ORACLE®

Avoiding SQL Performance Regressions – New Techniques for Solving an Old Problem

Prabhaker Gongloor,
Product Management Director
Real Application Testing, Database Manageability and Diagnosability
Oracle Corporation

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.

Outline

- SQL Performance Regressions: Challenges
- SQL Performance Analyzer (SPA): Overview
- SPA Enhancements: Oracle Database 11*g* Release 2
- New Techniques for Avoiding SQL Regressions
- Real-world Customer Case Studies
- Conclusion

SQL Performance Regressions: Challenges

SQL performance regressions: #1 cause of poor system performance

SQL performance can regress due to many changes - DB upgrades, patch-sets, optimizer statistics refresh, schema, parameter, hardware, etc.

Changes need to be tested thoroughly to avoid regressions

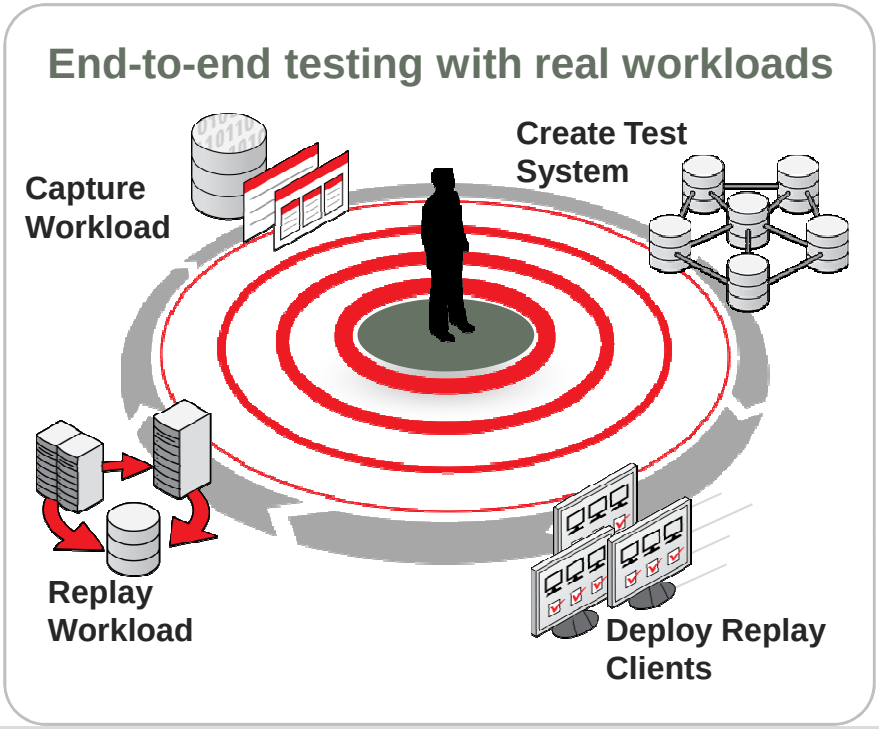
Traditional testing techniques suffer from many limitations: large workloads (100K SQL), expensive, partial workload capture & manual point solutions

As a result, no testing or sometimes limited testing done in production

SQL Performance Analyzer (SPA)

- **Proactively detects ALL SQL regressions BEFORE deploying actual change**
- **Provides integrated, comprehensive, and end-to-end solution**

Real Application Testing Features:
SPA and Database Replay

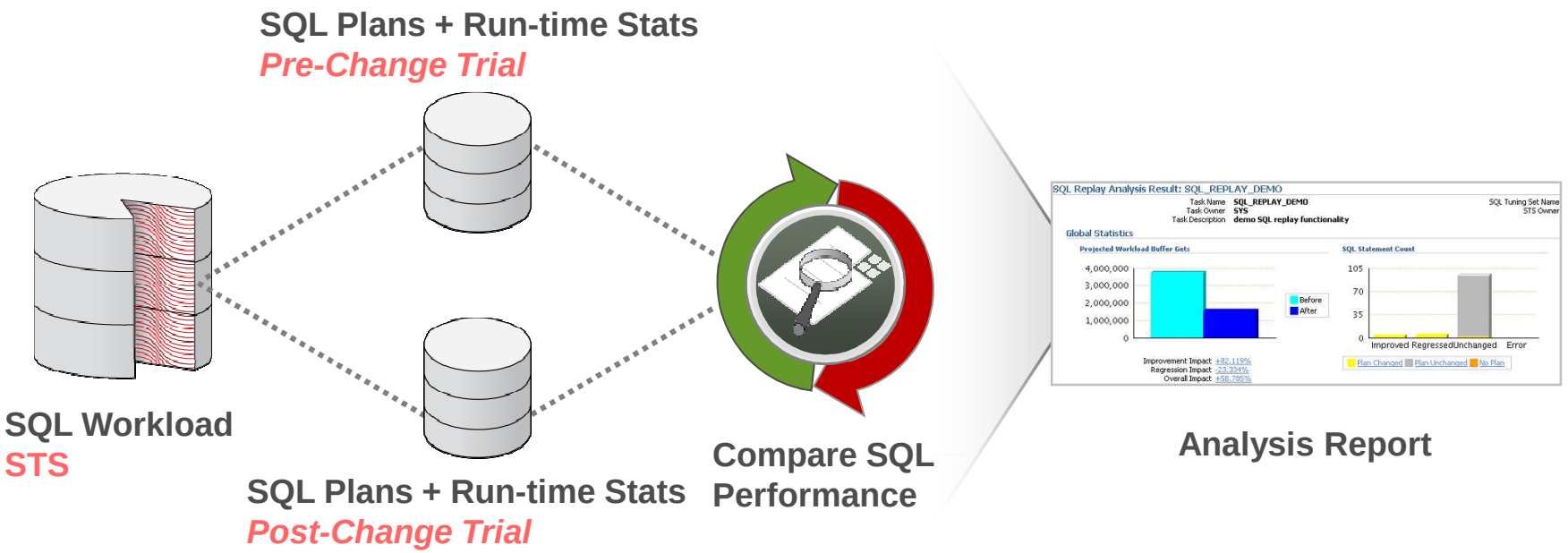


- **SQL Performance Analyzer**
  - SQL unit testing for response time
  - Identify and tune regressed SQL
- **Database Replay**
  - Load and performance testing for throughput
  - Remediate application concurrency problems

Outline

- SQL Performance Regressions: Challenges
- SQL Performance Analyzer (SPA): Overview
- SPA Enhancements: Oracle Database 11*g* Release 2
- New Techniques for Avoiding SQL Regressions
- Real-world Customer Case Studies
- Conclusion

Oracle Real Application Testing: SPA



- Test and predict impact of system changes on SQL query performance
- Analyze performance changes for improvements and regressions
- Comprehensive performance analysis and reporting
- Re-execute SQL queries in the given environment
- End-to-end solution: STS, SQL Plan Baselines, and SQL Tuning Advisor

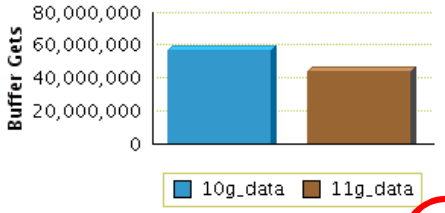
SPA Report (Example)

SQL Performance Analyzer Task Result: SYS.UPGRADE_10G11G

Task Name	**UPGRADE_10G11G**	SQL Tuning Set Name	OOW_54G	Replay Trial 1	**10g_data**
Task Owner	**SYS**	STS Owner	**SYS**	Replay Trial 2	**11g_data**
Task Description	**test upgrade to 11g**	Total SQL Statements	**54**	Comparison Metric	**Buffer Gets**
		SQL Statements With Errors	0		

Global Statistics

Projected Workload Buffer Gets



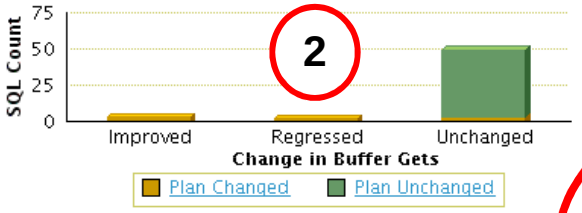
Improvement Impact 24%

Regression Impact -2%

Overall Impact 22%

1

SQL Statement Count



2

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

3

Top 10 SQL Statements Based on Impact on Workload

	SQL ID	Net Impact on Workload (%)	Buffer Gets 10g_data	Buffer Gets 11g_data	Net Impact on SQL (%)	% of Workload 10g_data	% of Workload 11g_data	Plan Changed
⇧	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

When to use SPA?

- Testing database upgrades and patch-set releases*
  - 9.2/10.1 →10.2 or 11g releases
  - 10.2.0.x →10.2.0.y or 11g releases
- SPA supports testing in Oracle Database Releases 10.2 and 11*g*
  - Optimizer statistics refresh
  - Database parameter changes
  - Database schema changes (e.g., add/drop indexes)
  - Implementation of tuning recommendations
  - I/O subsystem changes (e.g., ASM, Database Machine)
- SPA handles trials in a manner that does not change database data
  - Hence can be used for testing in production/standby environments

SPA Provides Broad Testing Coverage

- Across many releases of Oracle and for upgrades
- On test, standby, and production environments
- Extended to home-grown scripts, third-party testing tools, etc.
- Supports most applications - EBS, SAP, Siebel, home-grown, etc.

***MOS Note: 560977.1**

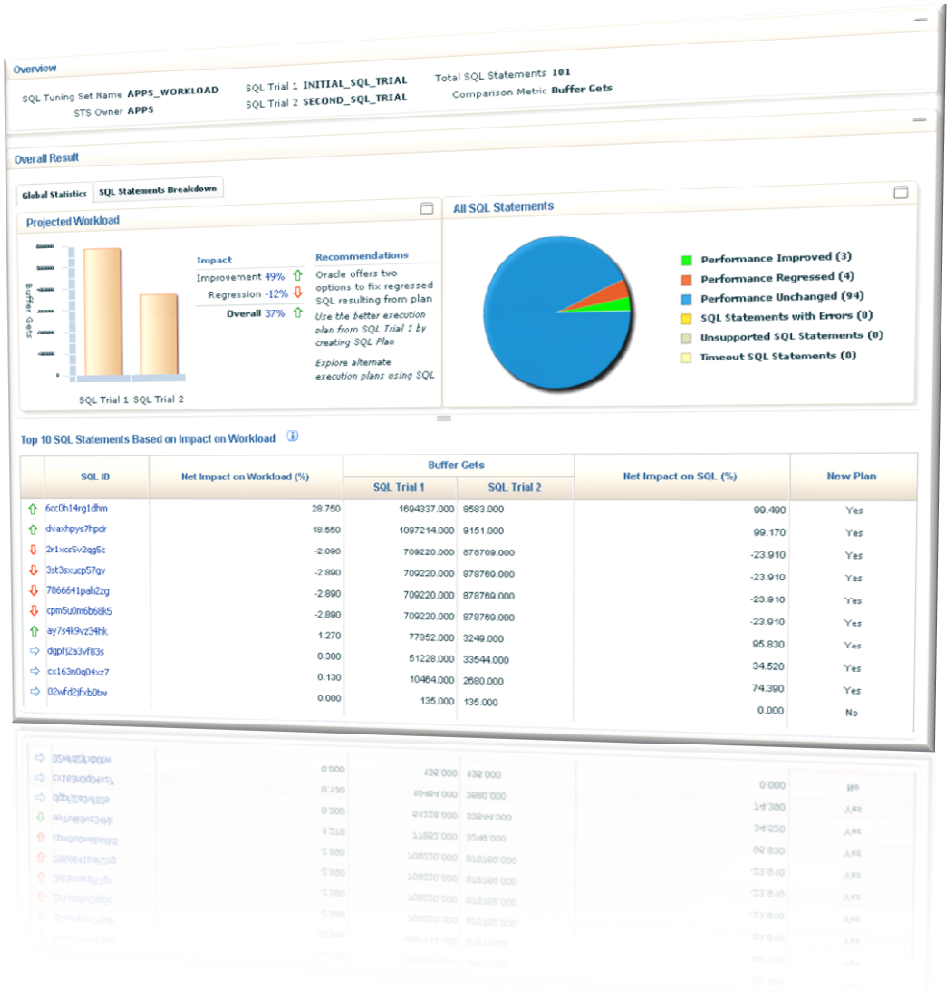
Outline

- SQL Performance Regressions: Challenges
- SQL Performance Analyzer (SPA): Overview
- SPA Enhancements: Oracle Database 11*g* Release 2
- New Techniques for Avoiding SQL Regressions
- Real-world Customer Case Studies
- Conclusion

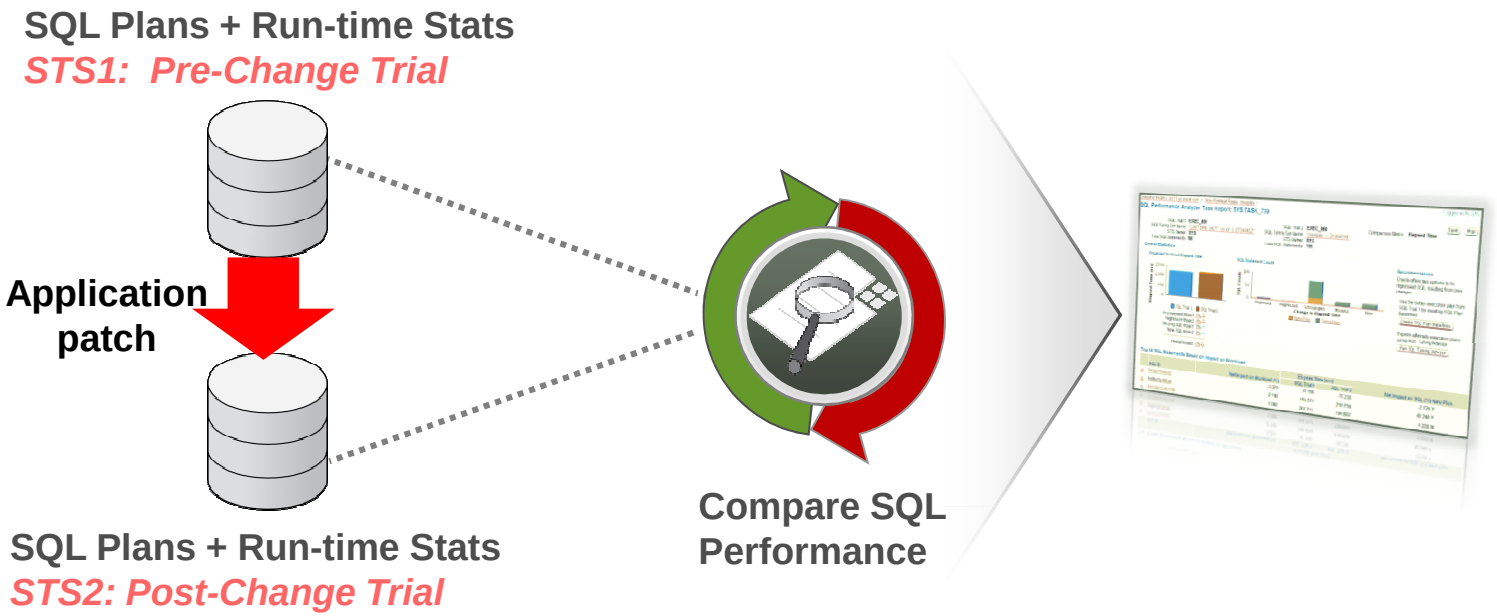
SPA Enhancements

New in Oracle Database 11*g* Release 2

- Better SPA trial accuracy through multiple test execution
- SPA Active Reports for offline viewing and analysis
- New workflows for db upgrade and optimizer statistics refresh
- Compare STS for comparing performance of two similar SQL workloads
- Leverage Oracle Active Data Guard for testing

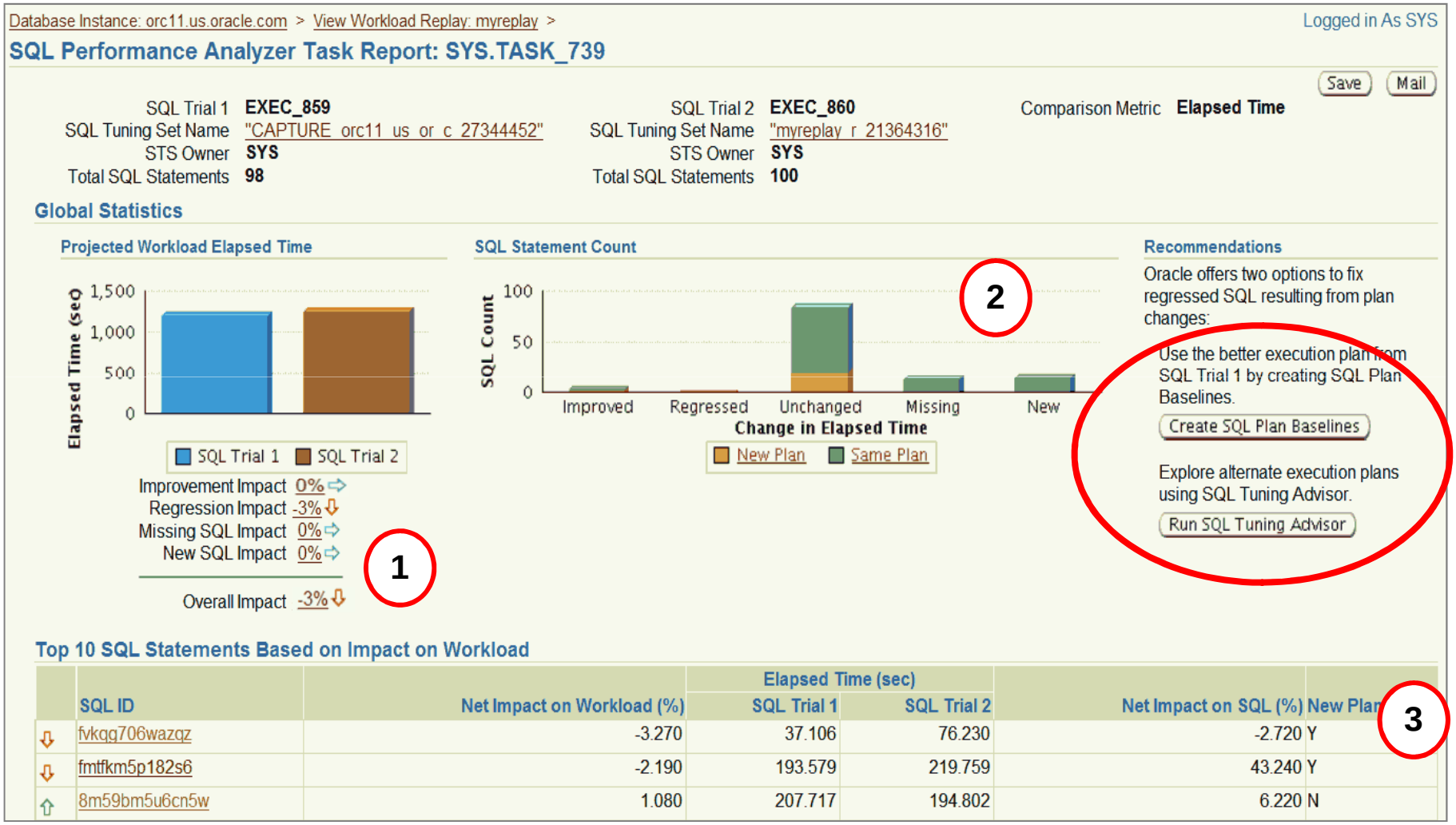


SPA Enhancements:
Compare STS



- SQL Tuning Set (STS) Compare feature
- Compares two related STSs and generates SPA report identifying
  - Common, Missing, New SQL
  - Multiple plans resulting from different binds or environments
- Enables performance impact analysis of application patches and upgrades
- Helps track workload drift

Example SPA Report – STS Compare



Outline

- SQL Performance Regressions: Challenges
- SQL Performance Analyzer (SPA): Overview
- SPA Enhancements: Oracle Database 11*g* Release 2
- New Techniques for Avoiding SQL Regressions
- Real-world Customer Case Studies
- Conclusion

New Techniques for Avoiding SQL Regressions

Create central “SPA System” for all testing

- Many database releases
- Test, production, standby databases

Minimize production impact

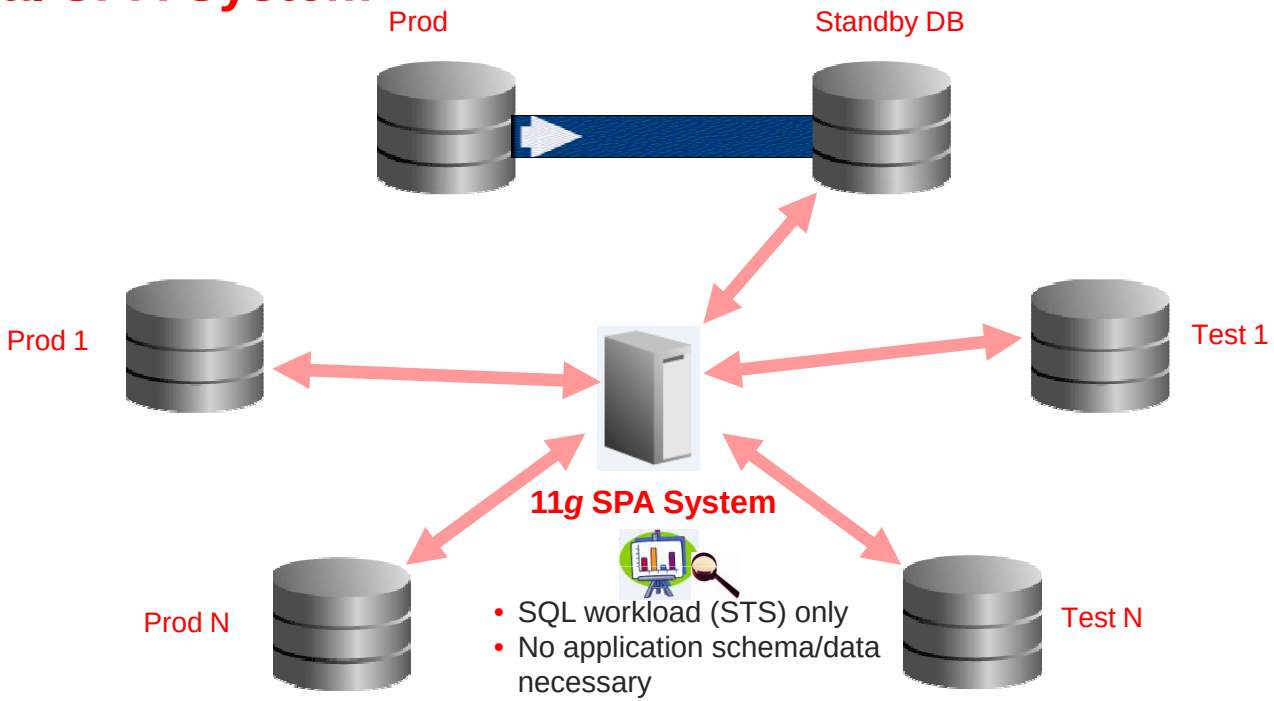
- Use scoped or private session testing where possible

Leverage Oracle Active Data Guard (Read-only Physical Standby) for testing

- Use idle standby resources for testing
- Provides full and current dataset

New Techniques for Avoiding SQL Regressions

Central SPA System



What is SPA system?

- Remote test executes SQL workload (STS)
- Performs reporting/analysis
- Any 11g database, preferably latest release
- Not mandatory except for testing pre-11g upgrades or 10.2.0.x → 10.2.0.y
- Same or higher version than the target database being tested

Benefits

- Use latest software for analysis
- Can be used as repository, helps persist results across database refreshes

Tips

- Use logon triggers to set environment or to make change on remote system

New Techniques for Avoiding SQL Regressions

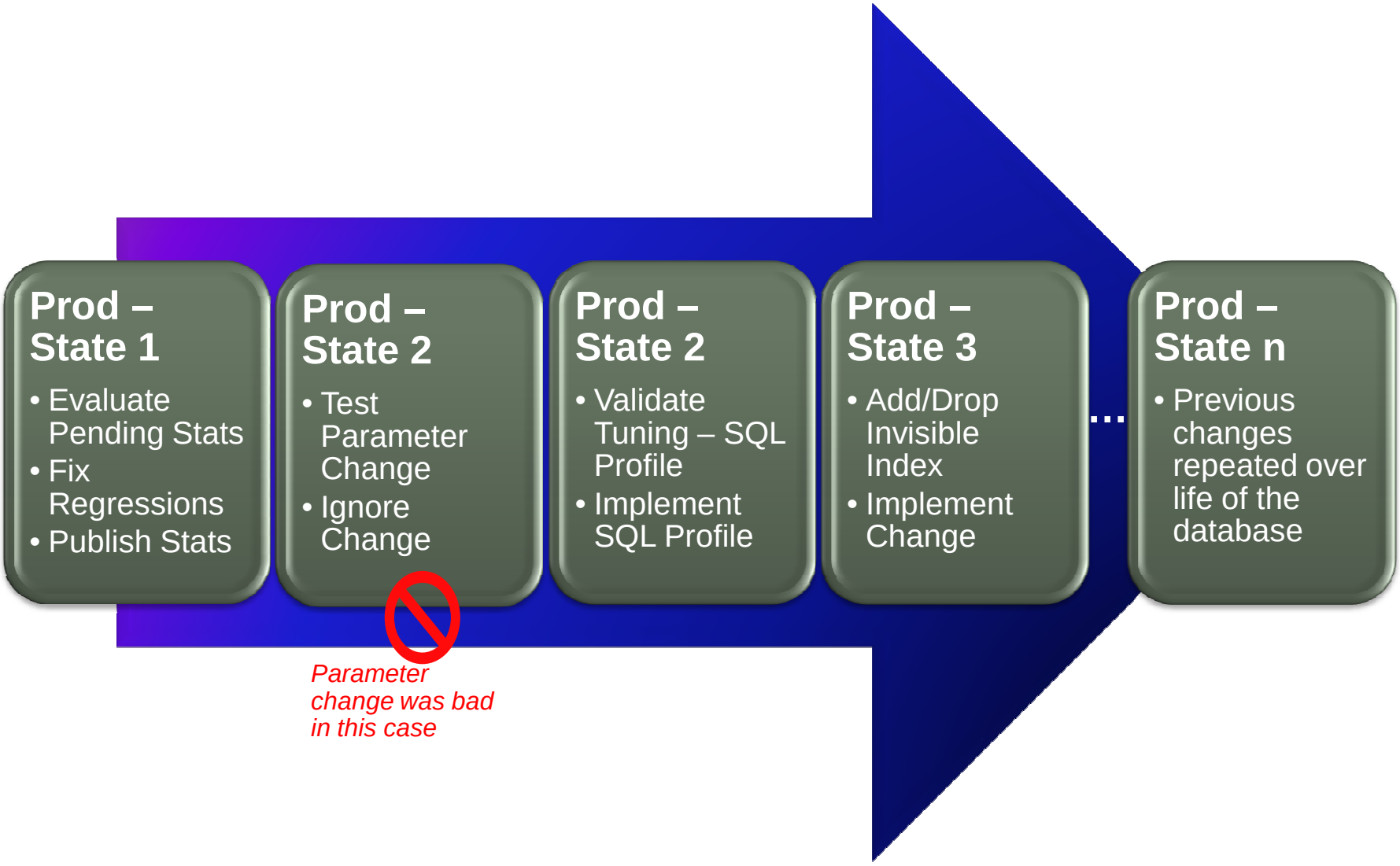
How to *Minimize* Impact on Production?

- Generate Plan Vs Test Execute
  - Use Generate Plan Trial Method to subset SQL with plan changes
  - Only test execute SQL with plan changes
- Limit testing scope to private session or schema where possible
  - Use alter session set <*parameter*> = <*value*>; (Vs system)
  - Example usage with SQL Profiles:

```
alter session set sqltune_category= 'TEST';
exec dbms_sqltune.accept_sql_profile( task_name => :stmt_task,
 category => 'TEST'); -- private scope, do testing !!
alter session set sqltune_category= 'DEFAULT';  -- Now SQL Profiles visible globally to all sessions
```

  - Similarly for pending statistics, invisible indexes
- Use SPA time limit to control resource usage
- Test during maintenance window or non-peak activity when spare resources are available

Example: Using SPA on Production



New Techniques for Avoiding SQL Regressions

Minimize Production Impact - Example

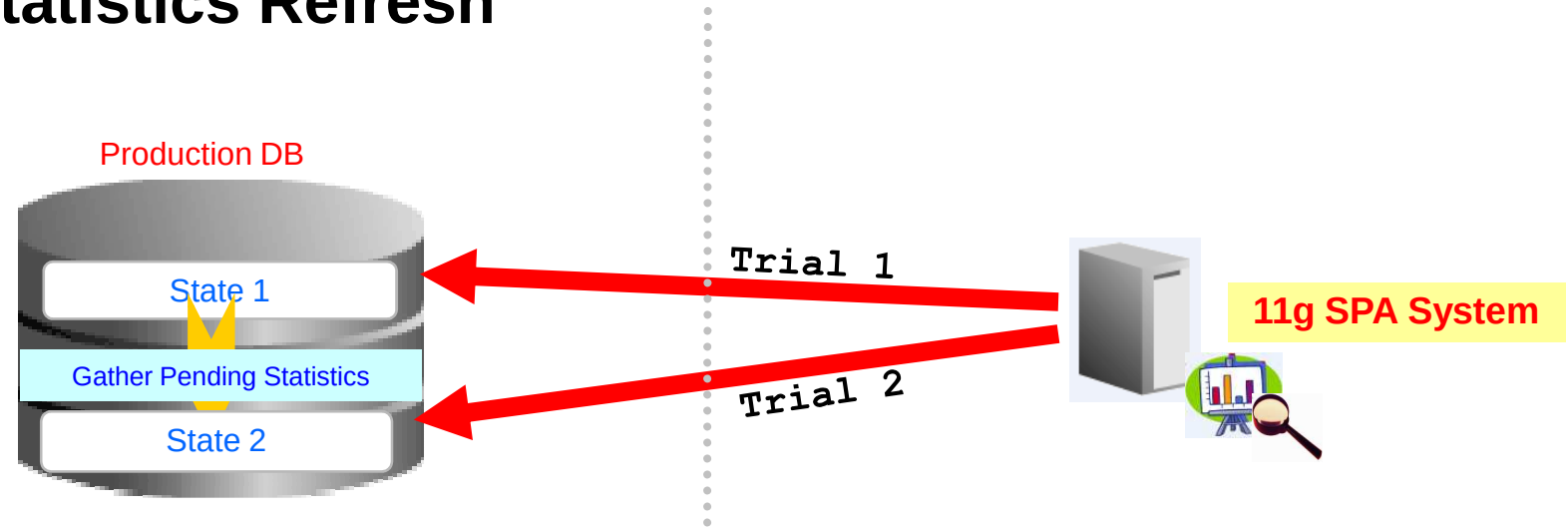
Scenario

- Refreshing optimizer statistics
- Database release is 10.2 or 11*g*
- STS captured on production and transferred to SPA system
- Use SPA to validate/assess impact

Goal

- Identify all SQL regressions
- Ensure no negative effects of change

Using SPA on Production: Validating Optimizer Statistics Refresh



- Use SPA Guided Workflow preferably or API
- Establish first trial remotely using current statistics – baseline
- Make change – gather stats in pending mode
- Establish second trial remotely using pending statistics
- Review SPA report and regressed SQL
- Tune regressed SQL remotely and publish pending statistics

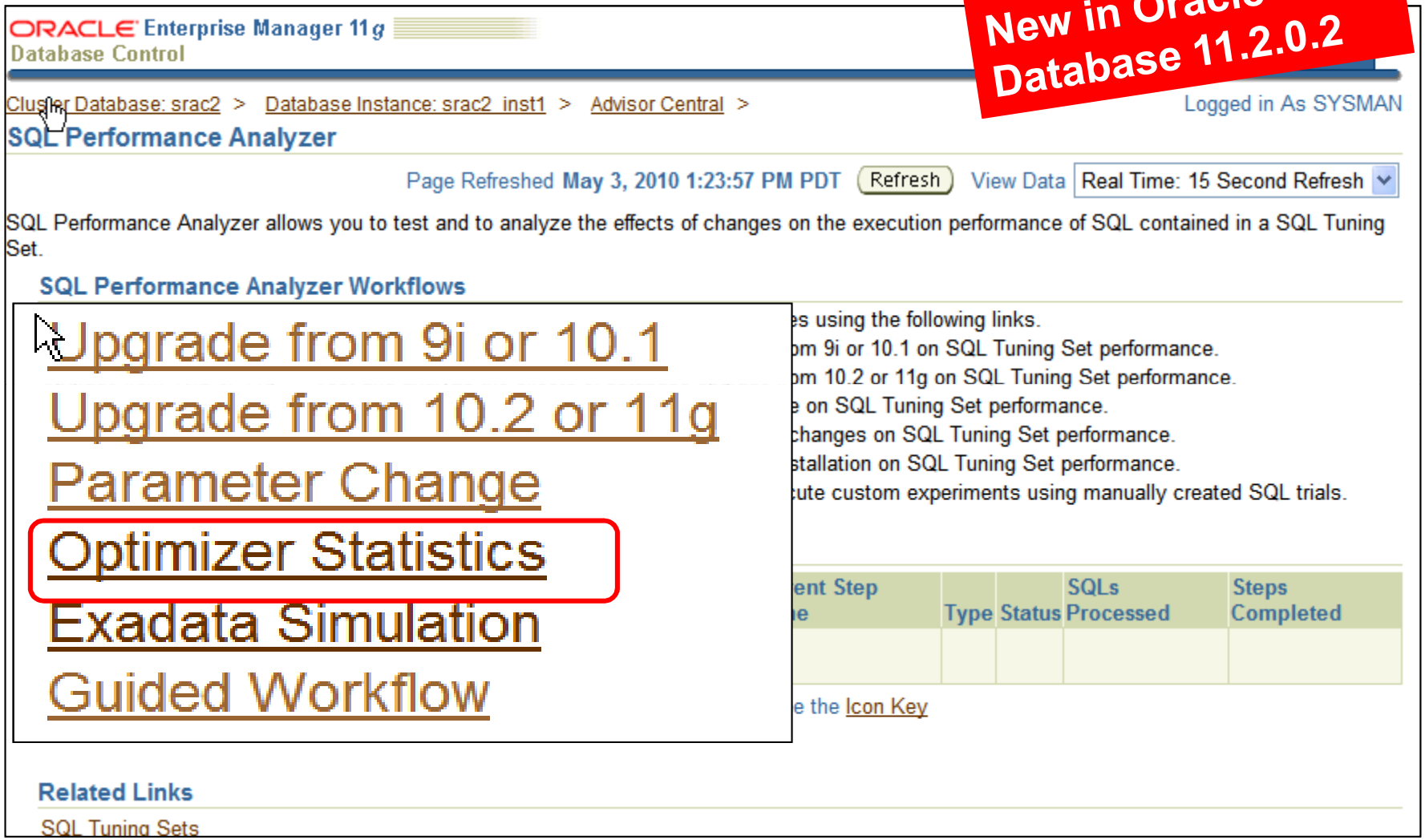
Using SPA on Production: Validating Optimizer Statistics Refresh

- For Oracle Database Release 11.2.0.2:
  - New simplified EM SPA workflow for statistics gathering (next slide)
- For Oracle Database 11g: Optimizer statistics refresh can be validated using:
  - EM: DB Home Page → Server tab -> Manage Optimizer Statistics -> Publish button: False OR

    API: exec dbms_stats.set_global_prefs('publish', 'FALSE');
  - Gather optimizer statistics
  - alter session set optimizer_use_pending_statistics = TRUE;
  - EM: DB Home Page → Server tab -> Manage Optimizer Statistics -> Publish button: True
- For Oracle Database Release 10.2:
  - Statistics are public once gathered, however SPA validation benefits outweigh risks
    - Pro-actively validate statistics refresh
    - For regressions, revert back to old statistics or tune SQL
    - Helps avoid firefights, down time, missed SLAs

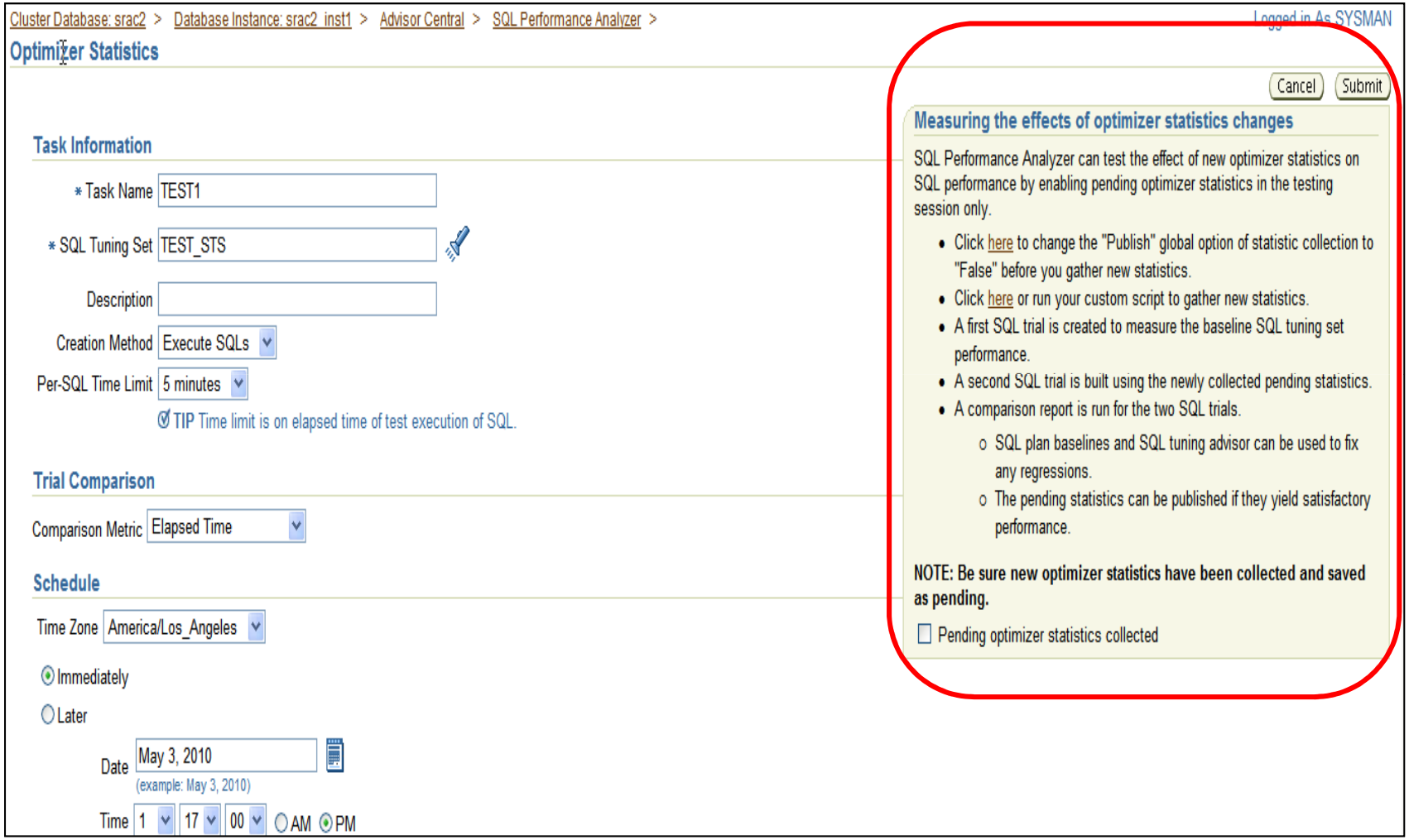
Enterprise Manager SPA Workflow: Validating Optimizer Statistics Refresh (1)

New in Oracle Database 11.2.0.2



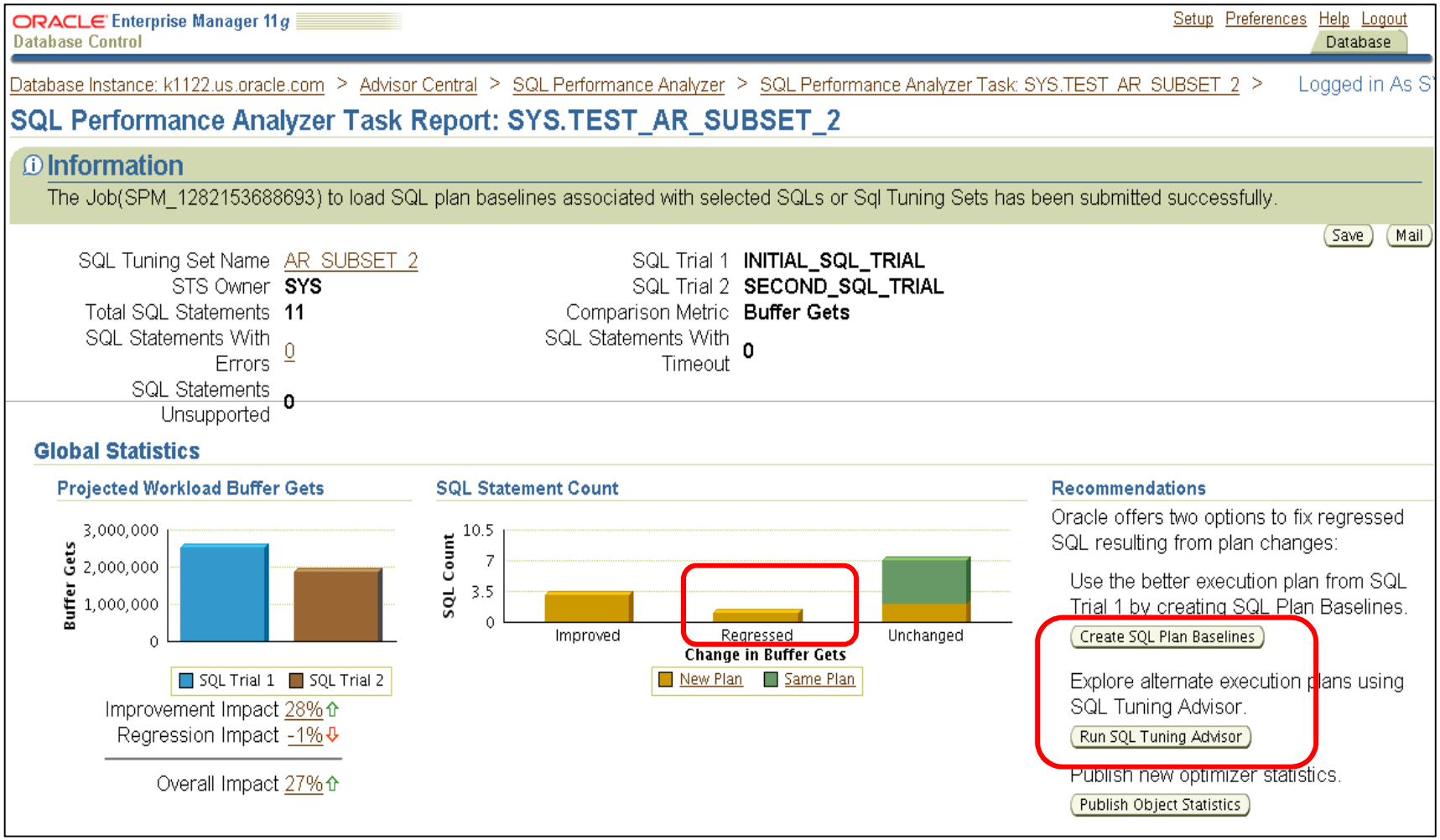
DB Home Page -> Software and Support tab -> SPA Link

Enterprise Manager SPA Workflow: Validating Optimizer Statistics Refresh (2)



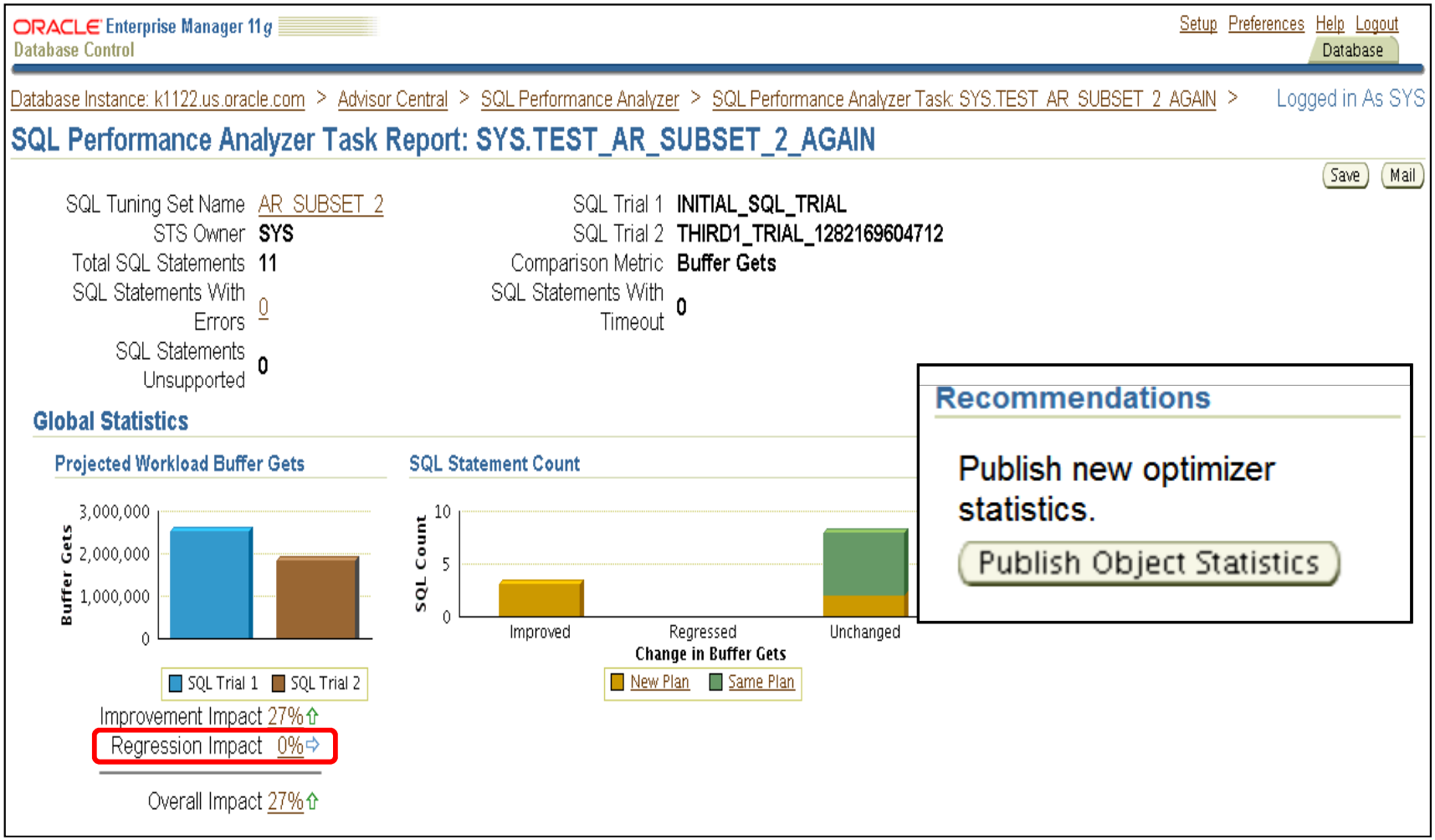
Enterprise Manager SPA Workflow: Validating Optimizer Statistics Refresh (3)

SPA Report: Remediating Regressed SQL



Enterprise Manager SPA Workflow: Validating Optimizer Statistics Refresh (4)

Publishing Pending Statistics



Using SPA on Production: Assess Parameter Changes

Scenario

- Assess parameter change (e.g., optimizer_features_enable, optimizer_mode, etc.) on workload
- Database release is 10.2/11g
- Captured STS is transferred to SPA System
- Use SPA system to validate/assess impact

Goal

- Identify all SQL regressions
- Ensure no negative effects of change

Using SPA on Production: Assess Parameter Changes

- Use SPA Guided Workflow preferably or API
- Establish first trial remotely – baseline
- Make scoped/private session change, e.g., optimizer_features_enable, optimizer_mode
- Establish second trial remotely
- Review SPA report and regressed SQL
- Tune regressed SQL remotely, and implement change at database level

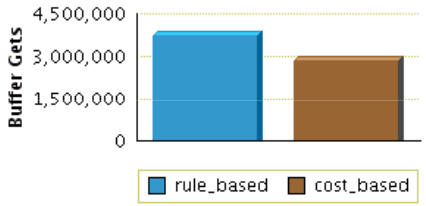
Using SPA on Production: Assess Parameter Changes: SPA Report

SQL Performance Analyzer Task Result: SYS.PARAM_CHANGE

Task Name	PARAM_CHANGE	SQL Tuning Set Name	HR_WORKLOAD	Replay Trial 1	rule_based
Task Owner	SYS	STS Owner	APPS	Replay Trial 2	cost_based
Task Description	test rule-based vs cost-based optimizer	Total SQL Statements	50	Comparison Metric	Buffer Gets
		SQL Statements With Errors	0		

Global Statistics

Projected Workload Buffer Gets

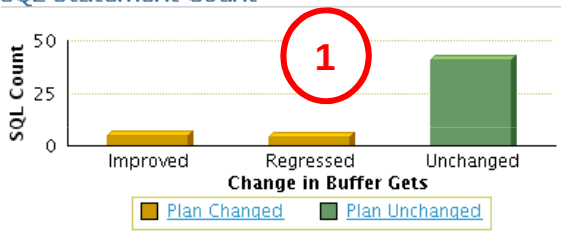


Improvement Impact 28%

Regression Impact -3%

Overall Impact 24%

SQL Statement Count



Regressed SQL tuning done on remote system

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
			rule_based	cost_based		rule_based	cost_based	
⇧	73s2sgy2svfrw	13.790	1,753,552.000	1,238,620.000	29.370	46.950	43.860	Y
⇧	gq2a407mv2hsy	13.790	1,753,552.000	1,238,620.000	29.370	46.950	43.860	Y
⇩	2wtgxbjz6u2by	-3.050	218,621.000	332,519.000	-52.100	5.850	11.780	Y
⇩	fbp9za0hqk2km	-0.070	6.000	2,721.000	-45,250.000	0.000	0.100	Y

Leverage Read-only Standby Databases for SPA Testing

Scenario

- Assess change such as optimizer statistics refresh, parameter, drop index
- Database release is 11g
- SQL workload (STS) exists on SPA system
- How to use SPA on available standby databases to validate/assess impact ?

Goal

- Identify all SQL regressions
- Ensure no negative effects of change
- Leverage idle resources and offload testing on standby databases

Leveraging Standby Databases for Testing - Snapshot Standby

Benefits of Standby Databases from testing perspective

- Provide full and current data set for testing
- Idle cycles/resources can be leveraged for Real Application testing

SPA and Database Replay work with Oracle Database 11*g* Snapshot Standby Database

- Snapshot database is opened for read/write testing
- Snapshot database has to catch-up on the logs that were not applied for duration of testing

SPA support for logical standby databases already exists in Oracle Database Releases 10*g* and 11*g*

Reference: "Beat-up Your Oracle Data Guard Standby with Oracle Real Application Testing – It's Payback Time!"

New Techniques for Avoiding SQL Regressions

Leverage Oracle Active Data Guard for Testing

SPA supports Oracle Active Data Guard (from Database Release 11.2.0.2)

- SPA enhanced to maintain read-only state on physical standby database
- Use SPA system to conduct remote trials on standby database (not mandatory, primary can also be used)
- No full DML test execution support

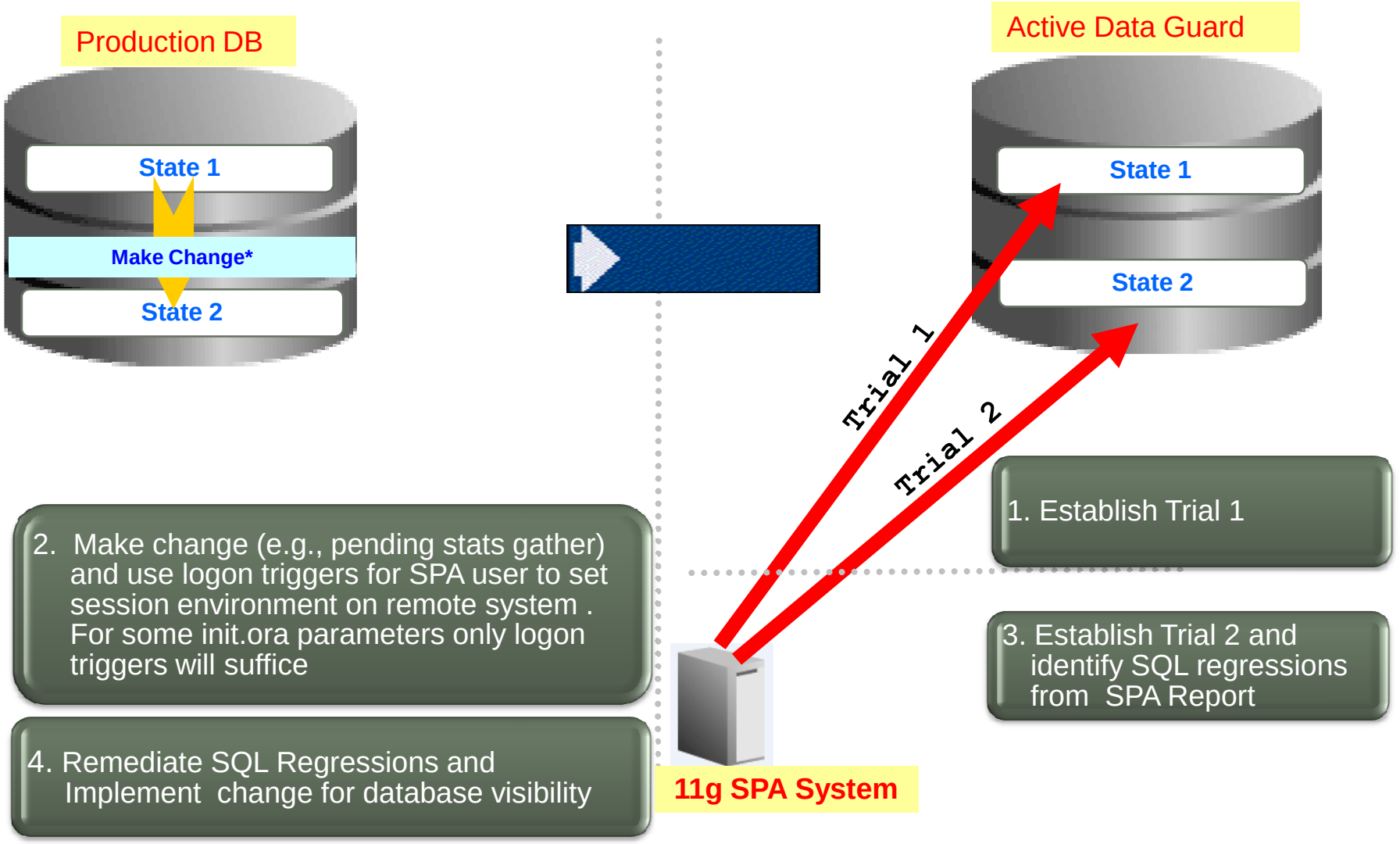
Supported changes - examples

- Optimizer statistics refresh, index add/drop, parameter changes, validation of SQL tuning – profiles, baselines

Testing Benefits

- Use full and current data set for testing
- Idle cycles/resources can be leveraged for Real Application testing

Using SPA with Oracle Active Data Guard



* Any change scoped/session change such as index drop/add using invisible indexes, optimizer statistics refresh in pending, parameter changes

Outline

- SQL Performance Regressions: Challenges
- SQL Performance Analyzer (SPA): Overview
- SPA Enhancements: Oracle Database 11*g* Release 2
- New Techniques for Avoiding SQL Regressions
- Real-world Customer Case Studies
- Conclusion

Customer Case Study (1): Real Application Testing Usage at DIRECTV

SPA Usage @ DIRECTV

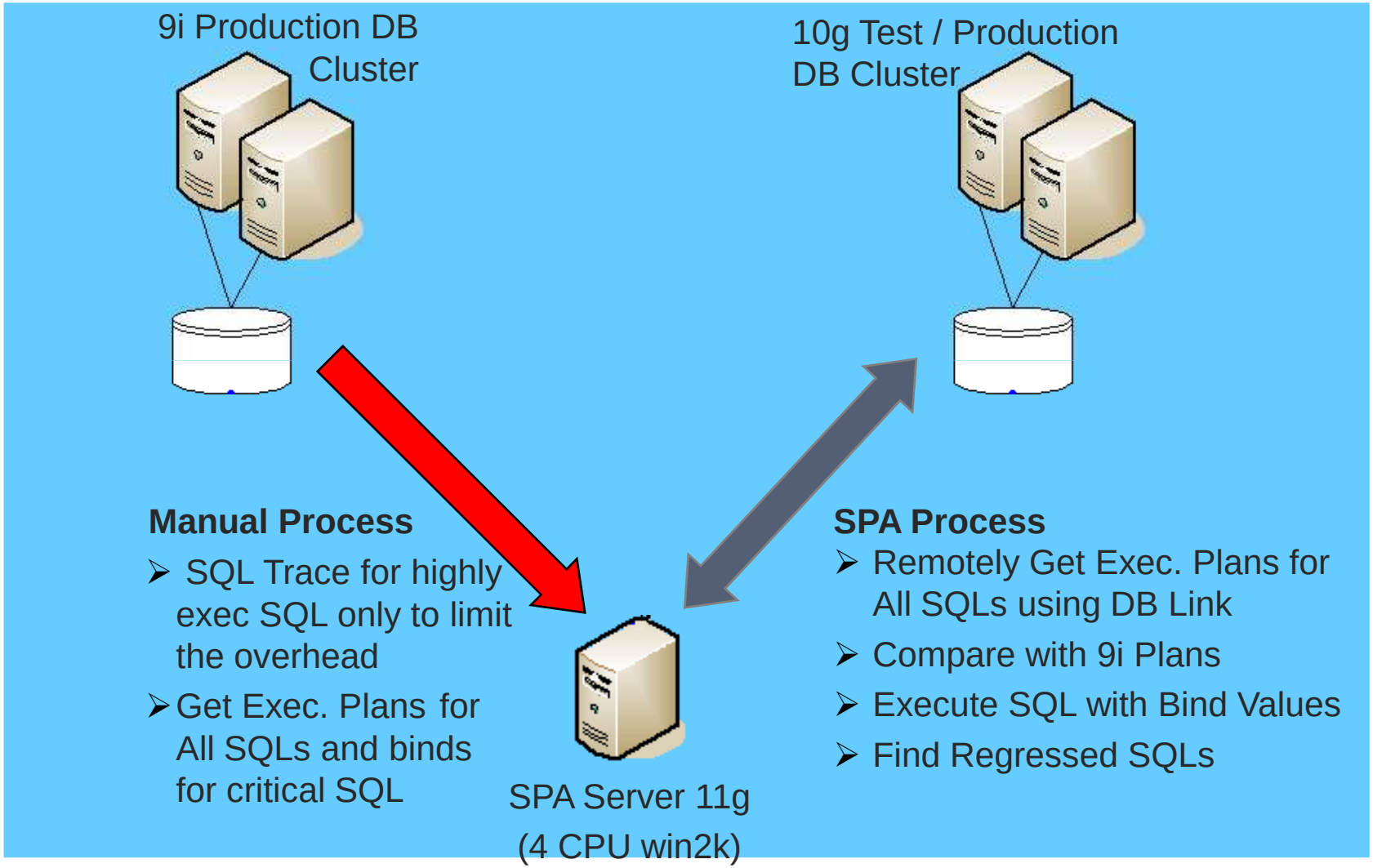
Sameer Marwa

Andrea Ngan

SPA Usage at DIRECTV

- **Background**
  - Large Siebel 7.7 call center implementation: 12.5 TB database
  - 2-Node HP Superdome server, total 224 CPUs, 25+ App Servers
  - High transaction volume, 18k concurrent users, business critical application
- **Challenges**
  - Expensive downtime: $$$/hr
  - Lengthy database/application restart:$$$/hr
  - Multiple team coordination
  - Application Upgrade and complexity: 109k SQL statements
- **SPA usage at DIRECTV**
  - Utilized to upgrade from Oracle Database 9i to 10g
  - Use it to validate all DB changes in Oracle Database 10g
    - CBO statistics refresh
    - Addition of new indexes
  - Few customizations done to handle complexity of the environment

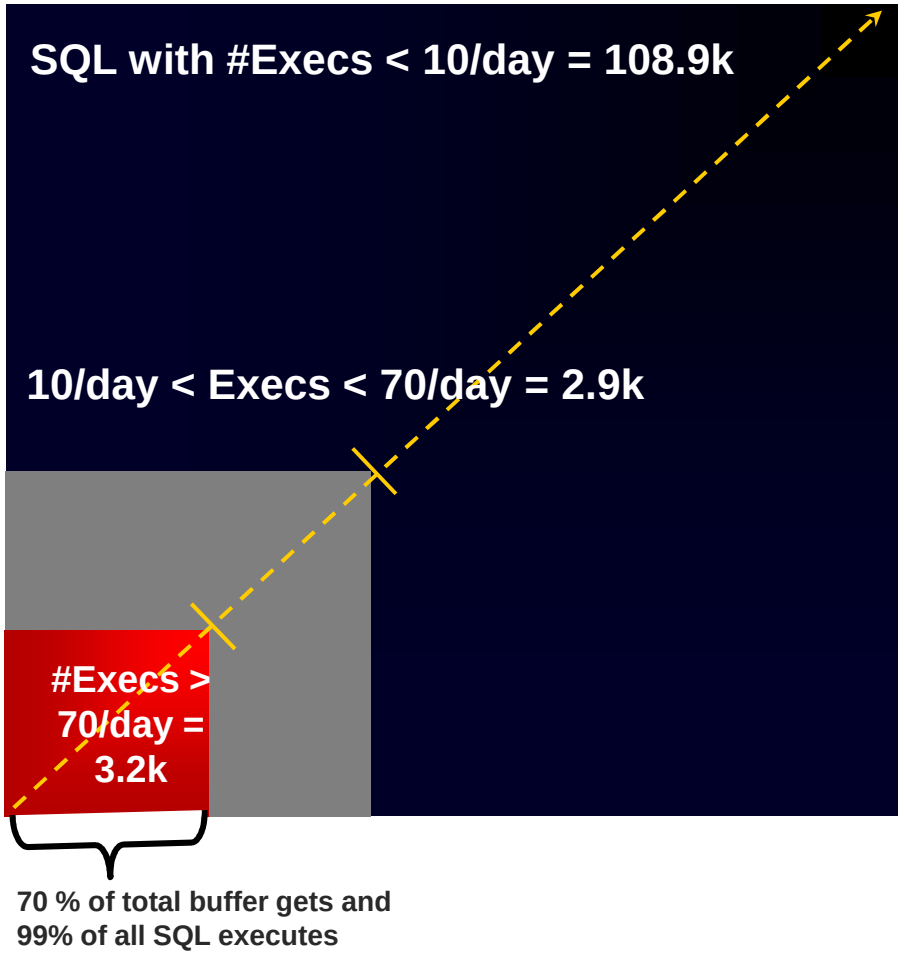
Oracle DB 9i to 10g Upgrade: Environment



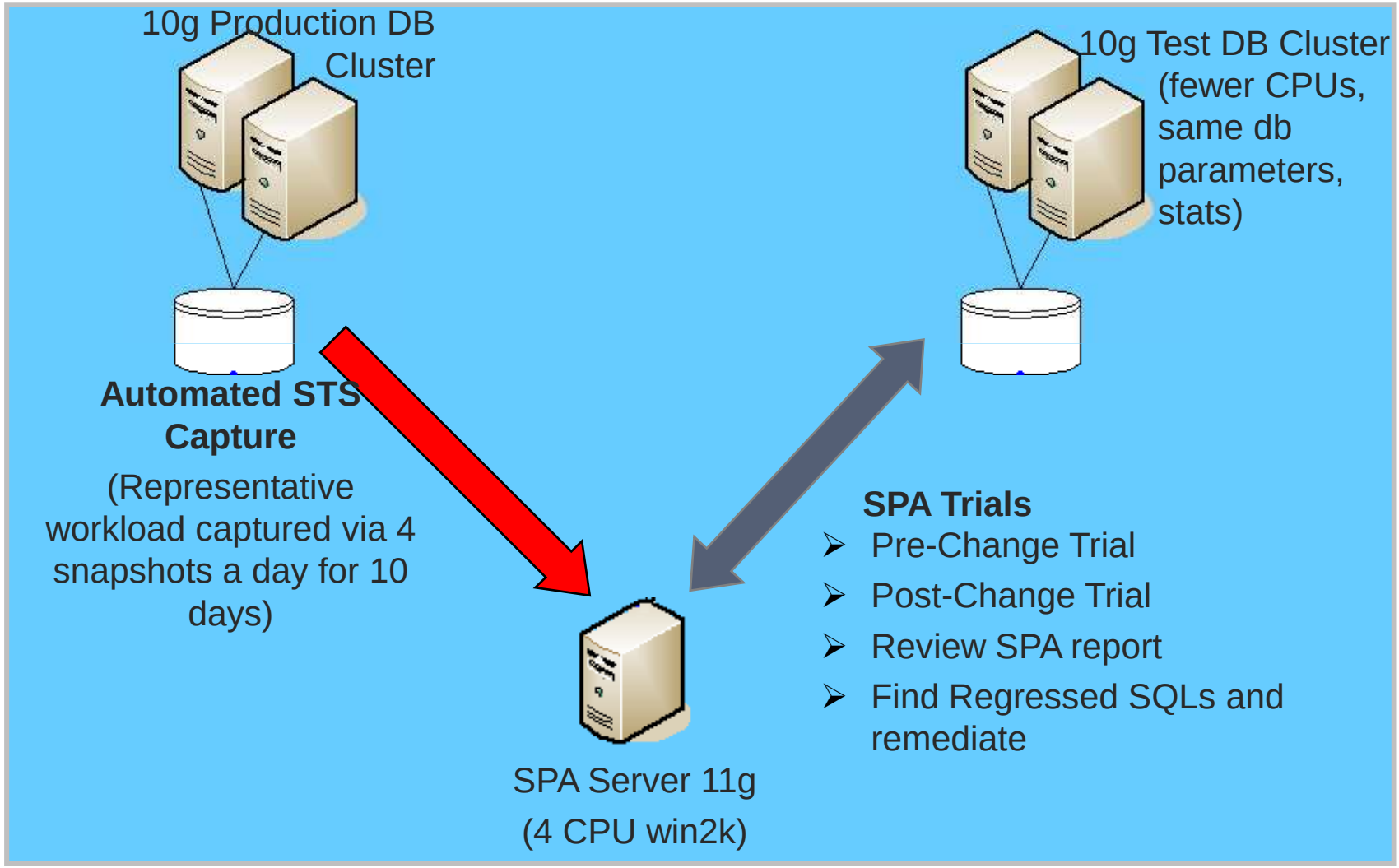
Refer to OOW 2009: Upgrading Oracle Database 9i to 10g for Siebel using SPA

Oracle DB 9i to 10g Upgrade: Summary

- **Focused on SQL with high executions to limit scope of work while covering majority of workload**
  - SQL with executions > 70/day constitute 99% of all SQL executions and 70% of all buffer gets
- **Only 6 out of 109k SQL statements had to be tuned post go-live on Oracle Database 10g**
  - SPA helped find the needle in the stack!
  - Regressed SQL had same execution plans as in 9i, but different in 10g due to bind peeking



Evaluate CBO Stats Refresh & Indexes on 10g: Environment



Evaluate CBO Stats Refresh & Indexes on 10g: Process

- **Step 1: Production Baseline: Execution statistics and plans captured from Production DB serves as “Prod-Baseline”**
- **Step 2: Test system: Pre-Change Trial: SQL Workload replayed, before change to establish “Test-Baseline”**
- **Step 3: Make changes on Test DB**
  - CBO STATS, new indexes
- **Step 4: Test System: Post-Change Trial: SQL Workload replayed post change**
- **Step 5: Pre-Change Vs Post-Change Trials on Test system compared**
  - For additional analysis compare against Prod-baseline

Evaluate CBO Stats Refresh & Indexes on 10g: Process

- **Only SELECT component of the workload replayed against the Test DB (about 90% of workload)**

- **SPA Analysis**
  - "Buffer-Gets" per execution* used to compare the performance of SQLs (since SQL exec time might be affected by slower Test DB)
  - Custom queries used to filter and analyze workload along multiple performance attributes
    - SQLs with > 25% impact, buffer gets/exec > 5k and executions > 10 per day
    - SQLs with > 0 % impact and differing in first step of the execution plan

Finding a Needle In the Haystack with SPA

Server Config:

Prod: HP Superdome (112 CPUs)
10.2.0.4

Test DB: HP Itanium (8 CPUs)
10.2.0.4

SPA DB: Win (4 CPUs)
11.1.0.6

SQL Workload

100k SQLs in Prod DB

40k DML SQLs
(10% of CPU load)

60k SELECT SQLs
(90% of CPU load & 95% of risk to change)

538 Negatively impacted SQLs

SQL with positive or no impact

SQLs with > 1 buffer gets impact

23 SQLs to investigate

Buffers> 5k & Impact>< 25%

SQL < 5k buffers/exec

Buffers> 5k & Impact> 25% & exec> 10/day & driving step difference

SPA Processing:

Only SELECT SQL are executed via SPA with 5 min limit
(6 hrs per trial)

538 SQLs with > 1 buffer difference are flagged as negatively impacted by SPA
(30 mins to run reports)

Custom filter used to narrow down the SQL to be investigated
(30 mins to run custom filter)

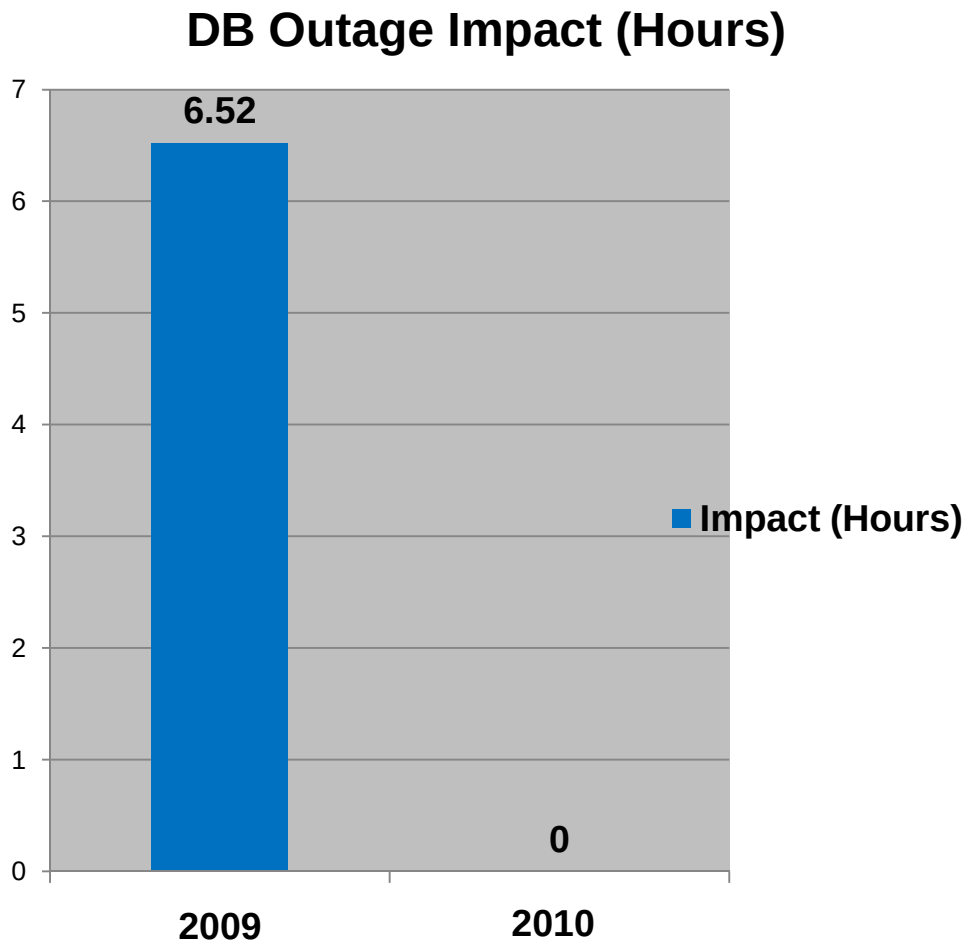
Total Time:
=2x 6hrs+ 30min+30min
=~13hrs

Evaluate CBO Stats Refresh & Indexes on 10g:
Lessons Learned and Summary

- **Most features worked as advertised**
  - Few limitations with BIND variables
    - SPA only executes one set of BIND DATA
    - More than 4k BINDS per SQL and BINDS of complex data types (CLOB,BLOB) are not captured by STS
  - *These can be handled through Database Replay, STS compare functionalities and SPA 11g R2*
- **Analysis of SQL is fast and efficient**
  - Trial of about 60k SQL completed in 6 hours
  - A change can negatively impact thousands of SQL – need to reduce to a manageable set by using custom filters
- **Huge success with SPA**
  - CBO stats refreshed on 10g with only 3 regressed SQL – all non critical
  - Addition of Indexes – improved workload performance

Evaluate CBO Stats Refresh & Indexes on 10g:
Lessons Learned and Summary

- SQL related impact to the business significantly reduced compared to 2009
- No SQL related Incidents since testing with SPA for last 6 months
- Workload SQL response time improved by 25%



GOODTV. BETTERTV. DIRECTV.

Customer Case Study (2): Real Application Testing Usage at Bank of America

Bank of America
Merrill Lynch

Risk Analysis & Management (RAM) Migration & Upgrade

September, 2010

Bill Rice
Vice President, DBA-Team Lead

Bank of America

RAM Application: Overview

What is RAM ?

- **Risk Analysis and Management**: 24x6 trading platform (150+ apps), global deployment
- Main function is position keeping for traders. Supported product types include: Options, swaps, common stock, convertibles and ETFs.
- Helps traders understand risk and minimize negative impact while helping them become aware of variables to optimize trading activities.
- RAM serves *ML front office but also provides back office* settlement and confirmation systems. RAM is a critical TIER 1 application.
- RAM supports global trading activities within Equity Linked, Portfolio, Global Equity Finance and Services (GEF&S) and Cash Trading in HK.
- **Main Business Functions**:

Trade Capture	**Creating and modifying instruments**	**P&L reporting**	**Scenario Analysis**	**Marking of books**
Risk Analysis and Calculation	**Settlement**	**Derivatives Pricing**	**Deal (edits and entry)**	**Workflow**

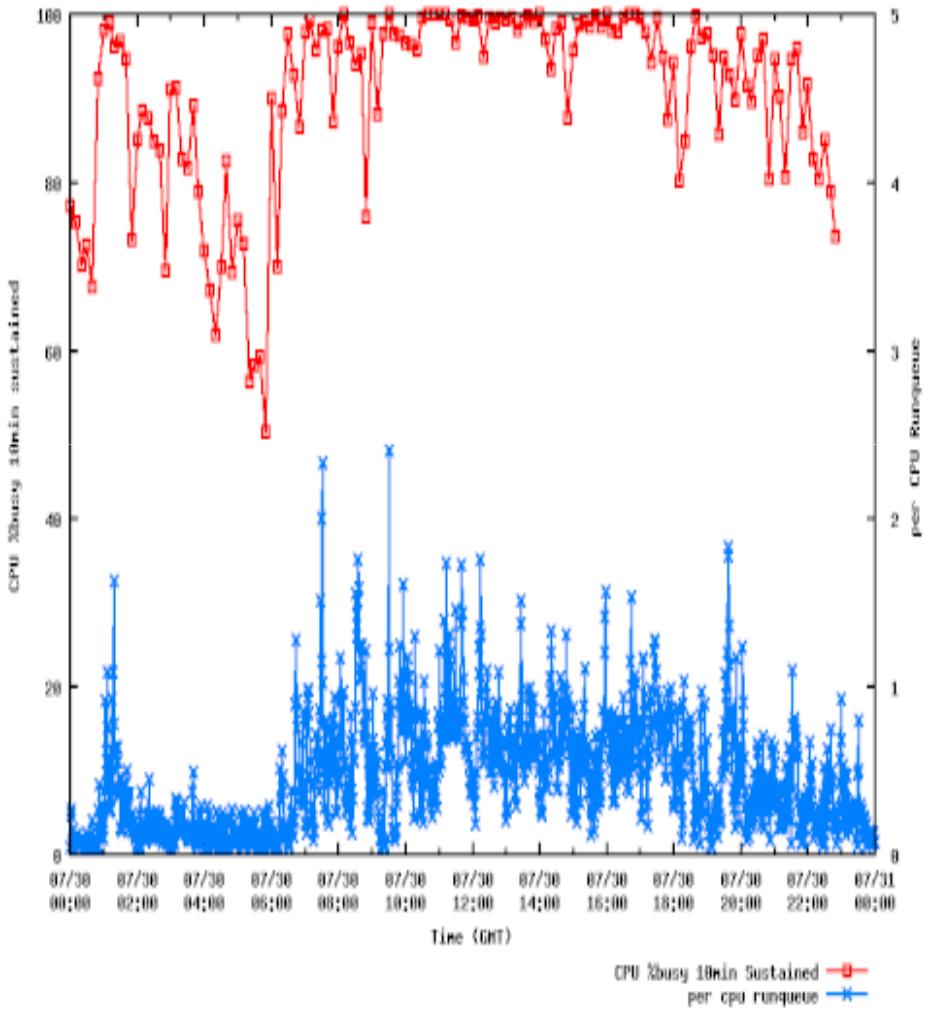
Bank of America

RAM Migration Challenge

Challenge

- Complexity - **7500 nightly batch jobs requiring 24x6 support**
- Scale - RAM **production** platform spread across **3 DB clusters** and **18 separate Oracle databases**
- Older Solaris Oracle 9i Platform was at or **near CPU capacity** – 700+ minutes of sustained DB utilization above 95%
- Critical business need: Upgrade for both support (since 9i) and capacity reasons
- Several hundred-thousand individual SQL Statements to potentially tune for 10g
- Minimal instrumentation to capture production SQL Statement and binds

Host CPU Utilization - %Busy



Bank of America

Solution

- A combined dual migration involving
  - Solaris → Linux
  - Oracle Database 9i → 10.2.0.4
- Phased migration plan: 1. Reporting, 2. Batch, 3. “Live” environment
- Captured production SQL using network appliance or sniffer
  - Since system utilization was near maximum capacity, enabling SQL Trace was not feasible
  - This will no longer be an issue for us from Oracle Database 10g
- Setup performance environment databases similar to production
- Re-played statements in performance environment against databases 9i and 10g using homegrown load scripts
- Performance environment provided flexibility to enable SQL trace, yet capture production SQL

Solution (contd.)

- Enable SQL Trace on Oracle Database 9i in the performance environment for bind capture
- Use SQL Performance Analyzer (SPA) methodology for Oracle Database 9i:
  - Convert SQL trace to SQL Tuning Set (STS)
  - Test execute on Oracle Database 10.2.0.4 in performance environment
  - Perform SQL and detailed plan change analysis
  - Tune identified regressions
- Results:
  - 50 regressions discovered out of 1 million SQL, cause of regression
  - Tuned through working with Oracle Support and several changes
    - Stored Outline (1 query)
    - _b_tree_bitmap_plans = FALSE (to force 9i ‘OR expansion’ behavior)
    - alter session set "_FIX_CONTROL"='4600710:OFF'; (for 9i in-list behavior)
    - Index creation (1 query)

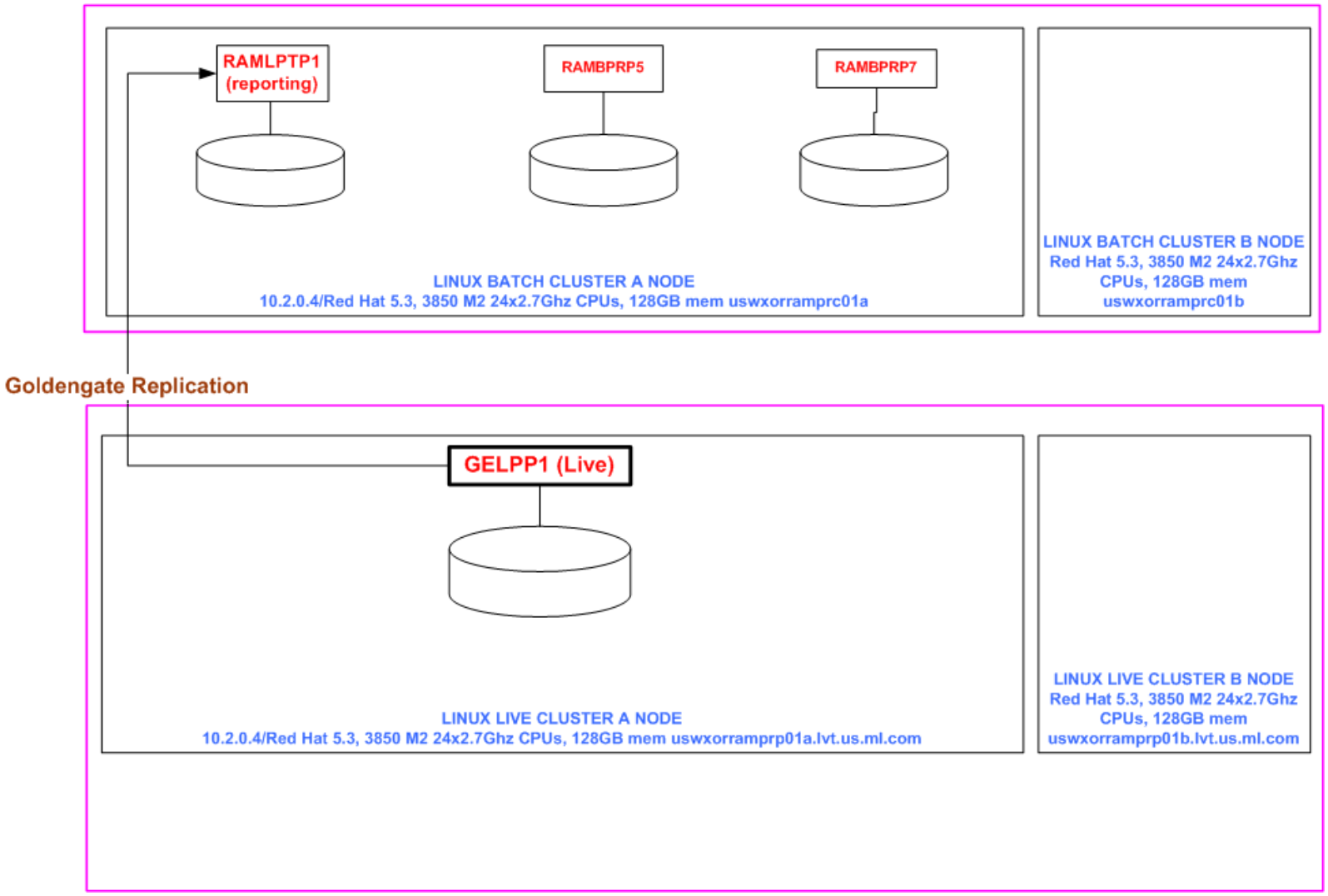
Summary of Success

- Nearly flawless transition to 10g
  - Only 1 undiscovered plan regression in 1 million unique SQL
- Achieved goal of minimizing risk to our business partners of slow or unresponsive application.

- Live for 1 month now with no issues.
- Book marking process was between 30-50% faster on 10g
- 10g test ran 169% faster than 9i (53 minutes vs. 143 minutes)
- SPA enabled
  - Improved productivity of DBA and Developers
    - Time to solve core issues, rather than file gathering, filtering, looking for individual plan changes and analysis
    - Focus on more strategic issues
  - More thorough analysis
  - More rounds of testing in a shorter timeframe due to efficient testing and analysis process
- SPA resulted in savings of 3-4 weeks of tedious SQL analysis, about 90% reduction in effort!

Bank of America

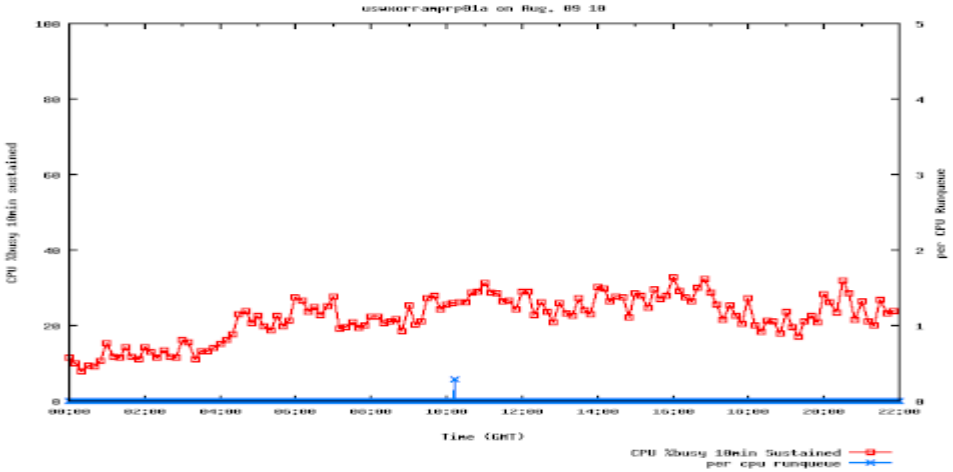
New System Architecture



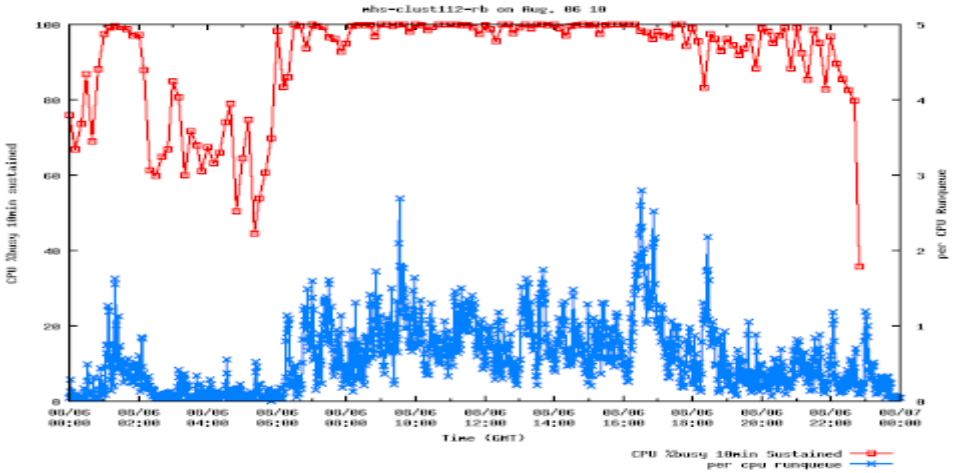
Bank of America

New System Architecture

After migration - CPU Utilization



Before migration - CPU Utilization



Bank of America

Customer Case Study (3): Real Application Testing Usage at NHN

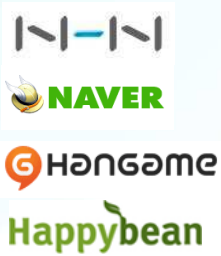
SPA

SQL Performance Analyzer
S317300 Monday

Juncheol Gim, Boonhoon Kim, NHN Korea

NHN Corporation

- South Korea's top internet company
  - Over $1 billion sales on 2009
  - Almost every South Korean uses our services
- www.naver.com
  - Korea's best search portal
  - 17,000,000 daily U.V (30% of South Korean)
  - 950,000,000 daily P.V
- www.hangame.com
  - Korea's leading online game portal
  - 3,000,000 daily U.V
  - 290,000 peek concurrent users
- happybean.naver.com
  - The first online donation portal
  - 4,000,000 users who participated in donation
  - NHN keeps donation and contribution to Korea Society because NHN's role in Korea is growing up

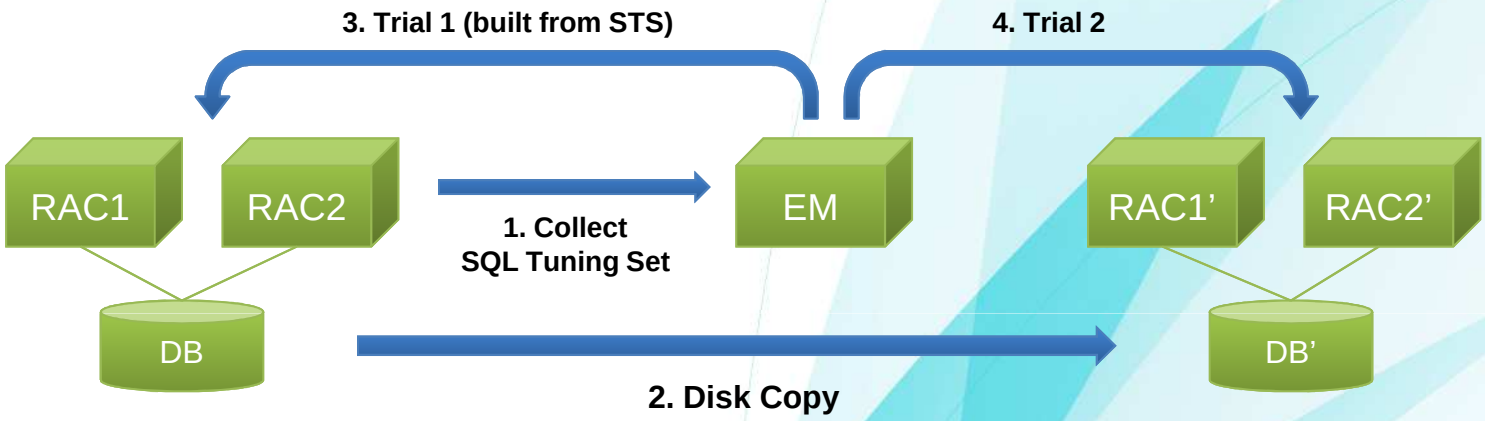


NHN Challenges and Solution

- Need to upgrade major DBs in NHN from 10gR2 to 11gR2
  - Upgrade performed in both Naver and Hangame simultaneously in different projects
- Why choose to upgrade 11gR2?
  - Need read only standby for service ➔ Oracle Active Data Guard provided best solution
  - Had new test infrastructure while all services were newly reorganized
    - Very good chance to test new system thoroughly
    - Minimize impact on production services due to 11g upgrade
- Upgrading to 11gR2 without full testing is very risky
  - Need thorough testing ➔ Need very novel reliable and effective testing method
  - **Considered RAT for this thorough testing solution**

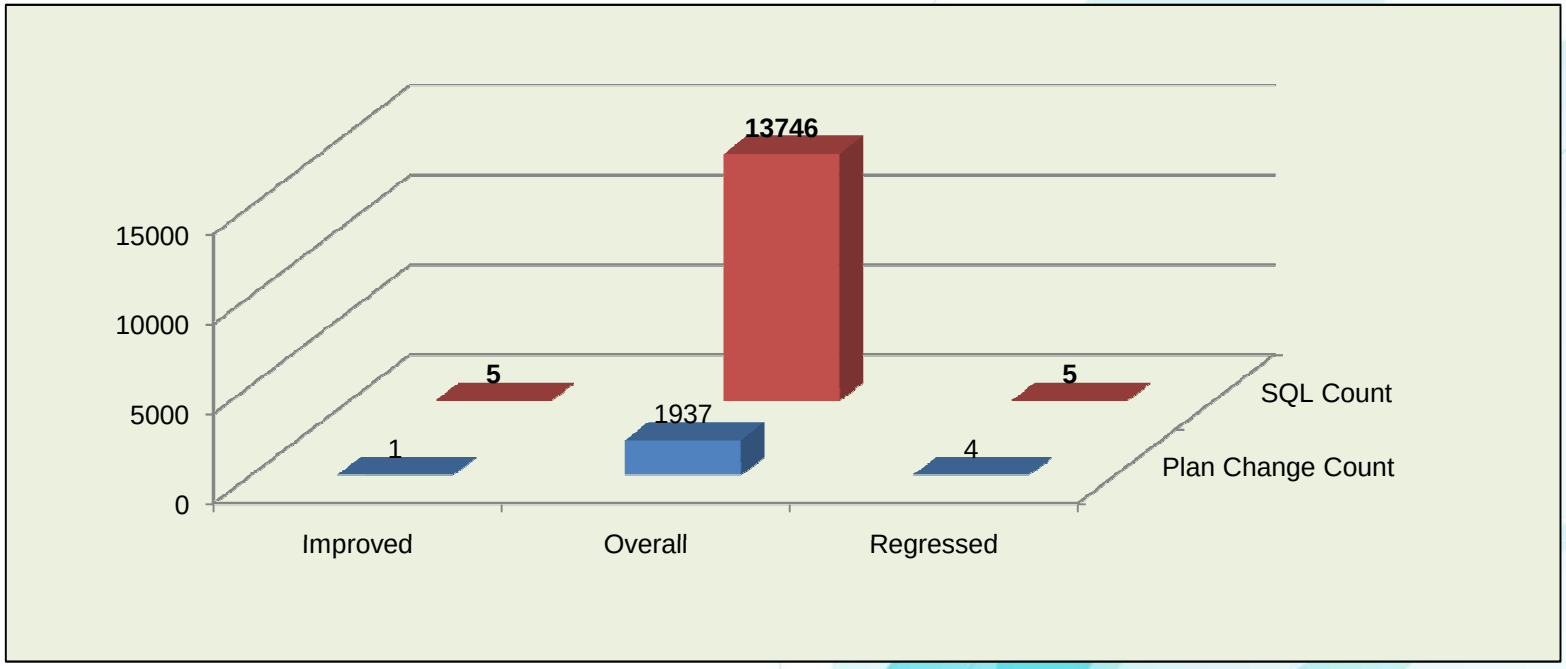
Tests for upgrading to 11gR2	**Database Replay**	**SPA**
Reliability test with real workload	Concurrency test using real workload	SQL-related issues, single user SQL response time test using production binds. Optimizer context
Reliability test of Active Data Guard	Yes	Test queries only, single user full DML testing also possible (11.2)
Performance test	System/Workload through put test	SQL focused testing: SQL Plans changes, single user response time

SPA Workflow



- Database Replay was also used at NHN but for purposes of this session we limit discussion to SPA
- Collected 13.7K queries for 3 days ➔ Those were all queries for that represented workload to be tested
- Test environment used the same types of machines as production system. Both used two nodes RAC and test DB were created through Disk Copy
- Each trial used 10min timeout and 10 executions and was compared with Buffer get.

SPA Results



- SPA Report showed very few query improvements(6) and regressions(9)
- Plan changes in both improvement and regression categories very negligible
- About 2000 queries changed plans but performance remained the same – good news…

SPA Results

- We tuned all 2000 queries because we thought that “Improved”/”Regressed” based on Buffer get was not meaningful
  - Because SPA found 2000 queries with changed plan, we were able to reduce the number of queries which might need tuning to 1/7 of total queries
  - Discussed with application development and manually tuned these statements
  - Staff has good performance tuning expertise
- We used SPA to test if each query was correctly executed as well as to check the performance of each query execution.
  - SPA executed actual query directly in target DB, just like in production
  - SPA helped detect and resolve on ORA-600 for which a fix was provided by Oracle
- Preferred PL/SQL to EM
  - Because report from PL/SQL provided more advanced functionality required for our detailed analysis than from EM
  - For basic reporting, EM reporting is sufficient

RAT - SPA at NHN: Summary

- Performance test with real workload
  - Not synthetic workload
  - Production binds, optimizer settings captured
  - Easier to create workload than Load Runner, captures plans, all relevant performance data easily and automatically
- Capturing workload on production database did not affect performance!
- We think if EM supports advanced reporting and finer level controls, applicability of RAT could be improved
  - In our environment, every plan change even with same performance was investigated to due criticality of application

Outline

- SQL Performance Regressions: Challenges
- SQL Performance Analyzer (SPA): Overview
- SPA Enhancements: Oracle Database 11*g* Release 2
- New Techniques for Avoiding SQL Regressions
- Real-world Customer Case Studies
- Conclusion

Conclusion

- SPA enables businesses to safely test and deploy system changes using real workloads
- Increases business agility and uptime
- Increases staff productivity – less firefighting
- More focus on strategic planning and execution
- Increased capital expenditure savings

- 232% ROI over 4 years*
- <12 months payback period *

*Based on POC and business case from a Fortune 500 Insurance Company

Stadtwerke Munich Services GmbH

What Stadtwerke Munich Services GMBH Is Saying

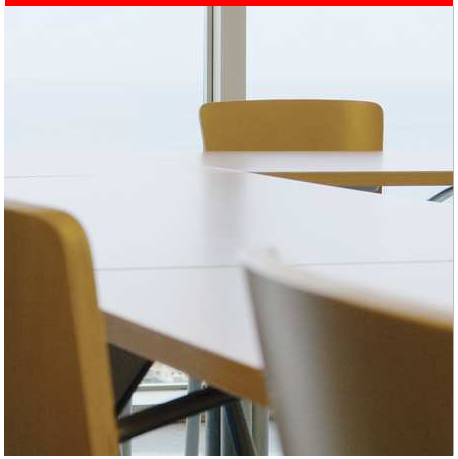
"SPA was particularly helpful in evaluating the performance of important queries at statement level, which must not be negatively impacted"

Manfred Fischer, Manager of SAP System Maintenance, Stadtwerke Munich, SAP customer

ORACLE

Source: Oracle for SAP Technology Update, Vol 19, May 2010

Appendix




ORACLE®