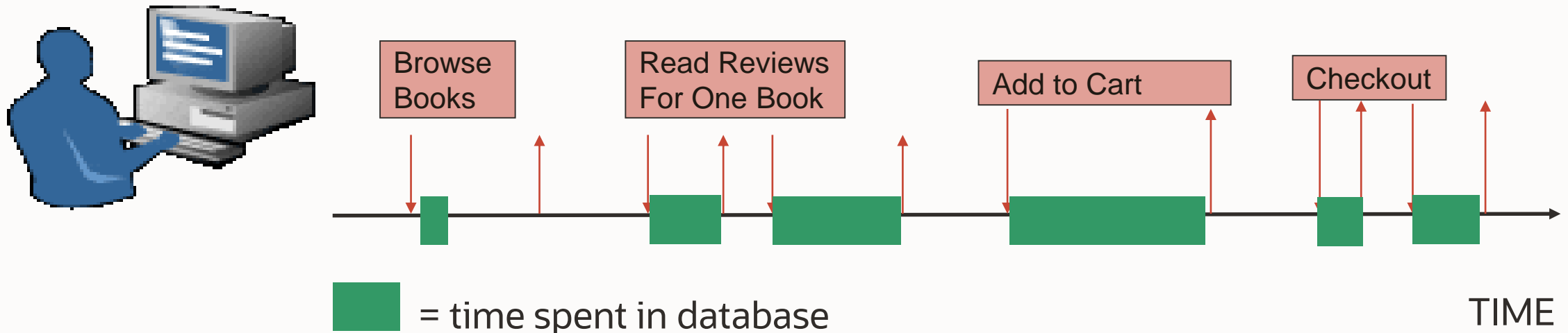
Enables targeted performance analysis of transient problems

ASH samples are proxy for DB Time

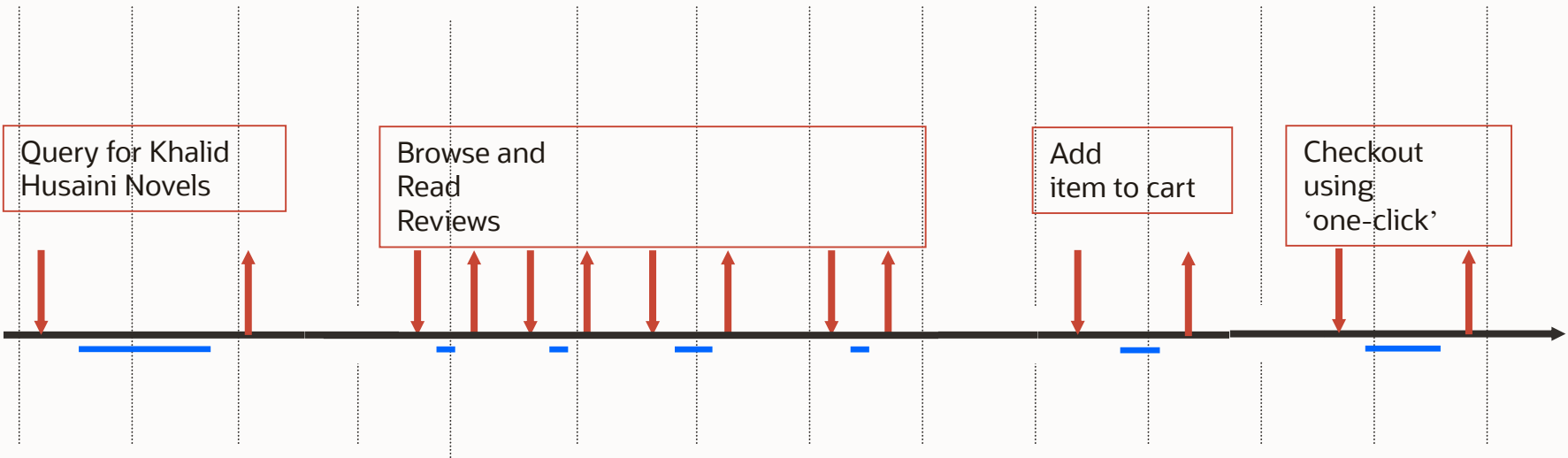
Each sample represents a second of session activity

Sum the seconds to compute DB Time

Average Activity of the Session (% Activity) =
DB Time/Elapsed Time



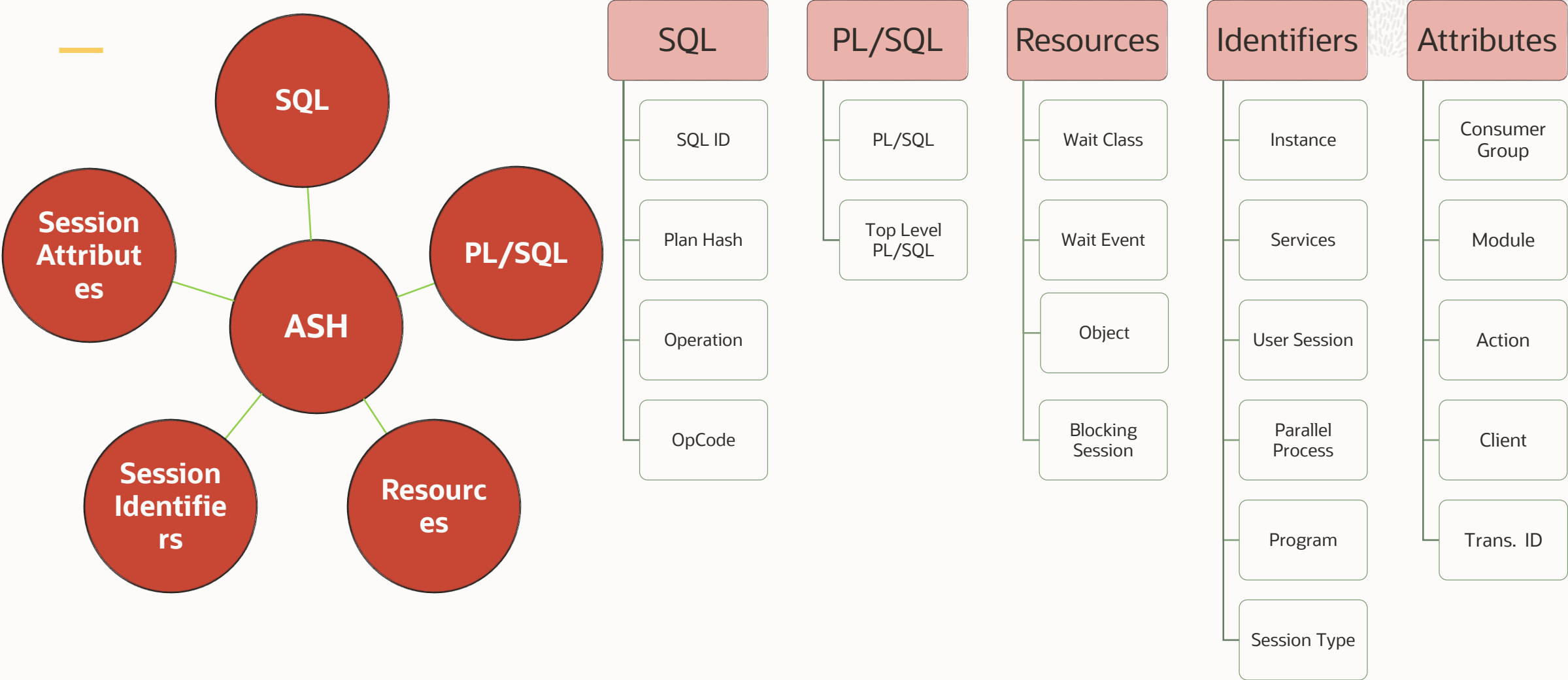
Session History (ASH)



Time	SID	Module	SQL ID	State	Event
7:38:26	213	Book by author	qa324jffritcf	WAITING	db file sequential read
7:42:35	213	Get review id	aferv5desfzs5	CPU	
7:50:59	213	Add to cart	hk32pekfcbdfr	WAITING	buffer busy wait
7:52:33	213	One click	abngldf95f4de	WAITING	log file sync



User Session Performance Dimensions



Automatic Database Diagnostics Monitor (ADDM)

Challenge

- Perform accurate and timely diagnosis of the problem before making any changes to a system
- Diagnose the root causes of performance problems

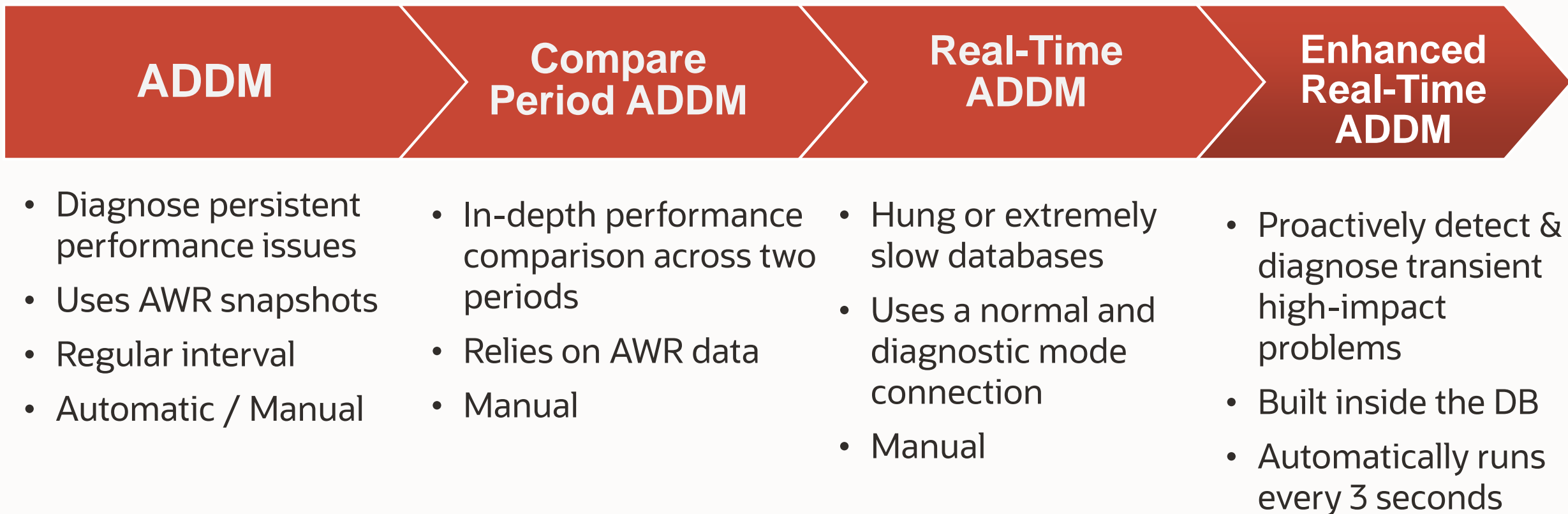


Solution

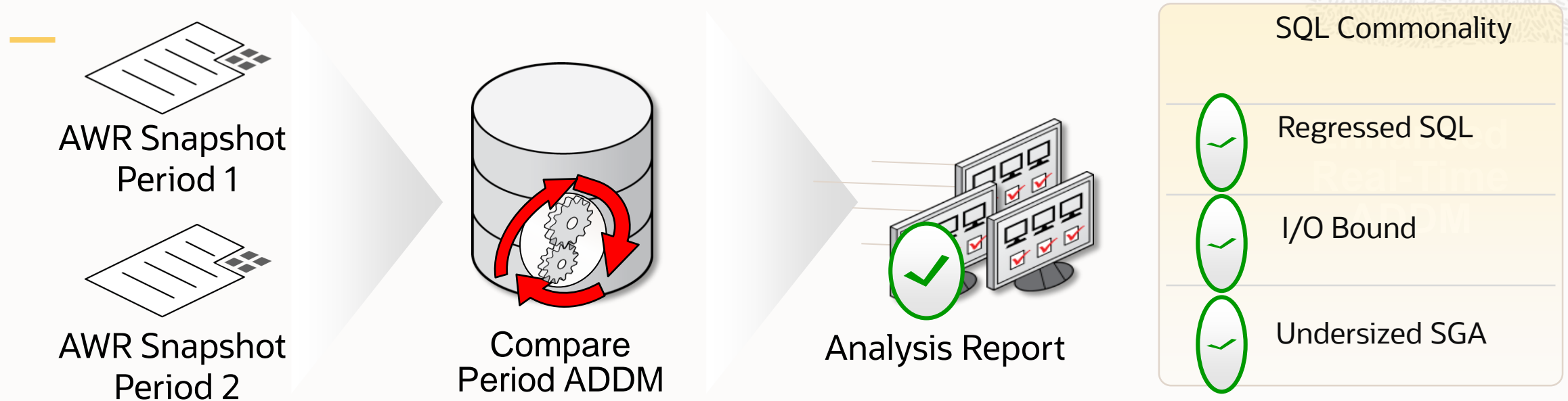
- ADDM
- **Benefit:**
ADDM makes it possible for the Oracle Database to diagnose its own performance and determine how any identified problems can be resolved.
- Root cause analysis,
Correction recommendations
Impact and benefits analysis

Automatic Database Diagnostics Monitor (ADDM)

Continuous Evolution in Database Performance Management



Compare Period ADDM

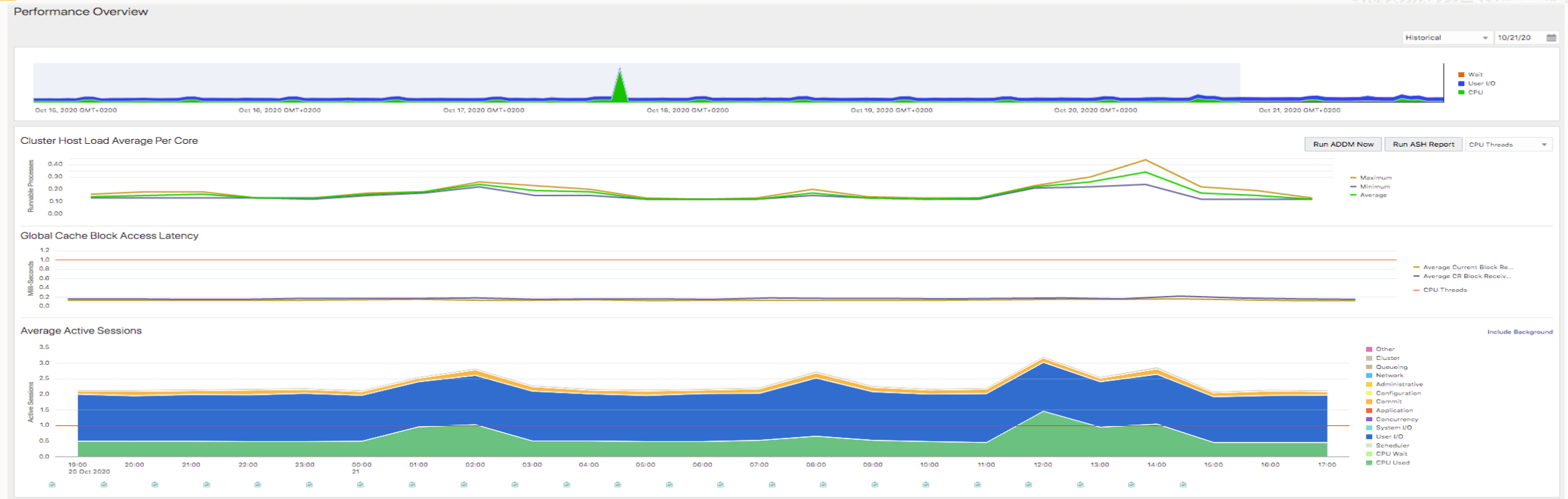


- Full ADDM analysis across two AWR snapshot periods
- Detects **causes**, measure **effects**, then **correlates** them
 - Causes: workload changes, configuration changes
 - Effects: regressed SQL, reach resource limits (CPU, I/O, memory, interconnect)
- Makes actionable recommendations along with quantified impact

ORACLE

FIND: Theory to Practice

DB Time Theory to Practice: EM Performance page



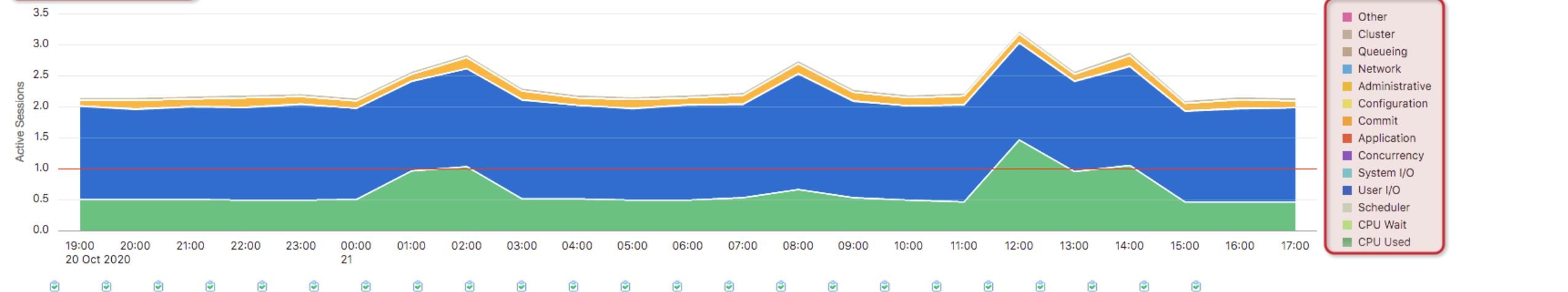
Active Sessions by wait class over time

Colored area = Amount of DB time , More the DB time , More the Problem

“Click on the big stuff...”

DB Time Theory to Practice: EM Performance page

Average Active Sessions

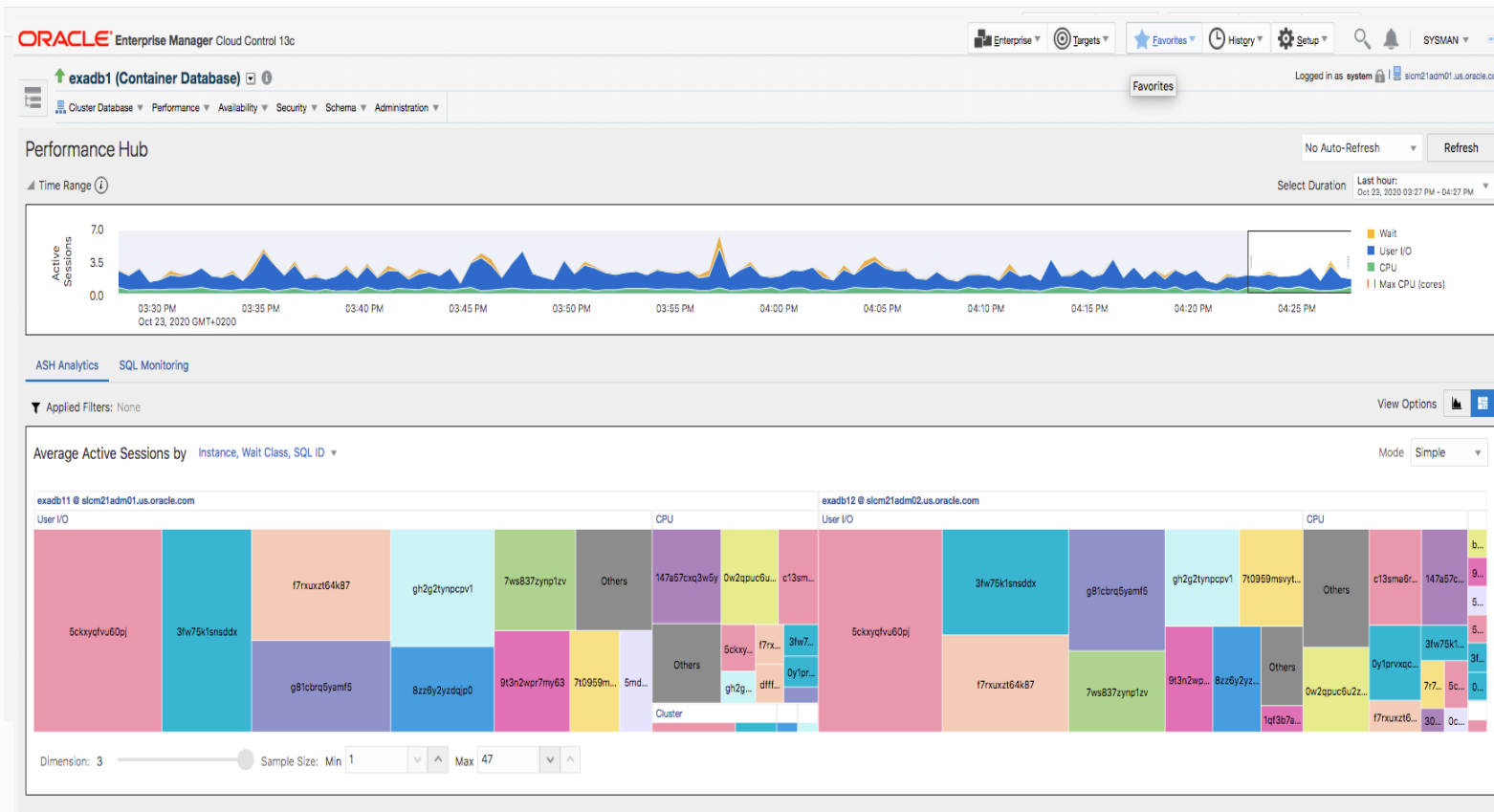


Active Sessions by wait class over time

Colored area = Amount of DB time , More the DB time , More the Problem

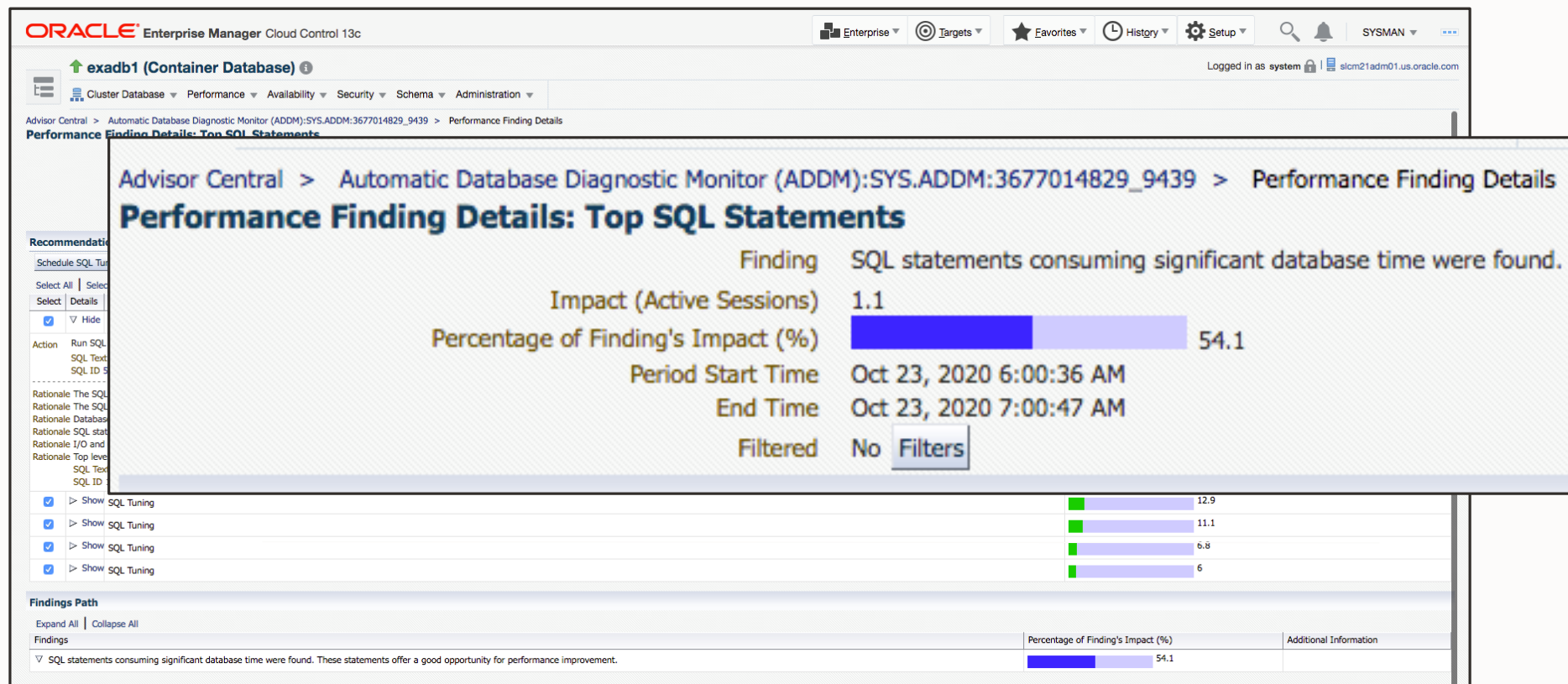
“Click on the big stuff...”

ASH Theory to Practice: EM ASH Analytics Page

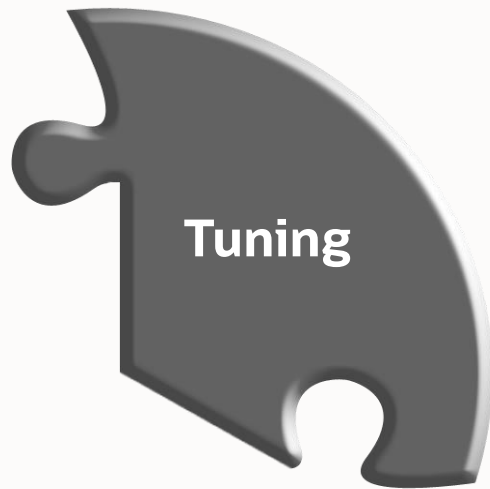


- Graphical ASH report for advanced analysis
- Select any time period for analysis
- Analyze performance across many dimensions
- Provides visual filtering for recursive drill-downs
- Different visualizations: Stacked chart or Tree Map
- Collaborate with others using Active Reports

ADDM to Practice: Automatic Diagnosis and Recommendations

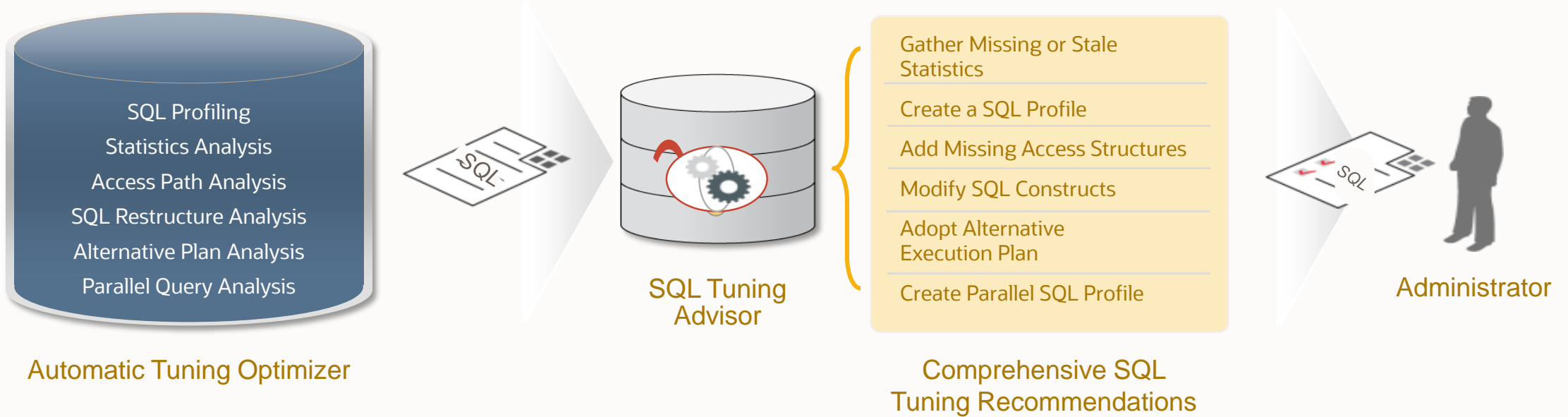


- 54.1% of the impact is from the SQL statements in the report
- Performance diagnostics data provided for SQL causing high DB resource usage



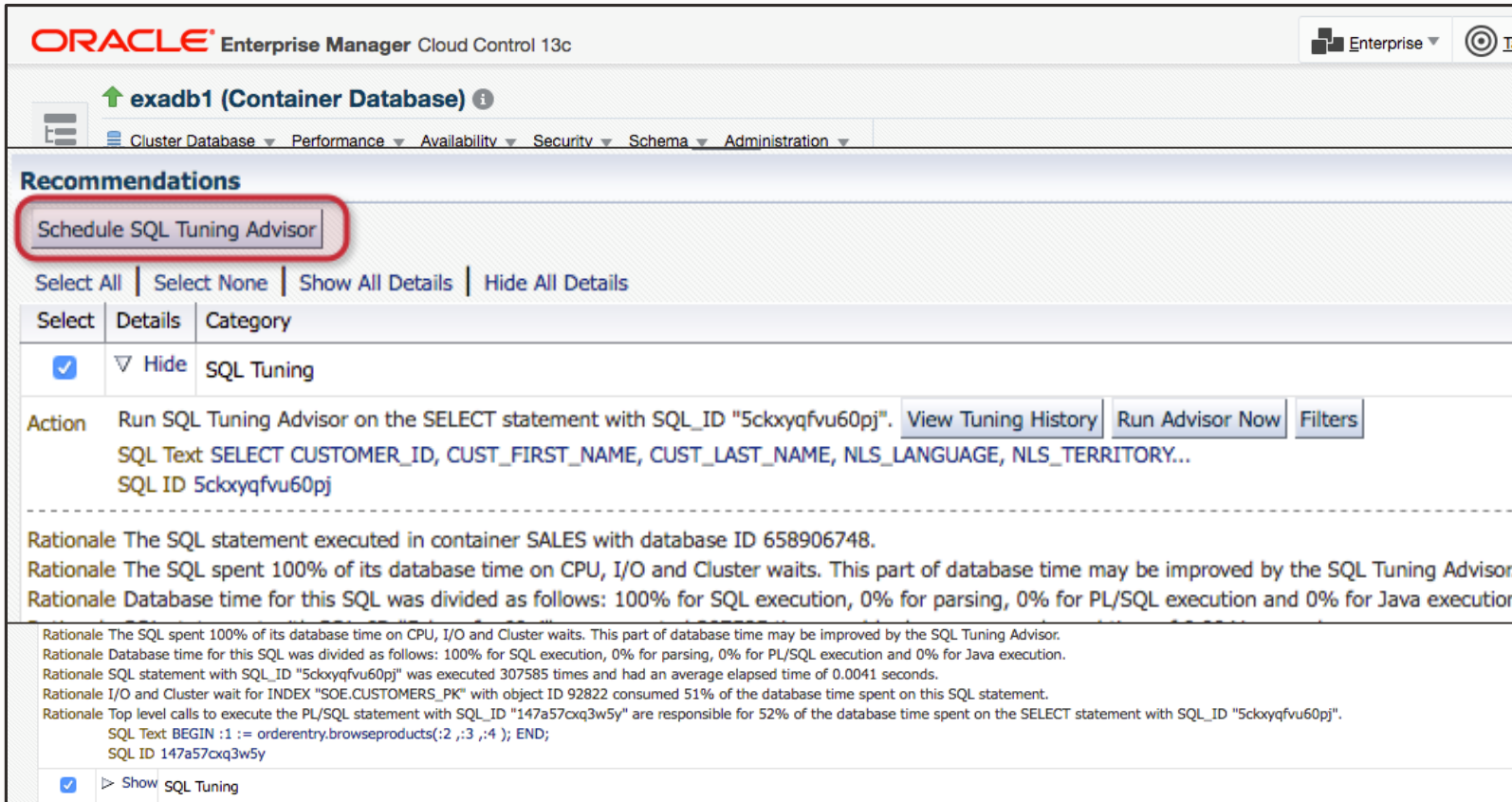
THE FIX

Automatic SQL Tuning



- 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)

The Fix: ADDM tuning -> SQL Tuning Advisor



The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface for a container database named 'exadb1'. The 'Recommendations' section is active, and the 'Schedule SQL Tuning Advisor' button is highlighted with a red box. Below this, there are links for 'Select All', 'Select None', 'Show All Details', and 'Hide All Details'. A table lists the recommendations, with the first one selected (checkbox checked) and categorized as 'SQL Tuning'. The recommendation details include the action to run the SQL Tuning Advisor on a specific SQL statement, the SQL text, and the SQL ID. The rationale explains that the SQL statement spent 100% of its database time on CPU, I/O, and Cluster waits, and that the SQL Tuning Advisor can improve this. The SQL text is 'SELECT CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME, NLS_LANGUAGE, NLS_TERRITORY...' and the SQL ID is '5ckxyqfvu60pj'.

ORACLE Enterprise Manager Cloud Control 13c

Enterprise

exadb1 (Container Database)

Cluster Database Performance Availability Security Schema Administration

Recommendations

Schedule SQL Tuning Advisor

Select All | Select None | Show All Details | Hide All Details

Select	Details	Category
<input checked="" type="checkbox"/>	Hide	SQL Tuning

Action Run SQL Tuning Advisor on the SELECT statement with SQL_ID "5ckxyqfvu60pj". [View Tuning History](#) [Run Advisor Now](#) [Filters](#)

SQL Text SELECT CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME, NLS_LANGUAGE, NLS_TERRITORY...

SQL ID 5ckxyqfvu60pj

Rationale The SQL statement executed in container SALES with database ID 658906748.

Rationale The SQL spent 100% of its database time on CPU, I/O and Cluster waits. This part of database time may be improved by the SQL Tuning Advisor.

Rationale Database time for this SQL was divided as follows: 100% for SQL execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java execution.

Rationale The SQL spent 100% of its database time on CPU, I/O and Cluster waits. This part of database time may be improved by the SQL Tuning Advisor.

Rationale Database time for this SQL was divided as follows: 100% for SQL execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java execution.

Rationale SQL statement with SQL_ID "5ckxyqfvu60pj" was executed 307585 times and had an average elapsed time of 0.0041 seconds.

Rationale I/O and Cluster wait for INDEX "SOE.CUSTOMERS_PK" with object ID 92822 consumed 51% of the database time spent on this SQL statement.

Rationale Top level calls to execute the PL/SQL statement with SQL_ID "147a57cxq3w5y" are responsible for 52% of the database time spent on the SELECT statement with SQL_ID "5ckxyqfvu60pj".

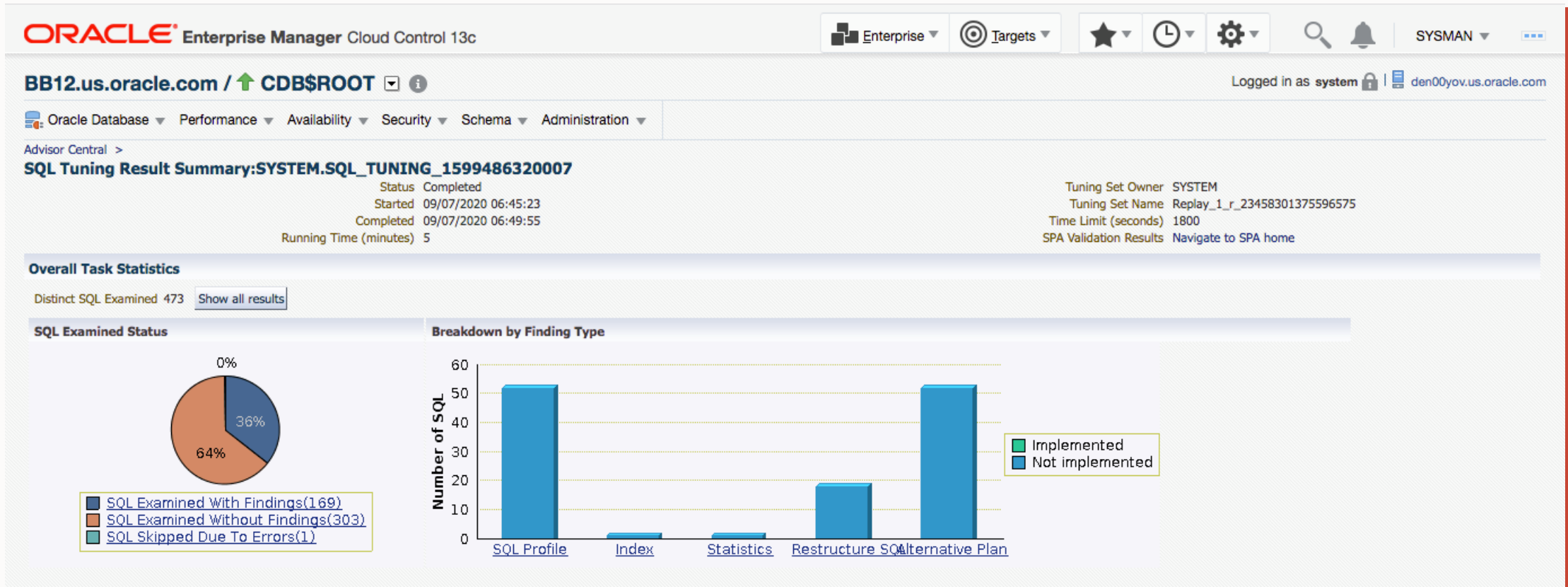
SQL Text BEGIN :1 := orderentry.browseproducts(:2 ,:3 ,:4); END;

SQL ID 147a57cxq3w5y

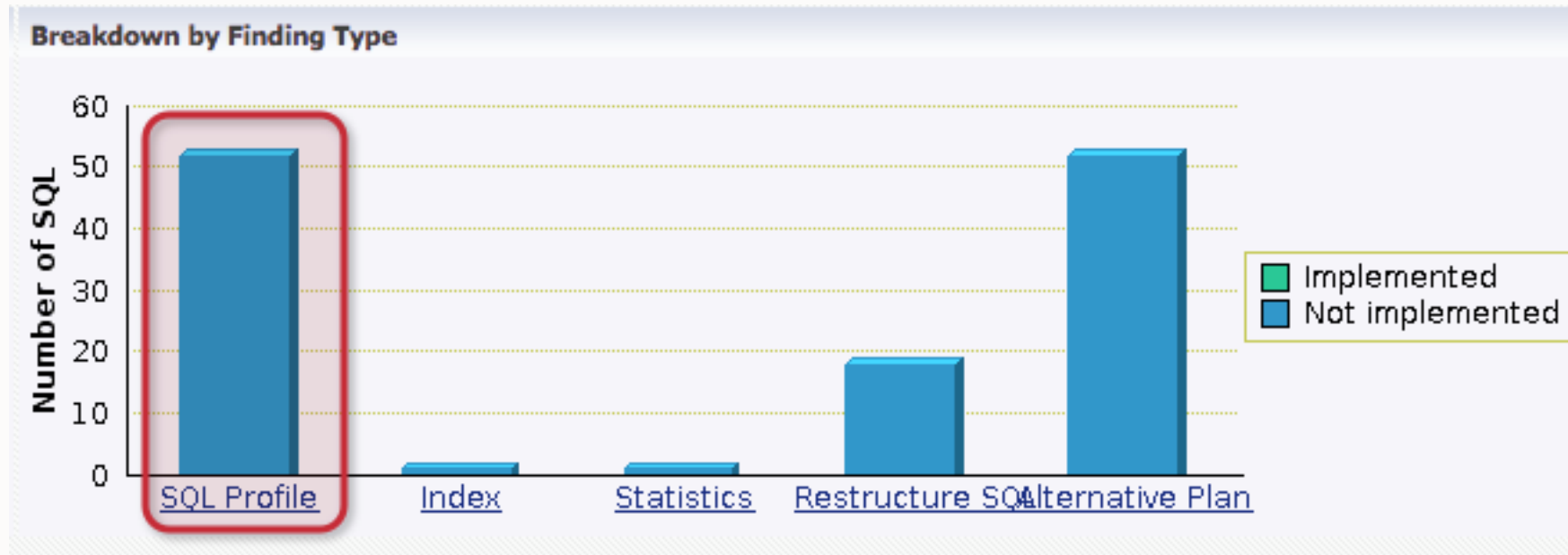
☒ Show SQL Tuning

- Run the problematic performing statements through Tuning Advisor
- We will automatically capture the problematic statement
- Top SQL statements are automatically captured in AWR
- Seamless integration between Diagnostics and Tuning

The Fix: SQL Tuning Advisor



The Fix: Tuning Advisor – SQL Profile



- Comprehensive analysis and multiple alternatives to improve performance
- SQL Profiles can be implemented transparently & non-intrusively to tune SQL statements
- As an example, let's gather statistics in pending mode...

What is SQL Profile?

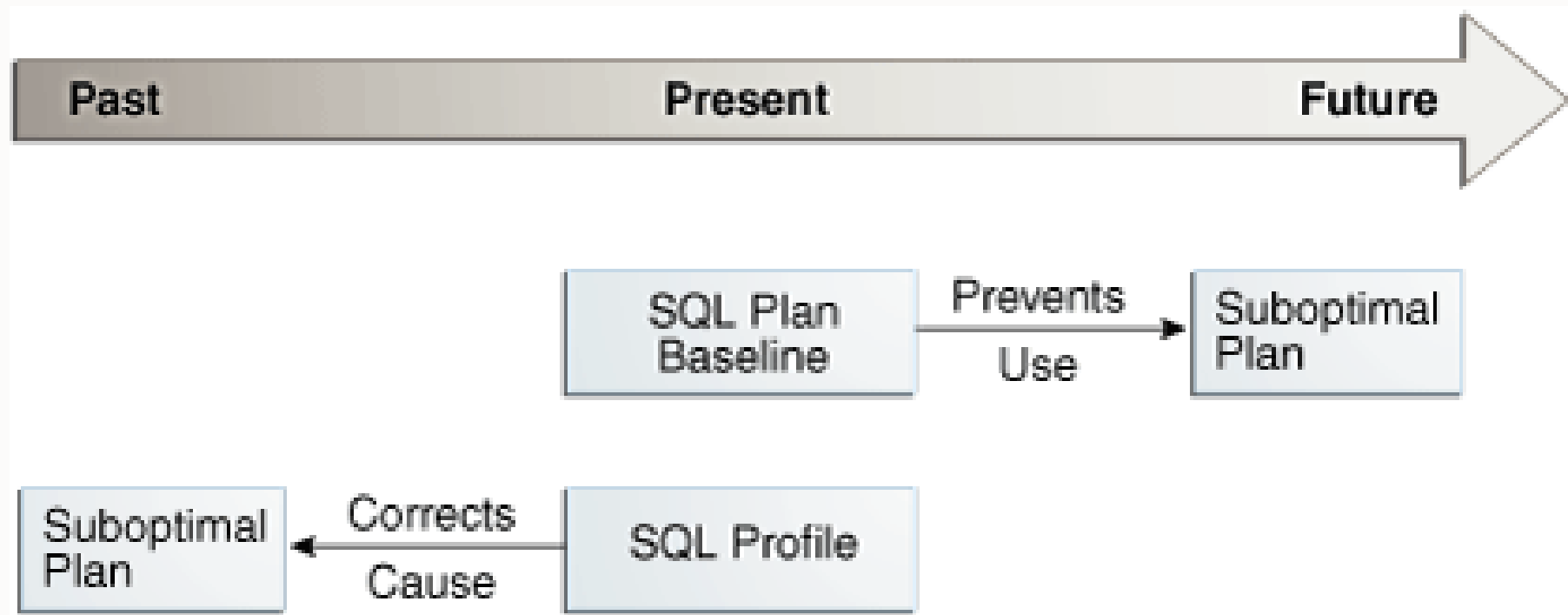
- Contains auxiliary information collected by the ATO for a SQL statement
 - Customized optimizer settings
- Based on past execution history (e.g., first_rows vs. all_rows)
 - Compensation for missing or stale statistics
 - Compensation for errors in optimizer estimates
- Estimation errors occur due to data skews and correlations, complex filters and joins
- Doesn't require any change to the SQL text
 - Ideal for Packaged Apps
- Persistent: Works across shutdowns & upgrades
- Transportable across databases (10.2)
- Force Matching for Literals: This setting may be useful for queries that use only literal values
- SQLTUNE_CATEGORY: Enables you to test a profile in a restricted environment before making it available to other sessions. Can be enforced at session level or system level, this defaults to system level

What SQL Plan Baselines?

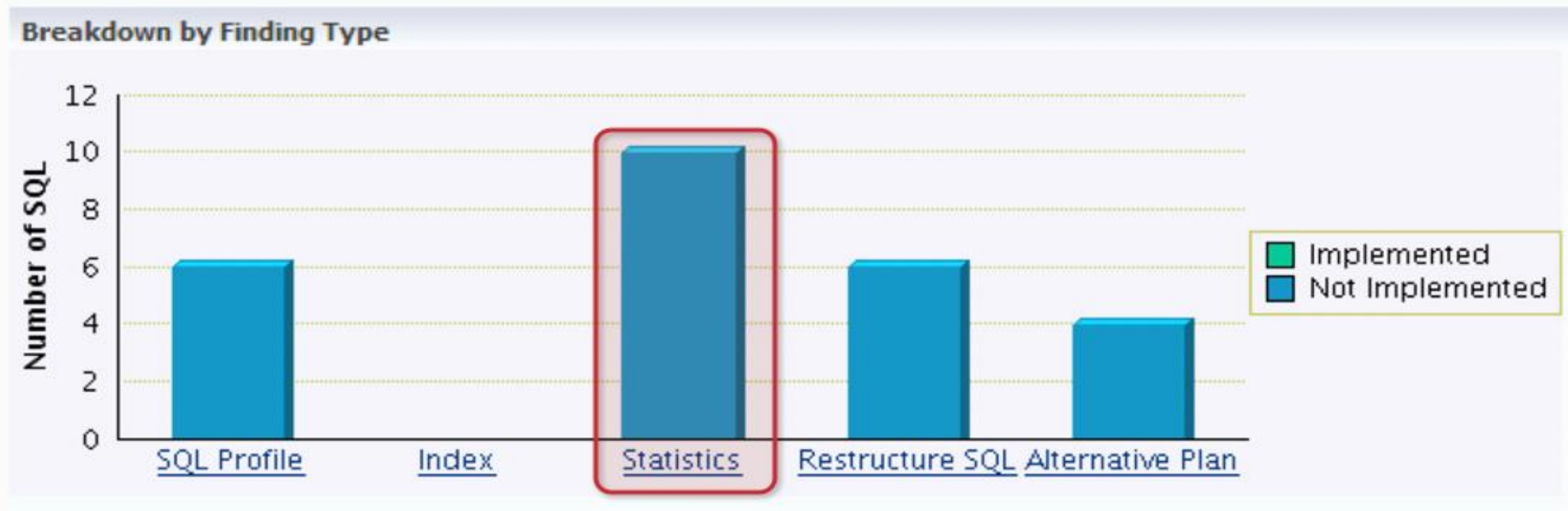
- A Plan Baseline is a set of accepted plans that the optimizer is allowed to use for a SQL statement
- A plan includes all plan-related information that the optimizer needs to reproduce an execution plan, such as)
 - SQL plan identifier
 - Set of hints
 - Bind Values
 - Optimizer environment
- The primary goal of SQL plan management is to prevent performance regressions caused by plan changes

Tips: Should I use SQL Baselines or SQL Profiles?

- SQL plan baselines are proactive but restricts your potential gain
- SQL profiles are reactive but can improve your performance



The Fix: Tuning Advisor - comprehensive recommendations

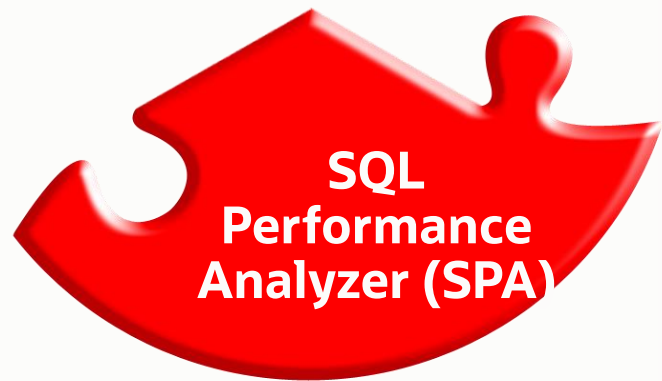


- Comprehensive analysis and multiple alternatives to improve performance
- SQL Profiles can be implemented transparently & non-intrusively to tune SQL statements
- As an example, let's gather statistics in pending mode...

The Fix: Tuning Advisor

Gather Optimizer Statistics Recommendation

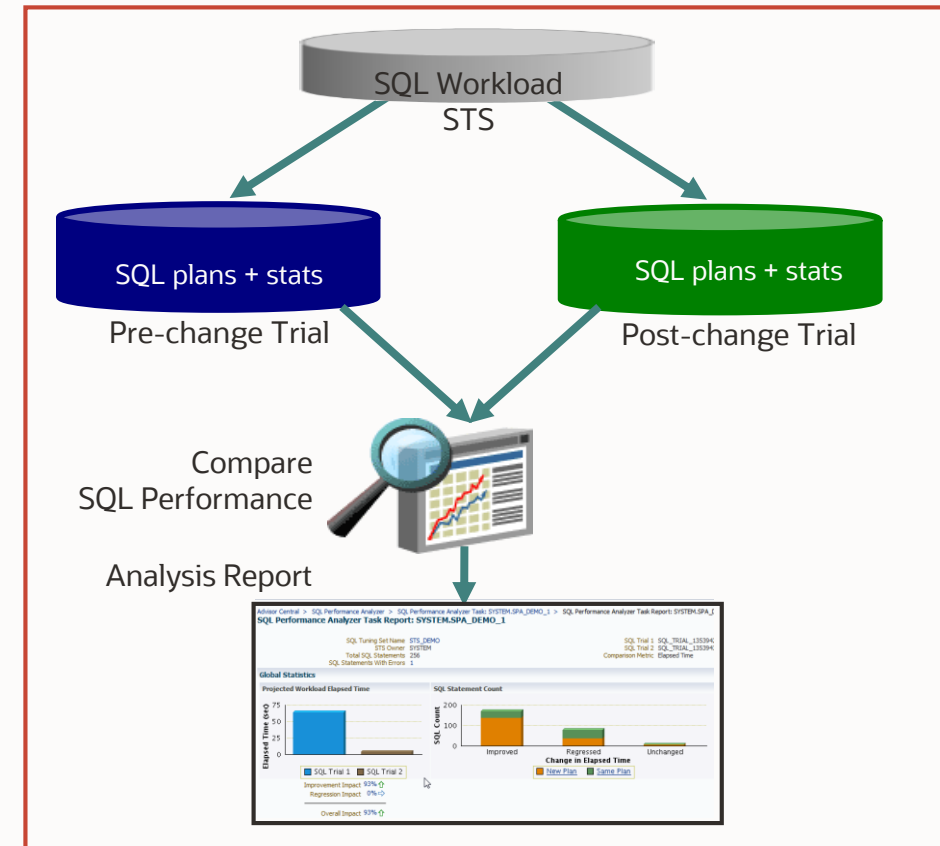
- The Fix – Gathering Optimizer Statistics
- However, this could negatively impact other SQL in the workload
- If only there was a fool-proof method to validate statistics gathering recommendation....



VALIDATE

SPA Overview

- Helps users predict the impact of system changes on SQL workload response time
- 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
- Analyzes performance differences
- Offers fine-grained performance analysis on individual SQL
- Integrated with STS, SQL Plan Baselines, & SQL Tuning Advisor to form an end-to-end solution



SPA Report Example

SQL Performance Analyzer Task Result: SYS.UPGRADE_10G11G

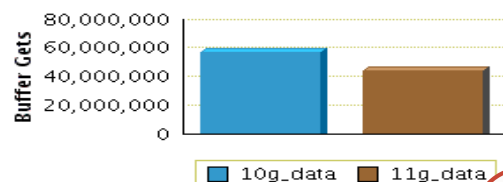
Task Name **UPGRADE_10G11G**
Task Owner **SYS**
Task Description **test upgrade to 11g**

SQL Tuning Set Name **OOW_54G**
STS Owner **SYS**
Total SQL Statements **54**
SQL Statements With Errors **0**

Replay Trial 1 **10g_data**
Replay Trial 2 **11g_data**
Comparison Metric **Buffer Gets**

Global Statistics

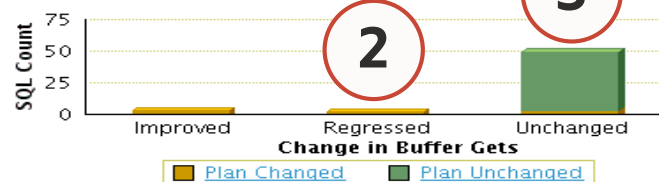
Projected Workload Buffer Gets



Improvement Impact **24%** ↑
Regression Impact **-2%** ↓

Overall Impact **22%** ↑

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.

Create SQL Plan Baselines

Explore alternate execution plans using SQL Tuning Advisor.

Run SQL Tuning Advisor

Top 10 SQL Statements Based on Impact on Workload

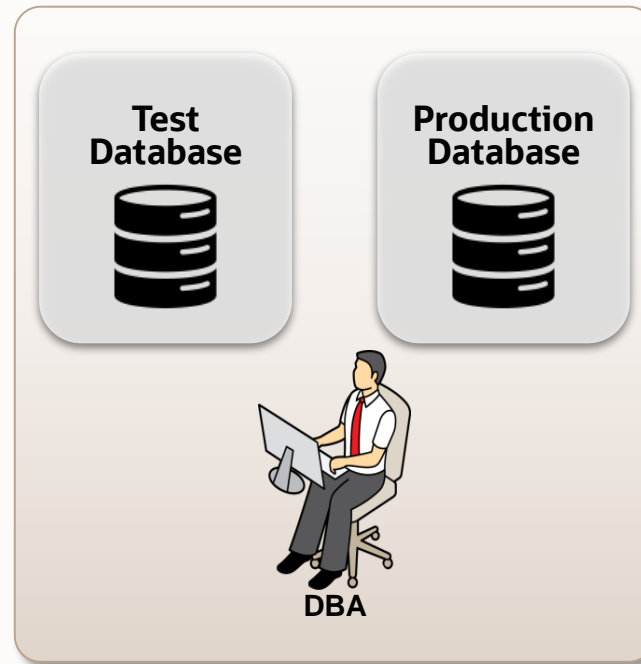
SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	% of Workload		Plan Changed
		10g_data	11g_data		10g_data	11g_data	
↑ g4dzf4ak4rus2	12.000	20,318,458.000	13,502,097.000	33.550	35.780	30.670	Y
↑ gfacm5jr3rz9j	11.990	6,990,541.000	180,401.000	97.420	12.310	0.410	Y
↓ 2ny751aat2vd9	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530	Y
↓ c2fb0ug5p7d4p	-0.750	12,740,524.000	13,165,998.000	-3.340	22.440	29.910	Y
↑ 2wtgxbjz6u2by	0.050	244,678.000	218,533.000	10.690	0.430	0.500	Y

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

Optimized

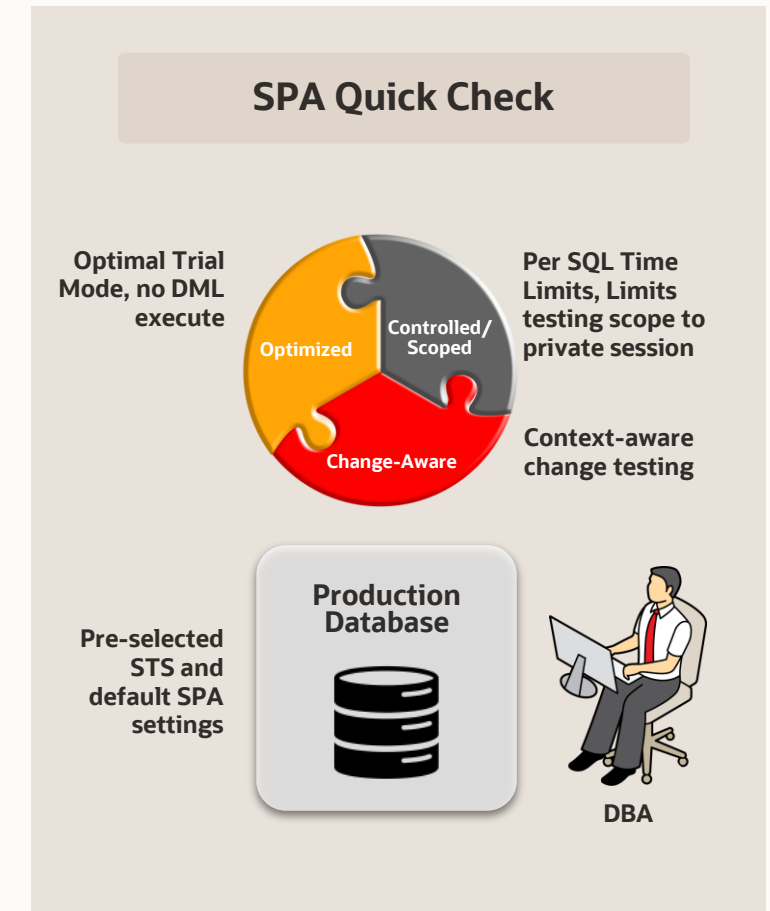
- Optimized for use on prod systems
- Optimal Trial or Explain Plan mode
- Disable multi-executions, full DML execute disabled

Controlled

- Per SQL time limits
- Testing scoped to private session
- Associate with Resource Consumer Group

Change-Aware

- Context-aware change testing workflows, such as,
 - Optimizer gather statistics
 - Init.ora parameter 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)

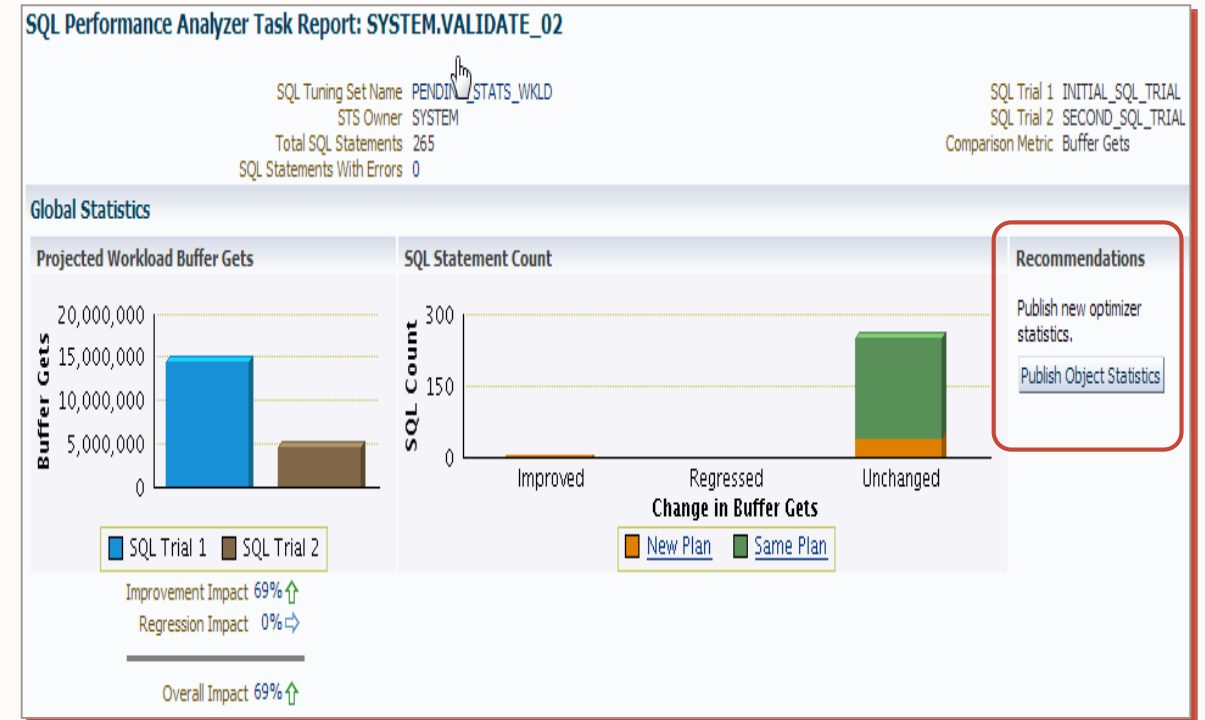
SPA Quick Check

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



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' 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

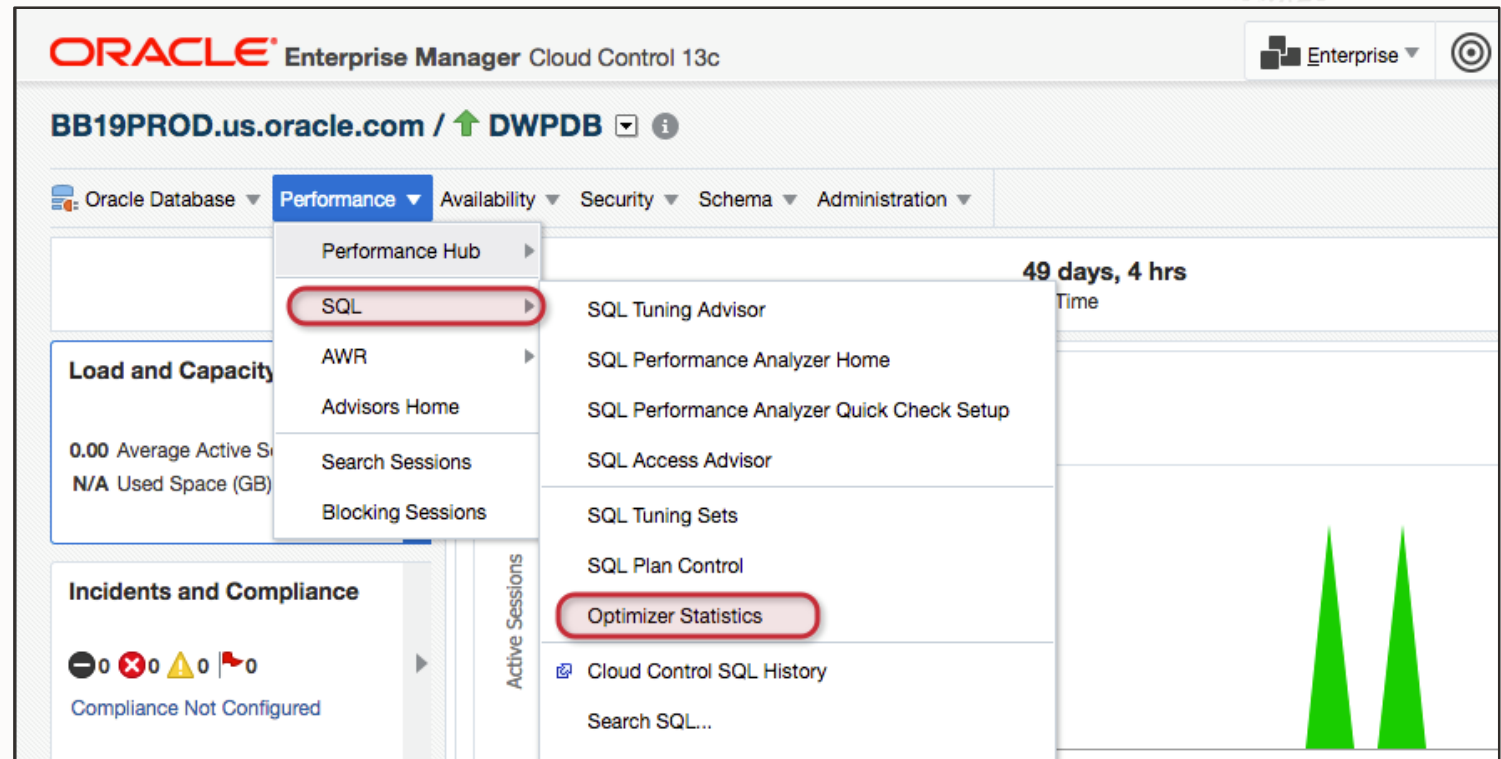
Use Resource Consumer Group ☒ Yes ☐ No

Resource Consumer Group

Save

Optimizer Statistics Refresh - Gather Statistics

Go to Optimizer Statistics Page



Optimizer Statistics Refresh - Gather Statistics

Go to Optimizer Statistics Page

- Choose Gather Statistics

ORACLE Enterprise Manager Cloud Control 13c

BB19PROD.us.oracle.com / DWPDB

Oracle Database Performance Availability BB19PROD.us.oracle.com_DWPDB Administration

Optimizer Statistics Console

Operations

Gather Lock View
Restore Unlock Delete

Statistics Status

1,593 36 2,979

Number of Objects

Fresh Stats
Stale Stats
No Stats

Configure

Global Statistics Gathering Options
Object Level Statistics Gathering Options

Configure Auto Task Optimizer Statistics
SPA Validation Results

Statistics Gathering Auto Tasks

80% 40% 0%

2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-... 2020-...

Failed%
Unknown%
Skipped%
Timed Out%
Pending%
In Progress%
Completed%

Statistics Gathering Job List

Operation Name	Job Name	Target	Status	Total Objects	Num Completed	Start Time	Duration
gather_database_stats (auto)	ORA\$AT_OS_OPT_SY_257	AUTO	COMPLETED	192	191	Oct 26, 2020 7:...	00:00:04
gather_database_stats (auto)	ORA\$AT_OS_OPT_SY_255	AUTO	COMPLETED	180	179	Oct 25, 2020 7:...	00:00:58
gather_database_stats (auto)	ORA\$AT_OS_OPT_SY_253	AUTO	COMPLETED	181	180	Oct 25, 2020 3:...	00:00:58

Optimizer Statistics Refresh - Gather Statistics

Go to Optimizer Statistics Page

- Choose Gather Statistics
- Choose the extent
- Choose to validate with SPA

ORACLE Enterprise Manager Cloud Control 13c

BB19PROD.us.oracle.com / DWPDB

Oracle Database Performance Availability Security Schema Administration

Scope Objects Customize Options Schedule Review

Gather Optimizer Statistics: Scope

Database: BB19PROD.us.oracle.com
Task Status: Enabled
Logged In As: system
Scope: Schema

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 performance tables.
- ☐ Dictionary Objects
Objects in 'SYS', 'SYSTEM' and all non-user defined schemas.
- ☐ System

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

Options for Scope: Schema

☒ **Validate with SQL Performance Analyzer**

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

Step 1 of 5

Optimizer Statistics Refresh - Gather Statistics

Go to Optimizer Statistics Page

- Choose Gather Statistics
- Choose the extent
- Choose to validate with SPA
- Choose object according to dialog

The screenshot displays the Oracle Enterprise Manager Cloud Control 13c interface. The top navigation bar includes the Oracle logo, 'Enterprise Manager Cloud Control 13c', and various utility icons. The breadcrumb trail shows 'BB19PROD.us.oracle.com / ↑ DWPDB'. The main navigation menu includes 'Oracle Database', 'Performance', 'Availability', 'Security', 'Schema', and 'Administration'. A progress bar indicates the current step is 'Objects' in a sequence of five steps: Scope, Objects, Customize Options, Schedule, and Review.

The page title is 'Gather Optimizer Statistics: Objects'. It shows the database as 'BB19PROD.us.oracle.com' and the user as 'system'. The task status is 'Enabled'. The page is at 'Step 2 of 5'.

A message states: 'This table contains the schemas for which optimizer statistics will be gathered. Click Add to add schemas to the table.' Below this is a table with one row containing the object name 'STAT1'. An 'Add' button is located to the right of the table. A 'Remove' button with a trash icon is also present next to the 'STAT1' row.

At the bottom, there is a checkbox labeled 'Lock optimizer statistics of objects after gather' which is currently unchecked. The page concludes with 'Cancel', 'Back', 'Step 2 of 5', and 'Next' buttons.

Optimizer Statistics Refresh - Gather Statistics

Go to Optimizer Statistics Page

- Choose Gather Statistics
- Choose the extent
- Choose to validate with SPA
- Choose object according to dialog
- Submit

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The top navigation bar includes the Oracle logo, 'Enterprise Manager Cloud Control 13c', and various icons. The breadcrumb trail indicates the current location: 'BB19PROD.us.oracle.com / ↑ DWPDB'. The user is logged in as 'system' with the session 'den00yov.us.oracle.com'. The main navigation menu shows 'Oracle Database' selected, with sub-menus for 'Performance', 'Availability', 'Security', 'Schema', and 'Administration'. The 'Performance' sub-menu is expanded, showing a progress bar with steps: 'Scope', 'Objects' (current), 'Customize Options', 'Schedule', and 'Review'. The 'Objects' step is highlighted, and a confirmation dialog is displayed. The dialog has a green header with a checkmark and the title 'Confirmation'. The message states: 'The Gather Optimizer Statistics job has been successfully submitted. SQL Performance Analyzer Task [SYSTEM.SPA_OPTSTATS_TASK_824074375](#)'. Below the message, it says: 'This table contains the schemas for which optimizer statistics will be gathered. Click Add to add schemas to the table.' There is an 'Add' button in the top right corner of the table area. The table has two columns: 'Name' and 'Remove'. The 'Name' column contains the text 'STAT1'. The 'Remove' column has a 'Remove' button and a pencil icon. At the bottom of the dialog, there is a checkbox labeled 'Lock optimizer statistics of objects after gather' which is currently unchecked. The bottom right corner of the dialog has 'Cancel', 'Back', 'Step 2 of 5', and 'Next' buttons.

ORACLE Enterprise Manager Cloud Control 13c

BB19PROD.us.oracle.com / ↑ DWPDB

Logged in as system | den00yov.us.oracle.com

Oracle Database ▾ Performance ▾ Availability ▾ Security ▾ Schema ▾ Administration ▾


Scope Objects Customize Options Schedule Review

✓ Confirmation

The Gather Optimizer Statistics job has been successfully submitted. SQL Performance Analyzer Task [SYSTEM.SPA_OPTSTATS_TASK_824074375](#)

This table contains the schemas for which optimizer statistics will be gathered. Click Add to add schemas to the table.

Add

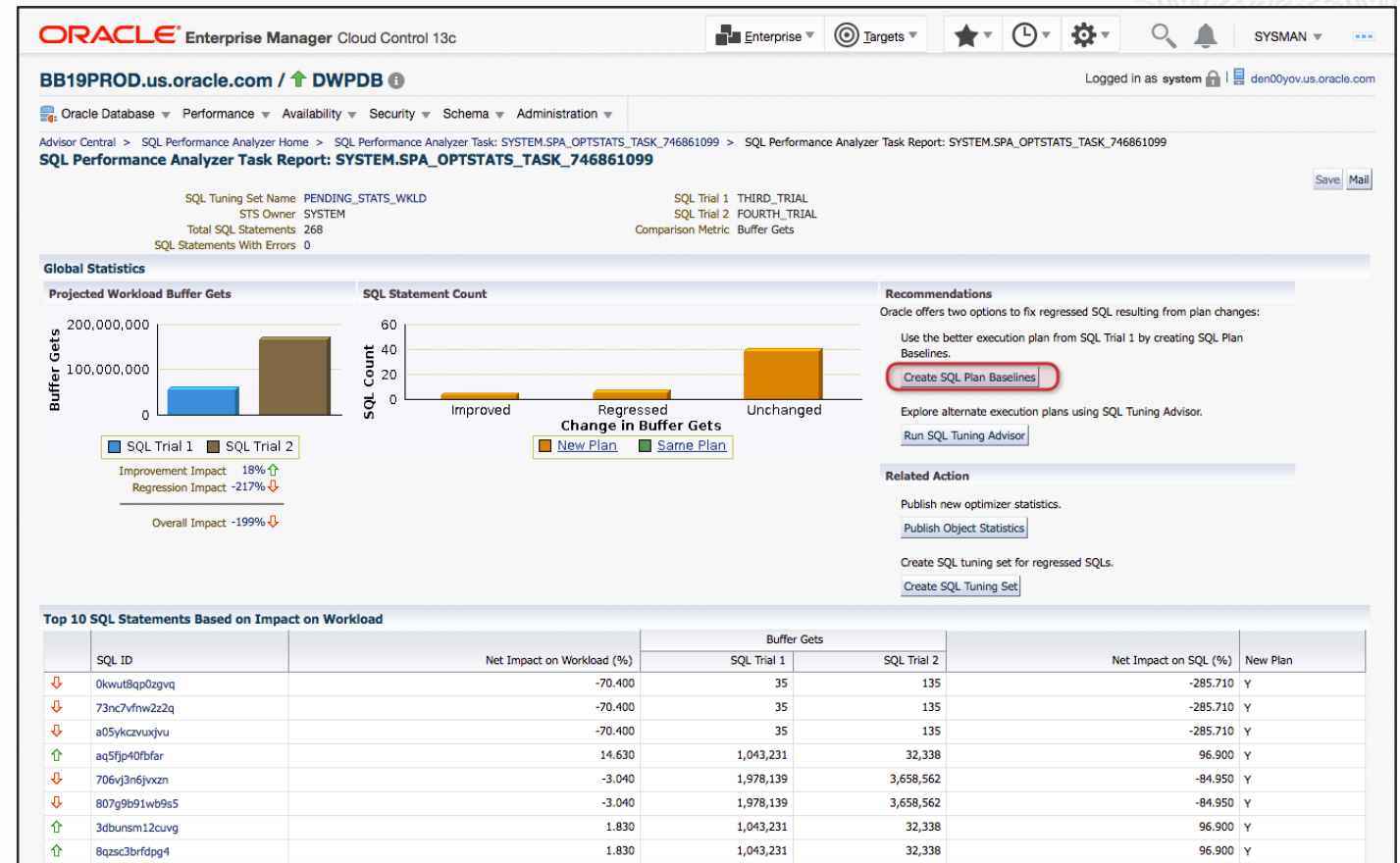
Name ▲	Remove
STAT1	

☐ Lock optimizer statistics of objects after gather

Cancel Back Step 2 of 5 Next

Optimizer Statistics Refresh- Remedy and publish

- Remedy regression
 - SQL Plan Baseline
 - Tuning advisor



Optimizer Statistics Refresh- Remedy and publish

- Remedy regression
 - SQL Plan Baseline
 - Tuning advisor
- Implement (for this example)
 - SQL Plan Baseline

ORACLE Enterprise Manager Cloud Control 13c

BB19PROD.us.oracle.com / DWPDB

Logged in as system

Advisor Central > SQL Performance Analyzer Home > SQL Performance Analyzer Task: SYSTEM.SPA_OPTSTATS_TASK_746861099 > SQL Performance Analyzer Task Report: SYSTEM.SPA_OPTSTATS_TASK_746861099

Regressed New Plan SQL Statements

SQL Performance Analyzer Task Report: SYSTEM.SPA_OPTSTATS_TASK_746861099

Create SQL Plan Baselines

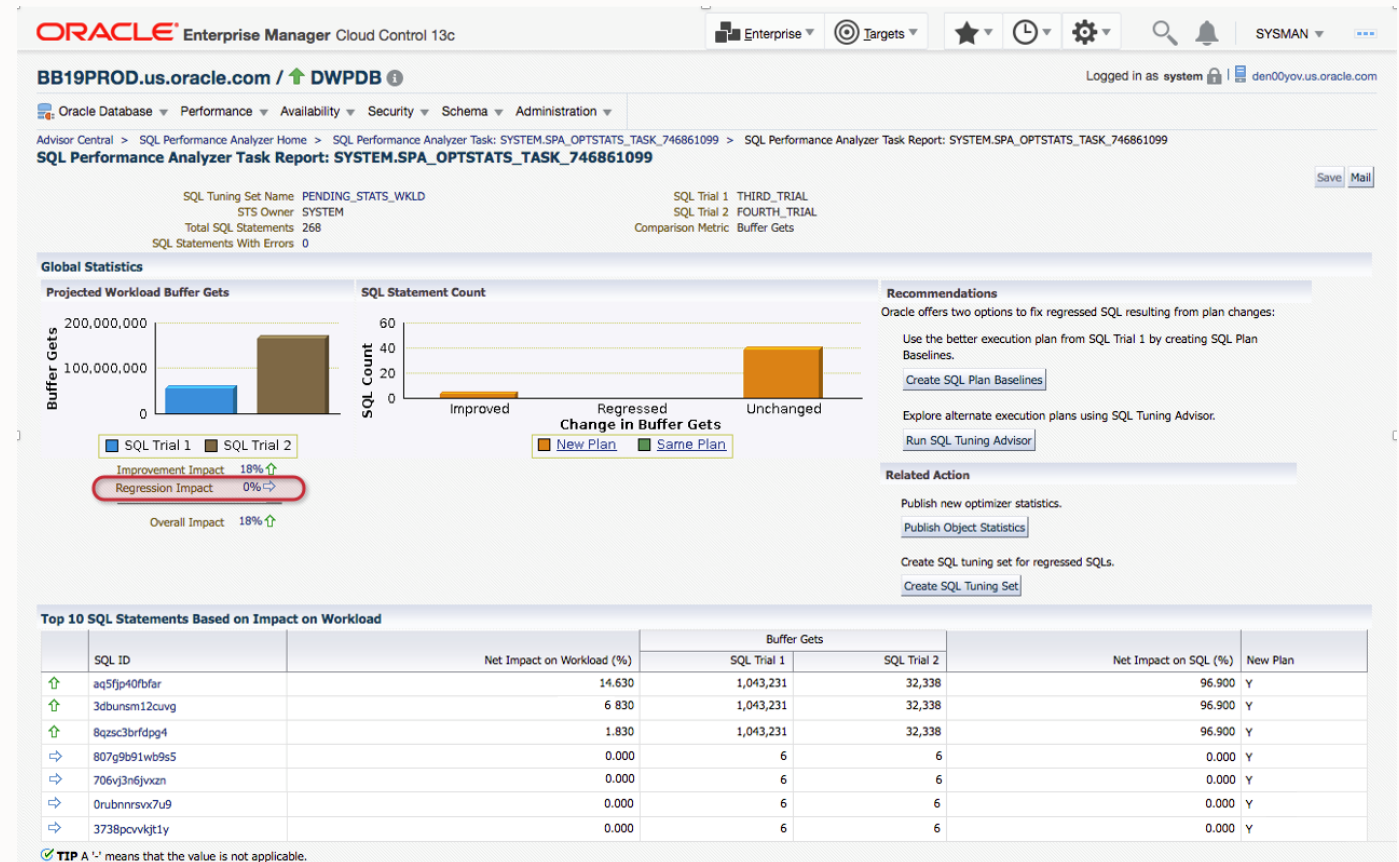
SQL Plan Baselines enable the optimizer to avoid performance regressions by requiring new plans to be at least as good as the better plans found in SQL trial 1.

Regressed New Plan SQL Statements

SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	% of Workload	
		THIRD_TRIAL	FOURTH_TRIAL		THIRD_TRIAL	FOURTH_TRIAL
0kwut8ap0zgvq	-70.400	35	135	-285.710		
73nc7vfw2z2q	-70.400	35	135	-285.710		
a05ykczzvuxju	-70.400	35	135	-285.710		
706vj3n6jyxzn	-3.040	1,978,139	3,658,562	-84.950		
807g9b91wb9s5	-3.040	1,978,139	3,658,562	-84.950		

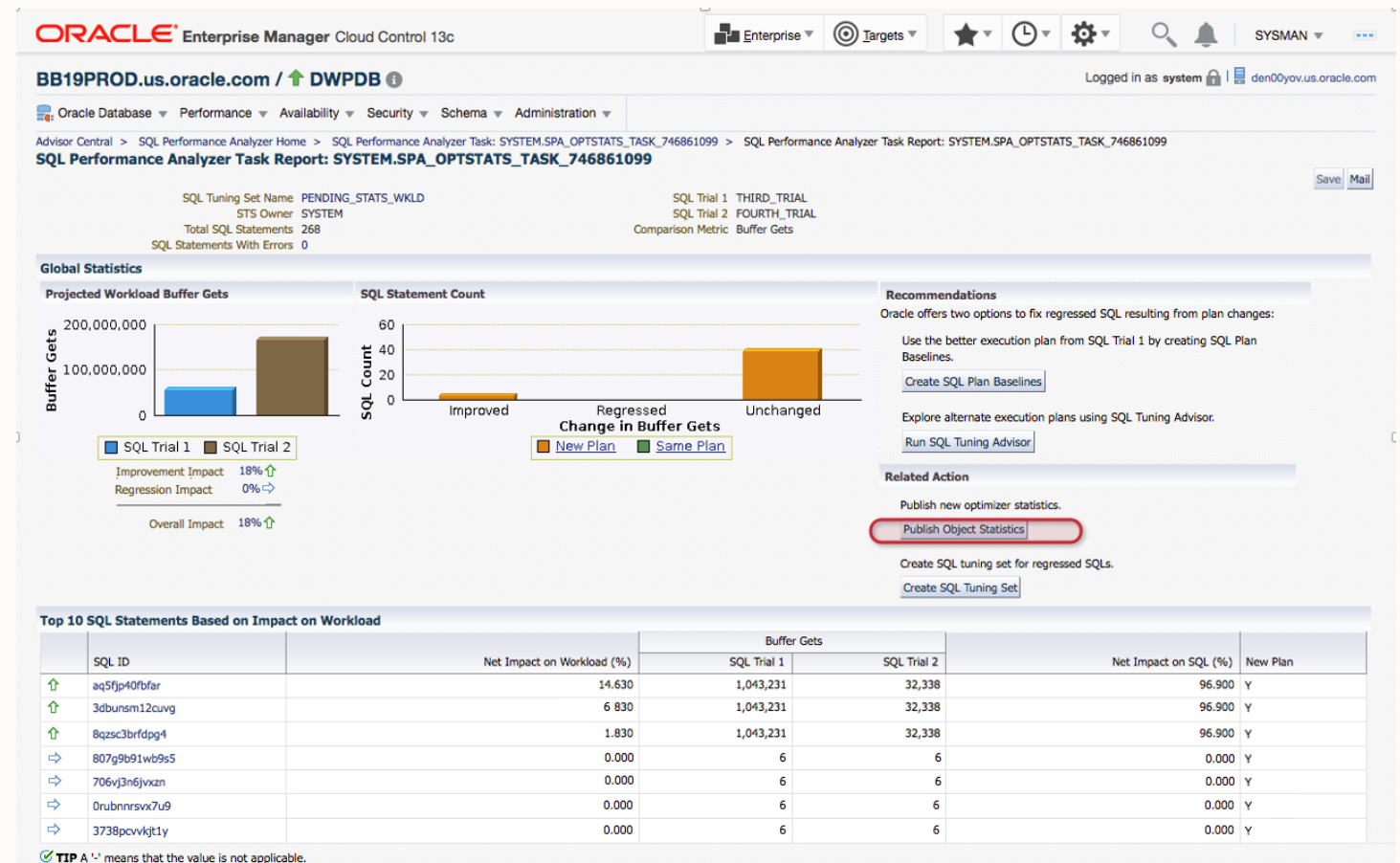
Optimizer Statistics Refresh- Remedy and publish

- Remedy regression
 - SQL Plan Baseline
 - Tuning advisor
- Implement (for this example)
 - SQL Plan Baseline
- Validate again



Optimizer Statistics Refresh- Remedy and publish

- Remedy regression
 - SQL Plan Baseline
 - Tuning advisor
- Implement (for this example)
 - SQL Plan Baseline
- Validate again
- Publish Statistics



Use Case 2: SPA Quick Check - Init.ora Parameter Validation

Problem:

- My manager wants me to improve database performance by tuning the optimizer_index_caching parameter. The assumption is that will help improve the performance significantly...
- However, I'm not sure if this parameter will help, how do I quickly and accurately validate the impact of the above parameter for my workload?

Solution: You can use SPA Quick Check to validate the performance impact of the parameter...

SPA Quick Check

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The top navigation bar includes the Oracle logo, 'Enterprise Manager Cloud Control 13c', and various icons for Enterprise, Targets, Favorites, Clock, Settings, Search, and Notifications. The user is logged in as 'system' at 'den00yov.us.oracle.com'. The breadcrumb trail shows 'Oracle Database > Performance > Availability > Security > Schema > Administration'. The 'Administration' menu is highlighted. Below the breadcrumb, there are buttons for 'Execute On Multiple Databases', 'Show SQL', 'Revert', 'Validate with SPA' (highlighted with a red box), 'SPA Validation Results', and 'Apply'. A message states: 'You are not logged on with SYSDBA privilege. Only controls for dynamic parameters are editable.' The main section is titled 'Initialization Parameters' and has tabs for 'Current' and 'SPFile'. A yellow information box contains the text: 'Succeed to submit task for validating parameters change using SQL Performance Analyzer. The task is SPA_OPTPARAMETER_TASK_99134608' and a link to 'SQL Performance Analyzer Task - SYSTEM.SPA_OPTPARAMETER_TASK_99134608'. Below this is a table of parameters:

Name	Help	Value	Comments	Type	Basic	Modified	Dynamic	Category
optimizer_index_caching	?	50	change from 0 to 50	Integer			✓	Optimizer
optimizer_index_cost_adj	?	100		Integer			✓	Optimizer
skip_unusable_indexes	?	TRUE		Boolean			✓	Optimizer

Buttons for 'Save to File' are located to the right of the table. At the bottom, there are buttons for 'Execute On Multiple Databases', 'Show SQL', 'Revert', 'Validate with SPA', 'SPA Validation Results', and 'Apply'.

SPA Quick Check

ORACLE Enterprise Manager Cloud Control 13c

Enterprise ▾ Targets ▾ ★ ▾ ⌚ ▾ ⚙ ▾ 🔍 🔔 SYSMAN ▾ ⋮

BB19PROD.us.oracle.com / ↑ DWPDDB ⓘ

Logged in as system ⓘ | den00yov.us.oracle.com

Oracle Database ▾ Performance ▾ Availability ▾ Security ▾ Schema ▾ Administration ▾

SQL Performance Analyzer Home

Page Refreshed **Oct 29, 2020 10:22:59 AM PDT** Refresh View Data Real Time: 15 Second Refresh ▾

SQL Performance Analyzer allows you to test and to analyze the effects of changes on the execution performance of SQL contained in a SQL Tuning Set.

SQL Performance Analyzer Workflows

Create and execute SQL Performance Analyzer Task experiments of different types using the following links.

Upgrade from 9i or 10.1	Test and analyze the effects of database upgrade from 9i or 10.1 on SQL Tuning Set performance.
Upgrade from 10.2 or higher releases	Test and analyze the effects of database upgrade from 10.2 or higher releases on SQL Tuning Set performance.
Parameter Change	Test and compare an initialization parameter change on SQL Tuning Set performance.
Optimizer Statistics	Test and analyze the effects of optimizer statistics changes on SQL Tuning Set performance.
Ignore Optimizer Hints	Test and analyze the effects of disabling optimizer hints on SQL Tuning Set performance.
Exadata Simulation	Simulate the effects of a Exadata Storage Server installation on SQL Tuning Set performance.
Guided Workflow	Create a SQL Performance Analyzer Task and execute custom experiments using manually created SQL trials.

SQL Performance Analyzer Tasks

Delete View Latest Report

Select	Name	Owner	Last Modified ▾	Current Step Name	Type	Last Run Status	SQLs Processed	Steps Completed	Task Type	Description
<input checked="" type="radio"/>	SPA_OPTPARAMETER_TASK_99134608	SYSTEM	Oct 29, 2020 10:09:13 AM	COMPARISON_IMPACT	Compare	Completed	1 of 1	7 of 7	Optimizer Parameter	Validate optimizer init parameter change with one-click SPA flow. Init Parameters: optimizer_index_caching (0 -> 50)
<input type="radio"/>	SPA_OPTSTATS_TASK_982889627	SYSTEM	Oct 29, 2020 7:58:19 AM	COMPARISON_IMPACT	Compare	Completed	4 of 4	7 of 7	Optimizer Stats	Validate pending stats with one-click SPA flow. Schemas: STAT1, STAT2
<input type="radio"/>	SPA_OPTSTATS_TASK_982743058	SYSTEM	Oct 29, 2020 7:55:57 AM	COMPARISON_IMPACT	Compare	Completed	8 of 8	7 of 7	Optimizer Stats	Validate pending stats with one-click SPA flow. Schemas: STAT1, STAT2
<input type="radio"/>	SPA_OPTSTATS_TASK_981423739	SYSTEM	Oct 29, 2020 7:35:38 AM	COMPARISON_IMPACT	Compare	Completed	8 of 8	7 of 7	Optimizer Stats	Validate pending stats with one-click SPA flow. Schemas: STAT1, STAT2
<input type="radio"/>	SPA_OPTSTATS_TASK_977777941	SYSTEM	Oct 29, 2020 6:33:16 AM	COMPARISON_IMPACT	Compare	Completed	8 of 8	7 of 7	Optimizer Stats	Validate pending stats with one-click SPA flow. Schemas: STAT1
<input type="radio"/>	SPA_OPTSTATS_TASK_977297266	SYSTEM	Oct 29, 2020 6:15:07 AM	COMPARISON_IMPACT	Compare	Completed	6 of 6	7 of 7	Optimizer Stats	Validate pending stats with one-click SPA flow. Schemas: STAT1

SPA Quick Check

ORACLE Enterprise Manager Cloud Control 13c

EnterpriseTargetsStarClockSettingsSearchBellSYSMAN

BB19PROD.us.oracle.com / DWDPDB

Logged in as system | den00yov.us.oracle.com

Oracle DatabasePerformanceAvailabilitySecuritySchemaAdministration

Advisor Central > SQL Performance Analyzer Home > SQL Performance Analyzer Task: SYSTEM.SPA_OPTPARAMETER_TASK_74984772

SQL Performance Analyzer Task: SYSTEM.SPA_OPTPARAMETER_TASK_74984772

View Latest Report

Page Refreshed Oct 29, 2020 10:32:39 AM PDT Refresh

The SQL Performance Analyzer Task is a container for experimental results of executing a specific SQL Tuning Set under changed environmental conditions and assessing the impact of environmental changes on STS execution performance.

> SQL Tuning Set

SQL Trials

A SQL Trial captures the execution performance of the SQL Tuning Set under specific environmental conditions.

Create SQL Trial

SQL Trial Name	Description	Created	SQL Executed	Status
FIRST_TRIAL	Trial with current parameters. Only explain execution plan.	9/10/20 7:57 AM	No	COMPLETED
SECOND_TRIAL	Trial with parameters changed in session. Only explain execution plan so to find SQLs with plan change.	9/10/20 7:57 AM	No	COMPLETED
THIRD_TRIAL	Trial with no parameters changed in session, only executing the SQLs with plan change.	9/10/20 7:57 AM	Yes	COMPLETED
FOURTH_TRIAL	Trial with parameters changed in session, only executing the SQLs with plan change.	9/10/20 7:57 AM	Yes	COMPLETED

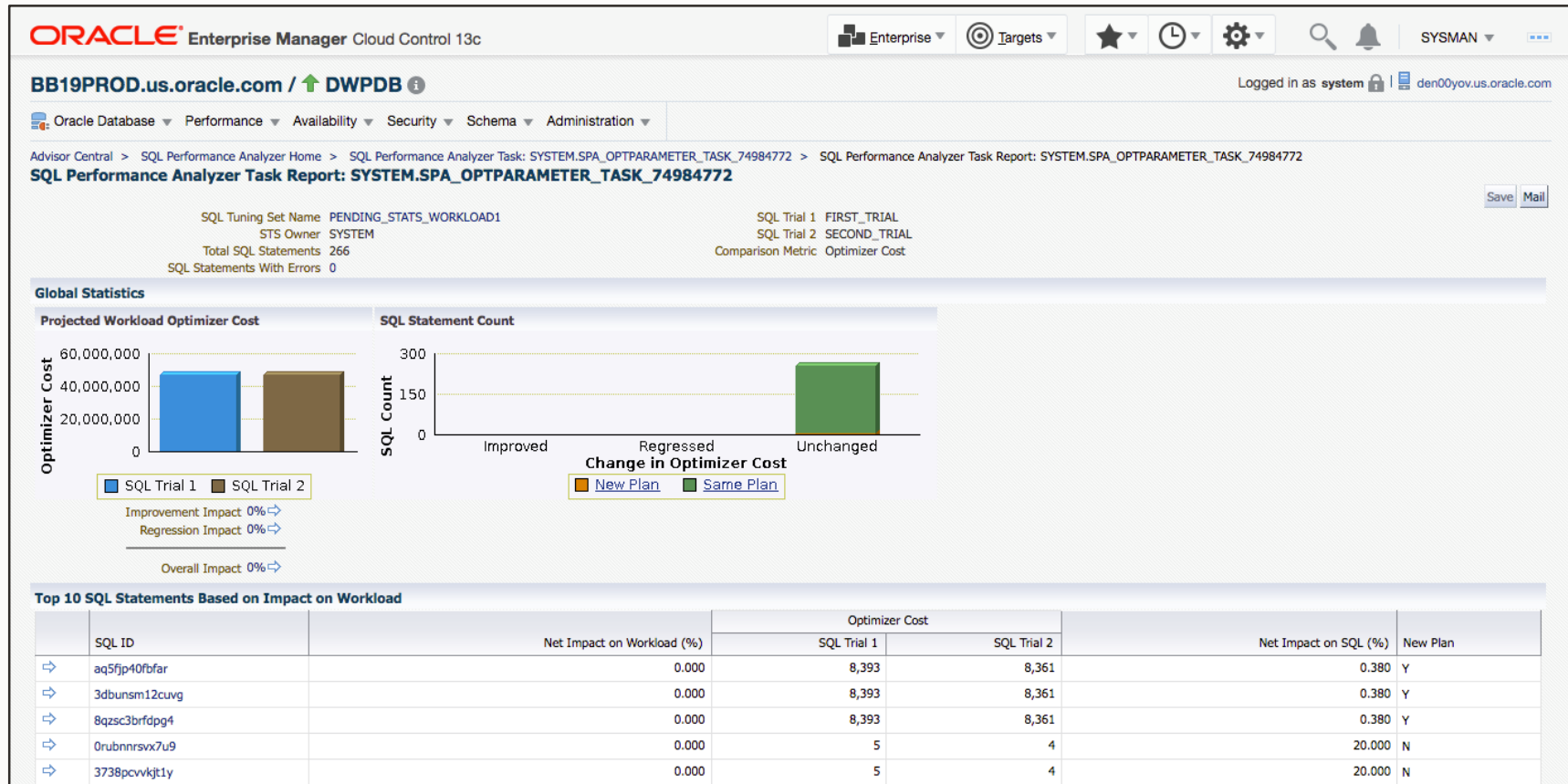
SQL Trial Comparisons

Compare SQL Trials to assess change impact of environmental differences on SQL Tuning Set execution costs.

Run SQL Trial Comparison

Trial 1 Name	Trial 2 Name	Compare Metric	Created	Status	Comparison Report	SQL Tune Report
FIRST_TRIAL	SECOND_TRIAL	Optimizer Cost	9/10/20 7:57 AM	COMPLETED		
THIRD_TRIAL	FOURTH_TRIAL	Buffer Gets	9/10/20 7:57 AM	COMPLETED		

SPA Quick Check



SPA Quick Check

ORACLE Enterprise Manager Cloud Control 13c

EnterpriseTargets★🕒⚙️🔍🔔SYSMAN...

BB19PROD.us.oracle.com / ⬆️ DWPDB ⓘ

Logged in as system | den00yov.us.oracle.com

Oracle Database ▾ Performance ▾ Availability ▾ Security ▾ Schema ▾ Administration ▾

Advisor Central > SQL Performance Analyzer Home > SQL Performance Analyzer Task: SYSTEM.SPA_OPTPARAMETER_TASK_74984772

SQL Performance Analyzer Task: SYSTEM.SPA_OPTPARAMETER_TASK_74984772

[View Latest Report](#)

Page Refreshed **Oct 29, 2020 10:32:39 AM PDT** [Refresh](#)

The SQL Performance Analyzer Task is a container for experimental results of executing a specific SQL Tuning Set under changed environmental conditions and assessing the impact of environmental changes on STS execution performance.

➤ **SQL Tuning Set**

▼ **SQL Trials**

A SQL Trial captures the execution performance of the SQL Tuning Set under specific environmental conditions.

[Create SQL Trial](#)

SQL Trial Name	Description	Created	SQL Executed	Status
FIRST_TRIAL	Trial with current parameters. Only explain execution plan.	9/10/20 7:57 AM	No	COMPLETED
SECOND_TRIAL	Trial with parameters changed in session. Only explain execution plan so to find SQLs with plan change.	9/10/20 7:57 AM	No	COMPLETED
THIRD_TRIAL	Trial with no parameters changed in session, only executing the SQLs with plan change.	9/10/20 7:57 AM	Yes	COMPLETED
FOURTH_TRIAL	Trial with parameters changed in session, only executing the SQLs with plan change.	9/10/20 7:57 AM	Yes	COMPLETED

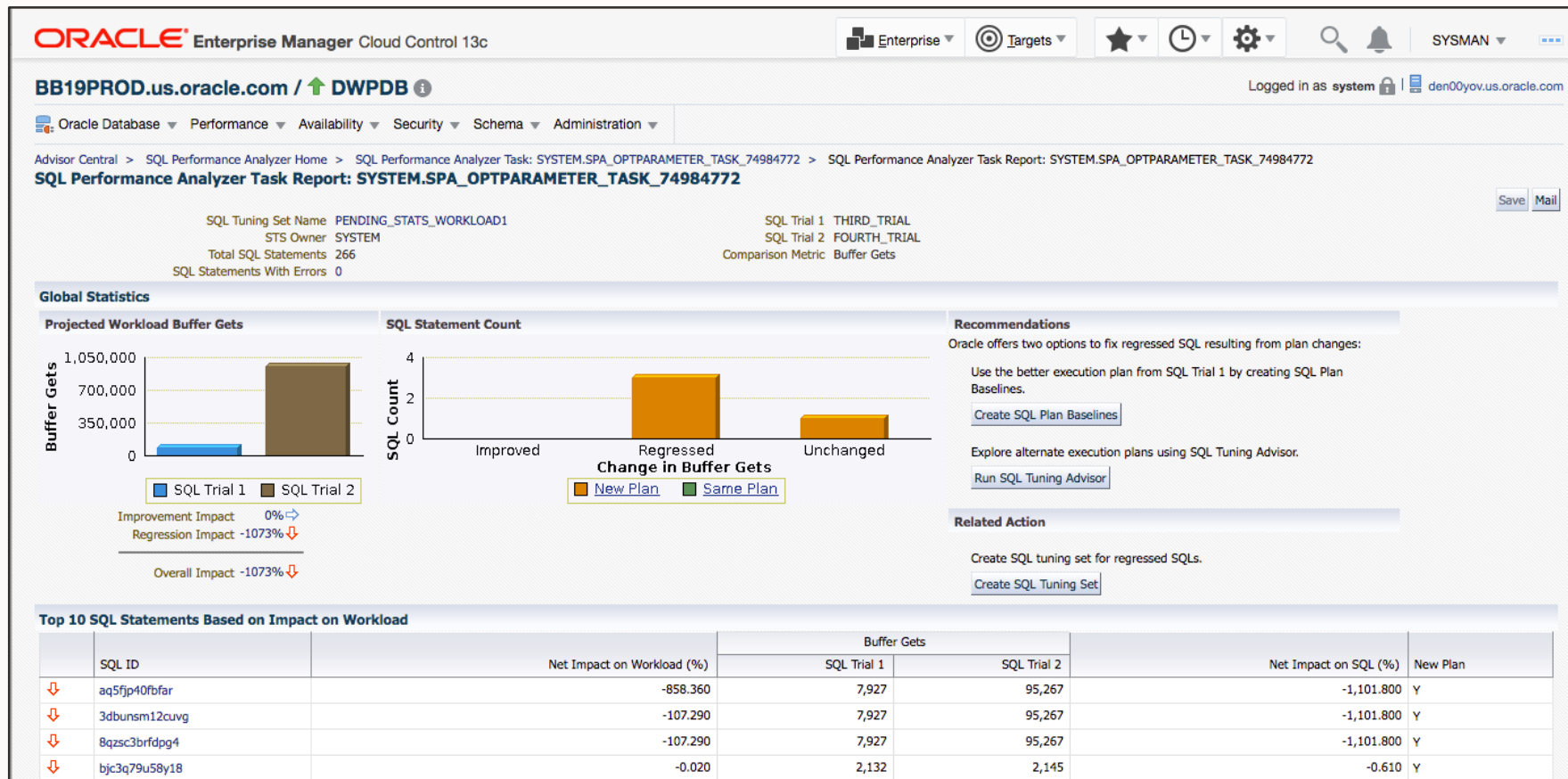
▼ **SQL Trial Comparisons**

Compare SQL Trials to assess change impact of environmental differences on SQL Tuning Set execution costs.

[Run SQL Trial Comparison](#)

Trial 1 Name	Trial 2 Name	Compare Metric	Created	Status	Comparison Report	SQL Tune Report
FIRST_TRIAL	SECOND_TRIAL	Optimizer Cost	9/10/20 7:57 AM	COMPLETED	🔗🔗	
THIRD_TRIAL	FOURTH_TRIAL	Buffer Gets	9/10/20 7:57 AM	COMPLETED	🔗🔗	

SPA Quick Check



SPA Quick Check - Init.ora Parameter Validation: Solution

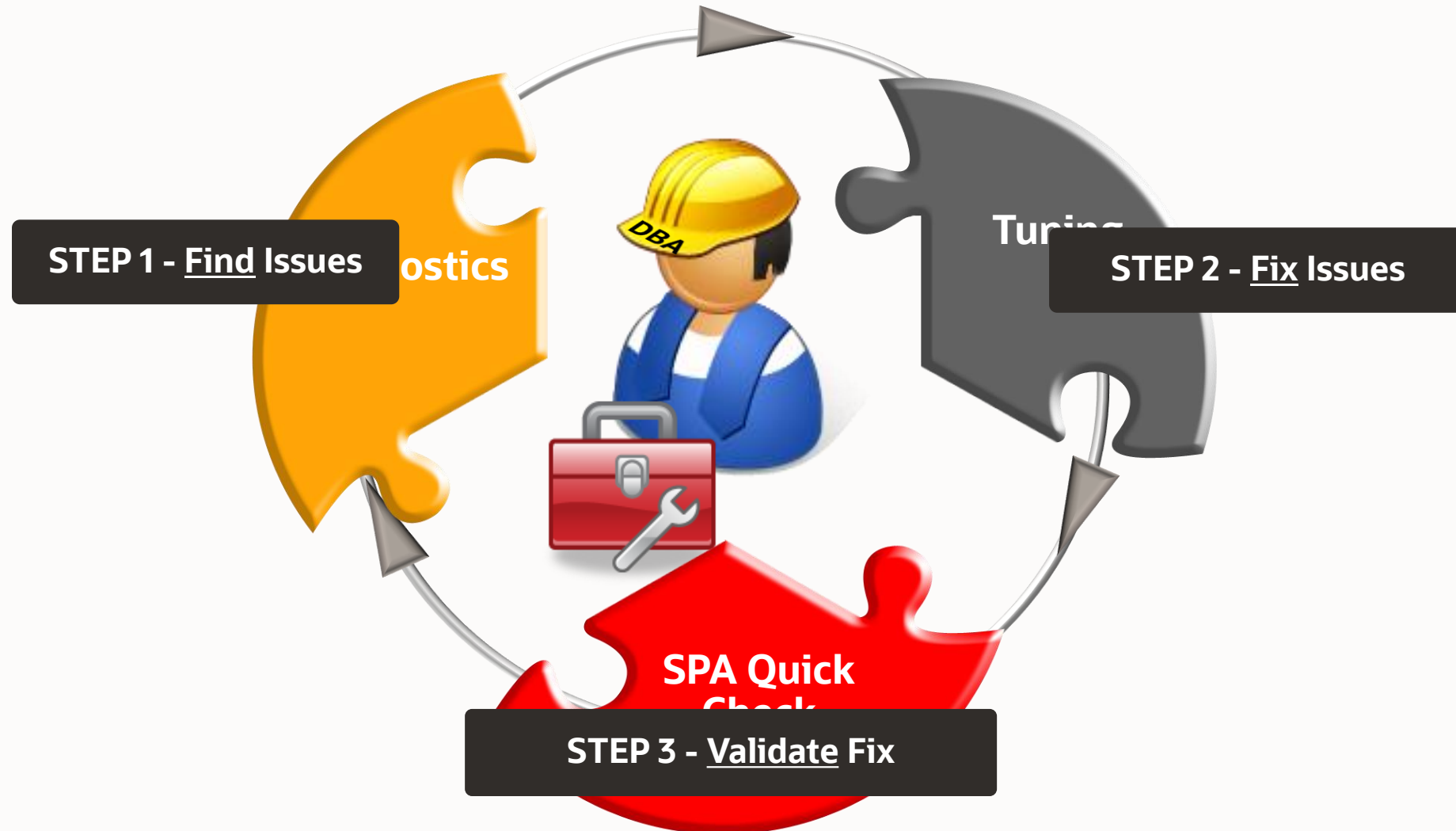
SPA Quick Check able to validate the impact of optimizer_index_caching on production

This change should **NOT** be implemented since it impacts the performance negatively

The change was tested on production in a controlled manner very efficiently without impacting production users!

Diagnostics, Tuning and SPA (D.T.S.)

Find → Fix → Validate



Thank you

