

Database Testing: Introducing DB Infrastructure Level Change with Full Confidence

Prabhaker Gongloor

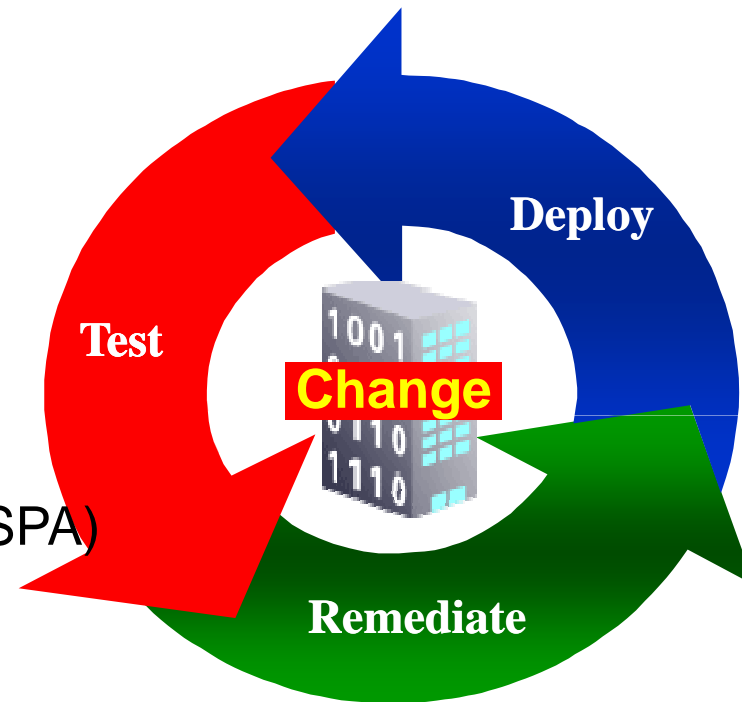
Ashish Agrawal

Presented by: Gary Huffman

Real Application Testing

Load and SQL impact testing solution providing highest quality testing for the database tier

- Benefit
 - Faster technology adoption
 - Lower cost and risk
- Features
 - Database Replay
 - SQL Performance Analyzer (SPA)



Business Agility through Superior Testing



Real Application Testing Features

Database Replay

- Replays *full* production workload to assess impact of system change
 - Concurrency/dependency preserved
 - DML and queries executed
- Types of changes:
 - DB, O/S upgrades
 - Configuration changes, e.g., SI to RAC
 - HW platform migration, etc.
- Comprehensive testing – Best solution for high confidence testing
- Assess overall system impact of change
- Provides performance, data, and error divergence reports

SQL Performance Analyzer (SPA)

- Re-executes *SQL queries* to assess impact of change on performance
 - Execute SQL serially
 - DML not executed
- Types of changes:
 - DB upgrades
 - Optimizer statistics refresh
 - New indexes, Materialized Views, etc.
- Limited to testing SQL response time impact
- Assess impact of SQL execution related changes,
- Provides fine-grained SQL impact analysis



SQL Performance Analyzer (SPA)



SPA Motivation

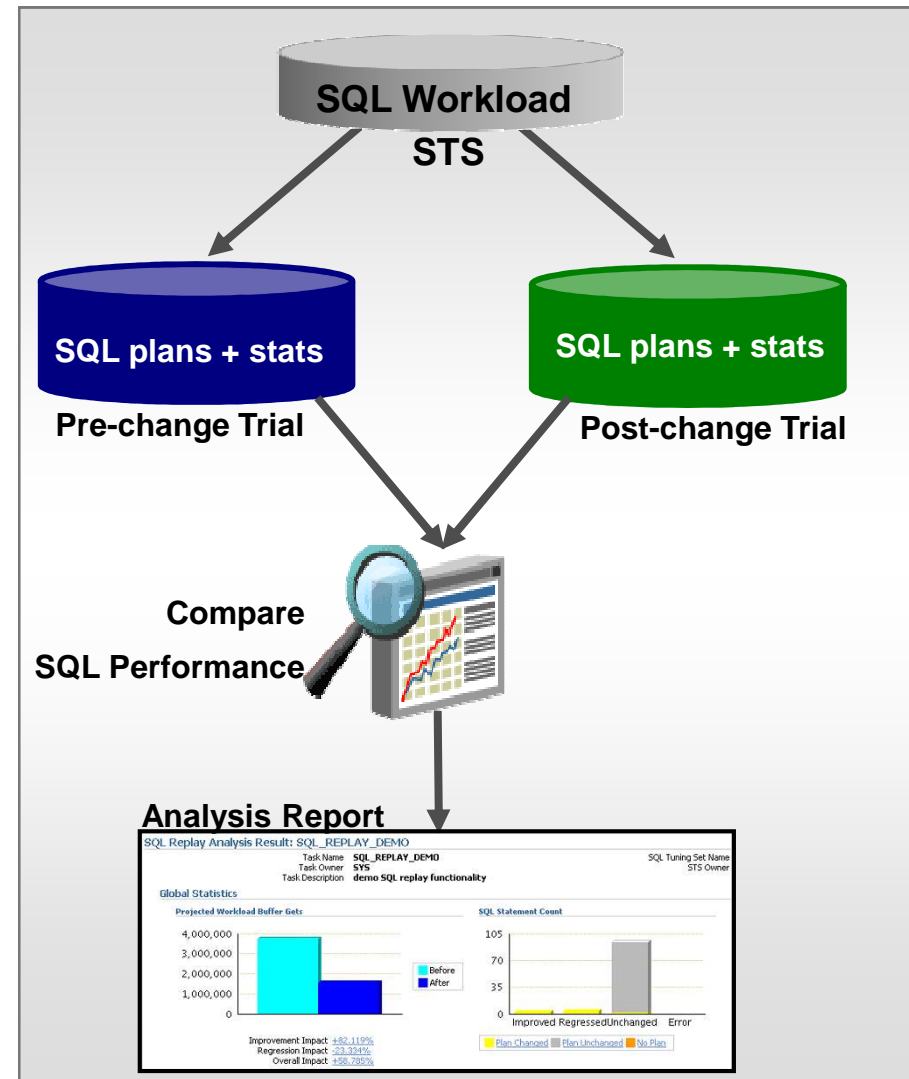
- Businesses need to adapt to changes to stay competitive, compliant and evolve
 - DB upgrades, schema, optimizer statistics refresh
 - SQL performance regressions: #1 cause of poor system perf.
- Current testing landscape and limitations
 - Expensive capture, partial workload, non-production optimizer context, binds
 - Large workloads (**100Ks** SQL stmts are common)
 - Manual and time consuming testing and regression tuning
 - No end-to-end testing solution
 - Test In Production is not too uncommon

SQL Performance Analyzer (SPA)

- Proactively detects ALL SQL regressions, BEFORE actual change is deployed
- Integrated comprehensive solution for end-to-end SQL workload testing

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: Common Usage Scenarios

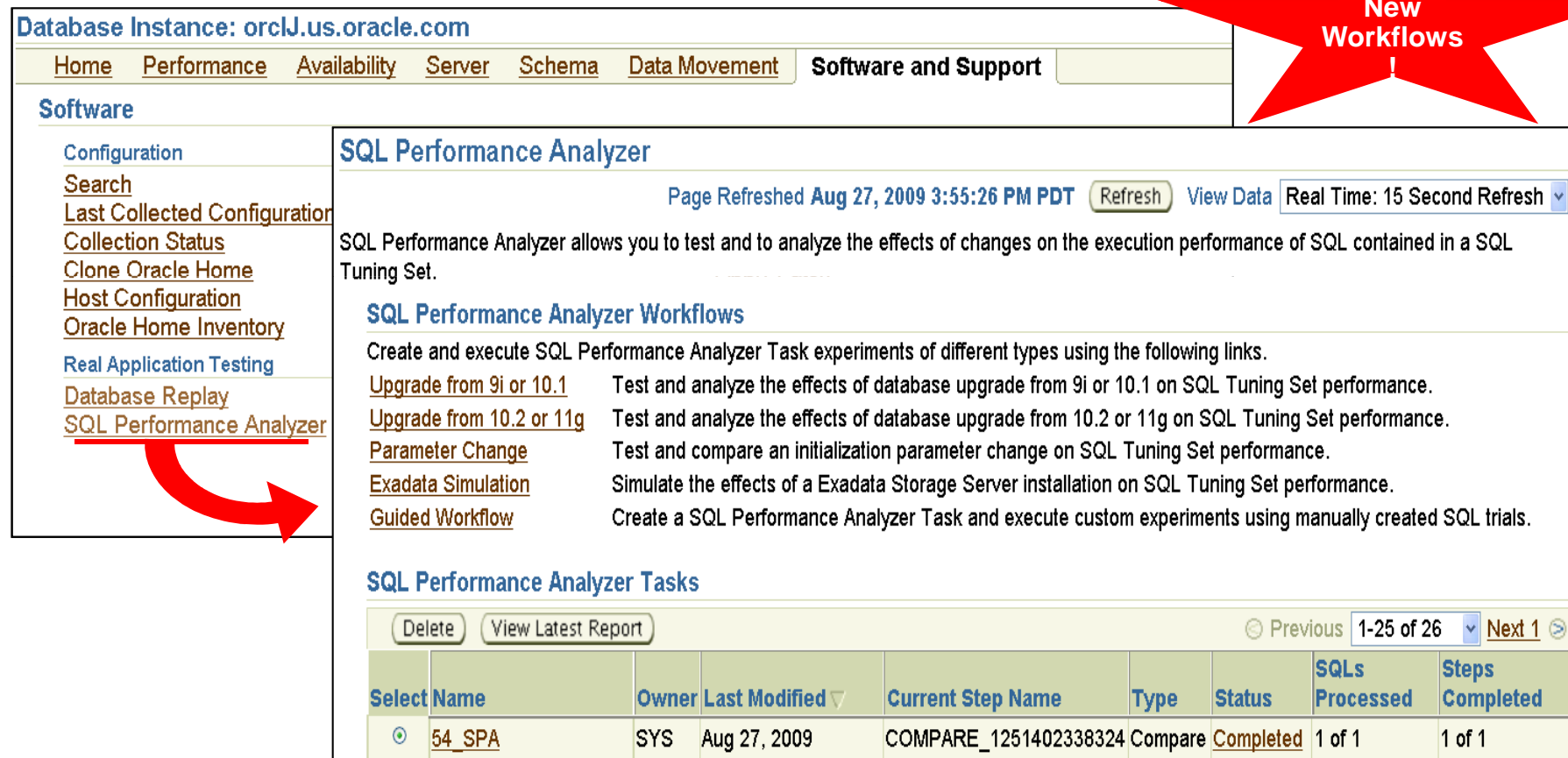
- Database upgrades and patch-set releases
 - 9.2/10.1 → 10.2 or 11g releases
 - 10.2.0.x → 10.2.0.y or 11g releases
- Optimizer statistics refresh
- Database parameter changes
- Database schema changes (e.g., add/drop indexes)
- Implementation of tuning recommendations
- I/O subsystem changes (e.g., ASM, Exadata)

SPA can be used for:

- any change that affects SQL execution plan & performance
- in production as well as test environments

SPA: Enterprise Manager Interface

- Rich GUI through Enterprise Manager – New workflows added!
- DBMS_SQLPA package PL/SQL API



Database Instance: orclj.us.oracle.com

Home Performance Availability Server Schema Data Movement **Software and Support**

Software

- Configuration
- Search
- Last Collected Configuration
- Collection Status
- Clone Oracle Home
- Host Configuration
- Oracle Home Inventory
- Real Application Testing
- Database Replay
- SQL Performance Analyzer**

SQL Performance Analyzer

Page Refreshed Aug 27, 2009 3:55:26 PM 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 11g	Test and analyze the effects of database upgrade from 10.2 or 11g on SQL Tuning Set performance.
Parameter Change	Test and compare an initialization parameter change 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 Previous 1-25 of 26 Next 1

Select	Name	Owner	Last Modified	Current Step Name	Type	Status	SQLs Processed	Steps Completed
<input checked="" type="radio"/>	54_SPA	SYS	Aug 27, 2009	COMPARE_1251402338324	Compare	Completed	1 of 1	1 of 1

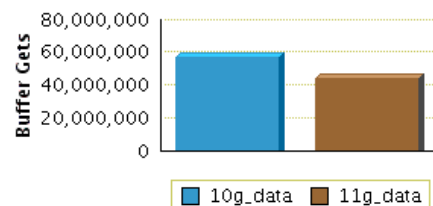
SPA Report

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

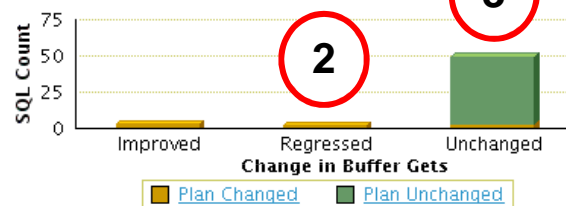
Projected Workload Buffer Gets



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

Overall Impact **22%** ↑

SQL Statement Count



Change in Buffer Gets
Plan Changed Plan Unchanged

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 Report

Regressed SQL Statements

Regressed SQL Statements							
SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	% of Workload		Plan Changed
		10g_data	11g_data		10g_data	11g_data	
2ny751aat2vd9	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530	Y
c2fb0u...7d4p	-0.750	12,					

SQL Details: 2ny751aat2vd9

Parsing Schema DWH_TEST Execution Frequency 1 [Schedule SQL Tuning Advisor](#)

►SQL Text

Single Execution Statistics

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

Plan Comparison

10g_data

Plan Hash Value 393503022

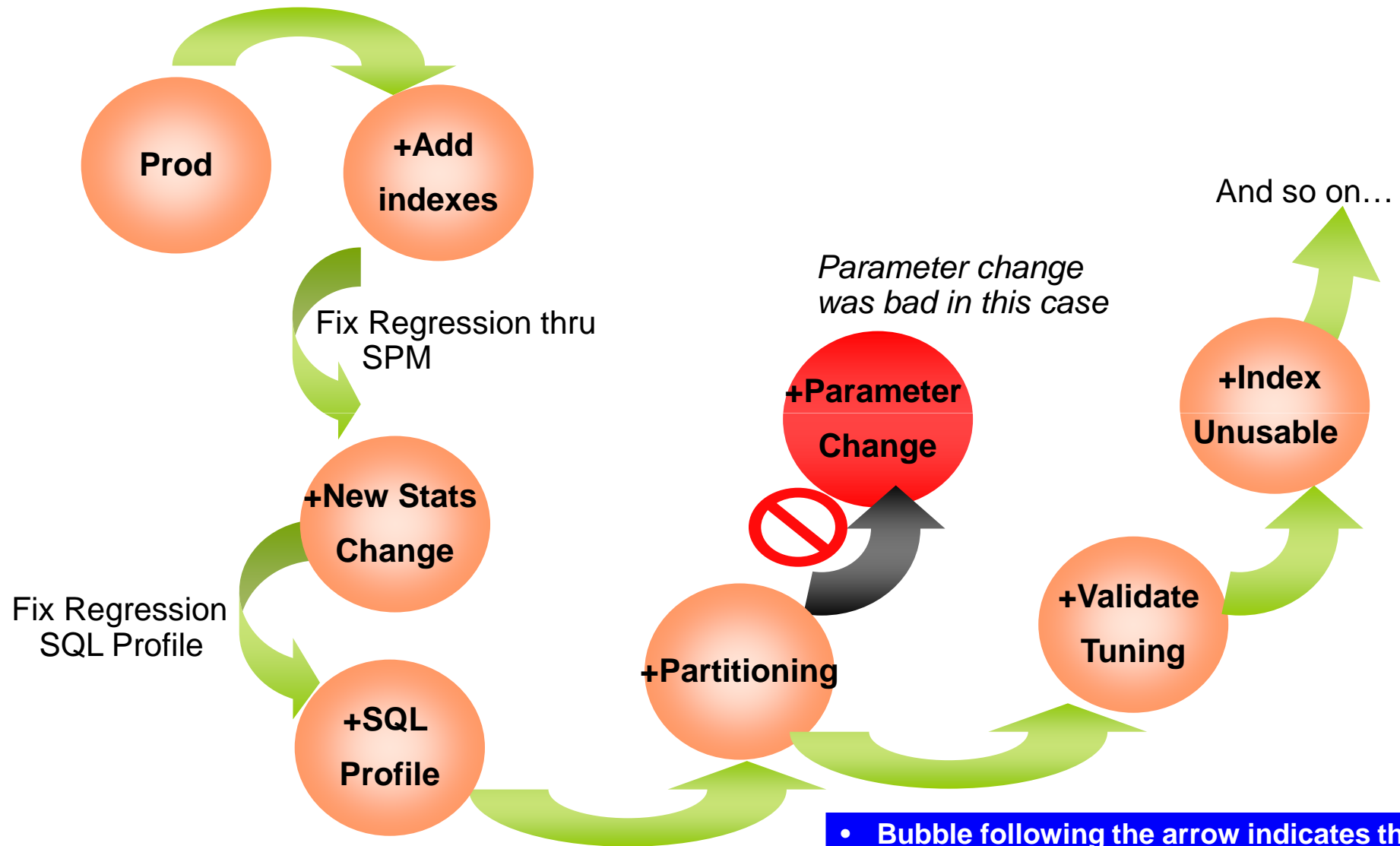
[Expand All](#) | [Collapse All](#)

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



Usage Scenario: Evaluating Changes On Production

Using SPA For Changes in Production: Example



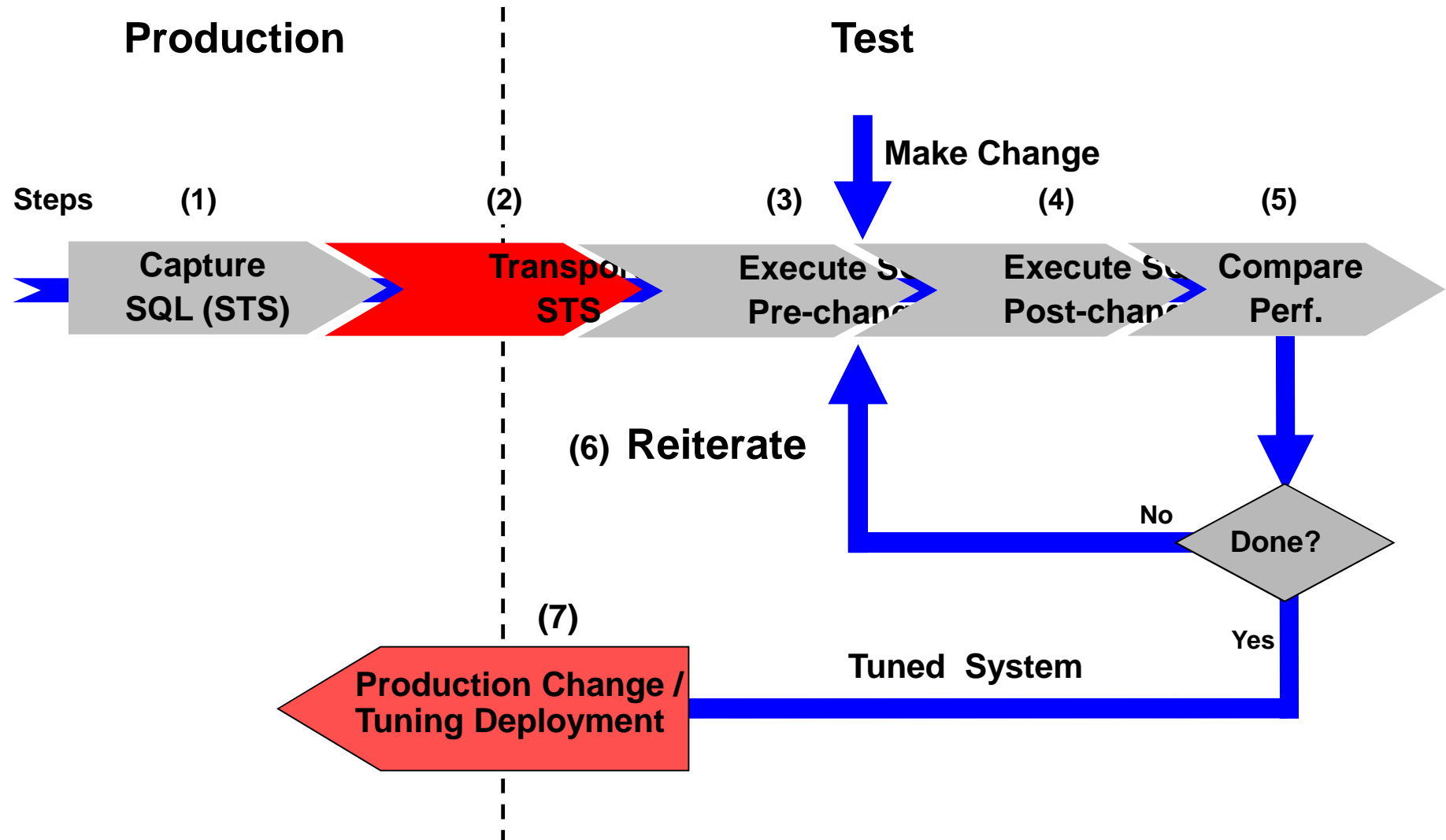
- Bubble following the arrow indicates the delta change on Production
- SPA is used for testing every change



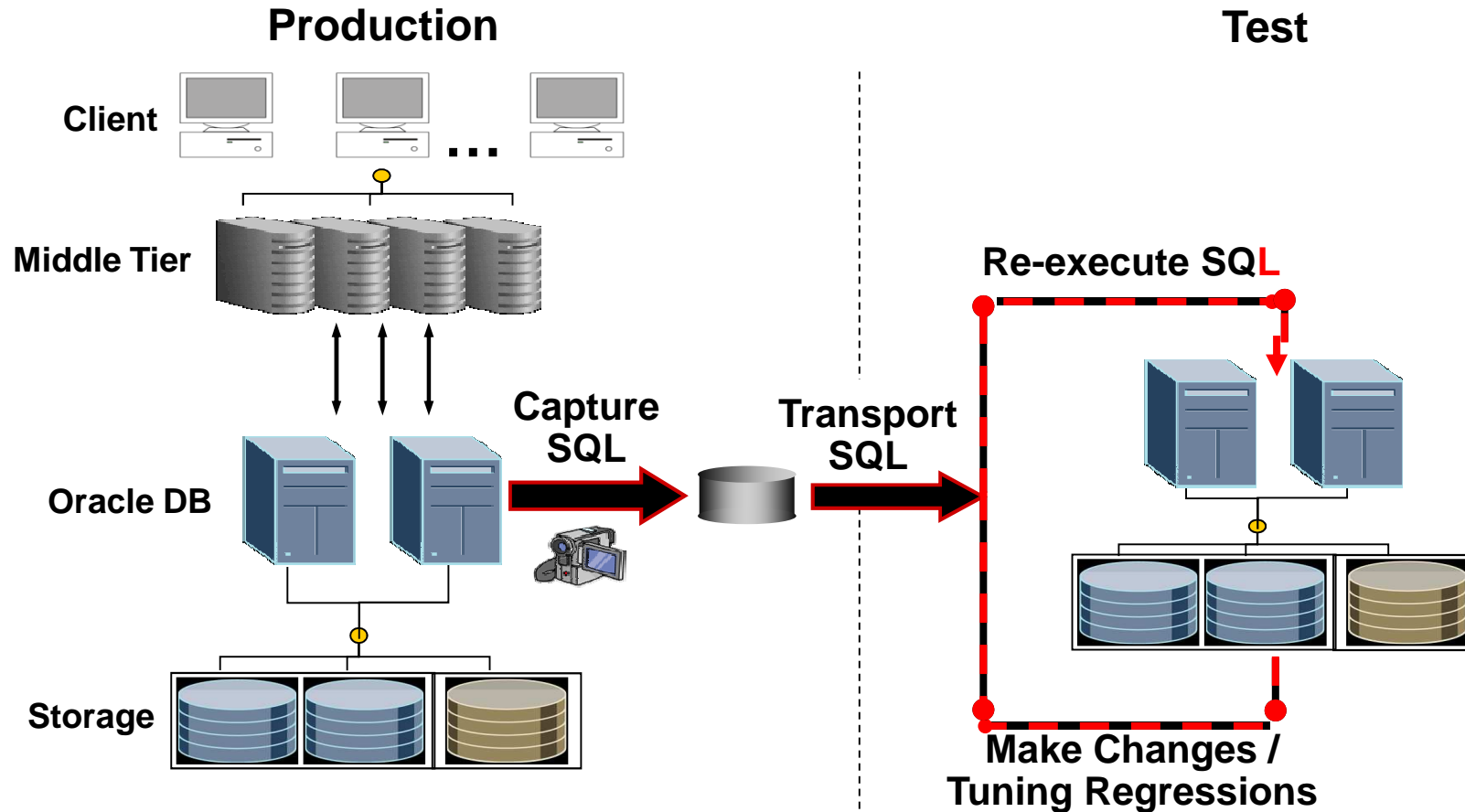
How to Minimize Impact on Production?

- 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 for SQL Profiles:
 - alter session set sqltune_category= 'TEST';
 - *Implement SQL Profiles and test Only sessions with "TEST" sqltune_category see these Profiles - private scope!!*
 - alter session set sqltune_category= 'DEFAULT'; -- Now SQL Profiles visible globally to all sessions
 - Similarly for Invisible Indexes, Pending Stats
- Use SPA time limit to control resource usage
- Test during maintenance window or non-peak activity when spare resources are available

SQL Performance Analyzer: Workflow



SQL Performance Analyzer: Overview



- If adequate spare cycles available, optionally execute SQL here

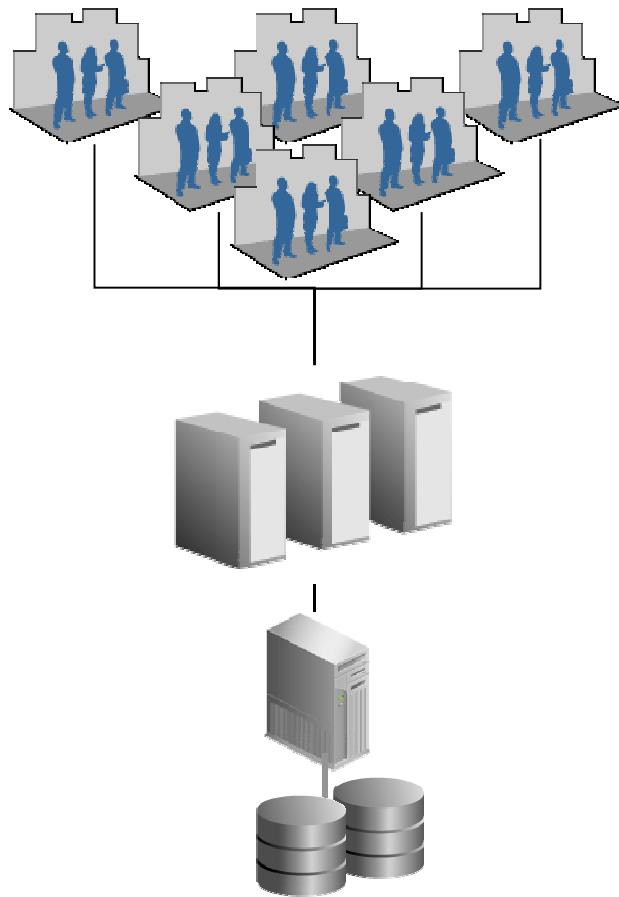
* No middle & application tier setup required



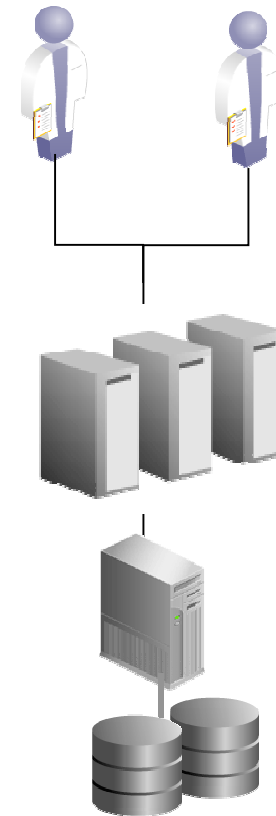
Database Replay

Testing Today

Test – 1-2 testers trying to be 1,000s of users



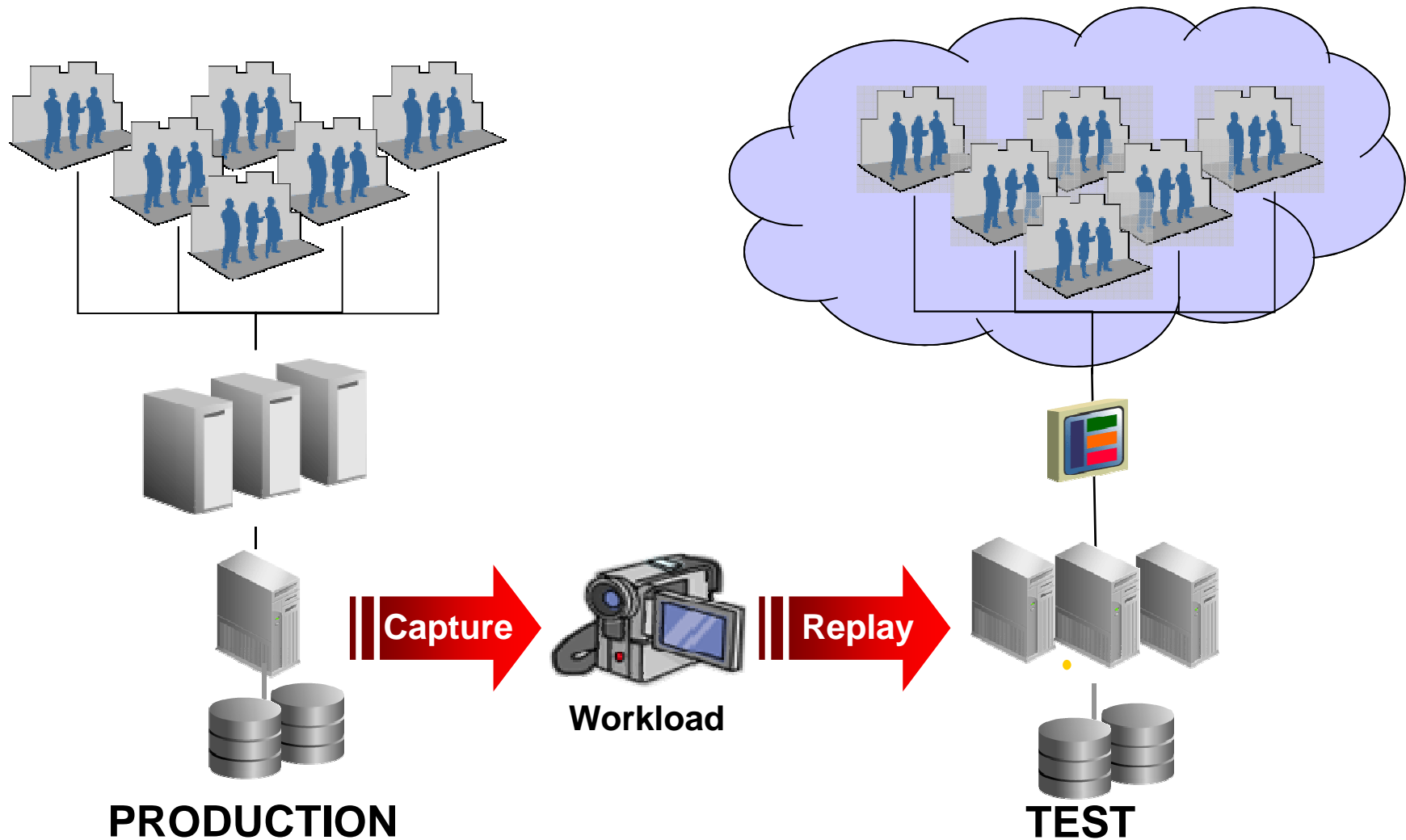
PRODUCTION



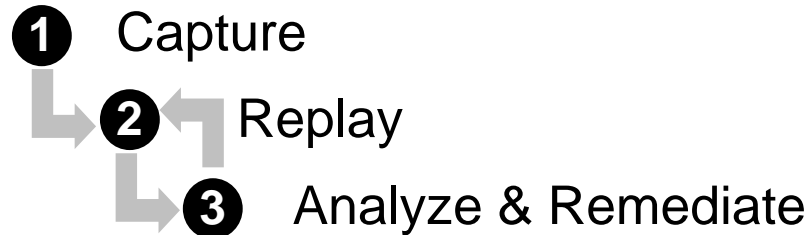
TEST

Database Replay

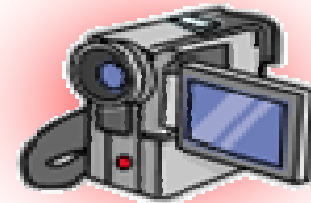
Test your system changes at production levels



Database Replay

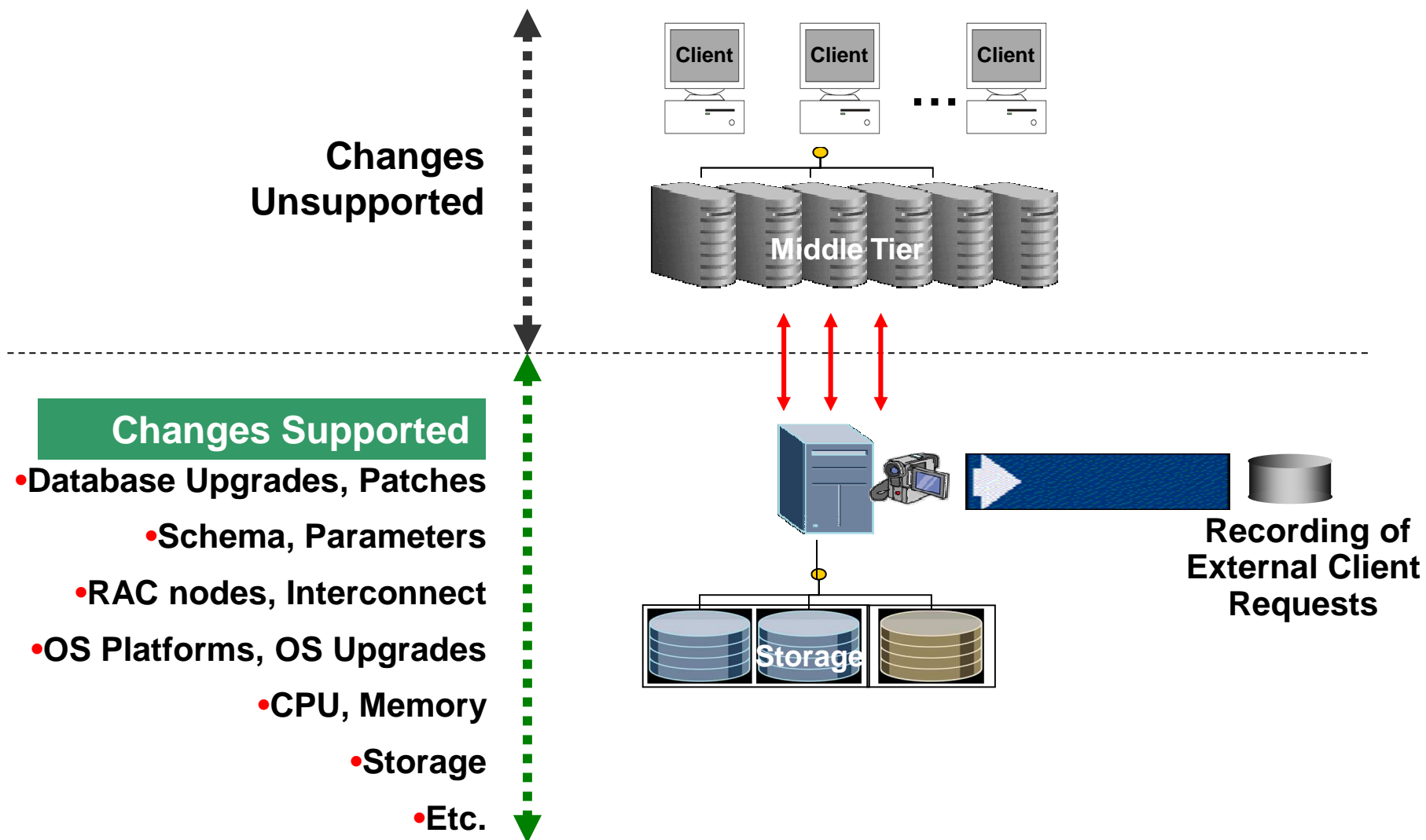


- Capture Workload in Production
 - Capture full production workload with real load, timing & concurrency characteristics
 - Move the captured workload to test system
- Replay Workload in Test
 - Make the desired changes in test system
 - Replay workload with full production characteristics
- Analyze & Report
 - Capture and Replay Reports
 - AWR, ASH, Replay Compare Period Reports



Analysis & Reporting

Supported Changes

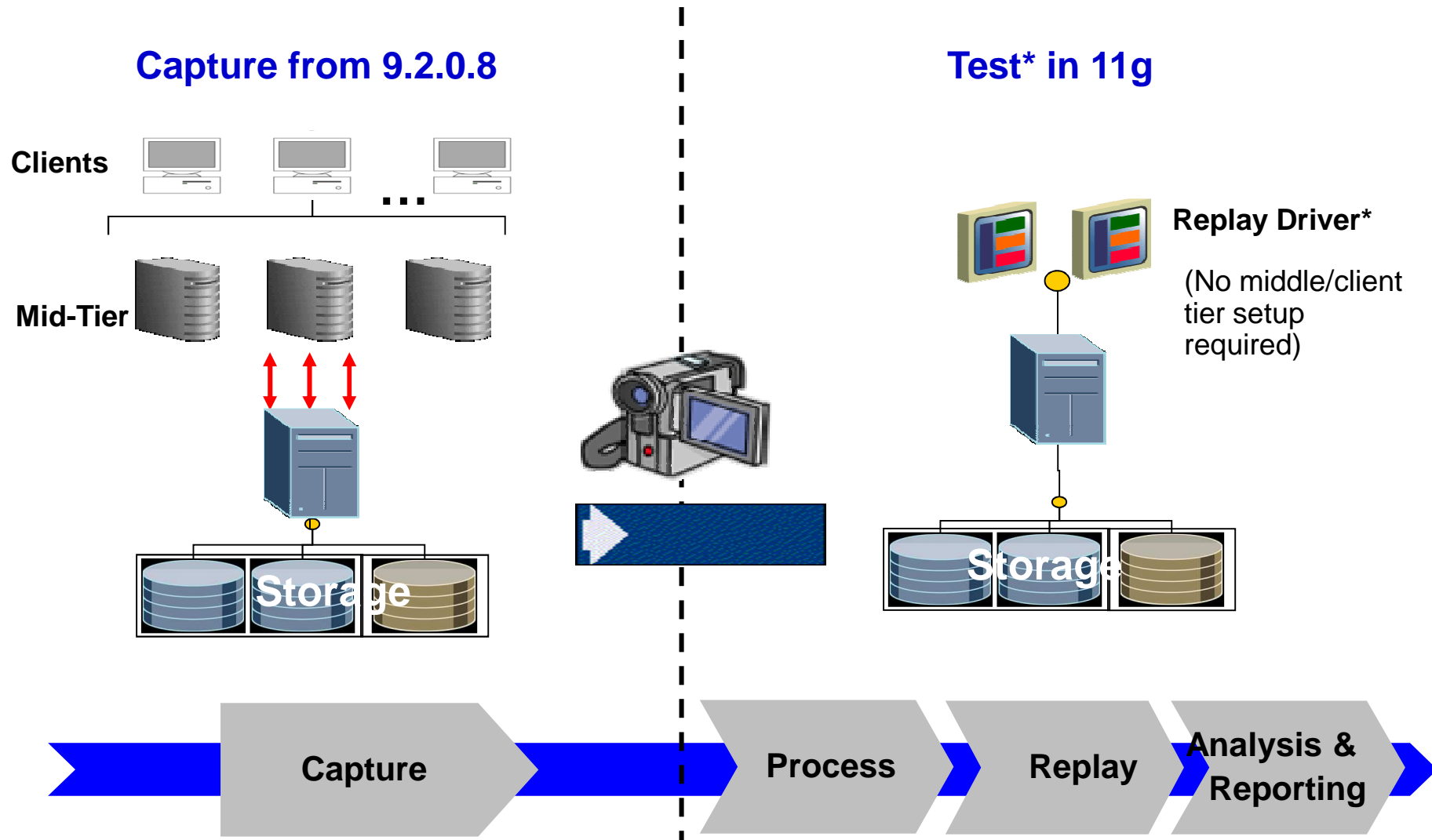


ORACLE

†Shared Server support from Oracle Database Release 11.2

Database Replay Workflow

Example: Upgrading from Oracle Database 9.2.0.8 to 11g



ORACLE

† ML 560977.1: Real Application Testing for Earlier Releases

* Use SPA for pre-11g release testing



Workload Types Supported

Supported

- All SQL (DML, DDL, PLSQL) with practically all types of binds
- Full LOB functionality (Cursor based and direct OCI)
- Local transactions
- Login/Logoffs
- Session switching
- Limited PL/SQL RPCs

Limitations

The following types of client requests are not captured in a workload:

- Direct path load of data from external files using utilities such as SQL*Loader
- Non-PL/SQL based Advanced Queuing (AQ)
- Flashback queries
- Oracle Call Interface (OCI) based object navigations
- Non SQL-based object access
- Distributed transactions (any distributed transactions that are captured will be replayed as local transactions)



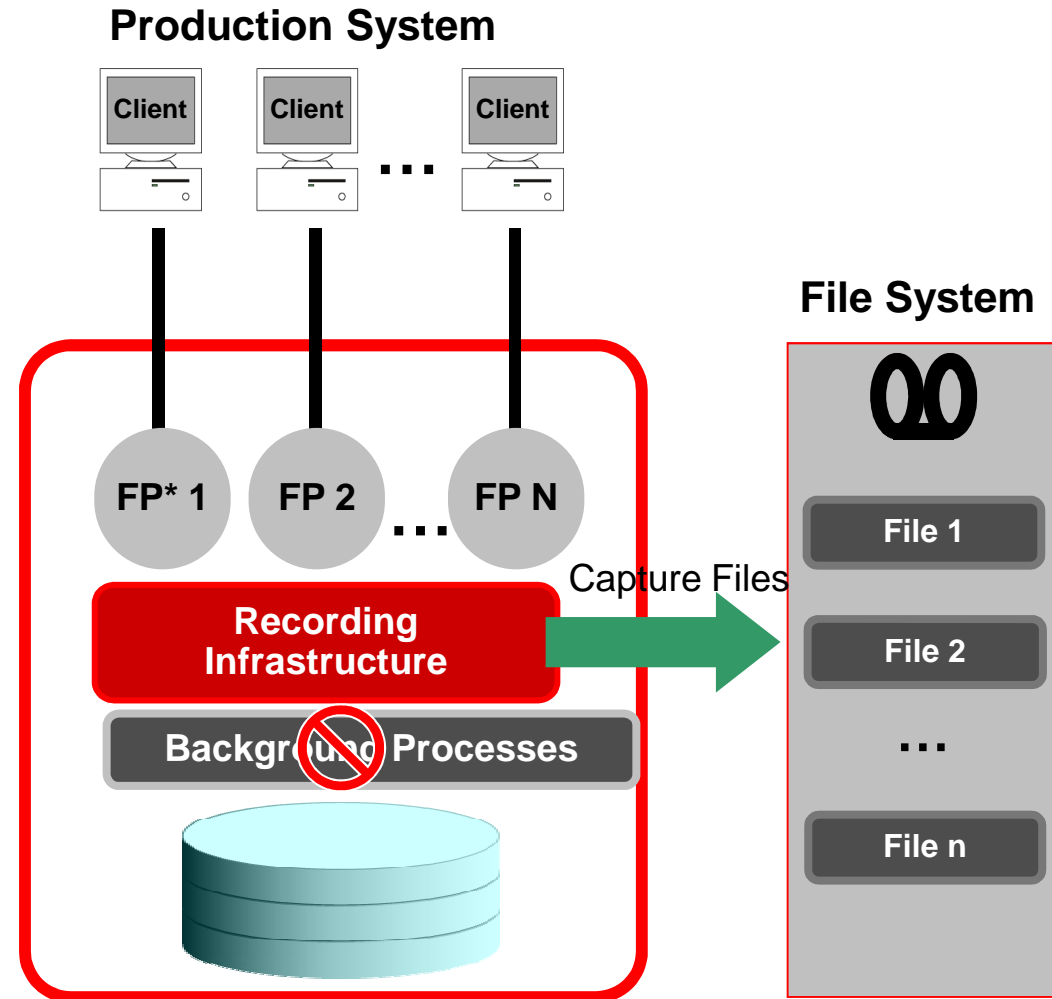
Database Replay Workflow

Database Replay usage comprises of 4 main steps

1. Capture workload on production system
2. Setup test system
3. Process workload files to prepare them for replay
4. Replay workload on test system and analyze results

Step 1: Workload Capture

- All external client requests captured in binary files
- System background, internal activity excluded
- Minimal performance overhead for capture
- For RAC, shared and local file system supported
- Specify interesting time period for capture, e.g., peak workload, month-end processing, etc.



ORACLE®

*FP: Foreground Process



Capture Options

- Workload can be filtered to customize what is captured
 - Filter Types
 - Inclusion Filters: Specifies which sessions should be captured
 - Exclusion Filters: Specifies which sessions should NOT be captured
 - Filter Attributes: Workload capture can be filtered using any of the following session attributes
 - User
 - Program
 - Module
 - Action
 - Service
 - Session ID
- Workload capture can be run on-demand or scheduled to run at later time

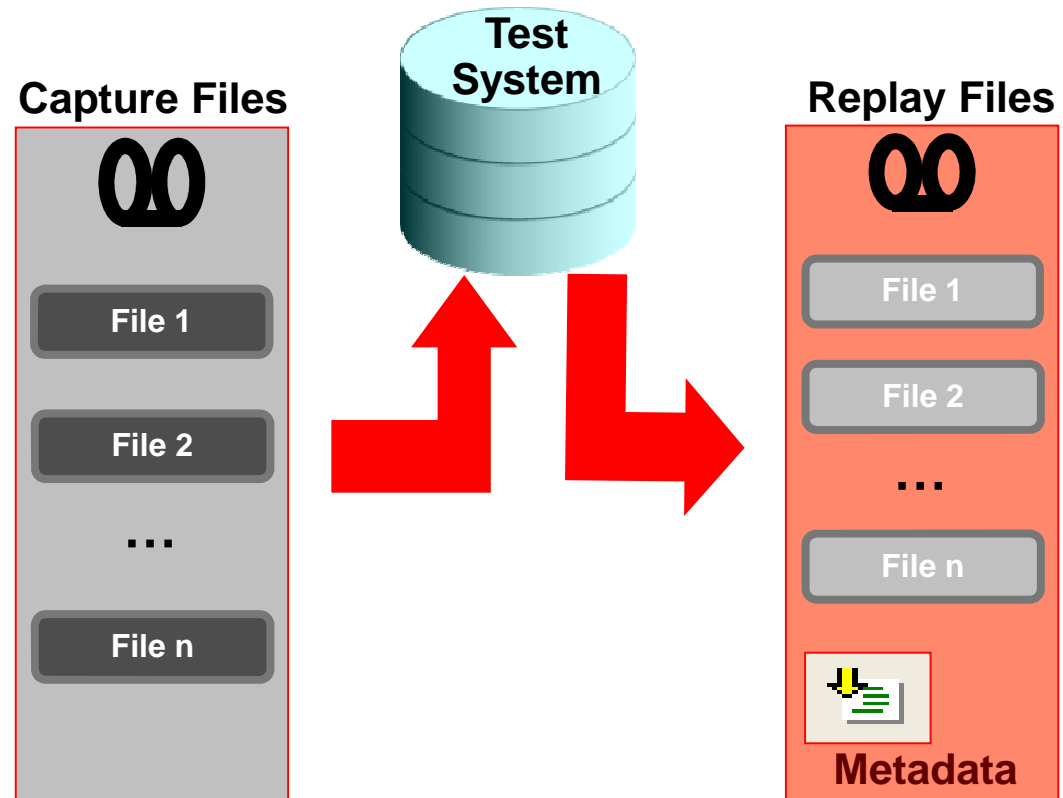


Step 2: Setup Test System

- Test system should have logically similar data as production system at the time capture is started
- Recommendation
 - Use RMAN to physically restore production database from backup (recovery scn/time provided)
 - Use Snapshot standby capabilities to create test system
 - Logically restore application data using import/export, Data Pump, etc.

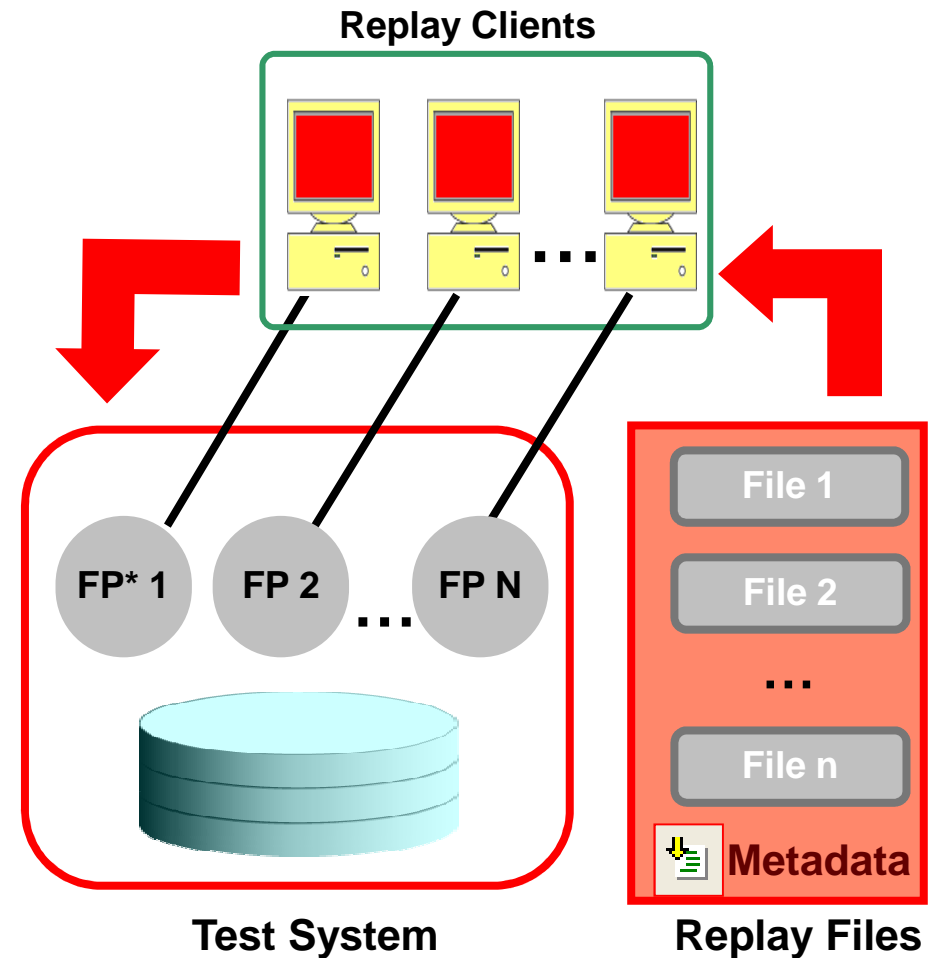
Step 3: Process Workload Files

- Transforms captured data into replay files and generates necessary meta-data for replaying workload
- Must be done on same version of database as replay system
- Recommended to process workload on test system as it can be resource intensive
- Once processed for a particular version, workload can be replayed many times
- For RAC, if using local file system, copy all capture files to single location for replay



Step 4: Replay Workload

- (wrc) is a multi-threaded client program that consumes processed workload and sends requests to the replay system
- Replay requests preserve timing, concurrency and dependencies seen on the capture system
- For workloads with high concurrency, multiple clients may be necessary. Calibration help provided.
- Automatic workload distribution among clients
- Once all replay clients are started, workload replay is then initiated by user





Replay Options

Synchronization

- Synchronized Replay
 - Workload is replayed in full synchronized mode
 - Exact same concurrency and timing as production workload
 - Transaction commit order is honored
 - Ensures minimal data divergence
- Unsynchronized Replay
 - Workload can be replayed in unsynchronized mode
 - Useful for load/stress testing
 - High Data Divergence



Replay Options

Think Time Control

- Controls think time between database calls
- Auto (Default): Adjusts think time so as to maintain captured request rate
- Percentage
 - 0% No think time, highest possible request rate
 - <100% Higher request rate
 - 100% Exact think time
 - >100% Lower request rate

Login Time Control

- Controls when sessions are created (login rate)
- Percentage
 - 0%: All session are connected immediately
 - 100% (Default): Sessions connect at same time as in captured system




Replay Options

- Number of Replay Clients
 - Configurable by user
 - Client Calibration Advisor recommends number of replay clients needed for specific workload
 - Replay clients are multithreaded clients that can drive multiple workload sessions each



Replay Analysis

- Application Level Validation: recommended method to determine if replay was successful.
- Comprehensive reports are provided for analysis purposes
- There (3) types of divergences are reported
 - **Data Divergence**: Number of rows returned by each call are compared and divergences reported
 - **Error Divergence**: For each call error divergence is reported
 - New: Error encountered during replay not seen during capture
 - Not Found: Error encountered during capture not seen during replay
 - Mutated: Different error produced in replay than during capture
 - **Performance Divergence**
 - Capture and Replay Report: Provides high-level performance information
 - ADDM Report: Provides in-depth performance analysis
 - AWR, ASH Report: Facilitates comparative or skew analysis



RAT Backport for Pre-11g Database Releases - more info

- For latest announcements, and get the one-off patch # for your release check WebIV note **560977.1**



Replay Technology

- **Functionally equivalent replay**
 - independent of clients/protocols in original setup
- **Server controlled replay**
 - scalable architecture: use an arbitrary number of replay clients
 - multiple multi-threaded replay clients to drive workload
- **Commit based synchronization**
- **Automatic Remapping of physical locators**
 - rowids
 - LOB locators
 - cursor numbers
- **Preservation of IDs during replay**
 - sequences
 - GUIDs
- **Platform Independent Replay**



Replay Data Divergence

Workload Characteristics that increase data/error divergence

- Implicit session dependencies by the Application (eg: use of dbms_pipe)
- Extensive use of multiple commits within PLSQL
- User Locks
- Use of non-repeatable functions, system dependent data.
- External interactions via urls, dblinks



Packages and Procedures

- file: `dbmswrr.sql`
- **DBMS_WORKLOAD_CAPTURE**
 - `START_CAPTURE:`
 - `FINISH_CAPTURE:`
 - `REPORT():`
 - `ADD_FILTER:`
 - `DELETE_FILTER:`
- **DBMS_WORKLOAD_REPLAY**
 - `PROCESS_CAPTURE`
 - `INITIALIZE_REPLAY`
 - `PREPARE_REPLAY()`
 - `START_REPLAY()`
 - `CANCEL()`
 - `REPORT`
 - `ADD_FILTER`
 - `REMAP_CONNECTION`



Data Dictionary Views:

- `DBA_WORKLOAD_CAPTURES` : Lists all the workload captures performed in the database
- `DBA_WORKLOAD_FILTERS` : Lists all the workload filters defined in the database
- `DBA_WORKLOAD_REPLAYS` : Lists all the workload replays that have been performed in the database
- `DBA_WORKLOAD_REPLAY_DIVERGENCE` : Used to monitor workload divergence
- `DBA_WORKLOAD_CONNECTION_MAP` : Used to review all connection strings used by workload replays



Replay Client [wrc]

- `$ORACLE_HOME/rdbms/bin/wrc`
- Part of Oracle Client and the Instant Client
- OCI based multi-threaded application
- Built in calibration advisor



DB Replay Security Model

- Any “Non-SYS” user with DBA Role
 - DBA role is not mandatory
 - Does not have to be the user whose workload is captured
- “Execute” privileges on DBMS_WORLOAD_CAPTURE/REPLAY procedures
 - DBMS_WORKLOAD_CAPTURE
 - START_CAPTURE, FINISH_CAPTURE, REPORT(), ADD_FILTER, DELETE_FILTER
 - DBMS_WORKLOAD_REPLAY
 - PROCESS_CAPTURE, INITIALIZE_REPLAY, PREPARE_REPLAY(), START_REPLAY(), CANCEL(), REPORT, ADD_FILTER, REMAP_CONNECTION
- Capture and Replay user can be different if they have appropriate privileges



Using Database Replay for Upgrade Testing



10.2 → 11g Database Upgrade

Scenario:

How can I use Database Replay to check if my peak workload will perform as expected after upgrade from 10.2 to 11g?

Goal:

Test the impact of Oracle Database 11g upgrade on the peak workload captured on production system & make sure are no negative effects of the change



Siebel Workload Description

- Siebel PSPP workload used for testing DB upgrade scenario
 - Used internally for upgrade certification and new feature uptake
- Siebel 8.0, 1300 users: 700 financial call center, 600 financial partner manager
- Financial call center scenario:
 - Creates new contact
 - Creates new opty for the contact
 - Add products to the opty
 - Creates quotes
 - Converts quotes to order
- Financial partner manager scenario
 - Creates a new service request
 - Assigns the service request



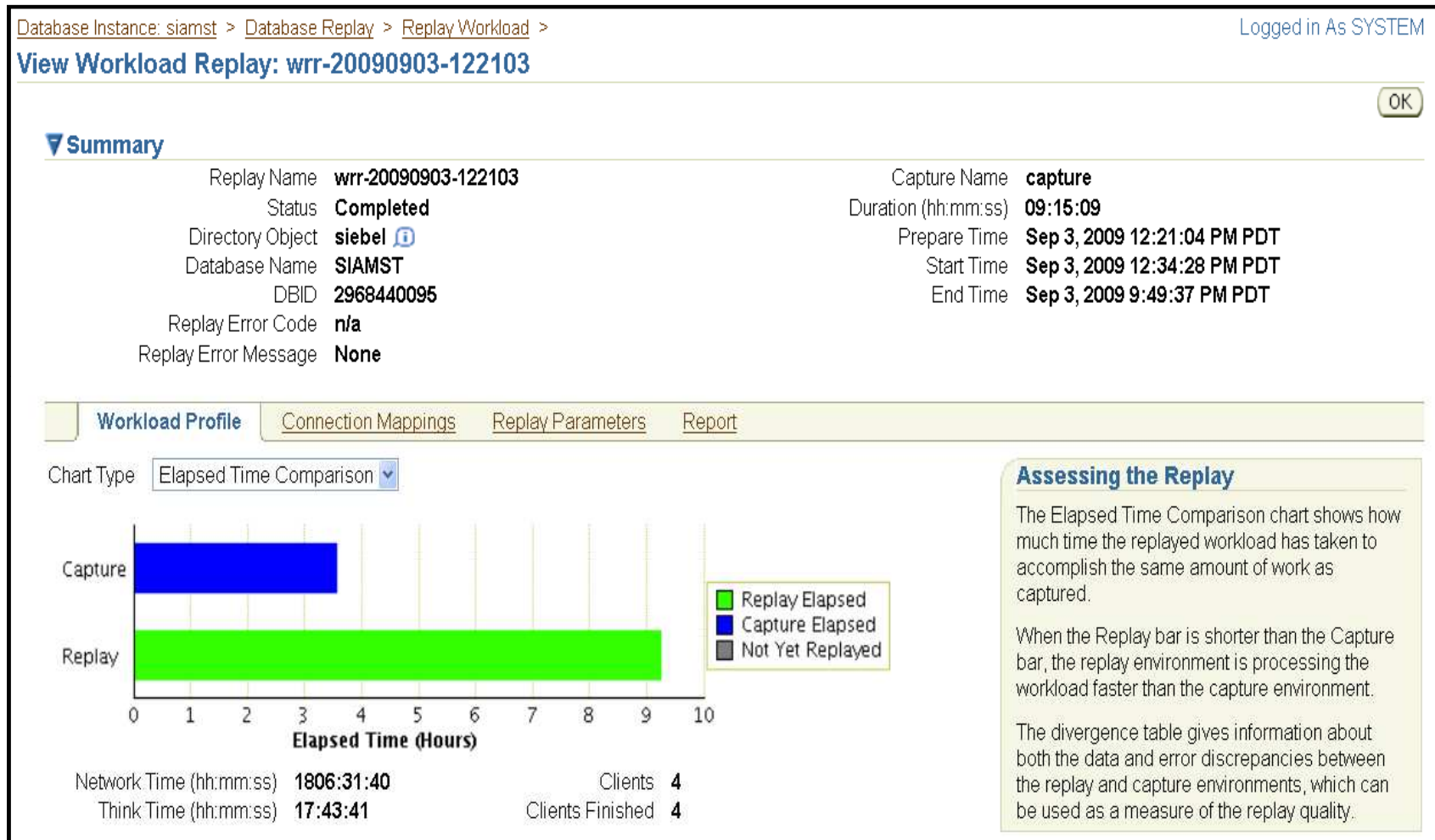
Replay Compare Period Report



Enhanced in 11.2 !

- Your new best friend in analyzing replay information!!
 - Provides holistic view of the experiment – covers functional and performance aspects of testing
 - “Replay Divergence Summary” categorization indicates if further analysis is necessary: LOW/MED/HIGH
- Two reports are available
 - Capture Vs Replay, Replay Vs Replay
- Identifies interference from other workloads, e.g., maintenance windows or other non-replayed workload
- Automatically runs ADDM
- Reports provide more accurate performance analysis
 - Uses enhanced ASH infrastructure for capture/replay sessions

Replay Summary (For Siebel Workload)



Replay Summary (Contd.) : Errors and Data Divergence

Divergence

	Number of Calls	Percentage of Total Calls
Error Divergence:		
Session Failures Seen During Replay	0	0.00
Errors No Longer Seen During Replay	24	0.00
Errors Mutated During Replay	2489	0.02
New Errors Seen During Replay	853	0.01
Data Divergence:		
DMLs with Different Number of Rows Modified	200	0.00
SELECTs with Different Number of Rows Fetched	425	0.00

▼ Detailed Comparison

	Capture	Replay	Percentage of Capture
Duration (hh:mm:ss)	03:33:47	09:15:09	259.68
Database Time (hh:mm:ss)	09:07:54	17:16:07	189.11
Average Active Sessions	2.56	1.87	72.82
User Calls	11,212,622	11,212,577	100.00

Replay Errors and Data Divergence Analysis

Database Instance: [siamst](#) > [Database Replay](#) > [Replay Workload](#) > [View Workload Replay: wrr-20090903-122103](#) > Logged in As SYSTEM

Diverged Calls During Replay: wrr-20090903-122103

View the most relevant set of replayed calls that have diverged from the capture by filtering out all but the ones of interest. A large number of calls may be relevant. If so, consider grouping them by an attribute value they have in common.

▼ Conditions for Displaying Diverged Calls

[Filter Conditions for Diverged Calls](#)

Only the calls that meet all the following filter conditions will be displayed among the results. Conditions with empty values will not contribute to filtering.

Type of Divergence: Select the blank choice to prevent filtering by type.

SQL ID:

Errors Mutated During Replay

Session Failures Seen During Replay

Errors No Longer Seen During Replay

Errors Mutated During Replay

New Errors Seen During Replay

DMLs with Different Number of Rows Modified

SELECTs with Different Number of Rows Fetched

[Fetch](#) [Diverge](#) [Below](#) [TIP](#) [Load](#) [Link is disabled when the all currently displayed.](#) [xt 15](#)

SQL ID	Timestamp ▲	Type of Divergence	Divergence Details	Rows Observed	Error Observed	Service	Module	Action	Session
	Sep 3, 2009 12:40:06 PM PDT	Error Mutated During Replay	Expected Error Number: 25228		15566	siamst	emagent_AQMetrics	DEQ	210:106
	Sep 3, 2009 12:40:06 PM PDT	Error Mutated During Replay	Expected Error Number: 25228		15566	siamst	emagent_AQMetrics	DEQ	210:106

Replay Errors and Data Divergence Grouping

Diverged Calls During Replay: wrr-20090903-122103

View the most relevant set of replayed calls that have diverged from the capture by filtering out all but the ones of interest. A large number of calls may be relevant. If so, consider grouping them by an attribute value they have in common.

▼ Conditions for Displaying Diverged Calls

Filter Conditions for Diverged Calls

Only the calls that meet all the following filter conditions will be displayed among the results. Conditions with empty values will not contribute to filtering.

Type of Divergence: Error Observed:

SQL ID:

Session (ID:Serial#): :

Service:

Module:

Action:

Grouping of Filtered Calls

Summarize the diverged calls displayed by filter using one of the following attributes:

Grouping Attribute:

Grouping on filtered attributes will be ignored.

[Fetch Diverged Calls](#)

Diverged Calls

Below is the grouped table of diverged calls that matches the specified attribute. Expand the table row to see the individual calls in the group.

TIP The SQL ID link for individual calls allows you to drill down to the call details. This link is disabled when the call details have not yet been loaded into the server. Click the link to load the call details for each call currently displayed.

Details	Session (ID:Serial#)	Completed Without Failure	Number of Calls	SQL ID	Timestamp	Type of Divergence	Divergence Details	Rows Observed	Error Observed	Service	Module	Action	Session
Show	210:106	✓	1		Sep 3, 2009 12:40:06 PM PDT	Error Mutated During Replay	Expected Error Number: 25228		15566	siamst	emagent_AQMetrics	DEQ	210:106
Show	12:380	✓	1		Sep 3, 2009 12:40:06 PM PDT	Error Mutated During Replay	Expected Error Number: 25228		15566	siamst	emagent_AQMetrics	DEQ	210:106
					Sep 3, 2009 12:40:06 PM PDT	Error Mutated During Replay	Expected Error Number: 25228		15566	siamst	emagent_AQMetrics	DEQ	210:106
					Sep 3, 2009 12:40:06 PM PDT	Error Mutated During Replay	Expected Error Number: 25228		15566	siamst	emagent_AQMetrics	DEQ	210:106

Replay Analysis: Reports

Database Instance: [siamst](#) > [Database Replay](#) > [Replay Workload](#) >
View Workload Replay: wrr-20090909-165137

[▶ Summary](#)

[Workload Profile](#) [Connection Mappings](#) [Replay Parameters](#) [Report](#)

Workload Replay Report

[Run Report](#)

Compare Period Report

First Workload Capture or Replay [mycapture \(Sep 16, 2009 10:20:52 AM\)](#)

Second Workload Capture or Replay [REPLAY-orclJ.us.oracle.com-20090929142400 \(Sep 29, 2009 2:25:16 PM\)](#)

[Run Replay Compare Period Report](#) [Run AWR Compare Period Report](#)


AWR Report


Workload Capture or Replay [REPLAY-orclJ.us.oracle.com-20090929142400 \(Sep 29, 2009 2:25:16 PM\)](#)

[Run Report](#)

ASH Report

Workload Capture or Replay [REPLAY-orclJ.us.oracle.com-20090929142400 \(Sep 29, 2009 2:25:16 PM\)](#)

Start Date 
(example: Oct 9, 2009)

End Date 
(example: Oct 9, 2009)

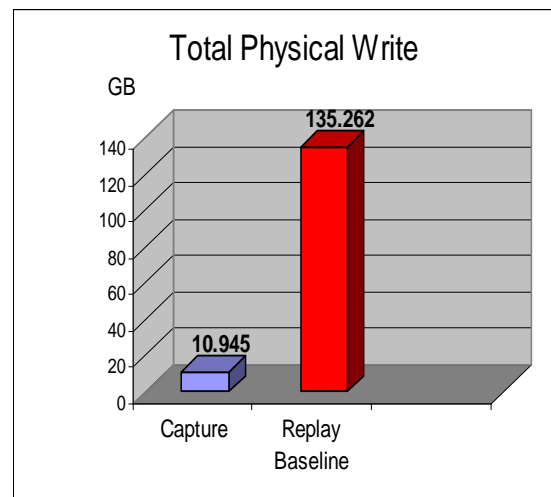
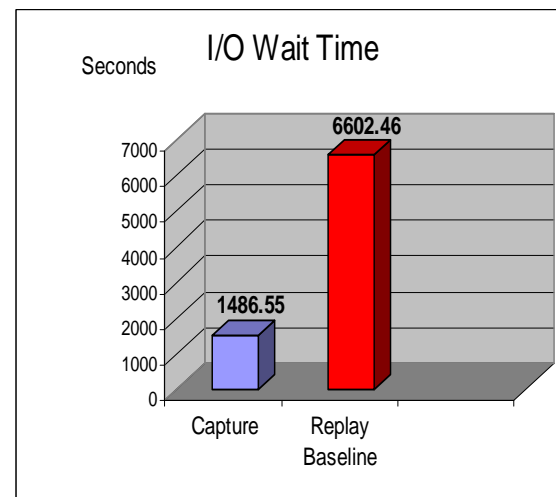
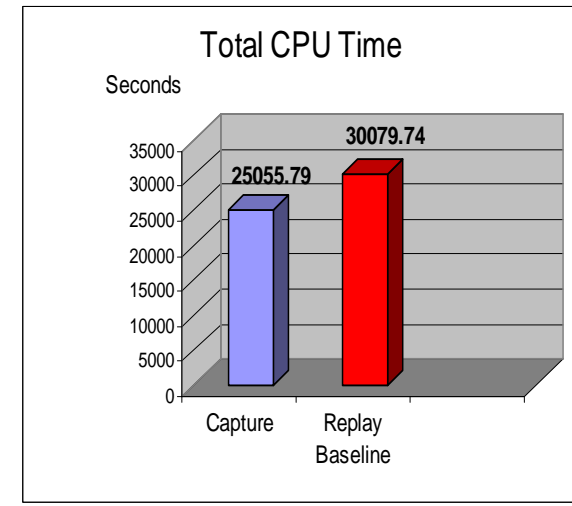
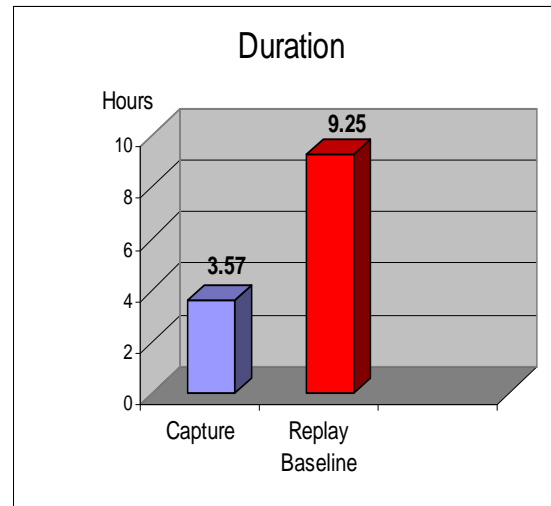
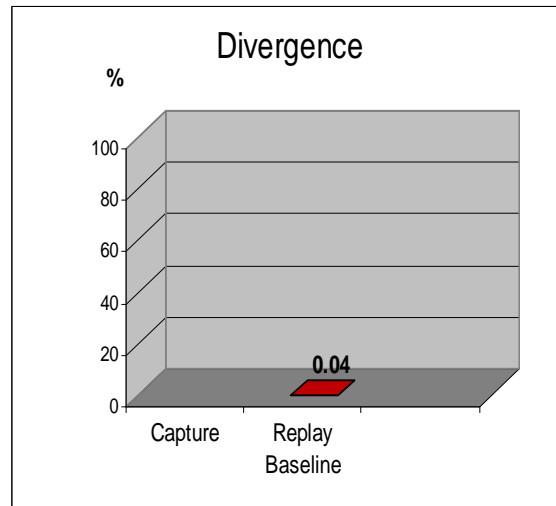
Start Time ☐ AM ☒ PM

End Time ☐ AM ☒ PM

Filter [SID](#)

[Run Report](#)

Replay Compare Period: 10.2.0.4 Capture Vs 11.2.0.1 Replay Baseline



[Compare Period Report Link](#)

Important Changes between Capture and Replay Baseline

(-) Changes to Important Parameters

	Capture	Replay
sessions	566	792
compatible	10.2.0.4	11.2.0

(-) Changes to Optimizer-Relevant Parameters

	Capture	Replay
optimizer_capture_sql_plan_baselines	NULL	FALSE
optimizer_use_invisible_indexes	NULL	FALSE
optimizer_use_pending_statistics	NULL	FALSE
optimizer_use_sql_plan_baselines	NULL	TRUE
optimizer_features_enable	10.2.0.4	11.2.0.1
result_cache_mode	NULL	MANUAL

No Changes to Memory Configuration Parameters

(-) Instances of the Capture Database

instance number	instance name	host name	number of CPU cores	number of CPU sockets	physical memory	instance type
1	siamst	stase07	8	8	31.73 G	RDBMS

(-) Instances of the Replay Database

instance number	instance name	host name	number of CPU cores	number of CPU sockets	physical memory	instance type
1	siamst	stase07	4	4	31.73 G	RDBMS

Replay Period Report: Top SQL Changes

5.9 hr

Force Matching Signature	example SQL_ID	Change in DB Time	Change in Average Response Time	Capture DB time	Replay DB time	example sql text
9011963922445766669	44u0kwtwmd7h5	21209.57 seconds	7175.79%	295.57 seconds	21505.14 seconds	(+) SELECT T9.CONFLICT_ID, T9.LAST_UPD, T9.CRE [...]
8567836812353523192	628g90748qrd5	-5947.74 seconds	-99.3%	5989.66 seconds	41.92 seconds	(+) SELECT T34.CONFLICT_ID, T34.LAST_UPD, T34. [...]
9389583114292788166	dry9f79q87ngv	-3150.3 seconds	-99.01%	3181.74 seconds	31.44 seconds	(+) SELECT T34.CONFLICT_ID, T34.LAST_UPD, T34. [...]
1542715800513830963	6fb88urfgyjsja	-2790.54 seconds	-100%	2790.54 seconds	0 seconds	(+) SELECT T34.CONFLICT_ID, T34.LAST_UPD, T34. [...]
11530247703233490999	8q1jjwp4u7pn6	-2547.13 seconds	-100%	2547.13 seconds	0 seconds	(+) SELECT /*+ BEGIN_OUTLINE_DATA IGNORE_OPTIM_EMBED [...]
17598323506657830469	5kcd99qf8y6yh	2386.13 seconds	6861.99%	34.77 seconds	2420.9 seconds	(+) SELECT T9.CONFLICT_ID, T9.LAST_UPD, T9.CRE [...]

0.6 hr

The Culprit!!

```
SELECT T9.CONFLICT_ID, T9.LAST_UPD, T9.CREATED, T9.LAST_UPD_BY, T9.CREATED_BY, T9.MODIFICATION_NUM,
T9.ROW_ID, T4.KEY_VALUE, T25.PR_DEPT_OU_ID, T1.INTEGRATION_ID, T1.PRTNR_FLG, T25.PR_REGION_ID,
T25.CITIZENSHIP_CD, T6.ATTRIB_07, T25.EMAIL_ADDR, T25.EMP_FLG, T25.FAX_PH_NUM, T25.FST_NAME,
T25.CITIZENSHIP_CD, T25.HOME_PH_NUM, T25.AGENT_FLG, T25.JOB_TITLE, T25.LAST_NAME, T25.SEX_MF,
T25.MEMBER_FLG, T25.MID_NAME, T25.OWNER_PER_ID, T9.NAME, T25.PERSON_UID, T25.PRIV_FLG, T20.STATUS,
T8.PR_EMP_ID, T3.NAME, T25.CURR_PRI_LST_ID, T25.PR_OU_ADDR_ID, T1.NAME, T20.PR_ADDR_ID,
T25.PR_EMAIL_ADDR_ID, T25.PR_ALT_PH_NUM_ID, T25.PR_REP_DNRM_FLG, T25.PR_REP_MANL_FLG,
T25.PR_REP_SYS_FLG, T25.PR_MKT_SEG_ID, T22.PR_EMP_ID, T8.PR_EMP_ID, T13.LOGIN, T18.LOGIN, T17.PR_FAX_NUM_ID,
T25.PR_GRP_OU_ID, T25.PR_INDUST_ID, T25.PR_NOTE_ID, T25.PR_OPTY_ID, T25.BU_ID, T25.PR_SYNC_USER_ID,
T25.PR_PER_ADDR_ID, T25.PR_PER_PAY_PRFL_ID, T25.PR_POSTN_ID, T25.PR_PROD_LN_ID, T25.PR_RESP_ID,
T17.PR_SMS_NUM_ID, T25.PR_SECURITY_ID, T5.NAME, T25.MED_SPEC_ID, T25.PR_STATE_LIC_ID, T25.PR_TERR_ID,
T25.PROVIDER_FLG, T12.OWN_INST_ID, T12.INTEGRATION_ID, T11.SHARE_HOME_PH_FLG, T25.CUST_SINCE_DT
```

Next Steps:

1. Tune Regressed SQL stmts using SQL Plan Baselines

2. Replay Workload

```
T16.PAR_ROW_ID, T12.ROW_ID, T12.PAR_ROW_ID, T12.MODIFICATION_NUM, T12.CREATED_BY, T12.LAST_UPD_BY,
T12.CREATED, T12.LAST_UPD, T12.CONFLICT_ID, T12.PAR_ROW_ID, T19.ROW_ID, T16.ROW_ID, T2.ROW_ID, T10.ROW_ID
FROM ORAPERF.S_ORG_EXT T1, ORAPERF.S_PARTY T2, ORAPERF.S_PRI_LST T3, ORAPERF.S_DQ_CON_KEY T4,
ORAPERF.S_MED_SPEC T5, ORAPERF.S_CONTACT_X T6, ORAPERF.S_POSTN T7, ORAPERF.S_POSTN T8, ORAPERF.S_PARTY
T9, ORAPERF.S_PARTY T10, ORAPERF.S_EMP_PER T11, ORAPERF.S_CONTACT_SS T12, ORAPERF.S_USER T13,
ORAPERF.S_USER T14, ORAPERF.S_USER T15, ORAPERF.S_PARTY T16, ORAPERF.S_CONTACT_LOYX T17, ORAPERF.S_USER
T18, ORAPERF.S_POSTN_CON T19, ORAPERF.S_POSTN_CON T20, ORAPERF.S_ORG_EXT T21, ORAPERF.S_POSTN T22,
ORAPERF.S_ADDR_PER T23, ORAPERF.S_ORG_EXT_FNX T24, ORAPERF.S_CONTACT T25 WHERE T25.PR_DEPT_OU_ID =
T1.PAR_ROW_ID (+) AND T1.PR_POSTN_ID = T22.PAR_ROW_ID (+) AND T25.CURR_PRI_LST_ID = T3.ROW_ID (+) AND
T25.PR_POSTN_ID = T8.PAR_ROW_ID (+) AND T9.ROW_ID = T20.CON_ID (+) AND T20.POSTN_ID (+) = :1 AND T22.PR_EMP_ID
= T13.PAR_ROW_ID (+) AND T8.PR_EMP_ID = T18.PAR_ROW_ID (+) AND T25.PR_PER_ADDR_ID = T23.ROW_ID (+) AND
T25.MED_SPEC_ID = T5.ROW_ID (+) AND T9.ROW_ID = T4.CONTACT_ID (+) AND T9.ROW_ID = T25.PAR_ROW_ID AND
T9.ROW_ID = T17.PAR_ROW_ID (+) AND T9.ROW_ID = T11.PAR_ROW_ID (+) AND T9.ROW_ID = T6.PAR_ROW_ID (+) AND
T9.ROW_ID = T12.PAR_ROW_ID (+) AND T19.POSTN_ID = :2 AND T25.ROW_ID = T19.CON_ID AND T16.ROW_ID =
T19.POSTN_ID AND T19.POSTN_ID = T7.PAR_ROW_ID (+) AND T7.PR_EMP_ID = T15.PAR_ROW_ID (+) AND
T25.PR_DEPT_OU_ID = T2.ROW_ID (+) AND T25.PR_DEPT_OU_ID = T21.PAR_ROW_ID (+) AND T25.PR_DEPT_OU_ID =
T24.PAR_ROW_ID (+) AND T25.PR_SYNC_USER_ID = T10.ROW_ID (+) AND T25.PR_SYNC_USER_ID = T14.PAR_ROW_ID (+)
AND ((T25.PRIV_FLG = 'N') AND (T19.CON_LAST_NAME >= :3)) AND (T9.ROW_ID IN ( SELECT SQ1_T1.PERSON_ID FROM
ORAPERF.S_PARTY_PER SQ1_T1, ORAPERF.S_PARTY SQ1_T2, ORAPERF.S_ORG_EXT SQ1_T3 WHERE ( SQ1_T2.ROW_ID =
SQ1_T3.PAR_ROW_ID AND SQ1_T1.PARTY_ID = SQ1_T2.ROW_ID) AND ((SQ1_T3.INT_ORG_FLG != 'Y' OR SQ1_T3.PRTNR_FLG
!= 'N') AND SQ1_T3.ACCNT_FLG != 'N') AND (SQ1_T3.NAME LIKE :4))) ORDER BY T19.POSTN_ID, T19.CON_LAST_NAME,
T19.CON_FST_NAME
```

ORACLE

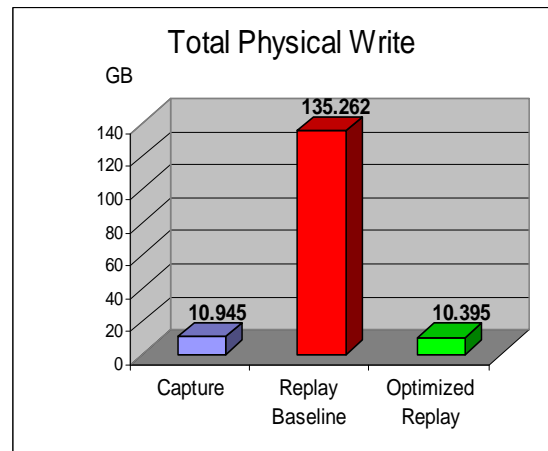
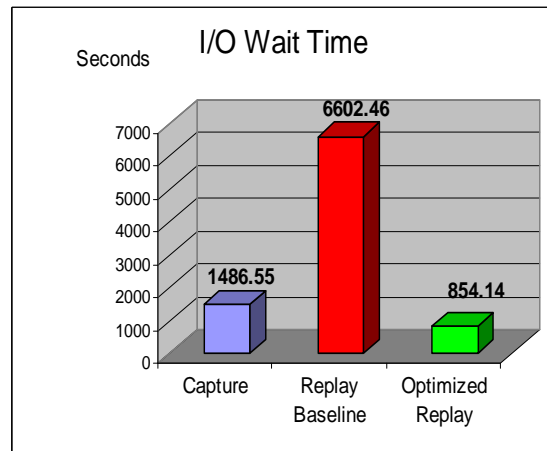
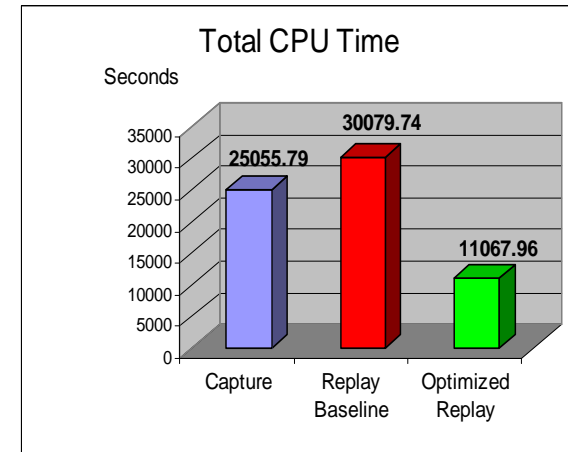
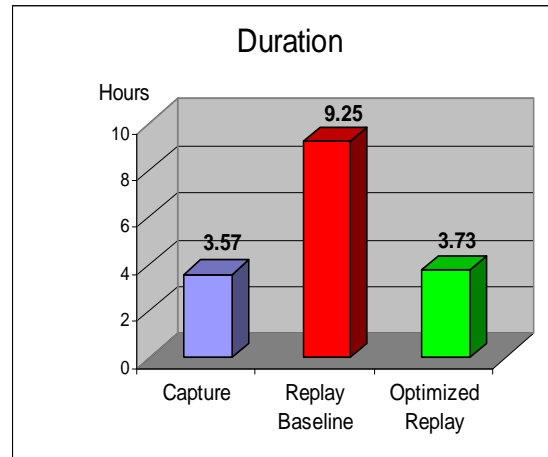
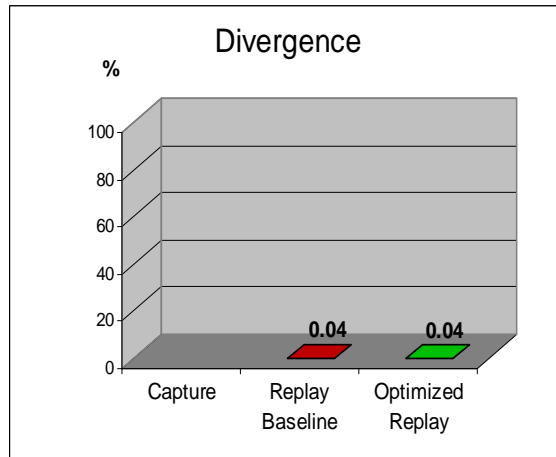
Replay Period Report: Validate Tuning

**Savings:
5.9 hr**

**Savings:
0.6 hr**

Force Matching Signature	example SQL_ID	Change in DB Time	Change in Average Response Time	1st Replay DB time	2nd Replay DB time	example sql text
9011963922445766669	44u0kwtwmd7h5	-21318.98 seconds	-99.13%	21505.14 seconds	186.16 seconds	(+) SELECT T9.CONFLICT_ID, T9.LAST_UPD, T9.CRE [...]
17598323506657830469	5kcd99qf8y6yh	-2366.15 seconds	-97.74%	2420.9 seconds	54.75 seconds	(+) SELECT T9.CONFLICT_ID, T9.LAST_UPD, T9.CRE [...]

Replay Compare Period: Replay Baseline Vs Optimized Replay



[Compare Period Report Link](#)



10.2 → 11g Upgrade: Summary



- Very low divergence rate, limited to background monitoring activity (EM)



Database Replay - Workload Capture: Best Practices

- Capture disk space estimation - maximum of
 - Extrapolated size from smaller duration capture (10-30min)
 - $2 * \text{Bytes received via SQL*Net from client (from AWR)}$
- Not mandatory to restart database
 - If many in-flight transactions or a busy system, consider longer duration capture or restart database
- Filter background activity, e.g., monitoring infrastructure – EM, OMS, Statspack, etc.
- Export AWR or create an AWR baseline after capture to avoid purging of AWR data



Database Replay – Test System Setup: Best Practices

- Apply recommended patches per Note 560877.1
- Disable maintenance windows and scheduled jobs (e.g., cron, dbms_job) on test system
- Validate that test system is setup properly – no missing synonyms, users, views, row counts etc
 - Use Enterprise Manager Change and Configuration Management Packs
- Resolve and correct external dependencies, e.g., db links, external files
- Use at least the number of replay clients as recommended by wrcl “calibrate” command.
- Replay clients should be isolated from database tier to avoid contention.



Database Replay – Test System Setup: Best Practices

- **Run Workload Analyzer tool on captured workload and follow recommendations to improve replay quality, for e.g., reset system clock if workload is sensitive to time. Available in 11.2.0.2.0**
- **Try both synchronized and unsynchronized replays.
Use `SYNCHRONIZATION=false` for workloads which have dependencies on `dbms_lock.sleep`, concurrent manager in E-BIZ, heavy PLSQL workload, Multiple commits within PLSQL scripts.**



Conclusion

- Real Application Testing provides comprehensive and easy to use solution for assessing impact of changes on workload
 - Higher quality testing
 - Full coverage of workflows
- Helps adopt technology faster by cutting down testing time from months for days
- Can be used to test many changes
 - Upgrade testing including 9.2, 10.2 to 11g releases, optimizer stats refresh
 - Enabling new features, RAC, schema, parameter, OS/HW changes
- As a result with Real Application Testing businesses can
 - Stay competitive
 - Improve profitability
 - Be compliant



Resources

- **OTN**
 - Real Application Testing for Earlier Releases
 - Testing Performance Impact of an Oracle Database 9i/10g Release 1 to Oracle Database 10g Release 2 Upgrade with SQL Performance Analyzer
 - Migration to Cost-Based Optimizer
 - Upgrading from Oracle 9i to 10g: What to expect from the Optimizer
 - Upgrade Companion: MySupport Note: 466181.1: One-stop shop for Upgrades
 - Real Application Testing Users Guide (11.2)



SPARKS

Q&A

ORACLE®



SPARKS

- Visit us at <http://my.oracle.com/SPARKS> for additional details and archived presentations