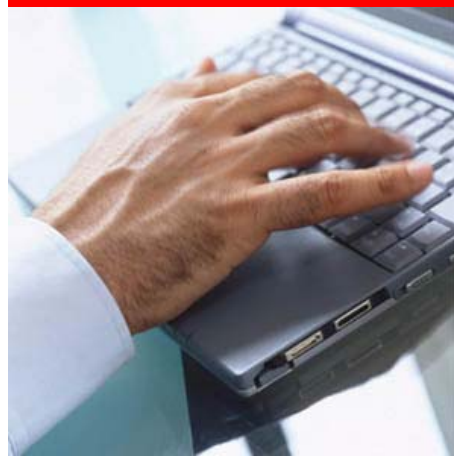
ORACLE®

Upgrading to 11g – Best Practices

Ashish Agrawal
Senior Principal Technical Support Engineer Oracle Center of Excellence

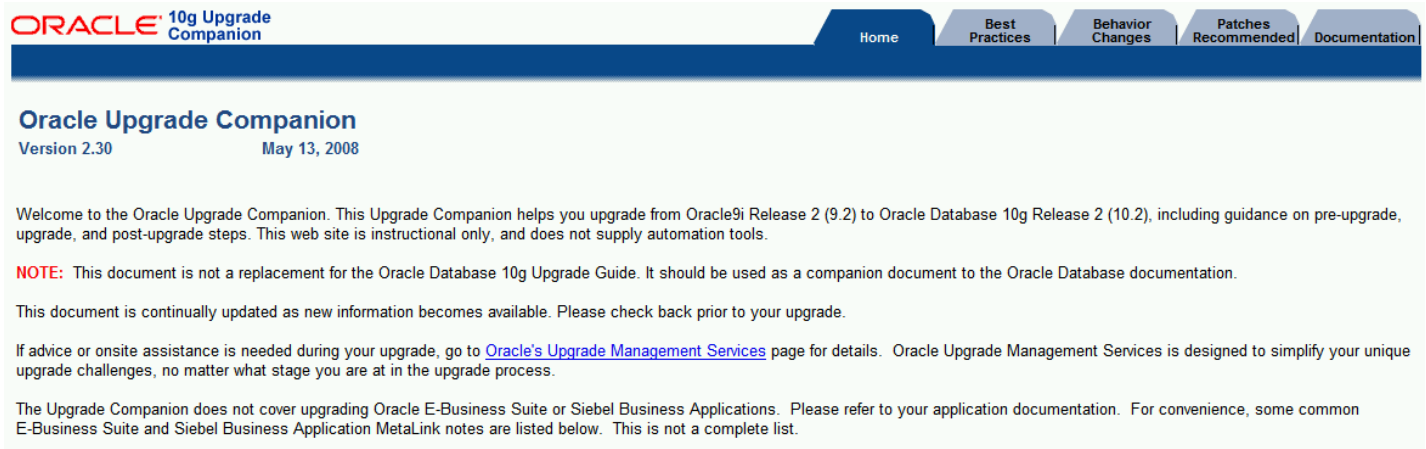
Agenda



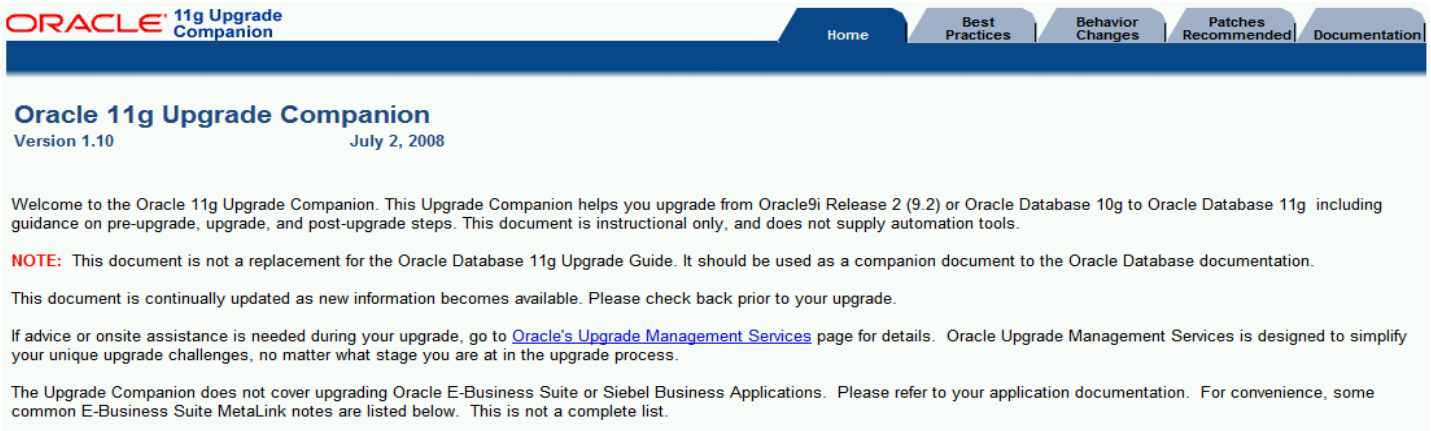
- Upgrade Companion
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
- Q&A

Documentation

- Note: 466181.1 Upgrade Companion 10gR2



- Note: 601807.1 Upgrade Companion 11g

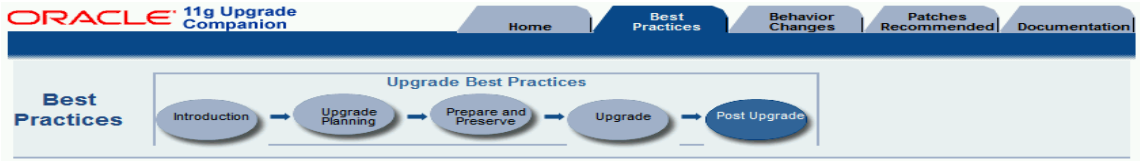


Documentation

- Note: 601807.1 Upgrade Companion 11g

Documentation

- Note: 601807.1



Database Performance

Perform Post-Upgrade Cost-Based Optimizer (CBO) Management Tasks

It is **critical** to properly manage CBO statistics after an upgrade to Oracle Database 11g. There are changes in Oracle Database 11g with regard to the CBO and having proper statistics gathered is essential to getting good performance.

See Managing CBO Stats during an upgrade 10g or 11g after upgrading from Oracle 9i or Oracle Database 10g to Oracle Database 11g to determine what should be done with statistics.

After you have addressed the CBO statistics, you are ready to begin validating the performance of the test system, or in the case of production, to begin monitoring production to catch any regressions that might have occurred.

Checking Database Performance

It is very important to check the performance of the database after upgrading the test and production databases. In TEST, this is accomplished by repeating the unit tests and load tests that were defined and executed before the upgrade (and discussed in the Best Practices > Preserve section). The unit tests should be done first so that any regressions can be addressed before going further.

After the unit tests are successful, the load tests should be performed and compared to the results of the load tests performed before the upgrade. Any regressions should be addressed before going further. **The production database should not be upgraded until performance regressions found in TEST are understood and resolved.**

See the following sections for additional details.

1. ***Checking the Performance of the TEST System with Unit Testing***
2. ***Checking the Performance of the TEST System with Load Testing***
3. ***Checking the Performance of the PRODUCTION System After an Upgrade***

Resolving Performance Regressions

The suggestions in this section may be used when resolving performance problems after upgrading test or production.

1. ***Overall Approach***
2. ***If Regression is Due to a SQL Execution Plan Change...***
3. ***If Regression is Due to Something Else***

Case Studies
- Resolving High CPU Usage on Oracle Servers

Documentation
- Oracle Database 11g Upgrade Guide > Ch. 4: After Upgrading
- Oracle Performance Tuning Guide, SQL Tuning Overview
- SQL Tuning Advisor

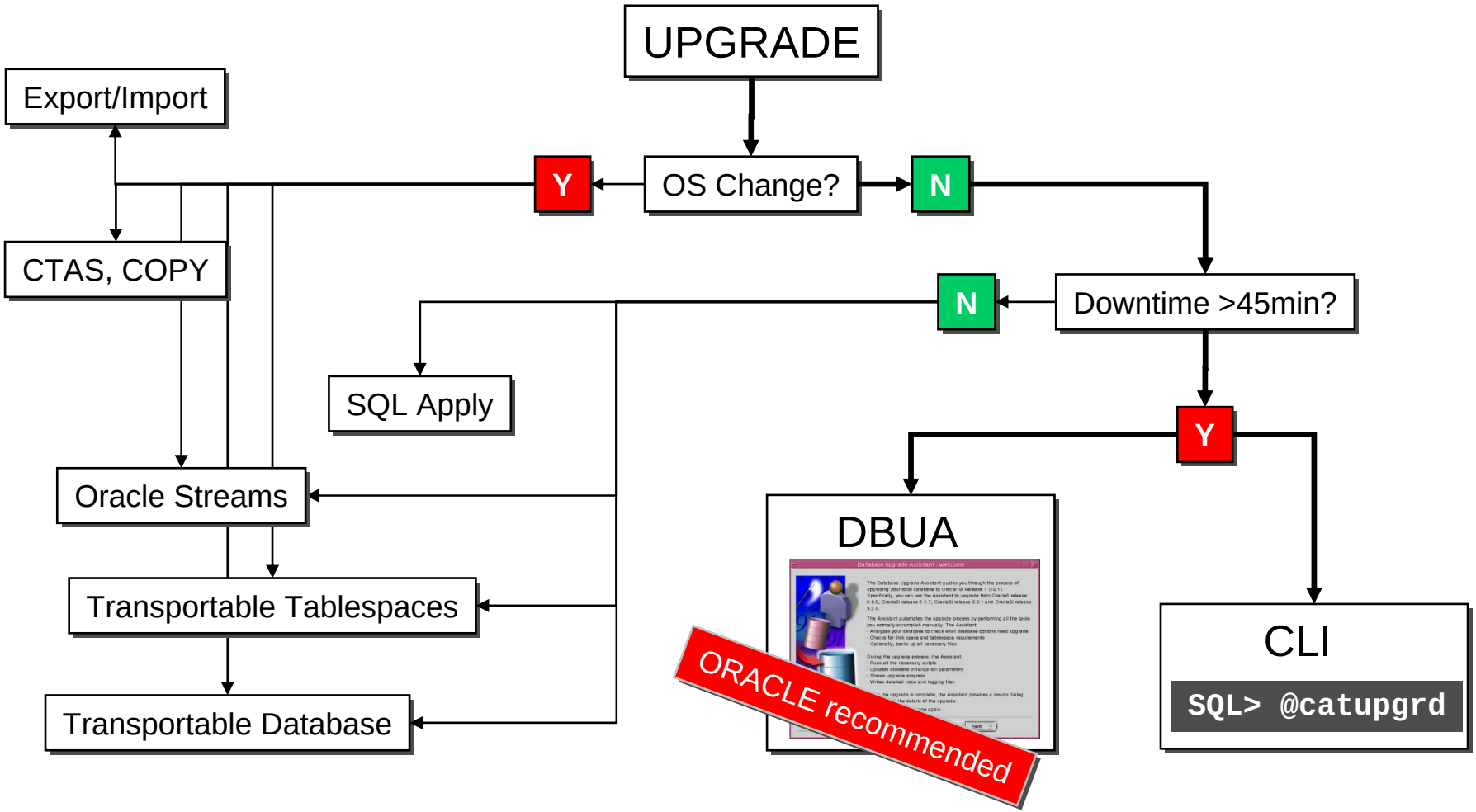
How-To
- Recommended Method for Obtaining 10046 trace for Tuning
- How to Identify Resource Intensive SQL for Tuning
- Managing CBO Stats during an upgrade to 10g or 11g
- Recording Explain Plans on 9i before an upgrade to 10g or 11g
- Oracle Performance Diagnostic Guide, Query Tuning

Notes
- Knowledge Browser, see Upgrade and Migrations
- Knowledge Browser, see Query Tuning
- Known RMAN Performance Problems

Scripts and Tools
- OS Watcher
- OSWg
- LTOM

White Papers
- Best Practices for Load Testing System Upgrades
- Determining CPU Resource Usage for Linux and Unix
- Measuring Memory Resource Usage for Linux and Unix

Upgrade Paths



Challenges

- >90% of highly visible problems attributed to an upgrade do not occur while upgrading but appear as unanticipated performance degradations days or weeks after the upgrade operation completed
  - Usually too late to schedule another extended outage to downgrade
  - Other changes have been introduced elsewhere to make the downgrade even more complex
- Typical root causes:
  - Optimizer regression - execution plan chosen is worse
    - Queries are slow
    - Reports or batches take longer to complete
  - Increased overall resource requirements
    - Changed behavior / new bottlenecks under high load

Best Practices – Preserve

- Preserve as much information as possible BEFORE you upgrade the production environment to the new release - that's of vital importance!!!
  - Capture accurate performance statistics to act as baseline
    - Allows comparison: before ↔ after
    - Measure query and batch timings
    - System level performance stats (OS Watcher / statspack / AWR)
    - Save execution plans
  - Save old configuration information (RDA / OCM)

Best Practices – Preserve

- Keep a backup copy of your current optimizer statistics. See Note 465787.1
- Incorrect / inaccurate optimizer statistics is one cause of suboptimal execution plans.
- Save execution plans

Best Practices – Pre Upgrade

- Upgrade and stress test a test database first
  - Do functional testing
  - Always use real production data
    - Create a copy of the production database with RMAN
    - Or leverage a standby DB if multiple exist
  - Run performance tests with real life loads!!!
- Rehearse both upgrade and back out procedures
  - Consider other dependent components that have to be shut down too
  - Should be able to do steps in your sleep
    - Most production systems upgrade during off-hours
  - Use a buddy system to make sure checklists are followed

Pre Upgrade

- Create dictionary statistics prior to the upgrade - otherwise it will take significantly longer

```
SQL> EXECUTE dbms_stats.gather_schema_stats
    ('SYS',
     options          => 'GATHER',
     estimate_percent =>  DBMS_STATS.AUTO_SAMPLE_SIZE,
     method_opt       => 'FOR ALL COLUMNS SIZE AUTO',
     cascade          =>  TRUE);
```

- Detailed scripts can be found here:
  Oracle® Database Upgrade Guide Appendix B
- Or in 10g / 11g :

```
SQL> EXECUTE dbms_stats.gather_dictionary_stats;
```

Post Upgrade

- Create system statistics during a regular workload period - otherwise non-appropriate values for the CBO will be used:

```
SQL> EXECUTE dbms_stats.gather_system_stats('start');
...
SQL> EXECUTE dbms_stats.gather_system_stats('stop');
```

```
SQL> select pname NAME, pval1 WERT, pval2 INFO
     from aux_stats$;

NAME                      WERT INFO
------------------ ---------- ------------------------------
STATUS                         COMPLETED
DSTART                         04-07-2008 12:30
DSTOP                          05-07-2008 12:30
FLAGS                       1
CPUSPEEDNW           1392,39
IOSEEKTIM              11,405
IOTFRSPEED          25595,605
...
```

Post Upgrade

- Create fixed table statistics soon after the upgrade:

```
SQL> execute
       dbms_stats.gather_fixed_objects_stats;
```

Validate statistics gathering strategy

Default for DBMS_STATS is

```
DBMS_STATS.GATHER_SCHEMA_STATS(OWNNAME=>'NAME',-
  ESTIMATE_PERCENT=>DBMS_STATS.AUTO_SAMPLE_SIZE,-
  METHOD_OPT=>'FOR ALL COLUMNS SIZE  AUTO',-
  GRANULARITY =>'AUTO',-
  CASCADE=>DBMS_STATS.AUTO_CASCADE,-
  NO_INVALIDATE=>DBMS_STATS.AUTO_INVALIDATE);
```

ESTIMATE_PERCENT is set to AUTO_SAMPLE_SIZE.

In 11g You are encouraged to use AUTO_SAMPLE_SIZE for ESTIMATE_PERCENT .

In 11g AUTO_SAMPLE_SIZE is very fast compared to earlier versions. and gives accuracy of close to 100 % sample size.

AUTO_SAMPLE_SIZE uses a new Hash-based Sampling for Column Statistics.

Validate statistics gathering strategy

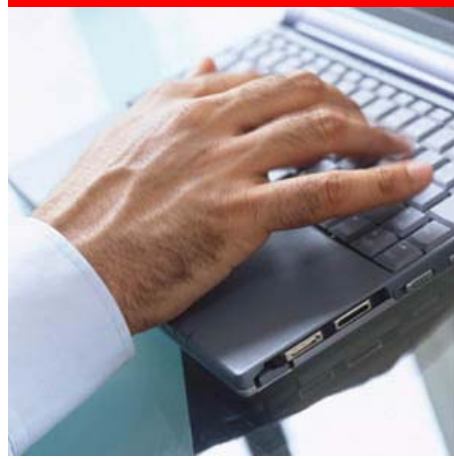
- New Features in dbms_stats in 11g

- Multicolumn Statistics
- Expression Statistics
- Statistic Preferences

Best Practices

- Possibilities for performance analysis and prediction:
  - STATSPACK
  - AWR
  - SQL Plan Management
  - Real Application Testing
    - SQL Performance Analyzer
    - Database Replay

Agenda



- Recap
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
- Q&A

Automatic Workload Repository

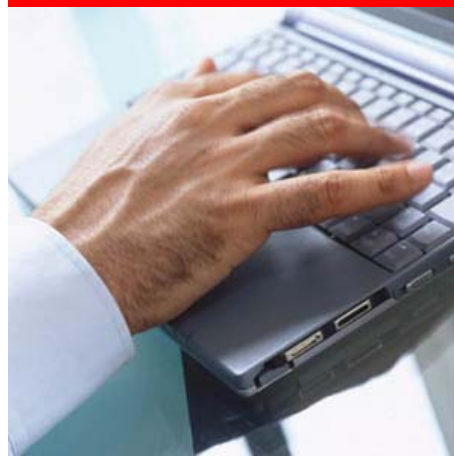
- Since Oracle 10*g*: AWR (Automatic Workload Repository)
  - Statistics repository in the SYSAUX tablespace
  - Contains AWR snapshots (current and past periods)
  - Requires a license for Diagnostic Pack
  - Command line report:
    - `@?/rdbms/admin/awrrpt.sql`
  - Export the AWR:

```
begin
 DBMS_SWRF_INTERNAL.AWR_EXTRACT(
      dmpfile => 'awr_data.dmp',
      dmpdir  => 'TMP_DIR',
      bid     => 302,
      eid     => 305);
end;
/
```

STATSPACK

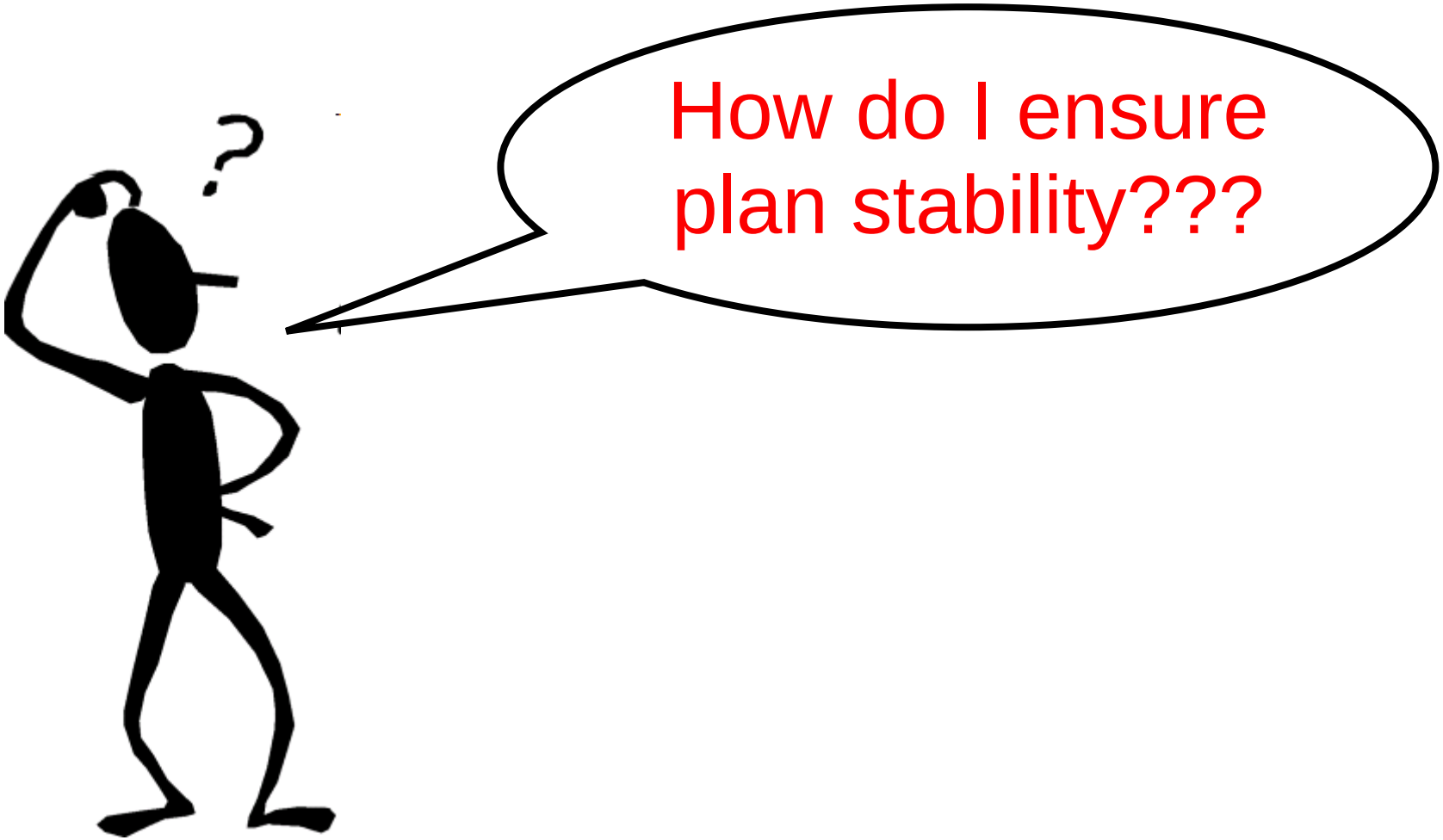
- Database performance tool - pre-10*g*
  - Note:394937.1 STATSPACK guide
  - Export the PERFSTAT user

Agenda



- Recap
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
- Q&A

Challenge

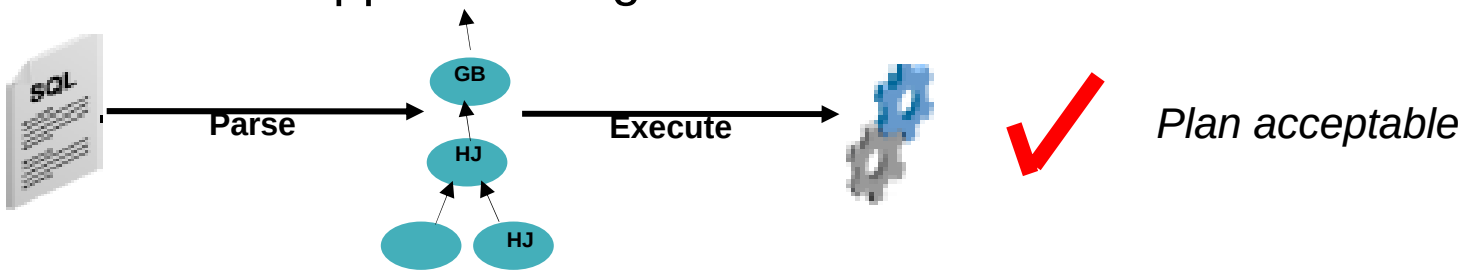


Plan Stability Techniques

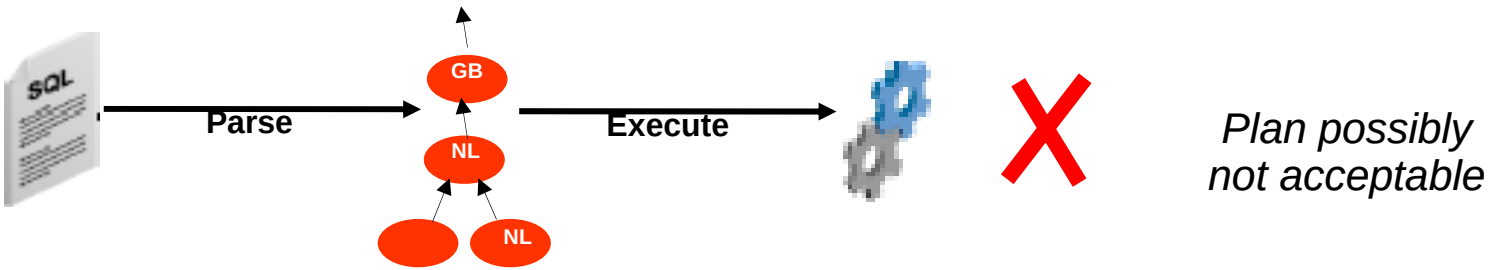
- Optimizer - prevent execution plan changes:
  - Classical approach:
    - Rule Based Optimizer (Please note: RBO desupport since Oracle 10*g*)
    - Hints
    - Stored Outlines
    - Rewriting SQL statements
    - optimizer_features_enabled=n.n.n
    - Change specific optimizer parameters
    - Import and fix object and system statistics
  - Modern, efficient and better resource consumption:
    - SQL Plan Management
    - SQL Profiling

Without SQL Plan Management

- Challenging to "freeze" execution plans and statistics
- Difficulty:
  - Statement has been parsed and a plan got created. Verification happens during execution:



  - Now some conditions get changed (e.g. upgrade)
    - A new plan will be created - is it better or worse???



SQL Plan Management

- First preventive and fully transparent database mechanism to ensure plan stability
- SQL execution plans will be recorded
- A SQL Baseline will be created
  - Consists of accepted execution plans
  - Contains only plans for statements being parsed/executed more than once
- Only known and verified and accepted plans will be used
- Package: **DBMS_SPM**

SQL Plan Management

- 3 phases for plan stability:
  - Capture
    - Create a SQL Baseline representing trusted execution plans
      - Baseline is stored in SQL Management Base in tablespace SYSAUX
      - There are two ways to capture execution plans in the SPM Management Base.
      - Automatic capture of execution plans by setting.
      - OPTIMIZER_CAPTURE_SQL_PLAN.
      - Manual Plan Loading or Bulk load of execution plans using
      - DBMS_SPM.LOAD_PLANS_FROM_SQLSET
        DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE
  - Selection
    - Only accepted plans will be used
    - New execution plans will be recorded in the plan history
  - Evolution
    - Evaluate all unverified plans for a given statement in the plan history to become either accepted or rejected

SQL Plan Management

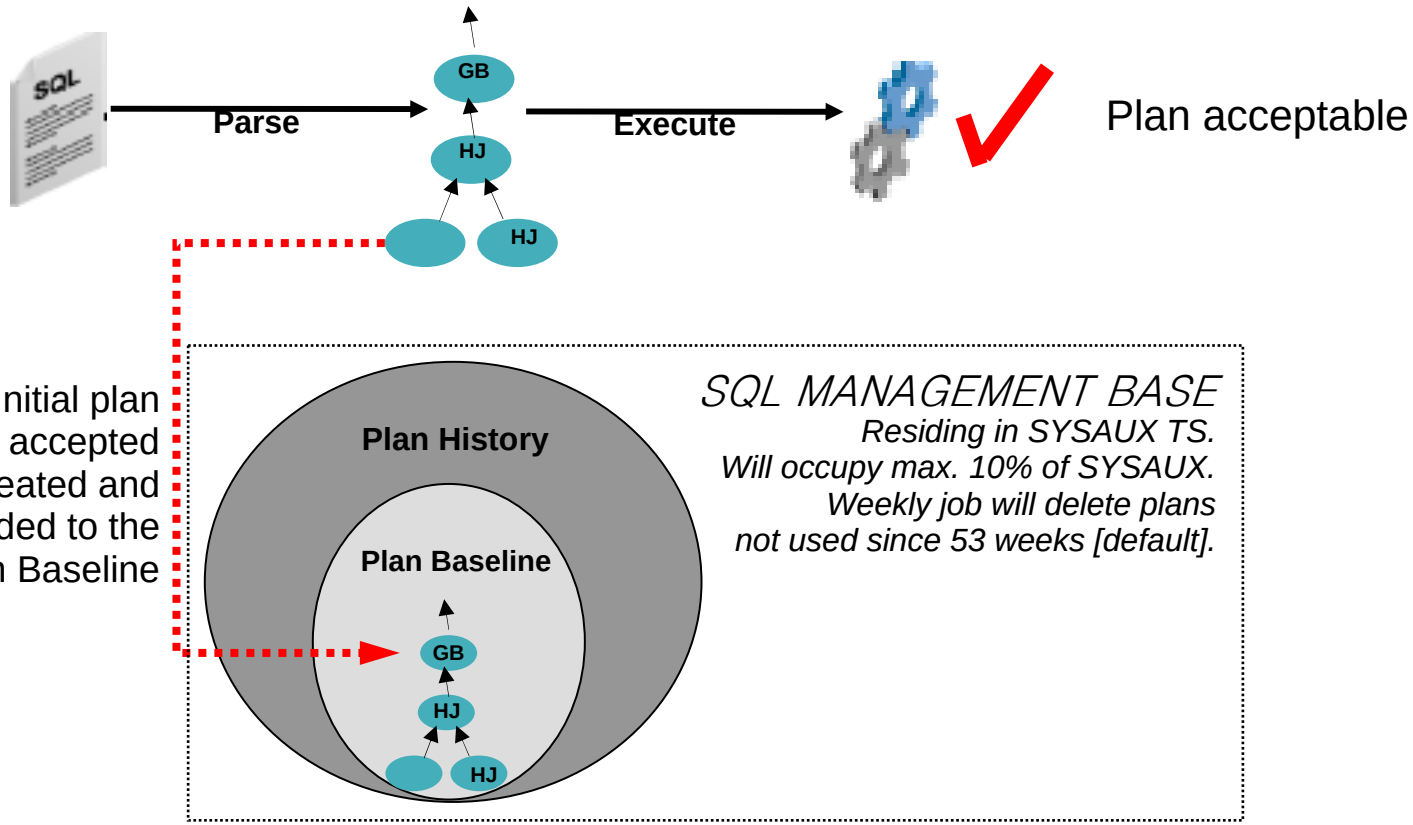
- White Paper:
  - **http://www.oracle.com/technology/products/manageability/database/pdf/ow07/spm_white_paper_ow07.pdf**


SQL Plan Management in Oracle Database 11g

An Oracle White Paper
June 2007

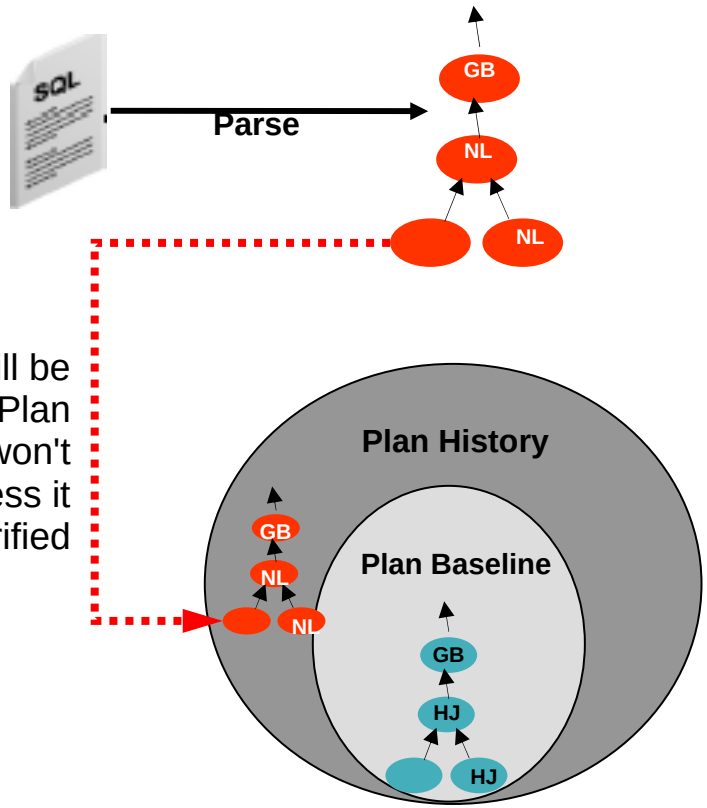
SQL Plan Management

- Phase 1 - Capture
  - **`OPTIMIZER_CAPTURE_SQL_PLAN_BASELINE=TRUE`** (Default: **`FALSE`**)



SQL Plan Management

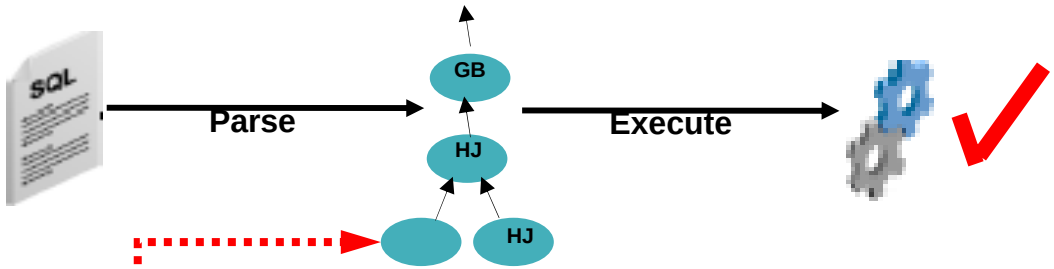
- Phase 2 - Selection
  - **`OPTIMIZER_USE_SQL_PLAN_BASELINE=TRUE`** (Default: **`TRUE`**)
  - Same statement parsed again but a <u>different</u> plan will be created



New plan will be added to the Plan History but it won't be used unless it has been verified

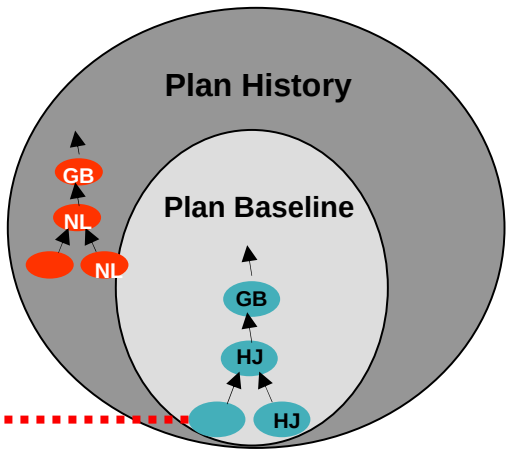
SQL Plan Management

- Phase 2 - Selection
  - **`OPTIMIZER_USE_SQL_PLAN_BASELINE=TRUE`**



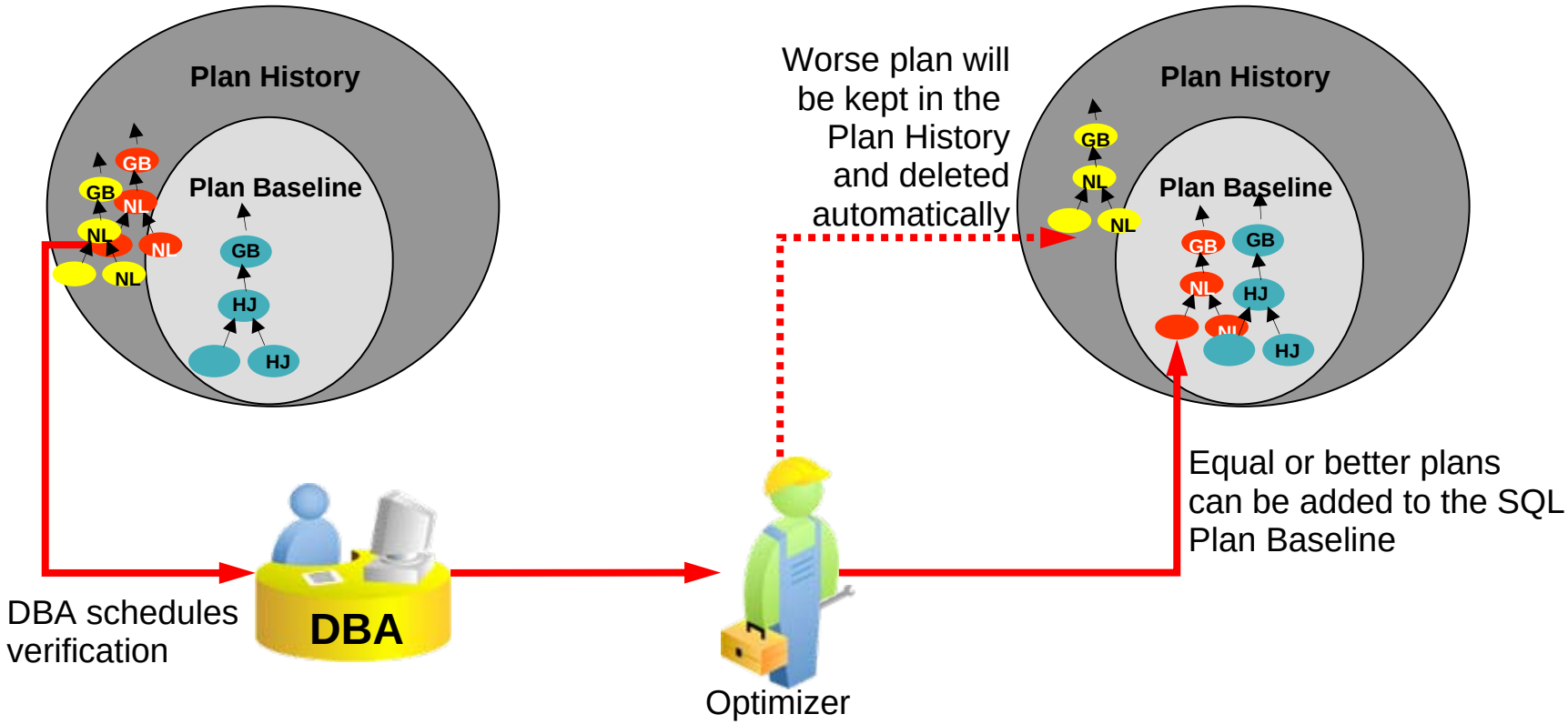
Plan known and acceptable

The optimizer will use only one of the VERIFIED plans stored in the SQL Baseline because only these will guarantee PLAN STABILITY



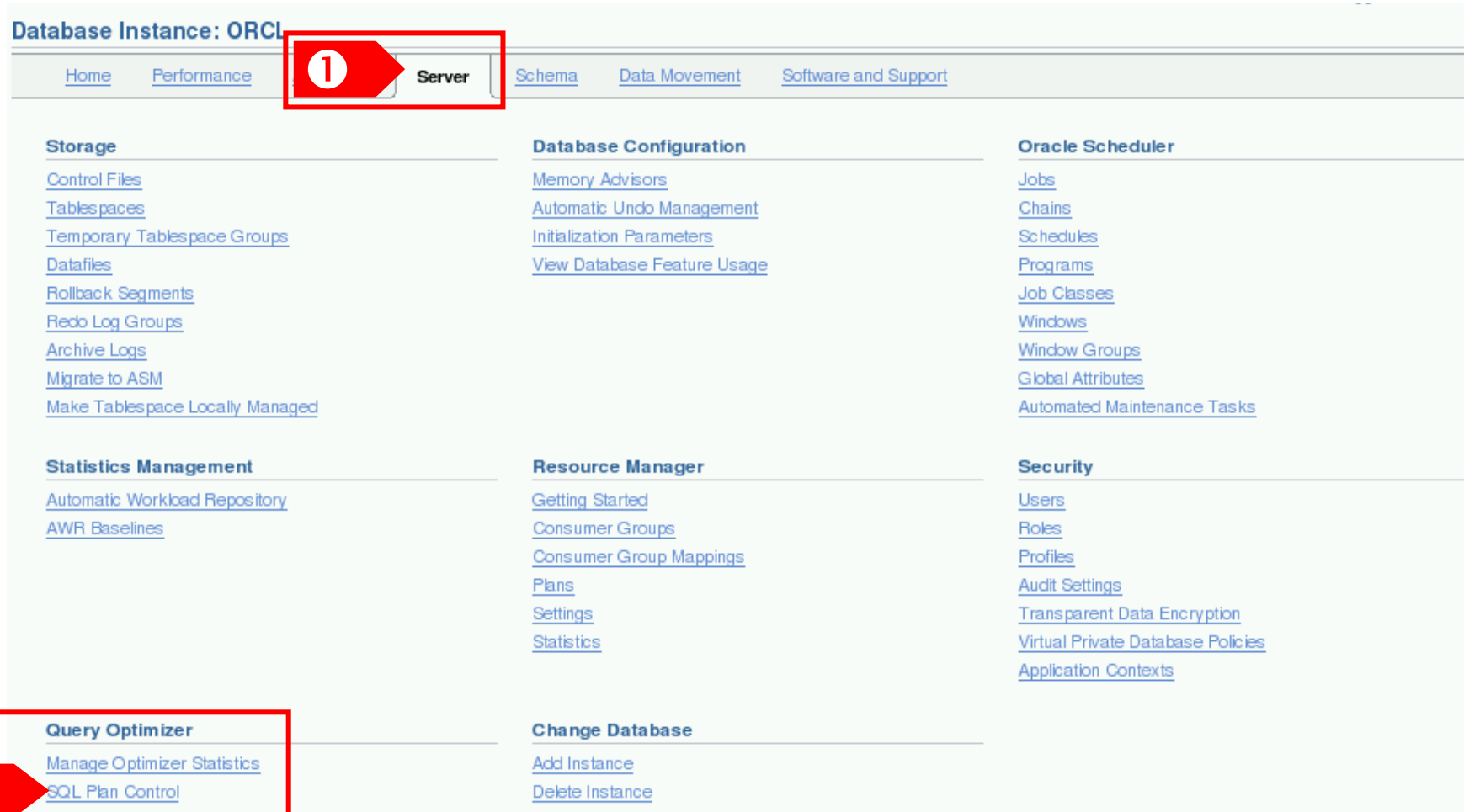
SQL Plan Management

- Phase 3 - Evolution



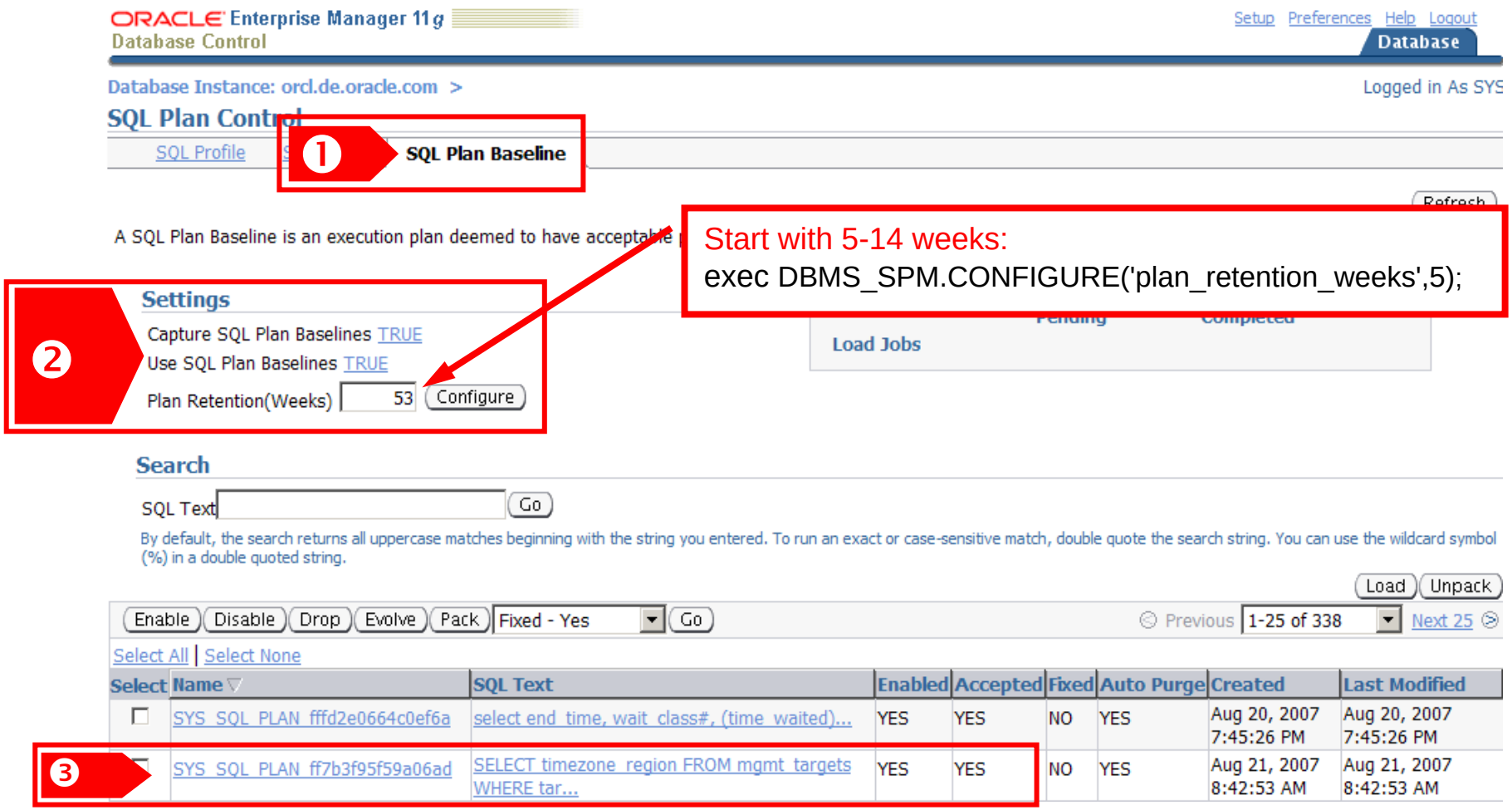
SQL Plan Management

- DatabaseControl:



SQL Plan Management

- DatabaseControl - Configuration:

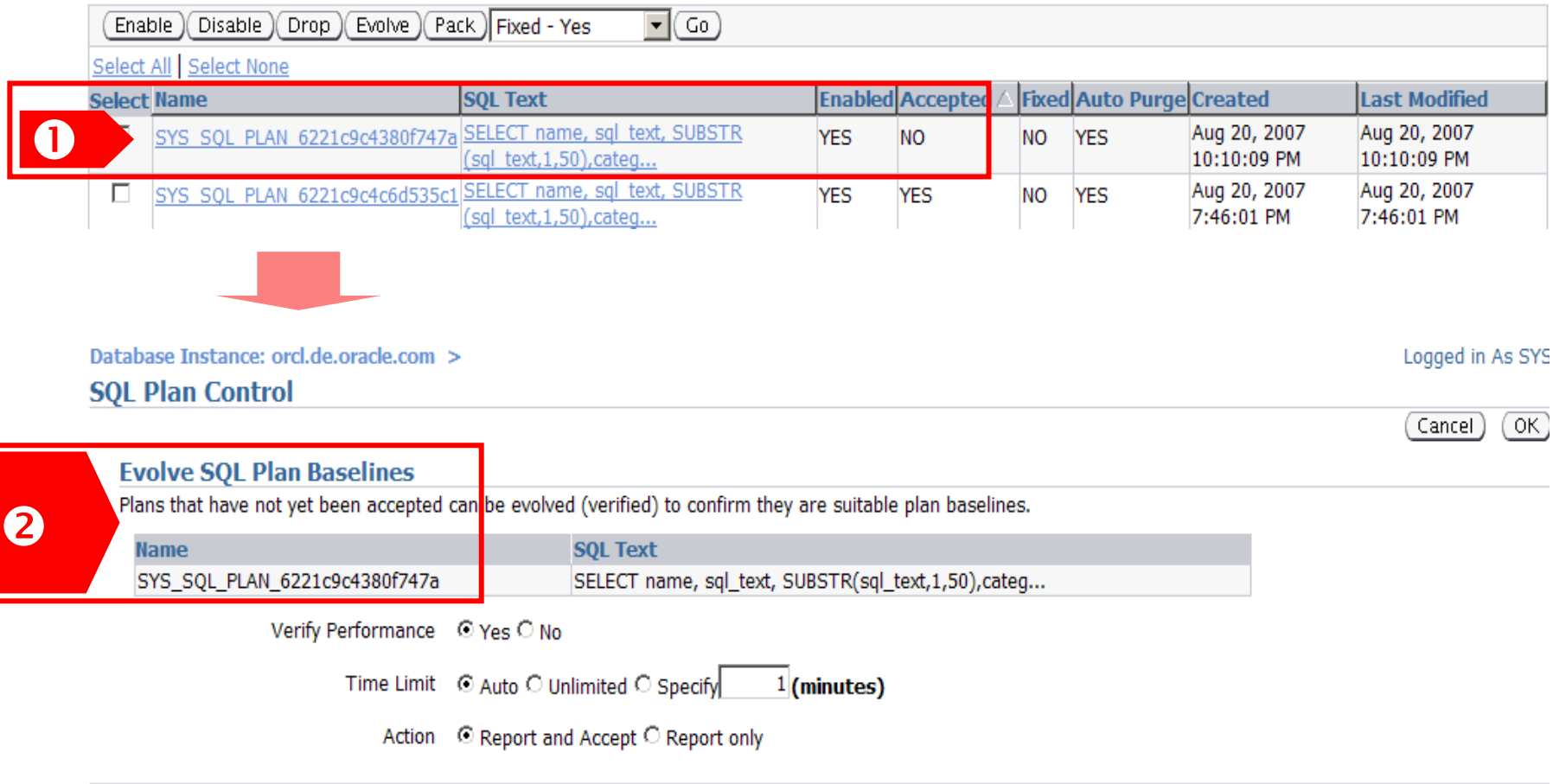


SQL Plan Management

- DatabaseControl - details:

SQL Plan Management

- DatabaseControl - evolve:

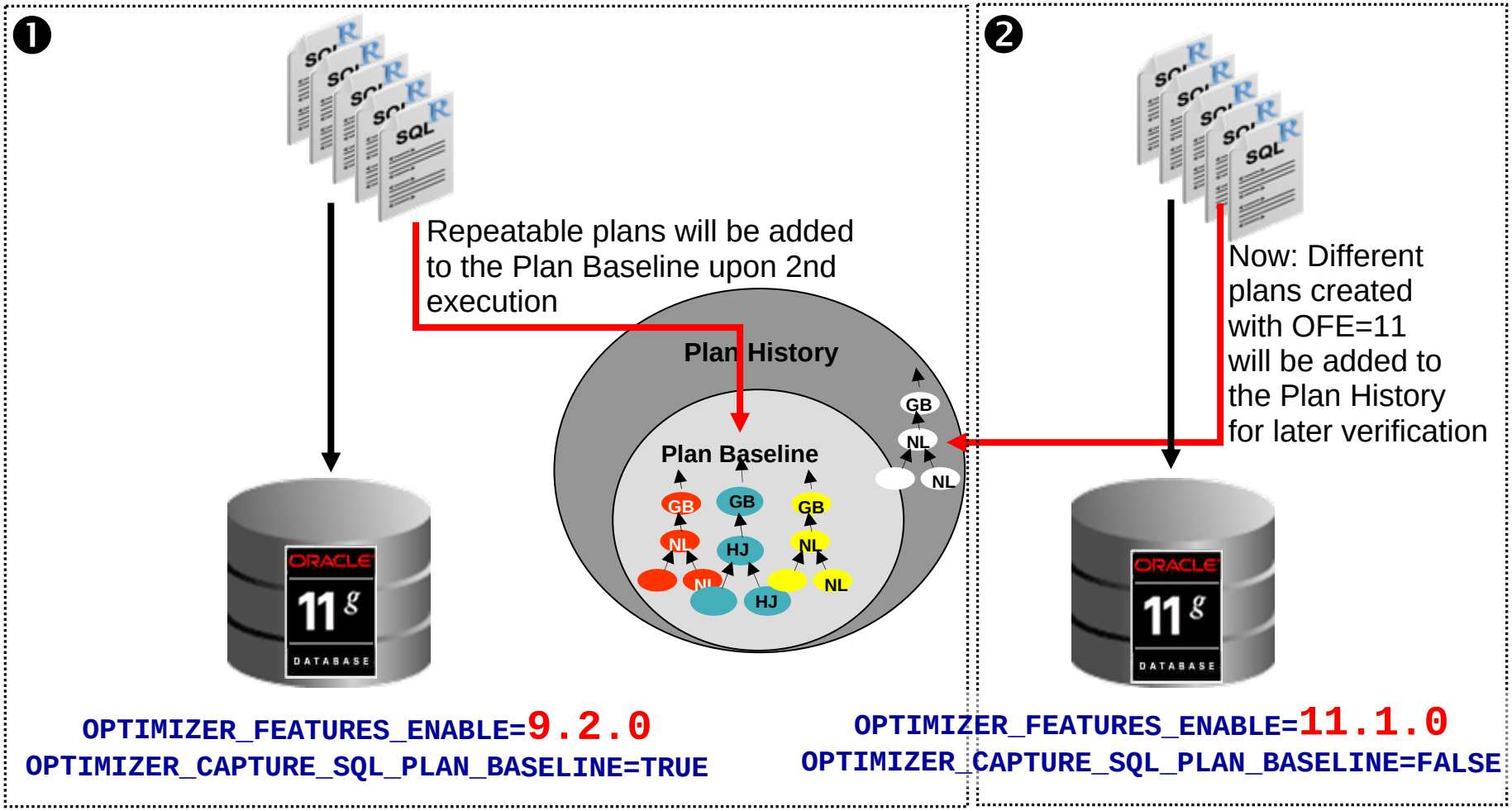


SQL Plan Management - Scenarios

- Upgrade scenario 1:
  - Parameterize the optimizer back to the "old" behaviour
  - Works for all database releases since 8.0.3
- Upgrade scenario 2:
  - Transport all well known plans to the new release
  - Works since 10*gR2*
- Ship a new application (module) along with appropriate SQL plan baselines:
  - Ship the best execution plans for the new statements within a staging table and ensure that they'll be used
  - Works since 10*gR2*

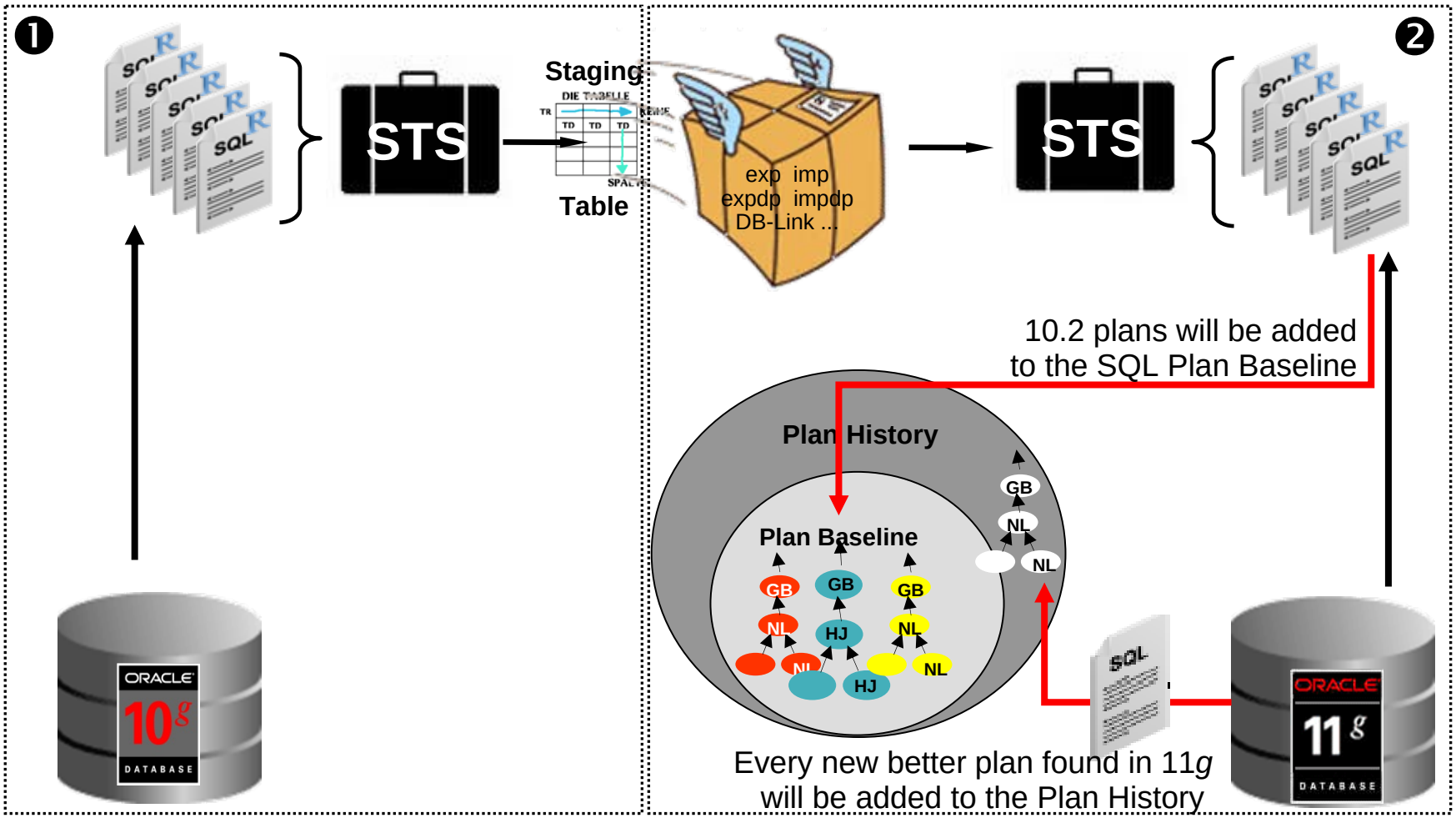
SQL Plan Management - Upgrade 1

- Upgrade scenario 1



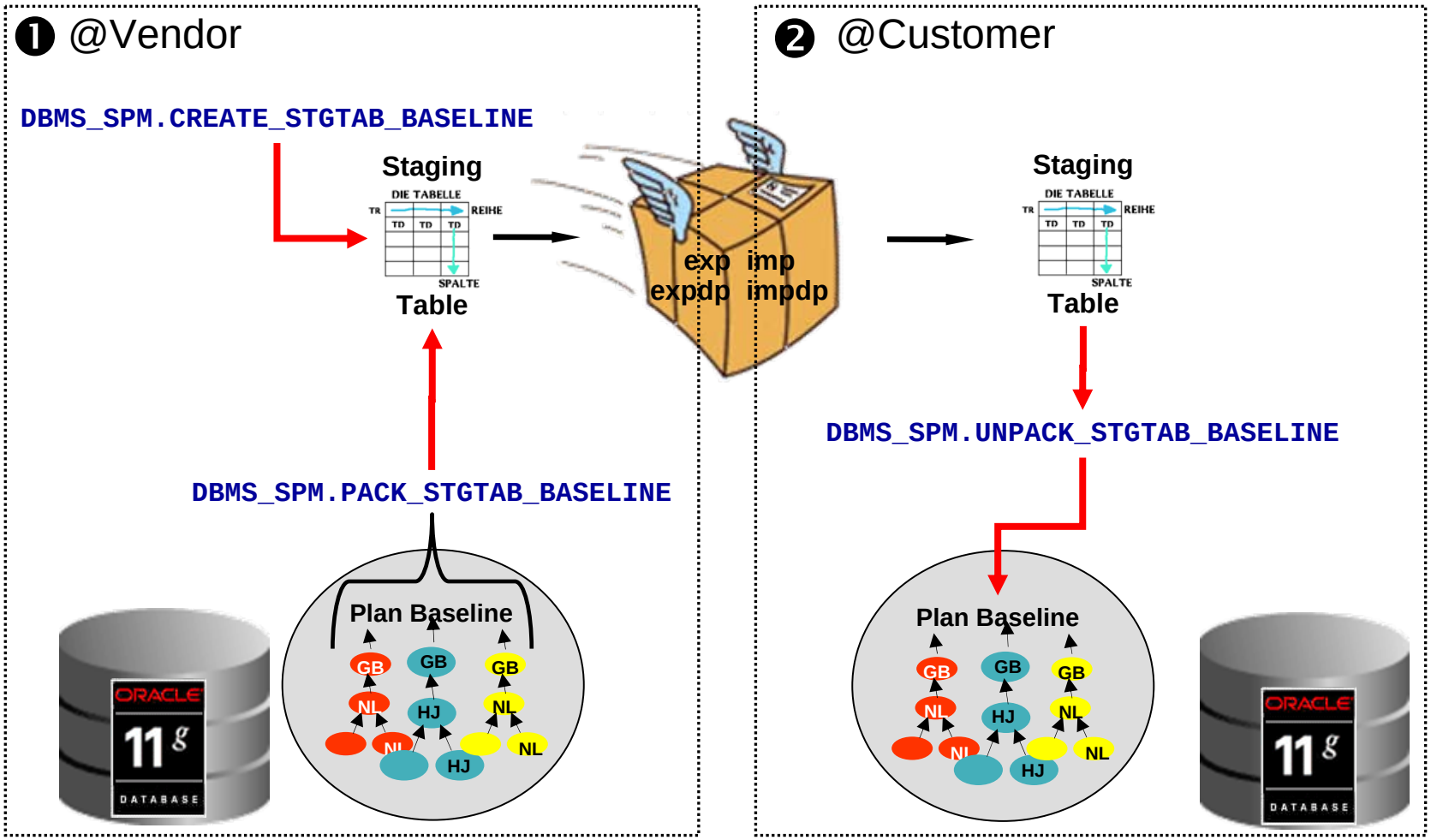
SQL Plan Management - Upgrade 2

- Upgrade scenario 2



SQL Plan Management - New Application

- New application (module) gets delivered



SPM NOTES

- **Note.789888.1**

  HOW TO LOAD SQL PLANS INTO SPM FROM AWR.

- **Note 801033.1**

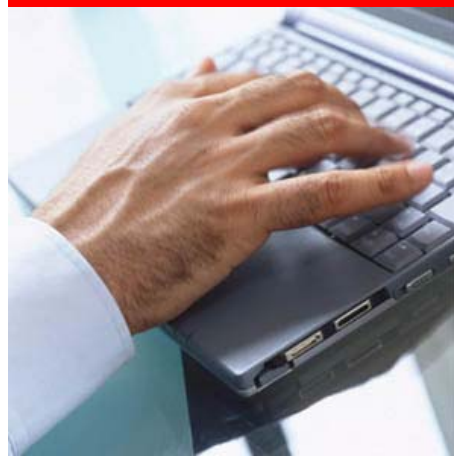
  HOW TO MOVE 10gR2 EXECUTION PLANS AND LOAD INTO 11g SPM.

- **Note.790039.1**

  HOW TO DROP PLANS FROM SPM REPOSITRY.

- **Note.456518.1 SQL PLAN MANAGEMENT.**

Agenda



- Recap
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
- Q&A

Real Application Testing

- Goal:
  - Enable successful adoption of new technology and features - specifically new releases of the database - at low-cost and low-risk by:
    - Record and replay a real workload to see how the new system performs
    - Finding regressions and changing plans before the upgrade

- Licensable database pack "Real Application Testing"
  - ⇨ Available with Oracle Database 11.1.0.6
  - ⇨ Available with patch set 10.2.0.4
  - ⇨ Available as single patch for 9.2.0.8 and 10.2.0.3
    - ⇨ For patch numbers please see Note:560977.1
  - ⇨ Can be used for upgrades from
    - ⇨ 9.2.0.8 to 10.2.0.3/4
    - ⇨ 9.2.0.8 to 11.1.0.x
    - ⇨ 10.2.0.3/4 to 11.1.0.x

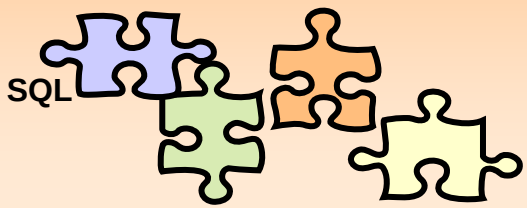
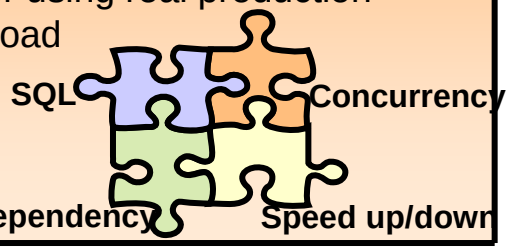
Real Application Testing

- Real Application Testing consists of:
  - Database Replay
    - Package **DBMS_WORKLOAD_CAPTURE**
    - Package **DBMS_WORKLOAD_REPLAY**
  - SQL Performance Analyzer (SPA)
    - Package **DBMS_SQLPA**
  - SQL Tuning Sets (STS)
    - Package **DBMS_SQLTUNE**

Real Application Testing

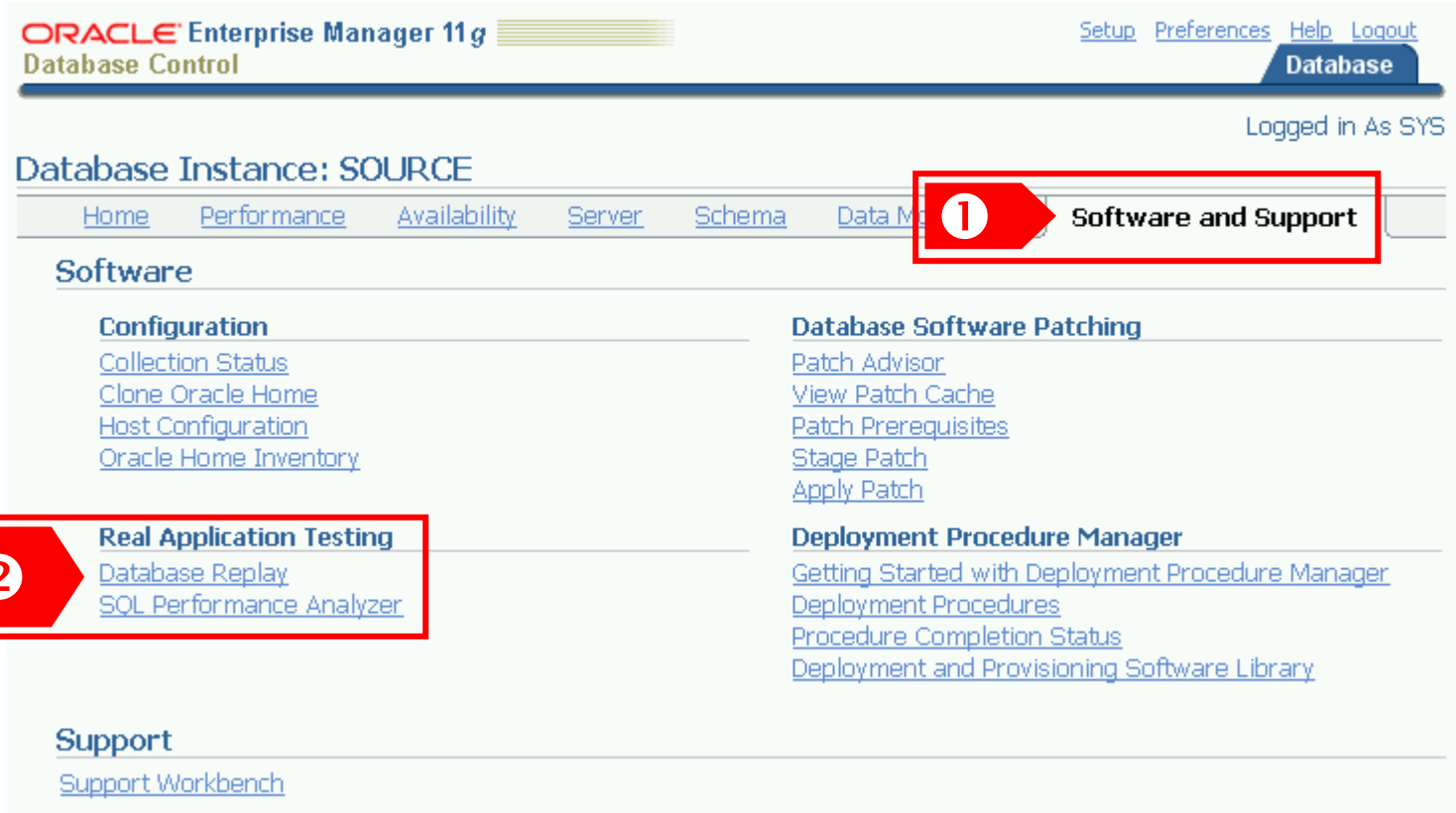
- White Paper:
  - SQL Performance Analyzer:
    - Techinical White Paper: SQL Performance Analyzer
      `http://www.oracle.com/technology/products/manageability/database/pdf/ow07/spa_white_paper_ow07.pdf`
  - Database Replay:
    - Techinical White Paper: Database Replay
      `http://www.oracle.com/technology/products/manageability/database/pdf/ow07/db_replay_white_paper_ow07_1.pdf`

Real Application Testing: Tools of the Trade

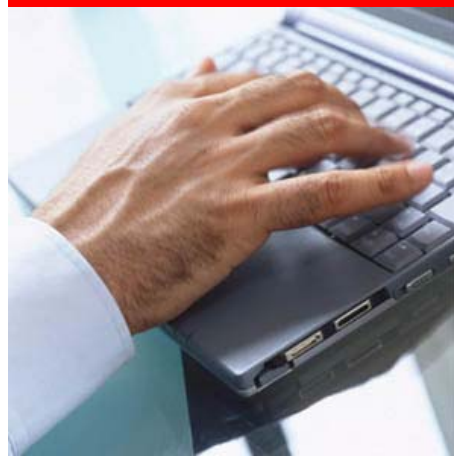
	SQL Performance Analyzer	*Database Replay*
What is it?	• Predicts SQL performance deviations before end-users can be impacted, helps assess impact of change on **SQL response time**	• Replays **real** database workload on test system, helps assess impact of change on **workload throughput**
How it works?	• Executes each SQL, stored in SQL Tuning Set, in **isolation** using **production context** and then compares before and after execution plans and run-time statistics	• Captures workloads and replays it with production characteristics including **concurrency**, **synchronization & dependencies**
When to use?	• **Unit testing** of SQL with the goal to identify the set of SQL statements with improved/regressed performance 	• **Comprehensive** testing of all sub-systems of the database server using real production workload 

Real Application Testing

- Database Control:

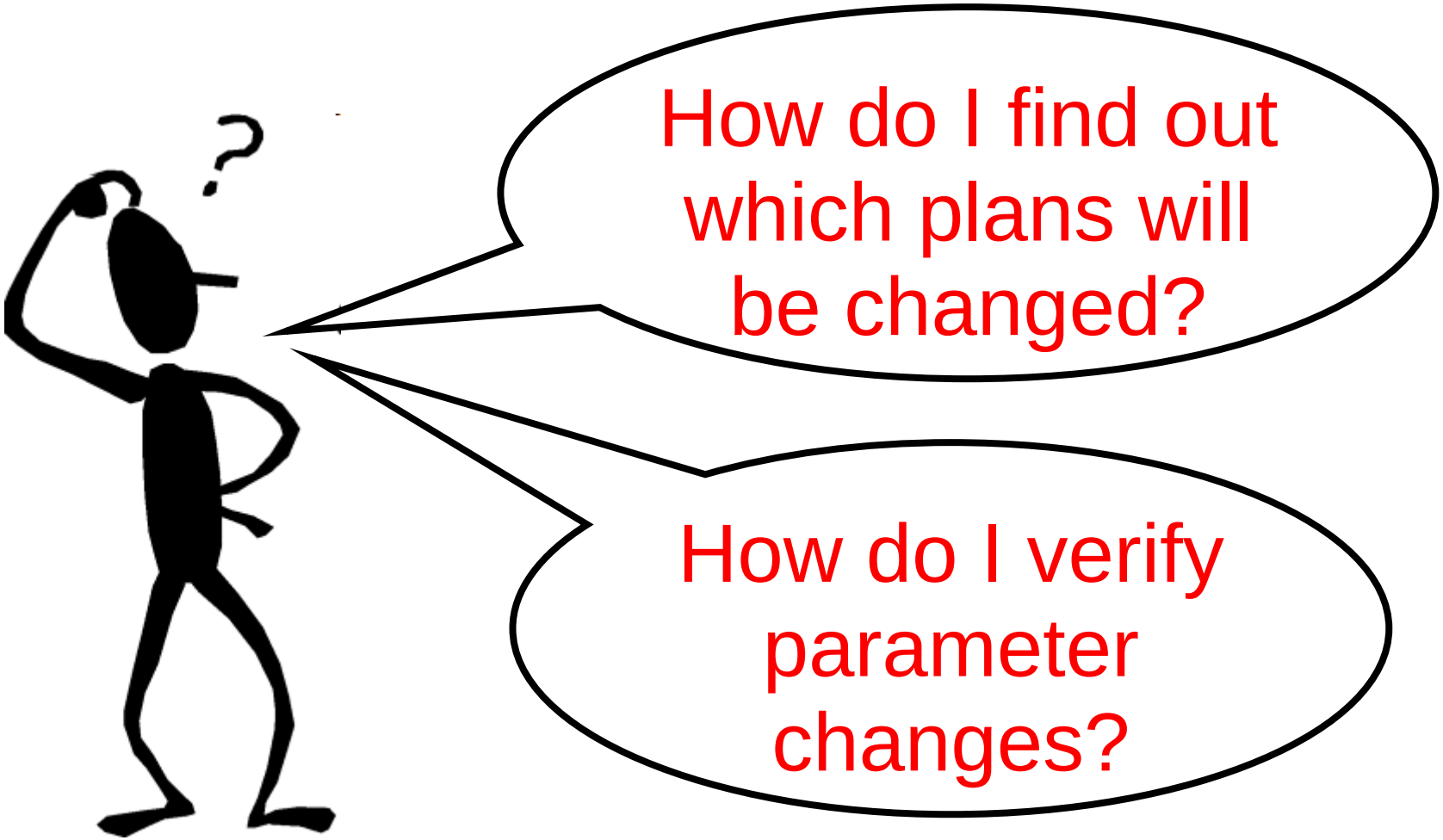


Agenda



- Recap
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
  - SQL Performance Analyzer
  - Database Replay
- Q&A

Challenge

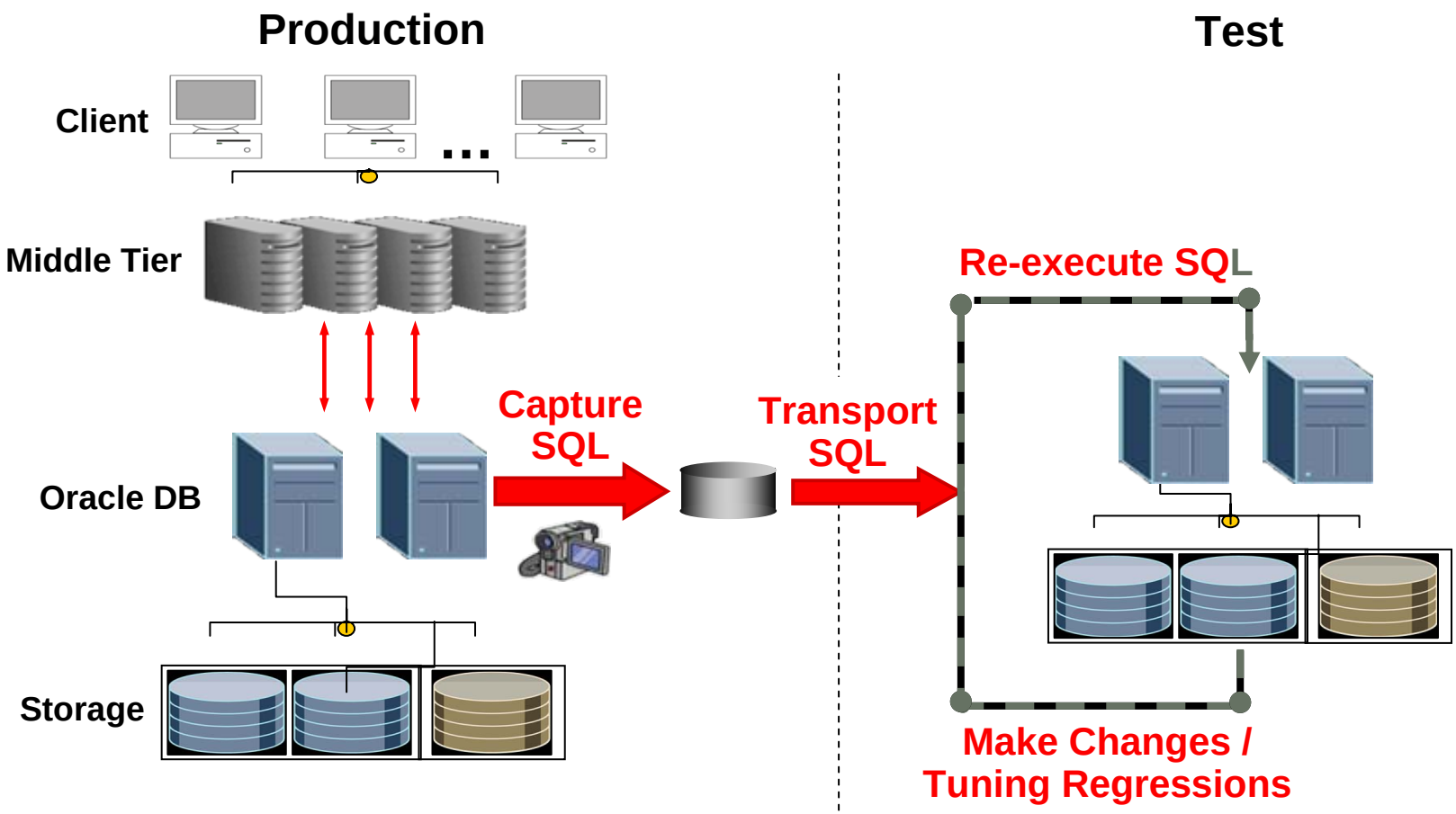

How do I find out which plans will be changed?
How do I verify parameter changes?

Common SPA Usage Scenarios

- Common SPA usage scenario examples
  - Database upgrades, patch set changes
  - Parameter changes
  - Schema changes
  - Optimizer statistics refresh, etc.

- SPA interface
  - Rich GUI through Enterprise Manager
  - Command line interface through **DBMS_SQLPA** PL/SQL package

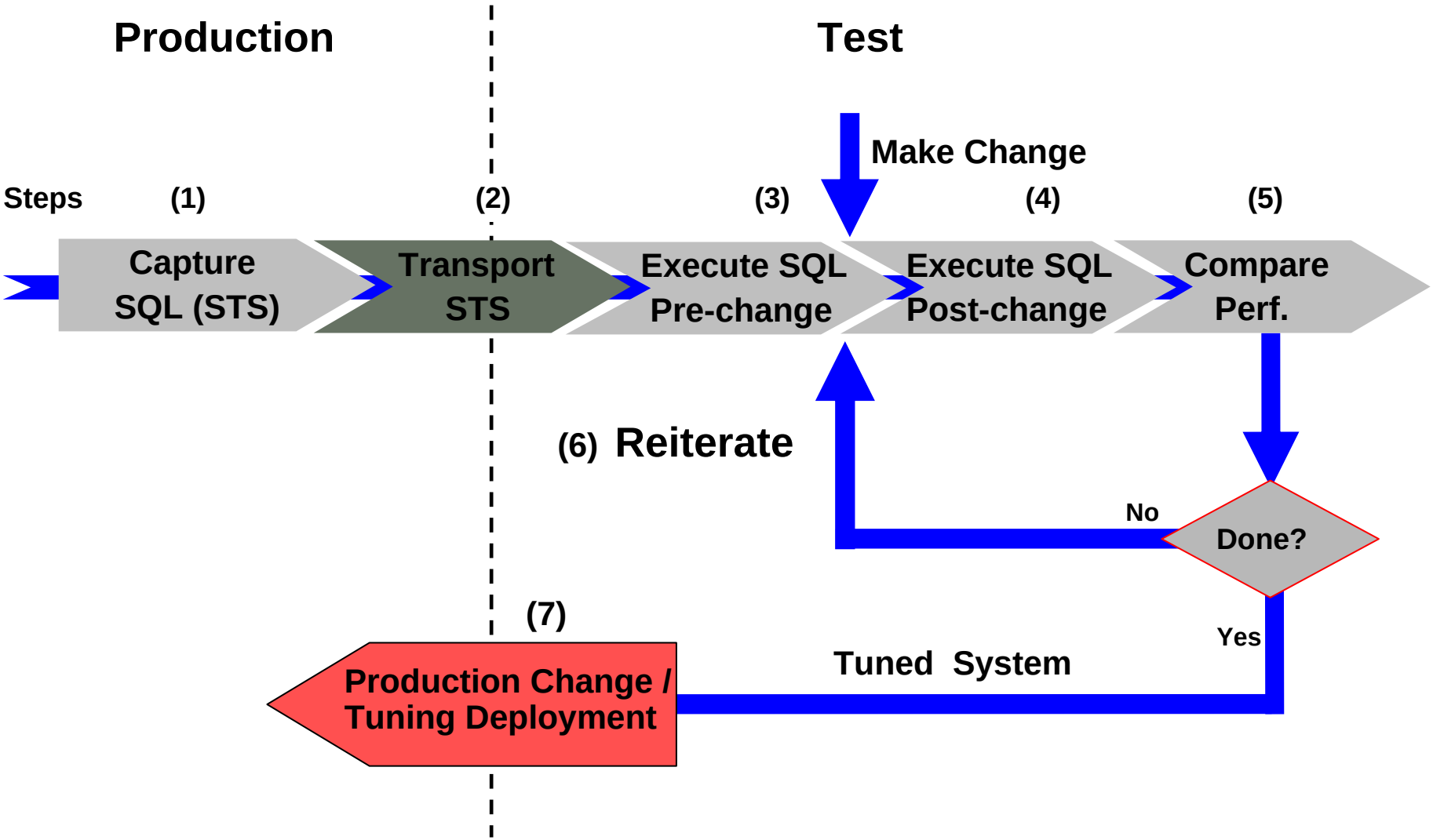
SQL Performance Analyzer: Overview



- If adequate spare cycles available, optionally execute SQL here

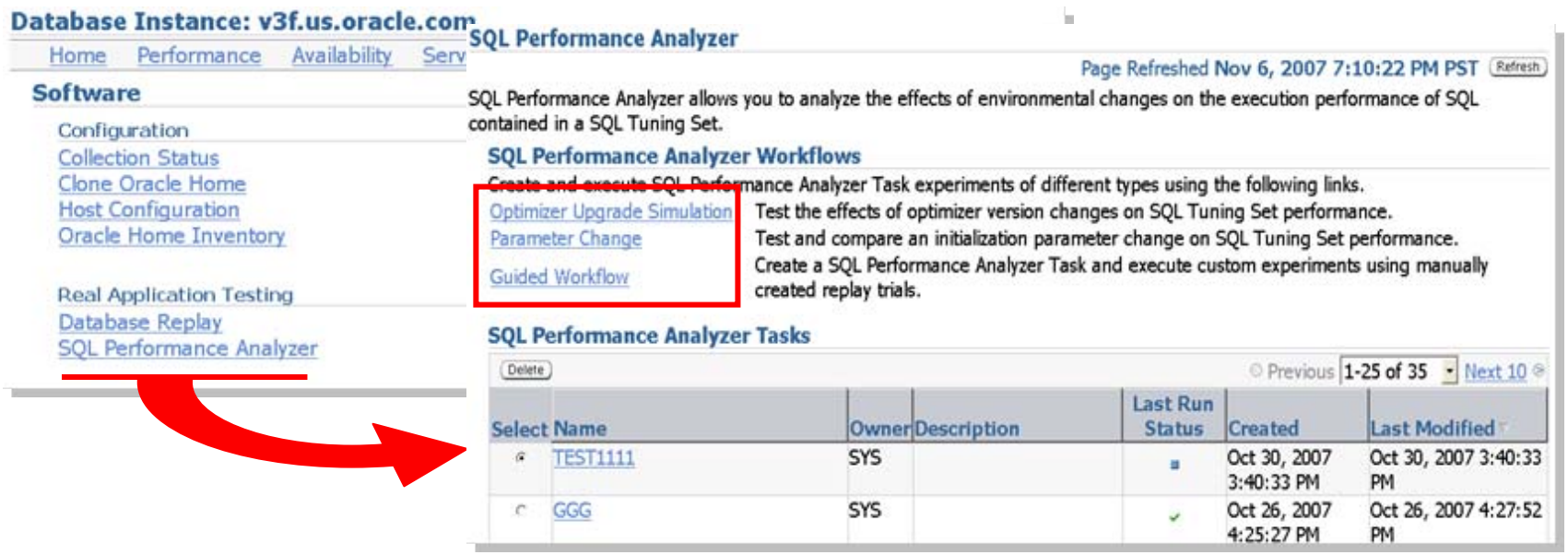
* No middle & application tier setup required

SQL Performance Analyzer: Workflow



SPA Workflows

- Possible workflows:
  - Optimizer upgrade simulation (Start: 8.0.0)
  - Init parameter change
  - Guided workflow for manual changes
- All are based on SQL Tuning Sets



Database Upgrade: 10*g* to 11*g*

***Scenario 1**:*

I want to upgrade from 10.x to 11.x database release to benefit from 11g functionality. How can I best accomplish the upgrade?

***Goal**:*

Assess impact of upgrade on SQL workload performance so that there are no surprises after upgrade. Once upgraded to 11*g* new features can be enabled one at a time.

Database Upgrade: 10g to 11g

- Step 1:
  - Capture workload into STS:
    Preferred method (> 10.2.0.1)

```
EXEC DBMS_SQLTUNE.CAPTURE_CURSOR_CACHE_SQLSET( .. )
```

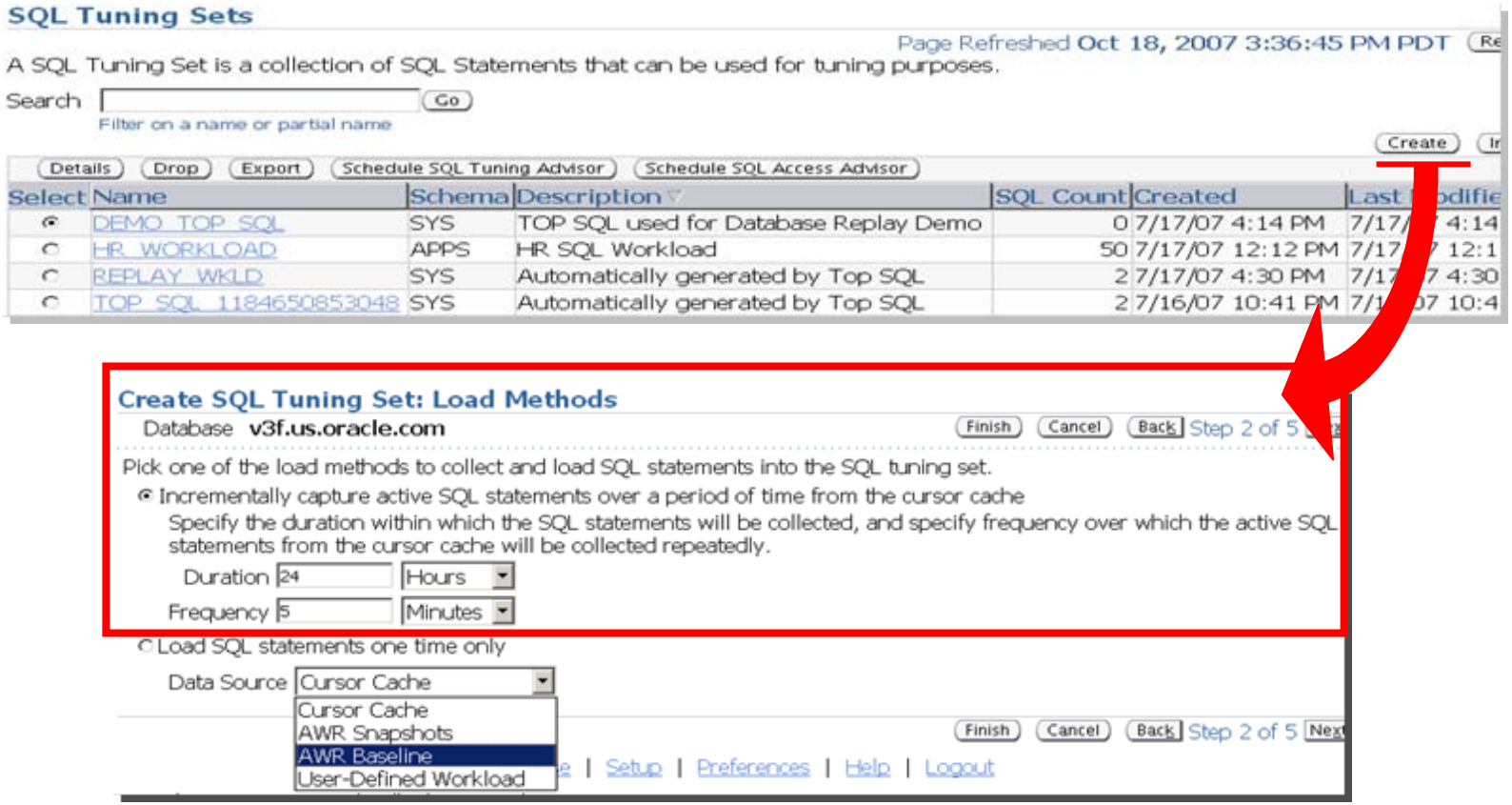
  - Create STS from Top SQL in AWR / AWR Baseline

  - For 9i database enable SQL tracing:

```
alter system set events
         '10046 trace name context forever, level 4';
```

Database Upgrade: 10*g* to 11*g*

- Step 1 in EM:

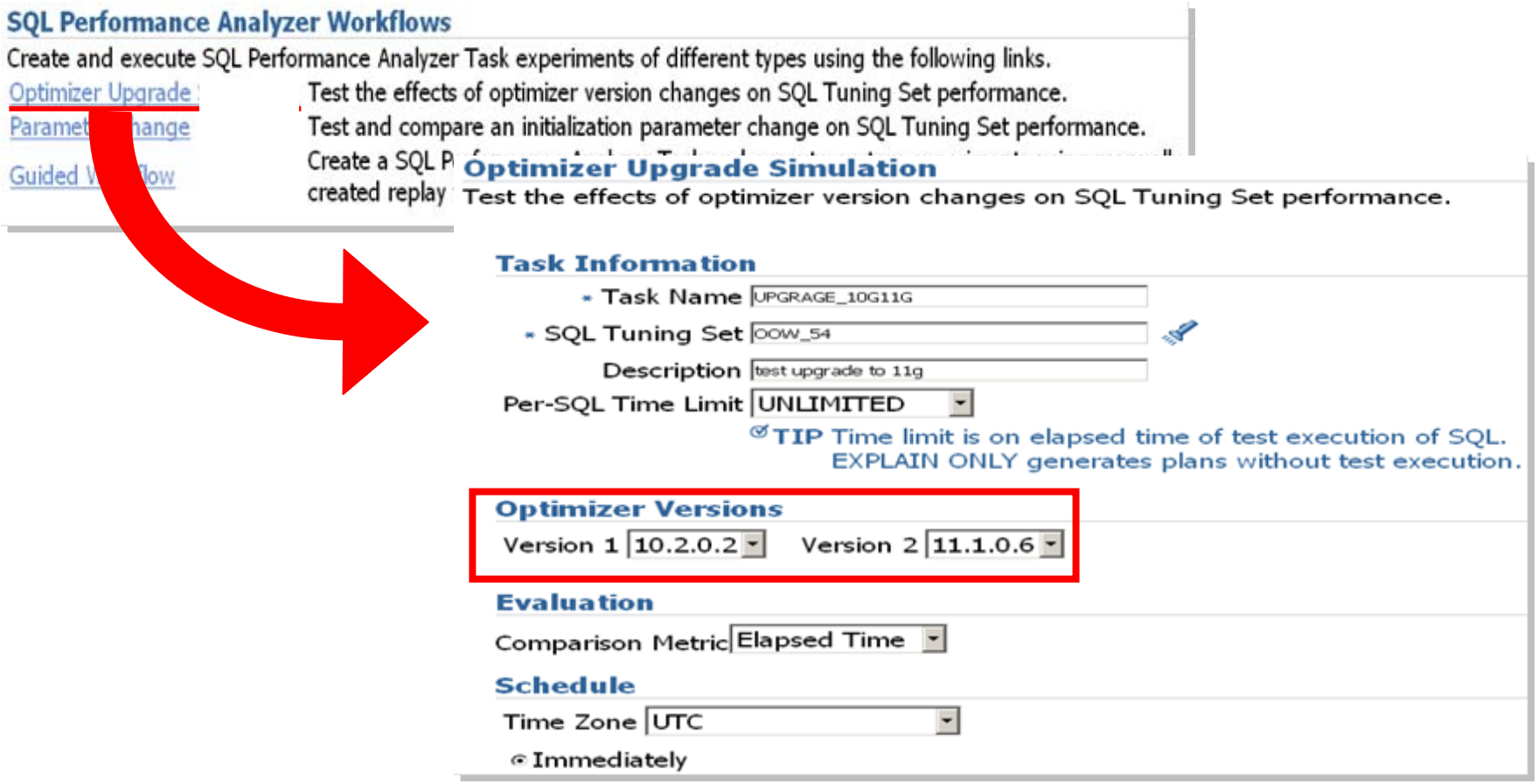


Database Upgrade: 10g to 11g

- Step 2:
  Upgrade a copy of 10*g* production to 11*g* test database
- Step 3:
  Run SPA in 11*g* test database
  - Select “Optimizer Upgrade” workflow link

Database Upgrade: 10g to 11g

- Step 3 in EM: Run SPA task



Database Upgrade: 10g to 11g

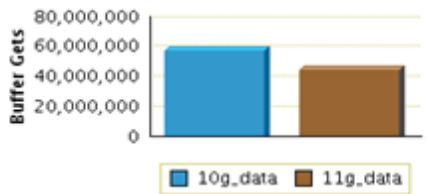
- SPA results

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

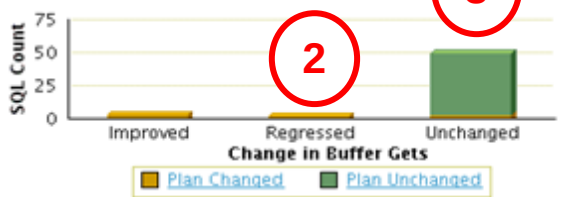
Projected Workload Buffer Gets



Improvement Impact 24%
Regression Impact -2%
Overall Impact 22%

1

SQL Statement Count



2

3

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

4

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 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

Database Upgrade: 10g to 11g

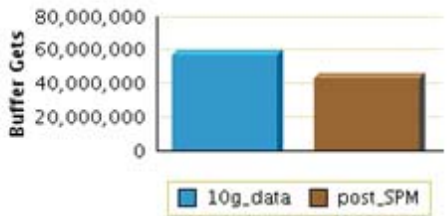
- After using SQL Plan Management with a baseline

SQL Performance Analyzer Task Result: SYS.UPGRADE_10G11G

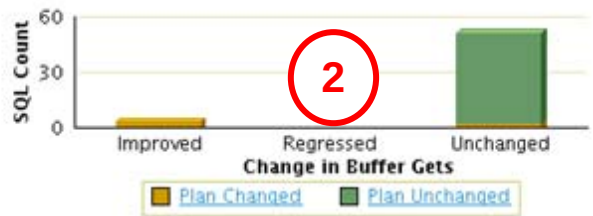
Task Name	UPGRADE_10G11G	SQL Tuning Set Name	OOW_54G	Replay Trial 1	10g_data
Task Owner	SYS	STS Owner	SYS	Replay Trial 2	post_SPM
Task Description	test upgrade to 11g	Total SQL Statements	54	Comparison Metric	Buffer Gets
		SQL Statements With Errors	0		

Global Statistics

Projected Workload Buffer Gets



SQL Statement Count



2

Improvement Impact 24%

Regression Impact 0%

1

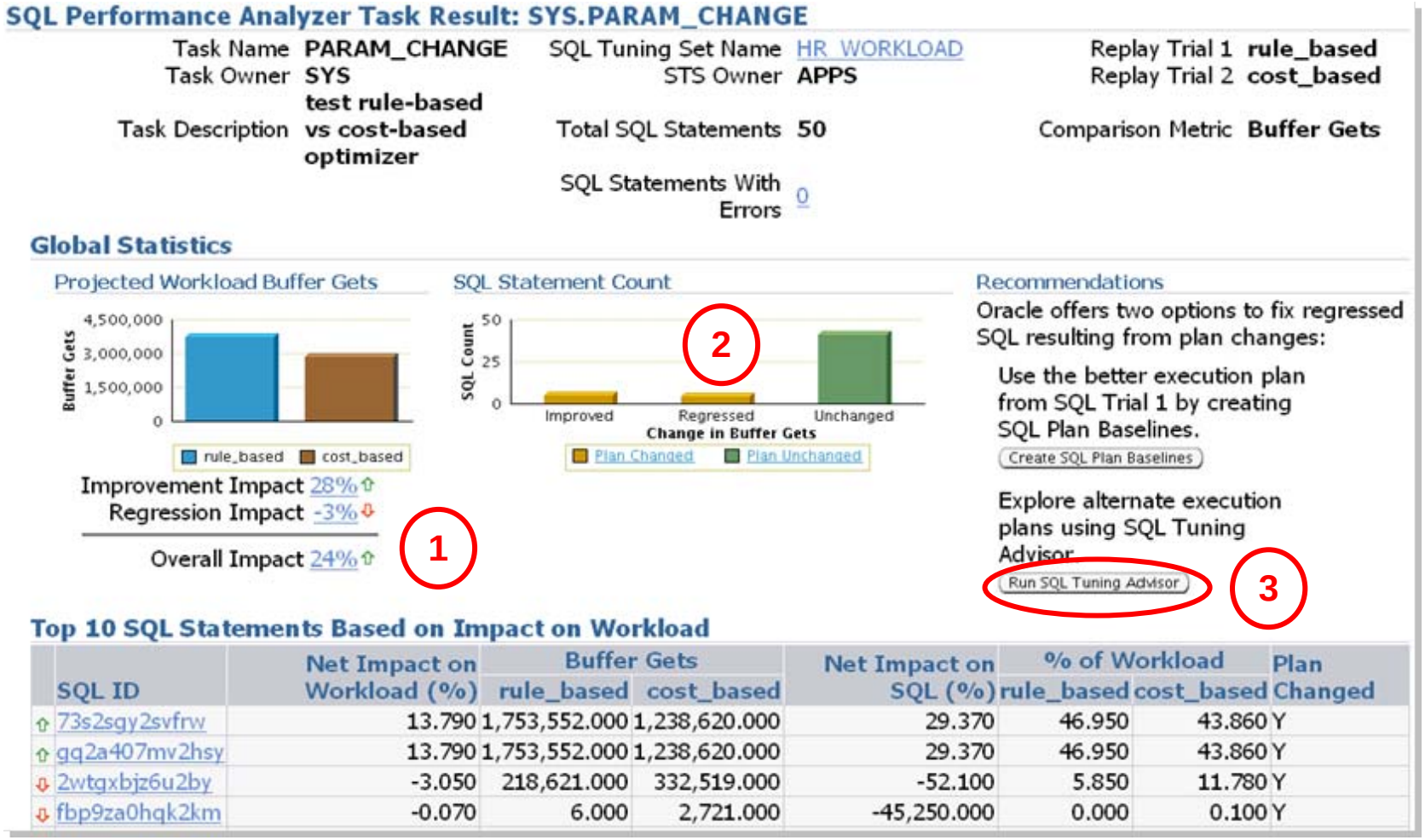
Overall Impact 24%

Top 10 SQL Statements Based on Impact on Workload

SQL ID	Net Impact on Workload (%)	Buffer Gets 10g_data	Buffer Gets post_SPM	Net Impact on SQL (%)	% of Workload 10g_data	% of Workload post_SPM	Plan Changed
g4dzf4ak4rus2	12.010	20,318,458.000	13,501,307.000	33.550	35.780	31.300	Y
gfacm5jr3rz9j	11.990	6,990,541.000	180,401.000	97.420	12.310	0.420	Y
2wtgxbjz6u2by	0.050	244,678.000	218,533.000	10.690	0.430	0.510	Y

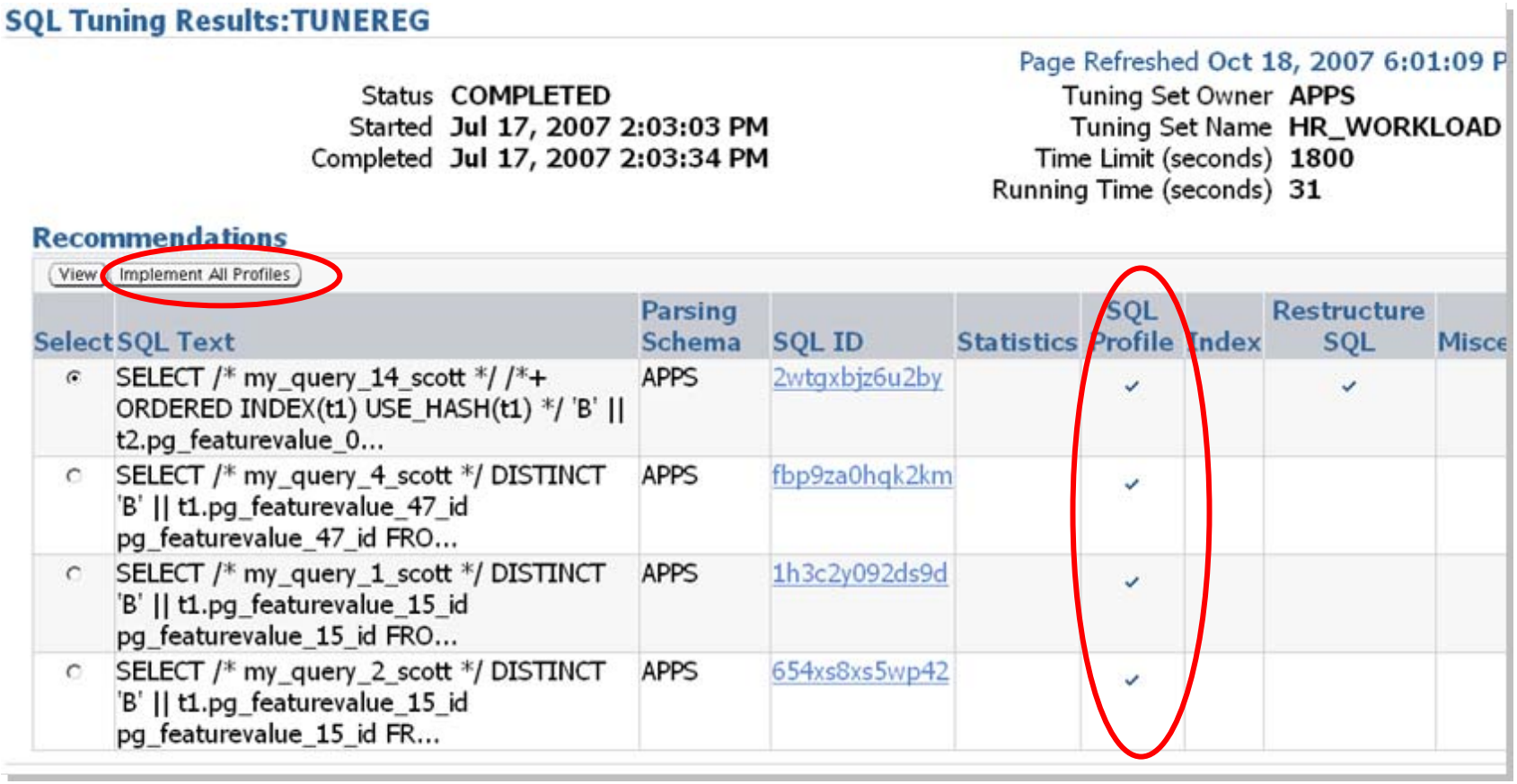
Parameter Changes

- SPA Recommendation: SQL Tuning Advisor



Parameter Changes

- SQL Tuning Advisor offers SQL Profiles



SQL Tuning Results:TUNEREG

Page Refreshed Oct 18, 2007 6:01:09 P

Status **COMPLETED**
Started **Jul 17, 2007 2:03:03 PM**
Completed **Jul 17, 2007 2:03:34 PM**

Tuning Set Owner **APPS**
Tuning Set Name **HR_WORKLOAD**
Time Limit (seconds) **1800**
Running Time (seconds) **31**

Recommendations

View | Implement All Profiles

Select	SQL Text	Parsing Schema	SQL ID	Statistics	SQL Profile	Index	Restructure SQL	Misce
◉	SELECT /* my_query_14_scott */ /*+ ORDERED INDEX(t1) USE_HASH(t1) */ 'B' || t2.pg_featurevalue_0...	APPS	2wtgxbjz6u2by		✓		✓	
○	SELECT /* my_query_4_scott */ DISTINCT 'B' || t1.pg_featurevalue_47_id pg_featurevalue_47_id FRO...	APPS	fbp9za0hqk2km		✓			
○	SELECT /* my_query_1_scott */ DISTINCT 'B' || t1.pg_featurevalue_15_id pg_featurevalue_15_id FRO...	APPS	1h3c2y092ds9d		✓			
○	SELECT /* my_query_2_scott */ DISTINCT 'B' || t1.pg_featurevalue_15_id pg_featurevalue_15_id FR...	APPS	654xs8xs5wp42		✓			

Parameter Changes

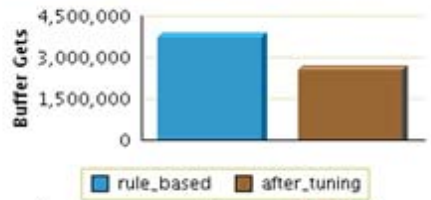
- SPA run after SQL Profiles had been applied

SQL Performance Analyzer Task Result: SYS.PARAM_CHANGE

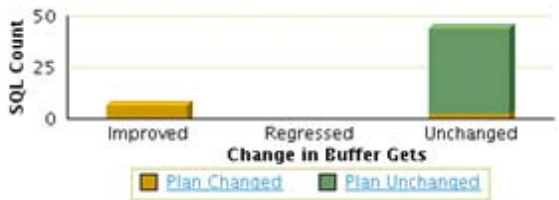
Task Name	PARAM_CHANGE	SQL Tuning Set Name	HR_WORKLOAD	Replay Trial 1	rule_based
Task Owner	SYS	STS Owner	APPS	Replay Trial 2	after_tuning
Task Description	test rule-based vs cost-based optimizer	Total SQL Statements	50	Comparison Metric	Buffer Gets
		SQL Statements With Errors	0		

Global Statistics

Projected Workload Buffer Gets



SQL Statement Count



Improvement Impact 33%
Regression Impact 0%
Overall Impact 33%

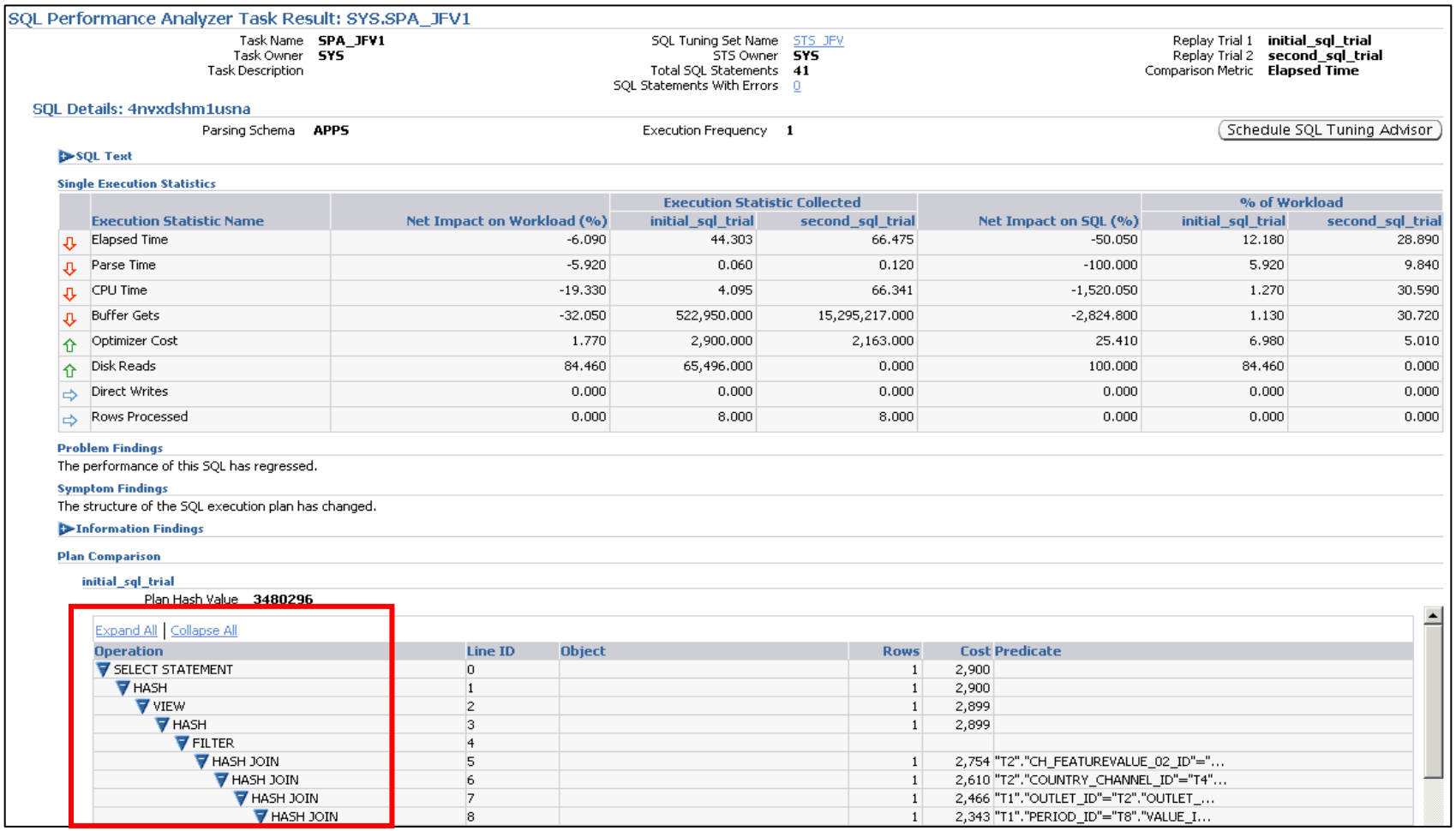
1

Top 10 SQL Statements Based on Impact on Workload

SQL ID	Net Impact on Workload (%)	Buffer Gets rule_based	Buffer Gets after_tuning	Net Impact on SQL (%)	% of Workload rule_based	% of Workload after_tuning	Plan Changed
73s2sgy2svfrw	13.790	1,753,552.000	1,238,620.000	29.370	46.950	49.300	Y
gq2a407mv2hsy	13.790	1,753,552.000	1,238,620.000	29.370	46.950	49.300	Y
2wtgxbjz6u2by	5.090	218,621.000	28,343.000	87.040	5.850	1.130	Y

SQL Performance Analyzer

- SPA: SQL statement details example



SQL Performance Analyzer Task Result: SYS.SPA_JFV1

Task Name SPA_JFV1
Task Owner SYS
Task Description

SQL Tuning Set Name STS_JFV
STS Owner SYS
Total SQL Statements 41
SQL Statements With Errors 0

Replay Trial 1 initial_sql_trial
Replay Trial 2 second_sql_trial
Comparison Metric Elapsed Time

SQL Details: 4nvxdshm1usna

Parsing Schema APPS
Execution Frequency 1
Schedule SQL Tuning Advisor

SQL Text

Single Execution Statistics

Execution Statistic Name	Net Impact on Workload (%)	Execution Statistic Collected: initial_sql_trial	Execution Statistic Collected: second_sql_trial	Net Impact on SQL (%)	% of Workload: initial_sql_trial	% of Workload: second_sql_trial
Elapsed Time	-6.090	44.303	66.475	-50.050	12.180	28.890
Parse Time	-5.920	0.060	0.120	-100.000	5.920	9.840
CPU Time	-19.330	4.095	66.341	-1,520.050	1.270	30.590
Buffer Gets	-32.050	522,950.000	15,295,217.000	-2,824.800	1.130	30.720
Optimizer Cost	1.770	2,900.000	2,163.000	25.410	6.980	5.010
Disk Reads	84.460	65,496.000	0.000	100.000	84.460	0.000
Direct Writes	0.000	0.000	0.000	0.000	0.000	0.000
Rows Processed	0.000	8.000	8.000	0.000	0.000	0.000

Problem Findings

The performance of this SQL has regressed.

Symptom Findings

The structure of the SQL execution plan has changed.

Information Findings

Plan Comparison

initial_sql_trial

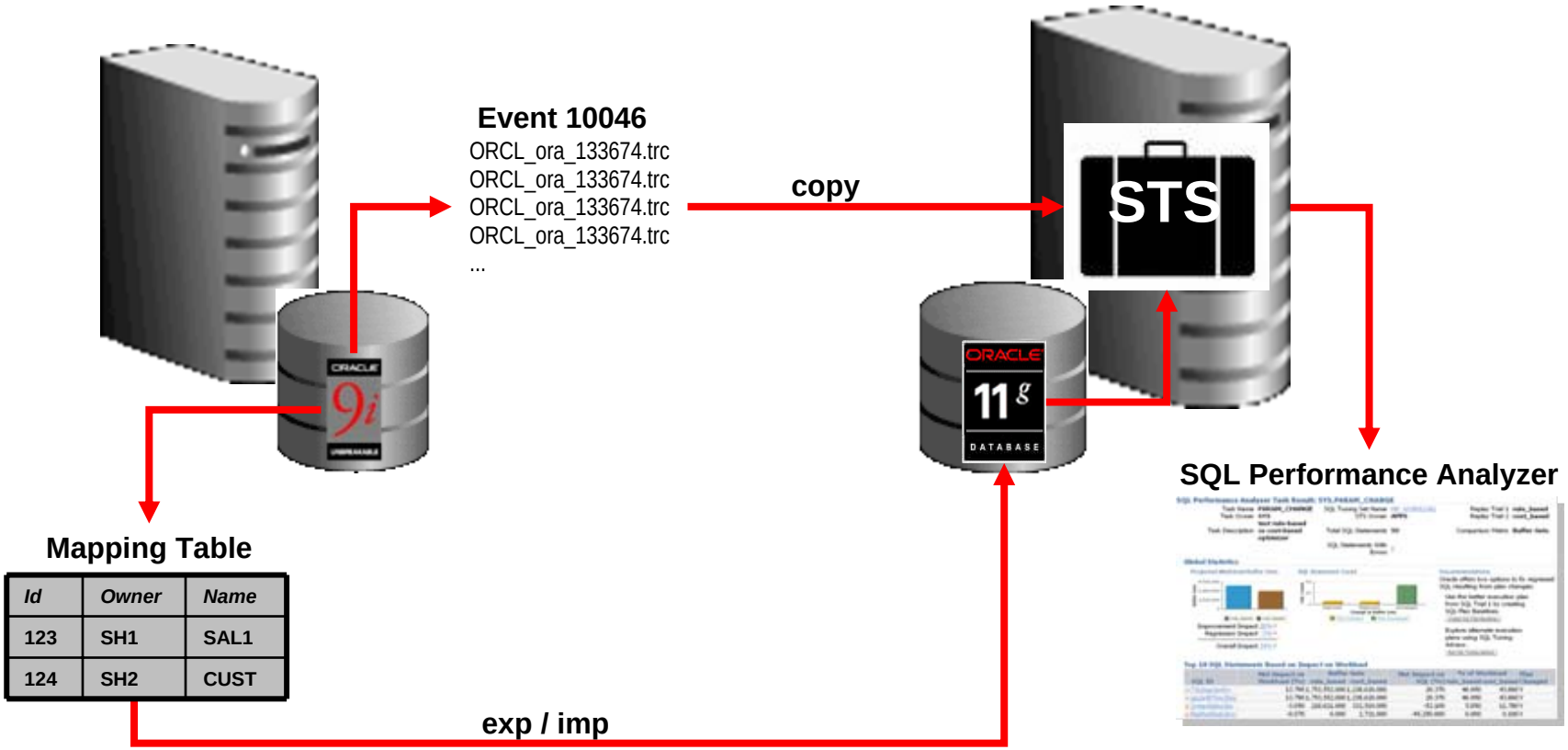
Plan Hash Value 3480296

Expand All | Collapse All

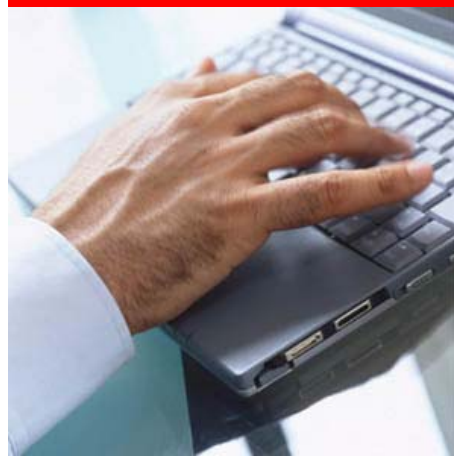
Operation	Line ID	Object	Rows	Cost	Predicate
SELECT STATEMENT	0		1	2,900	
HASH	1		1	2,900	
VIEW	2		1	2,899	
HASH	3		1	2,899	
FILTER	4				
HASH JOIN	5		1	2,754	"T2"."CH_FEATUREVALUE_02_ID"="...
HASH JOIN	6		1	2,610	"T2"."COUNTRY_CHANNEL_ID"="T4"...
HASH JOIN	7		1	2,466	"T1"."OUTLET_ID"="T2"."OUTLET_...
HASH JOIN	8		1	2,343	"T1"."PERIOD_ID"="T8"."VALUE_I...

SPA for a 9*i* workload against 11*g*

- Upgrade scenario from 9*i* directly to 11g
  - See Note: 562899.1 for the White Paper and Note:455889.1 for a SPA example

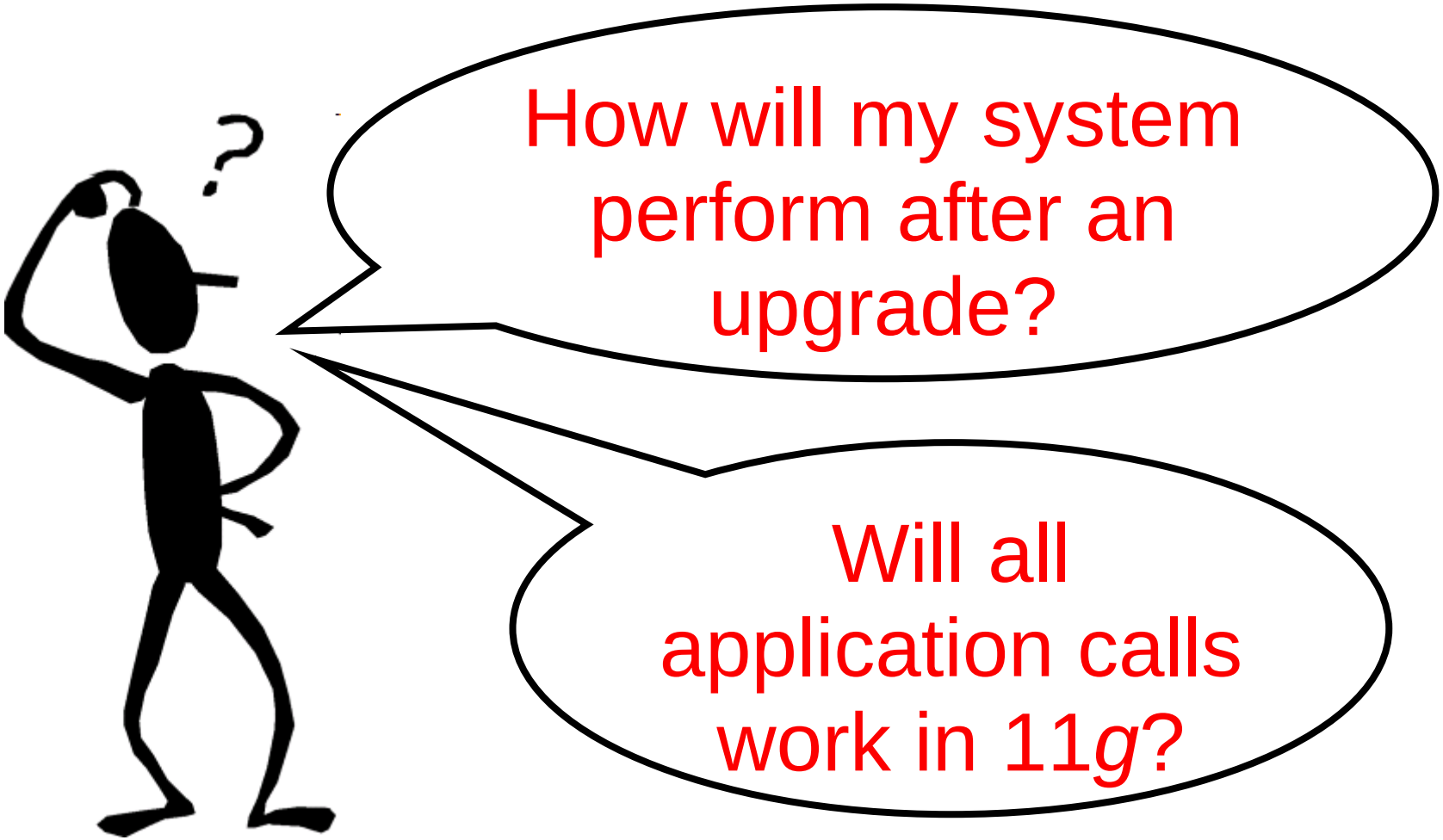


Agenda



- Recap
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
  - SQL Performance Analyzer
  - Database Replay
- Q&A

Challenge


How will my system perform after an upgrade?
Will all application calls work in 11g?

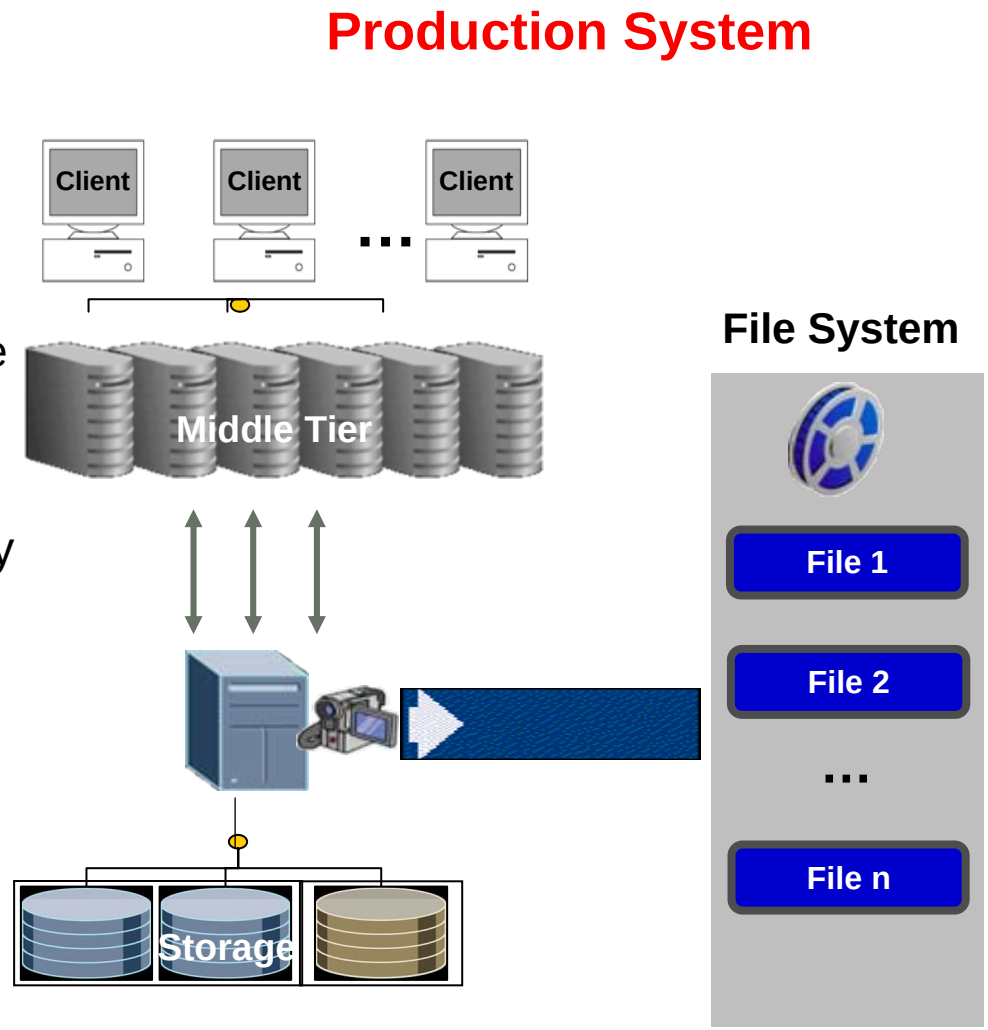
Database Replay

- Capture and replay an actual production database workload
- Identify, analyze and fix potential instabilities before making changes to production

- 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
  - Honor commit ordering
- Analyze & Report
  - Errors
  - Data divergence
  - Performance divergence

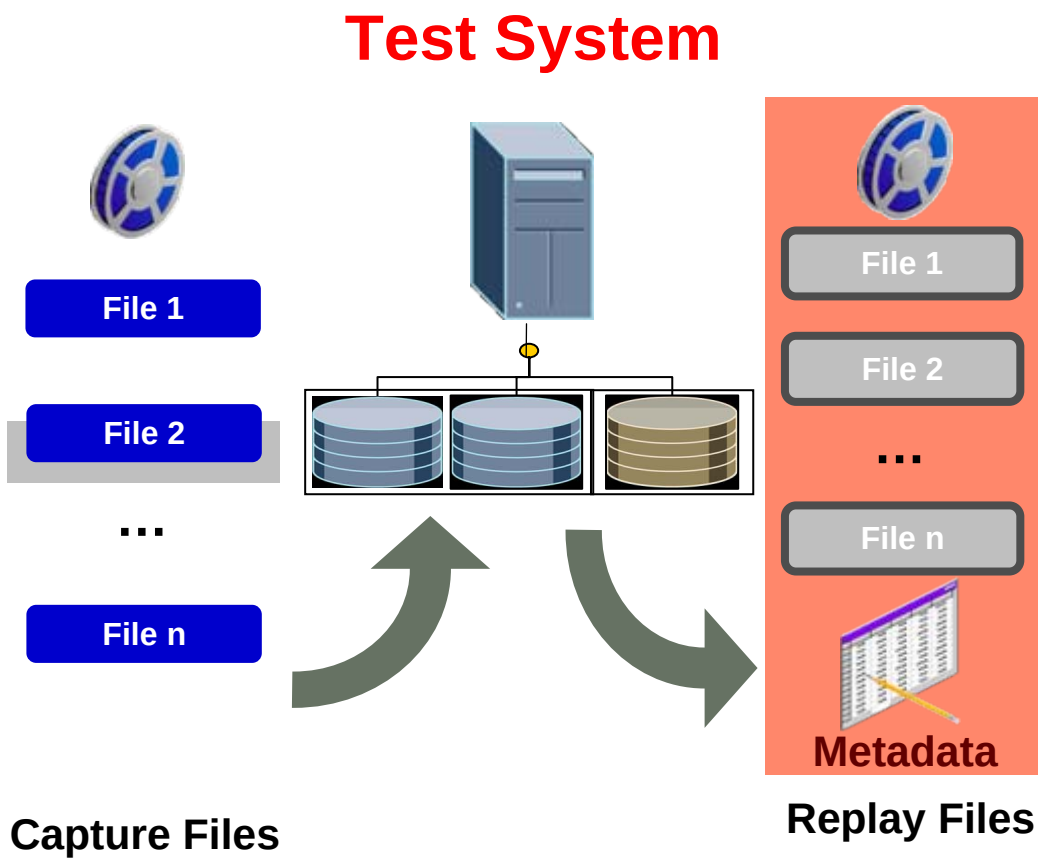
Step 1: Workload Capture

- All external client requests captured in binary files
- System background and internal activity excluded
- Minimal overhead
  - Avoids function call when possible
  - Buffered I/O
- Independent of client protocol
- Can capture on 9.2.0.8 and replay on 11g
- Capture load for interesting time period, e.g., peak workload, month-end processing, etc.



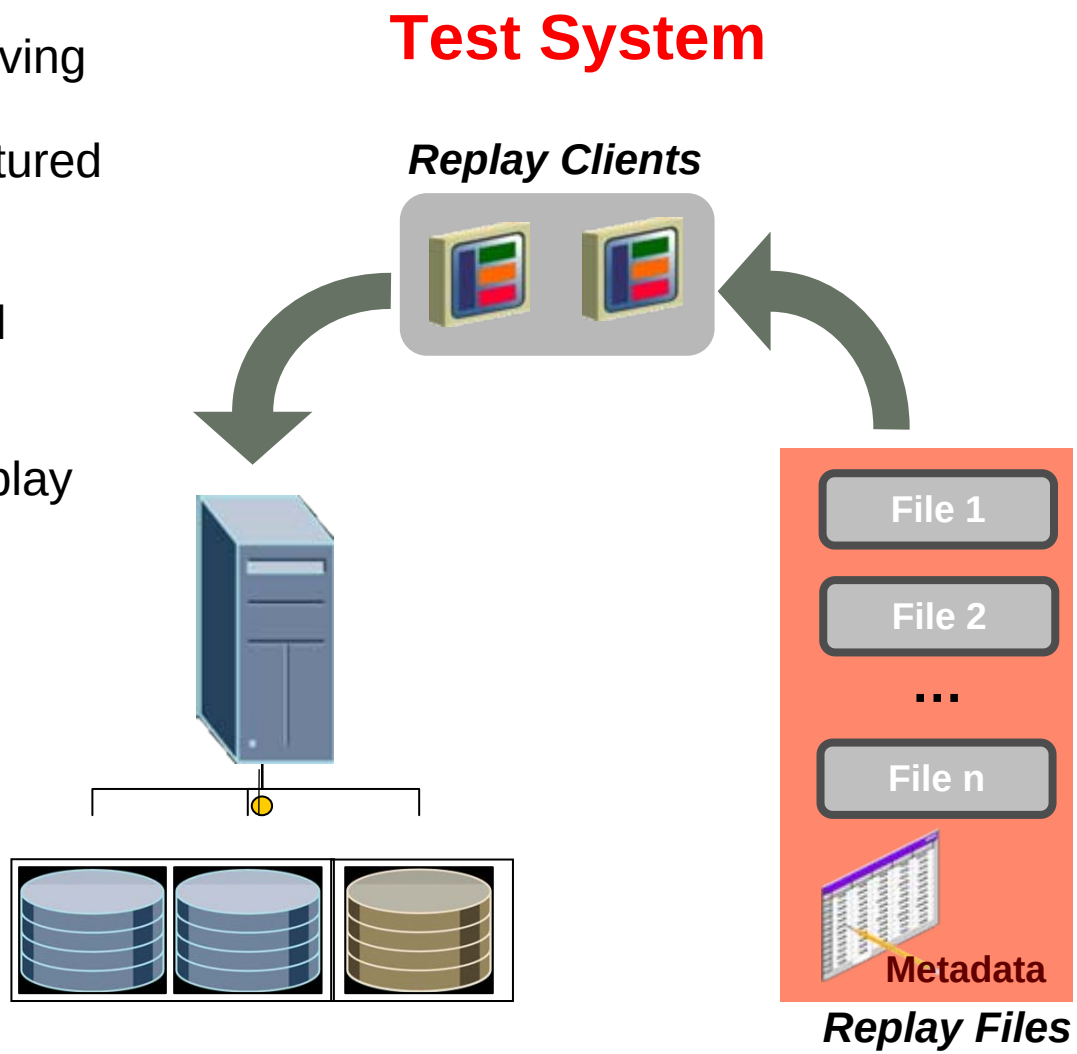
Step 2: Process Workload Files

- Setup test system
  - Application data should be same as production system as of capture start time
  - Use RMAN, Snapshot Standby, imp/exp, Data Pump, etc. to create test system
  - Make change: upgrade db and/or OS, change storage, migrate platforms, etc.
- Pre-processing transforms captured data into replayable format
- Once pre-processed, workload can be replayed many times



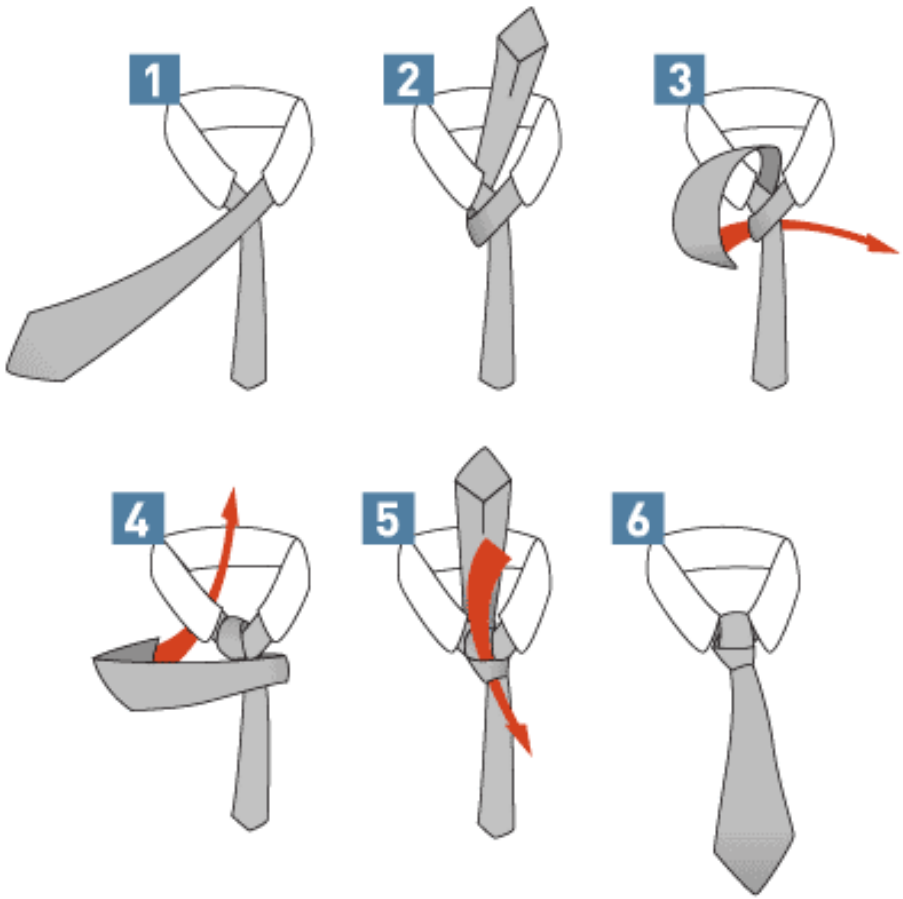
Step 3: Replay Workload

- Replays workload preserving timing, concurrency and dependencies of the captured system
- Replay Client is a special program that consumes processed workload and sends requests to the replay system
- For high concurrency workloads, it may be necessary to start multiple clients



Database Replay

- Step-By-Step

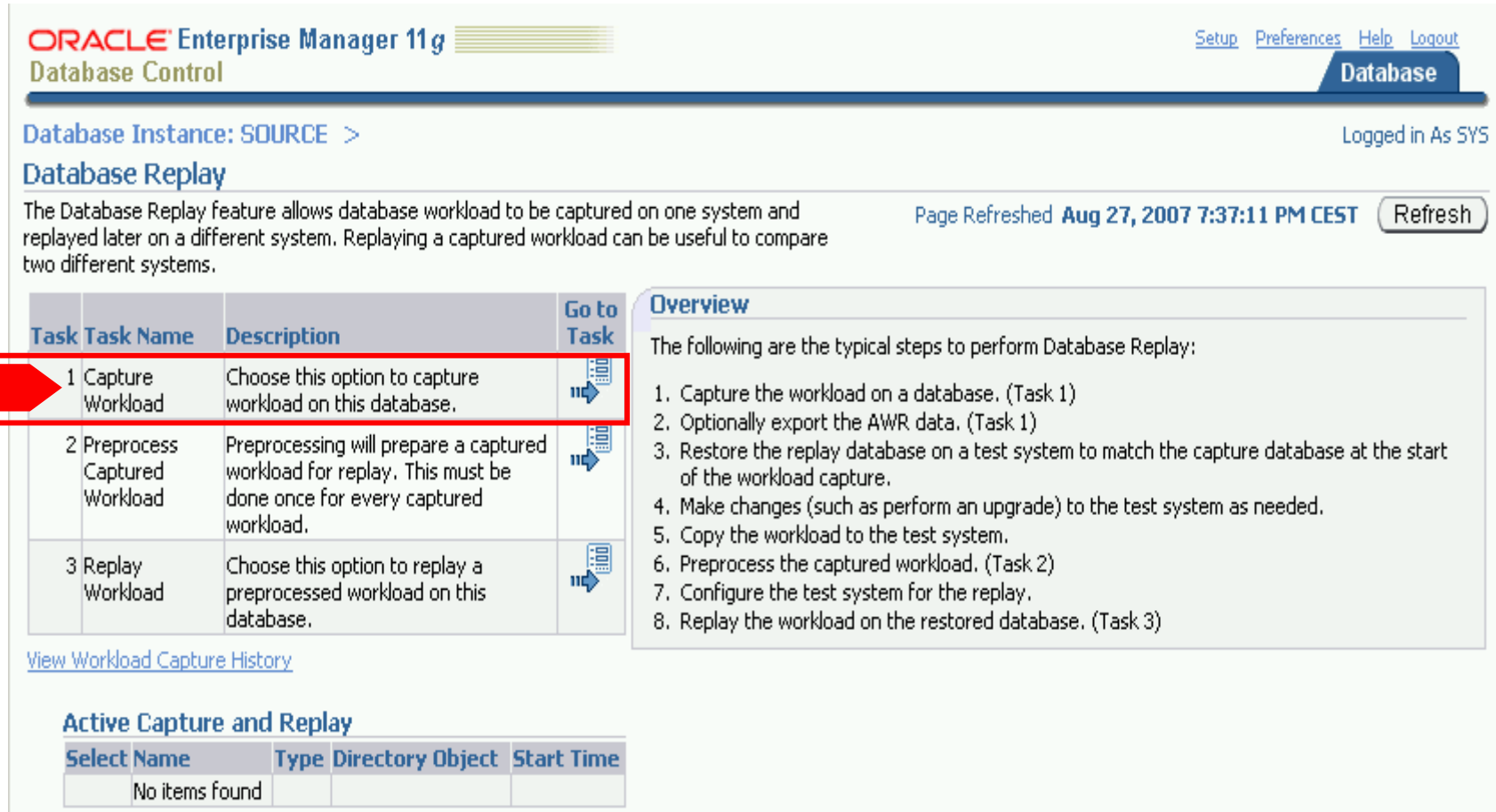


Database Replay

1. Create a copy of the database to replay the workload
   - RMAN Duplicate
   - Snapshot Standby

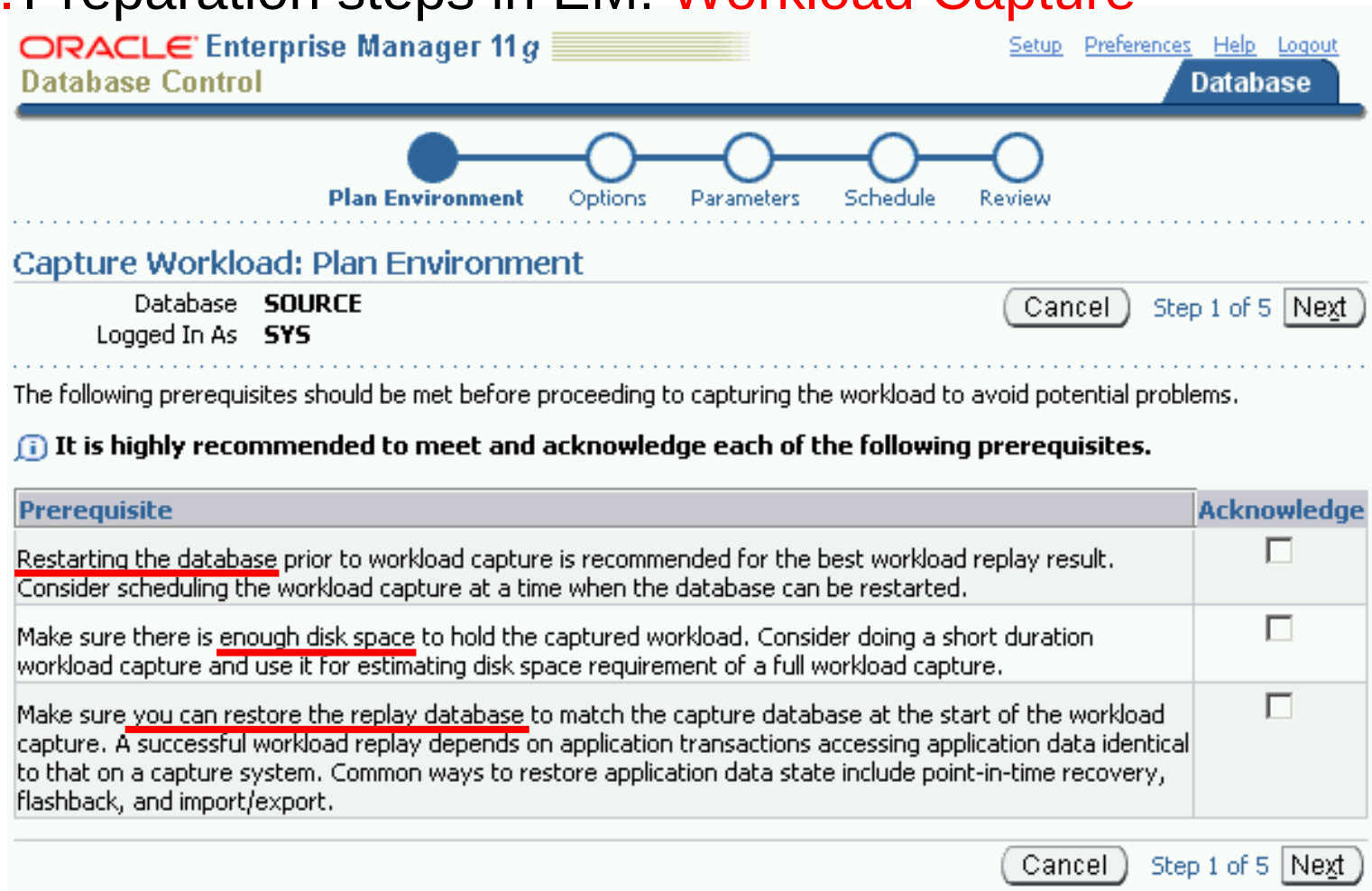
Database Replay

2. Workflow in EM: Workload Capture



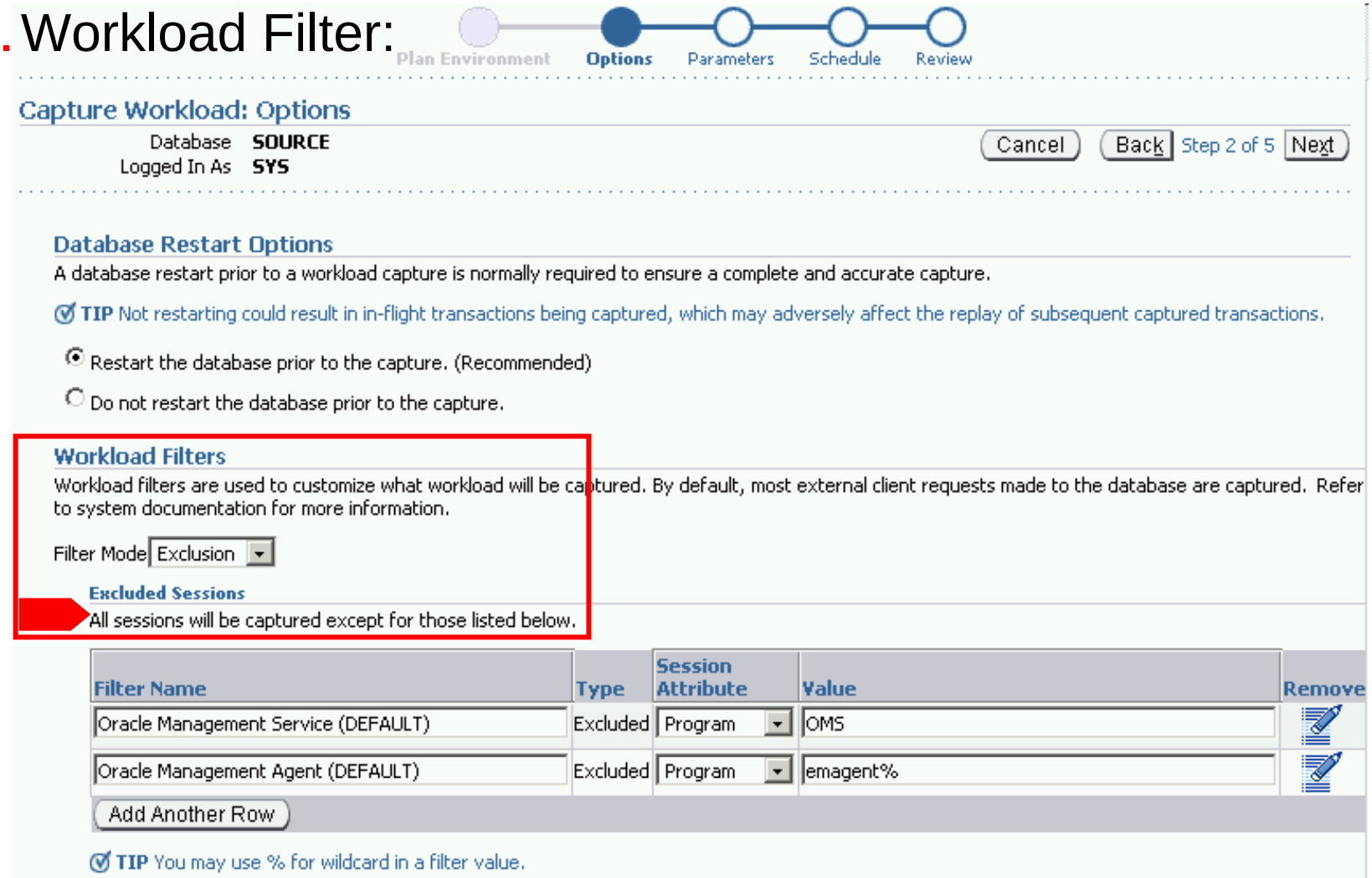
Database Replay

2. Preparation steps in EM: Workload Capture



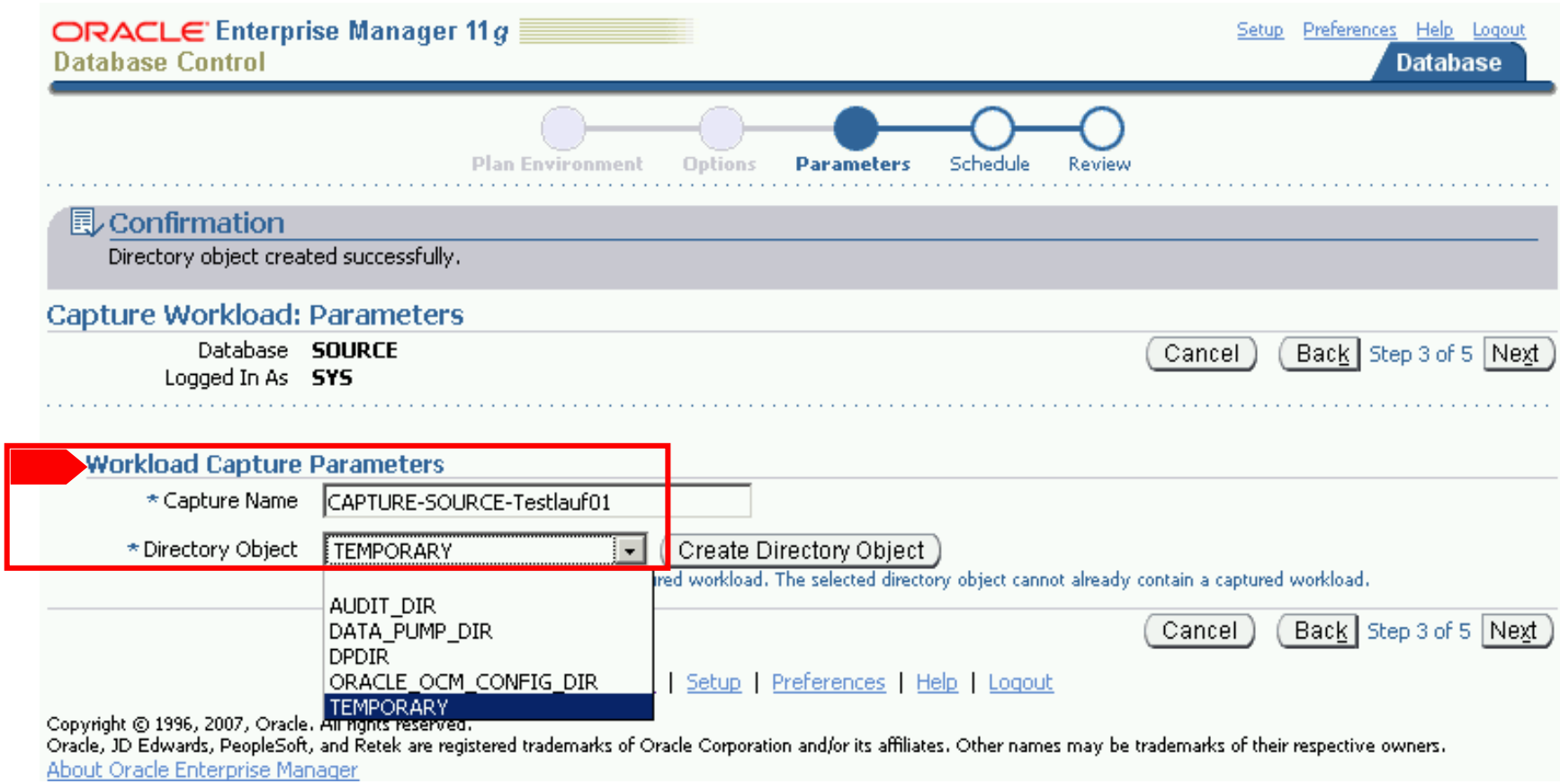
Database Replay

3. Workload Filter:



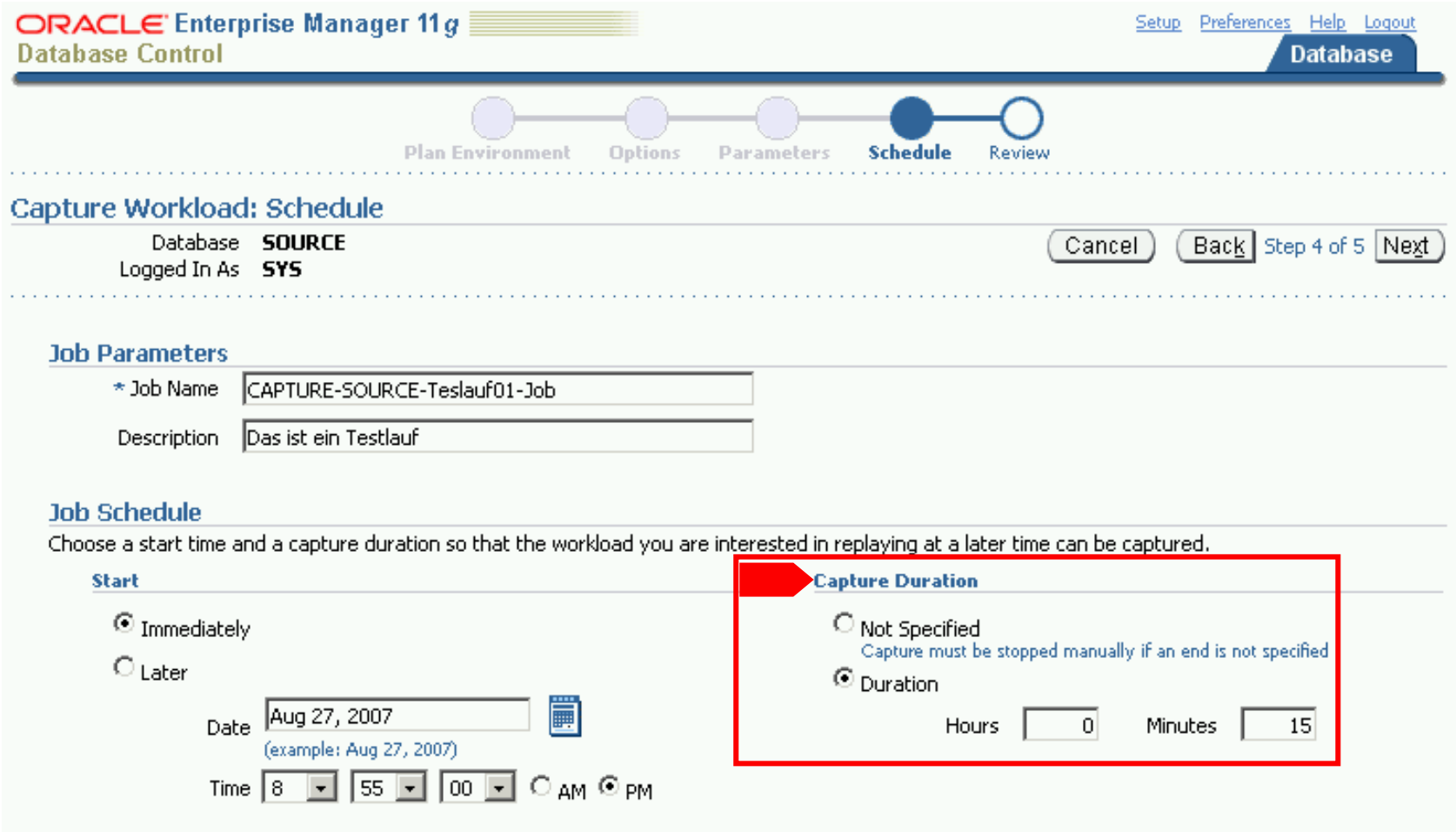
Database Replay

4. Prepare an empty directory for capture files



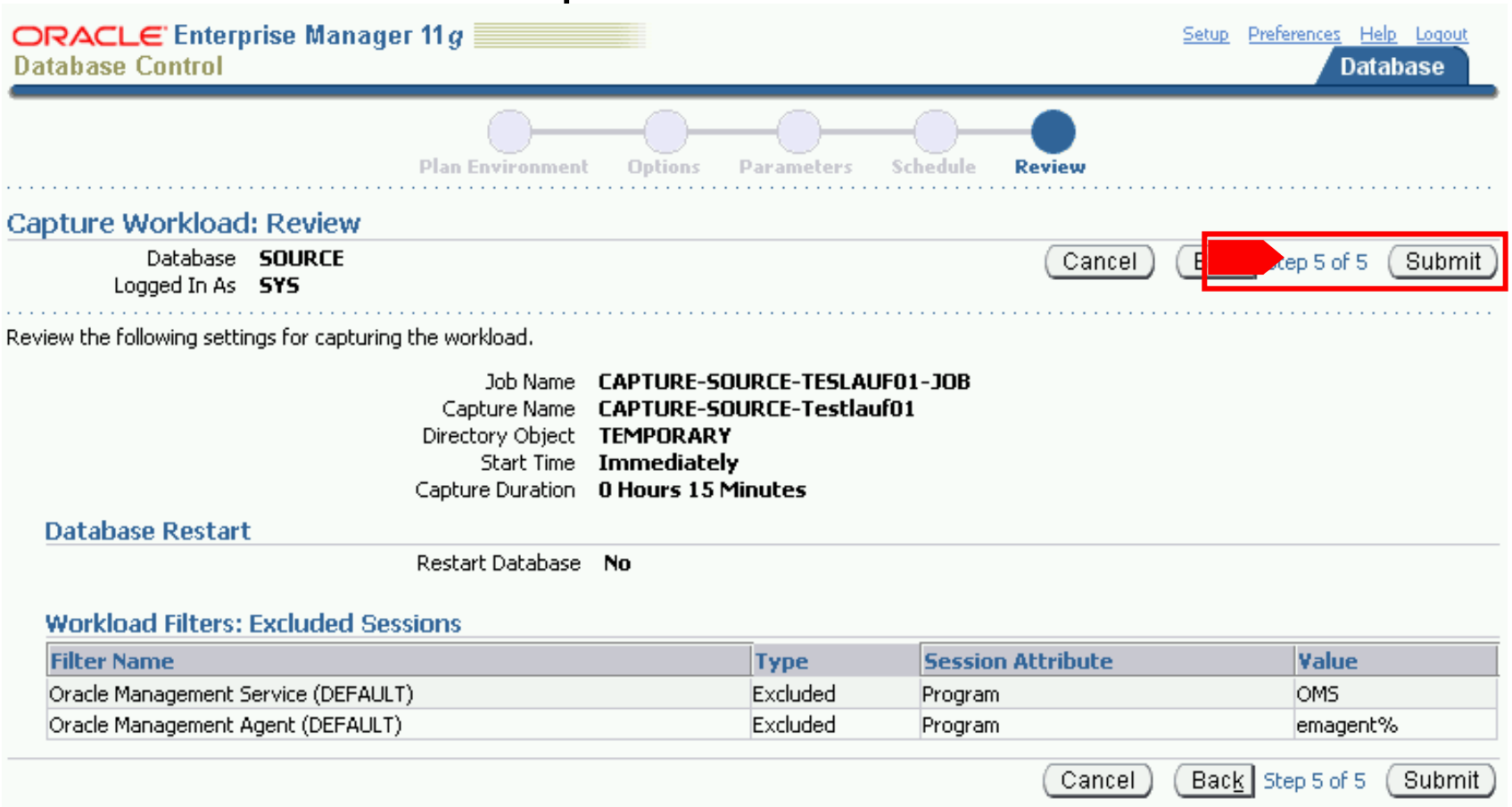
Database Replay

- Job scheduling - specify time frame for capture:



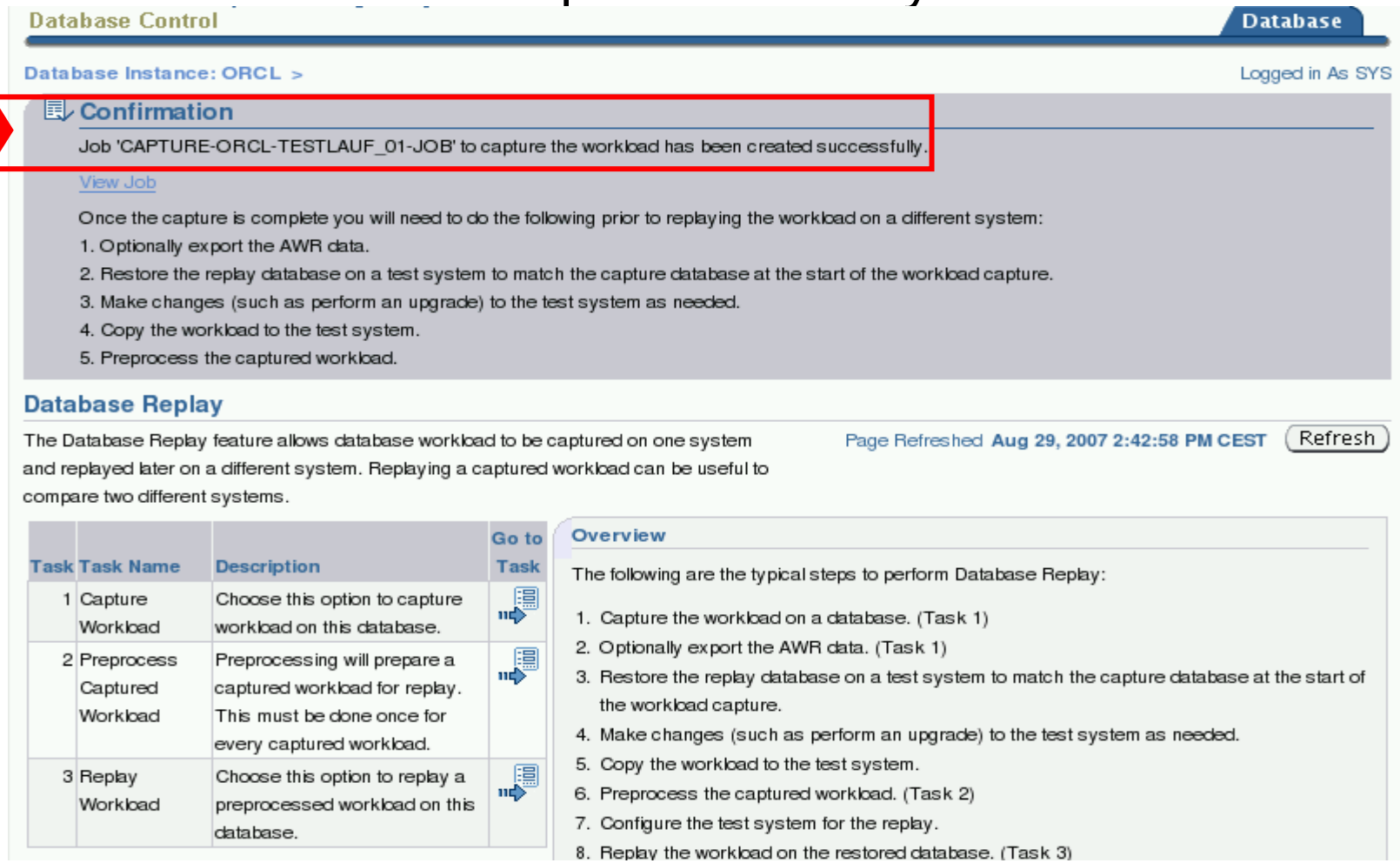
Database Replay

- Database Control - capture review and submit:



Database Replay

- Database Control - capture summary:



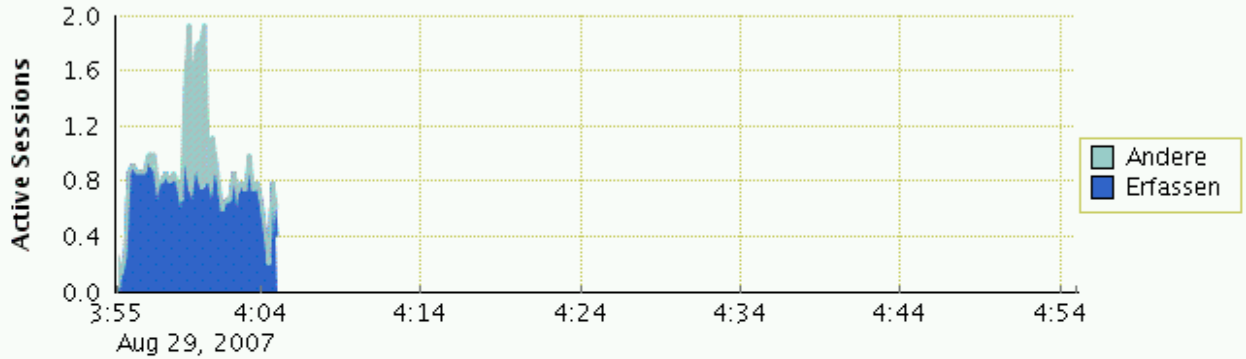
Database Replay

- Database Control - capture statistics:

Workload Profile | Workload Filters

View Workload Capture Report

Average Active Sessions

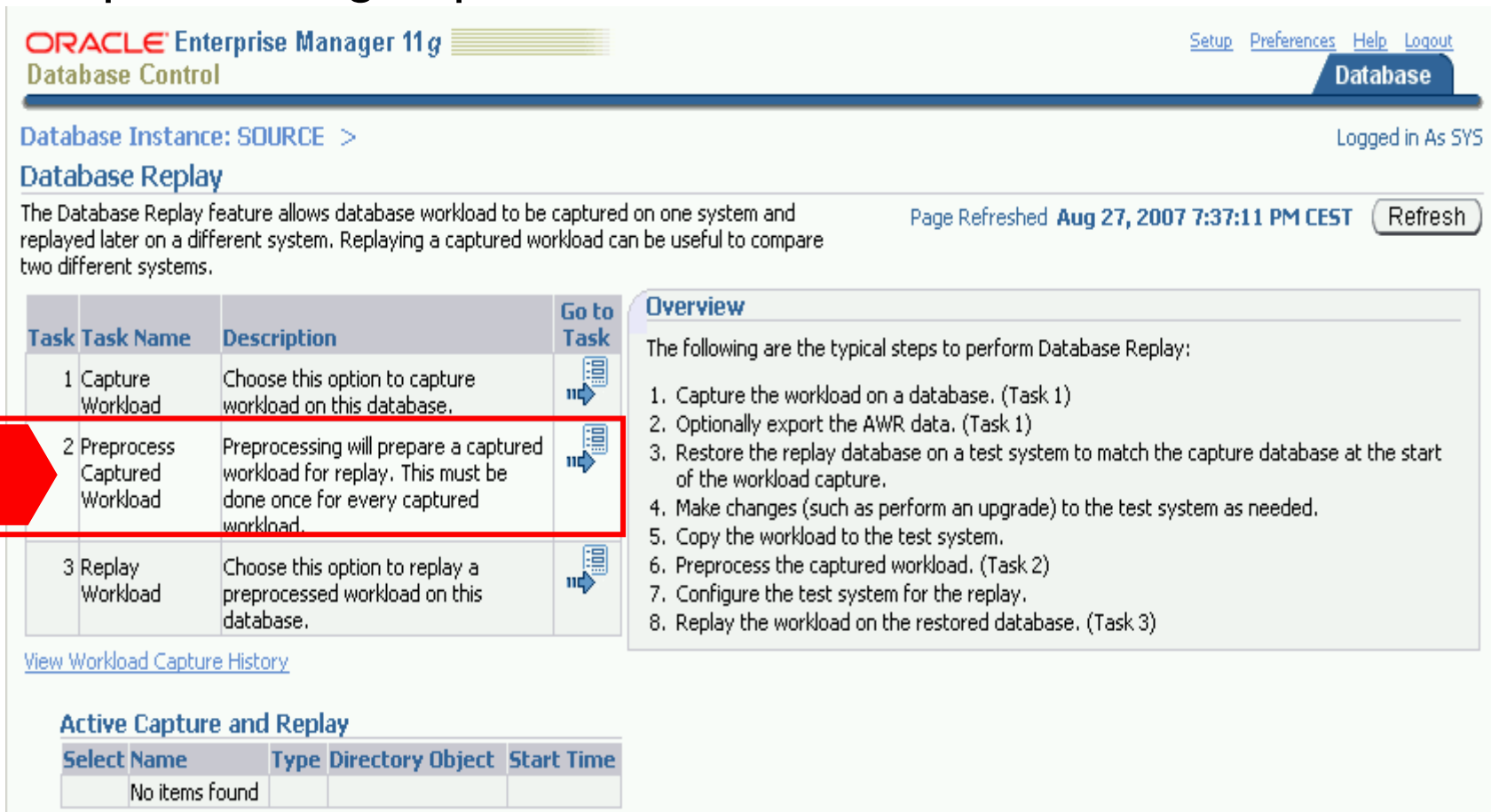


Comparison

	Total	Capture	Percentage of Total
Database Time (hh:mm:ss)	00:11:19	00:07:59	70.54
Average Active Sessions	1.13	0.80	70.54
User Calls	109,251	102,625	93.94
Transactions	289	54	18.69
Connects	37	12	32.43
Application Errors	N/A	51,100	N/A

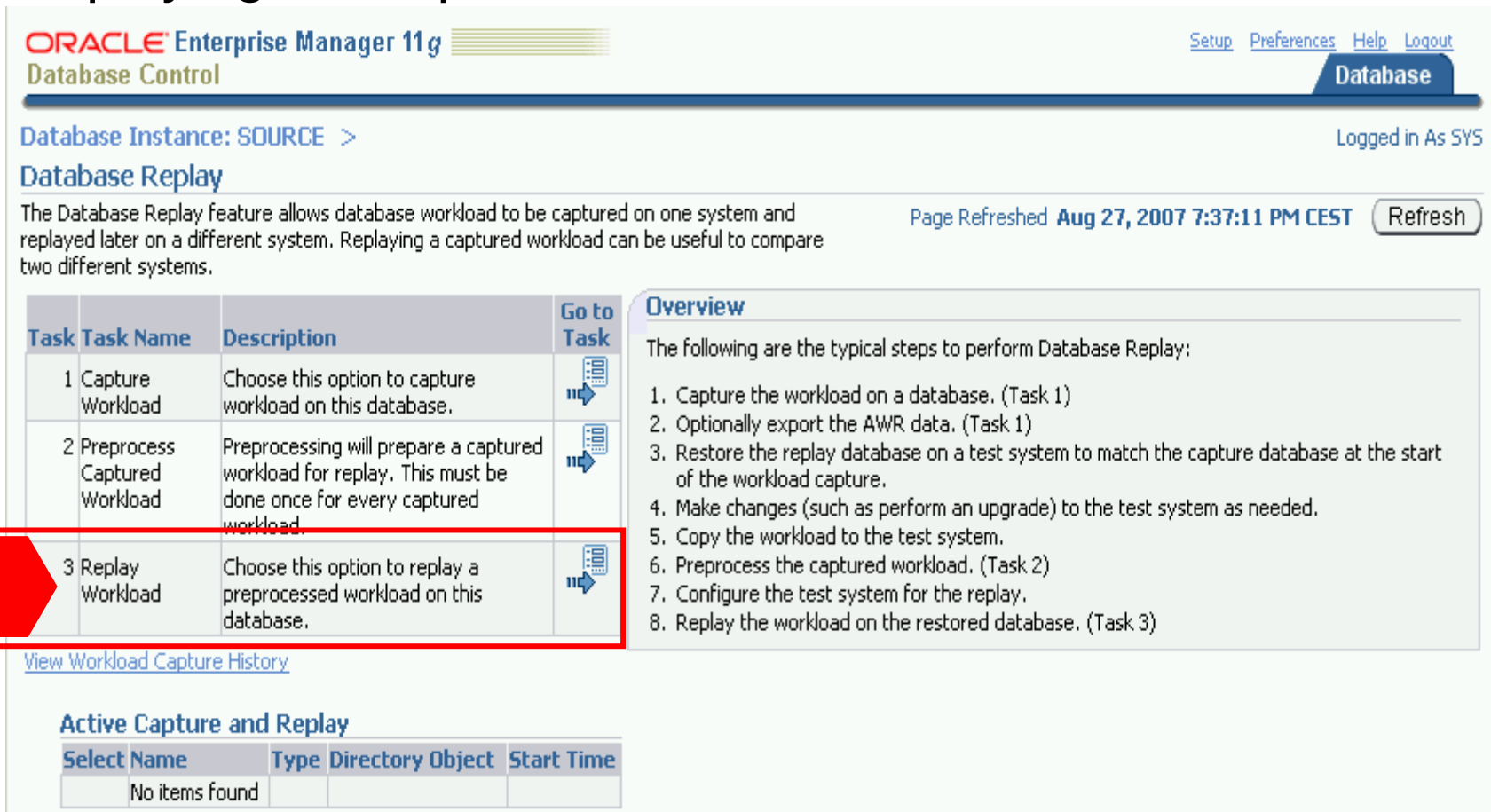
Database Replay

- Preprocessing captured workload:



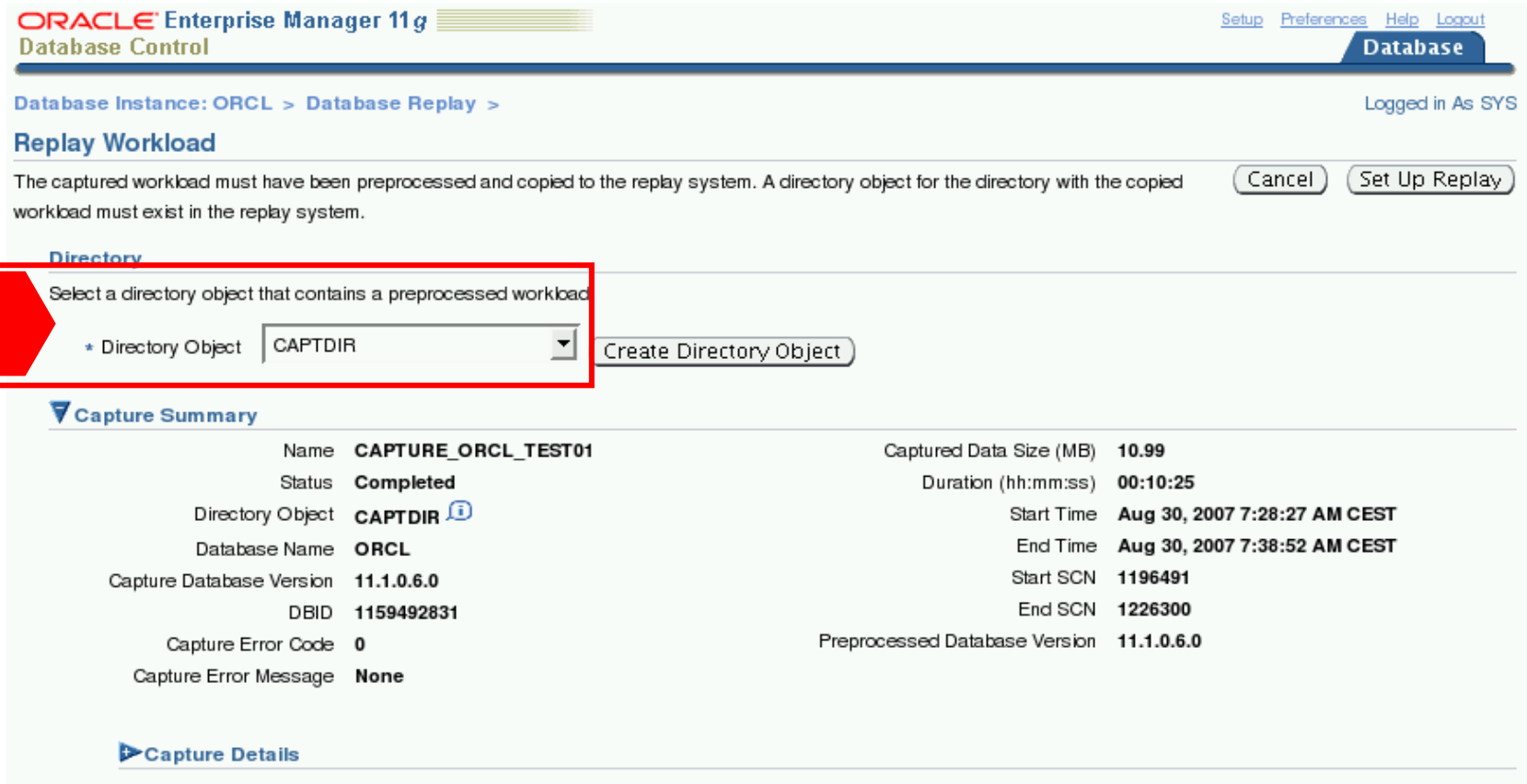
Database Replay

- Replaying the captured workload:



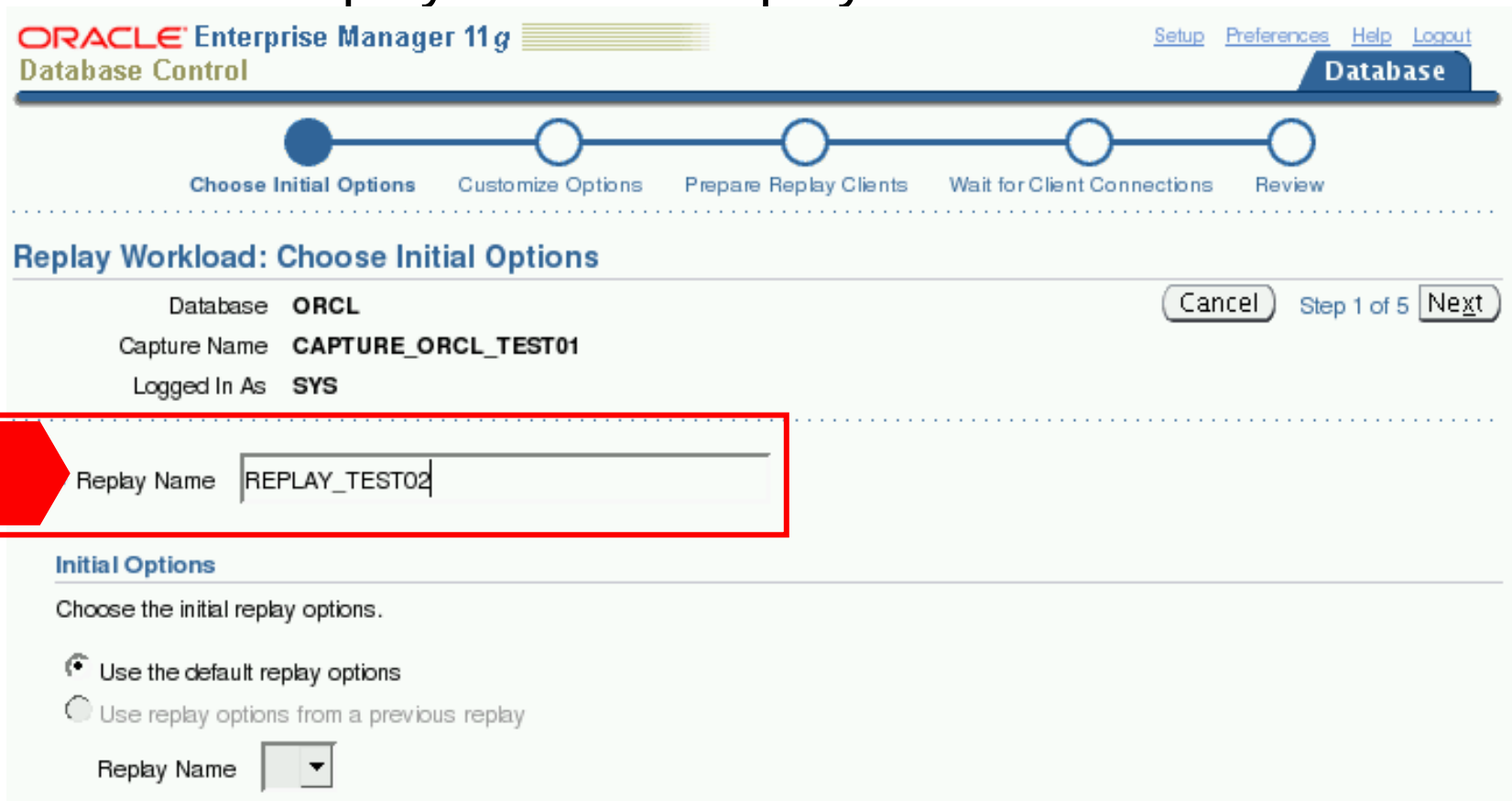
Database Replay

- Choose directory containing preprocessed capture files



Database Replay

- Workload Replay - choose replay name:



Database Replay

- Workload Replay: Wait for replay clients to connect

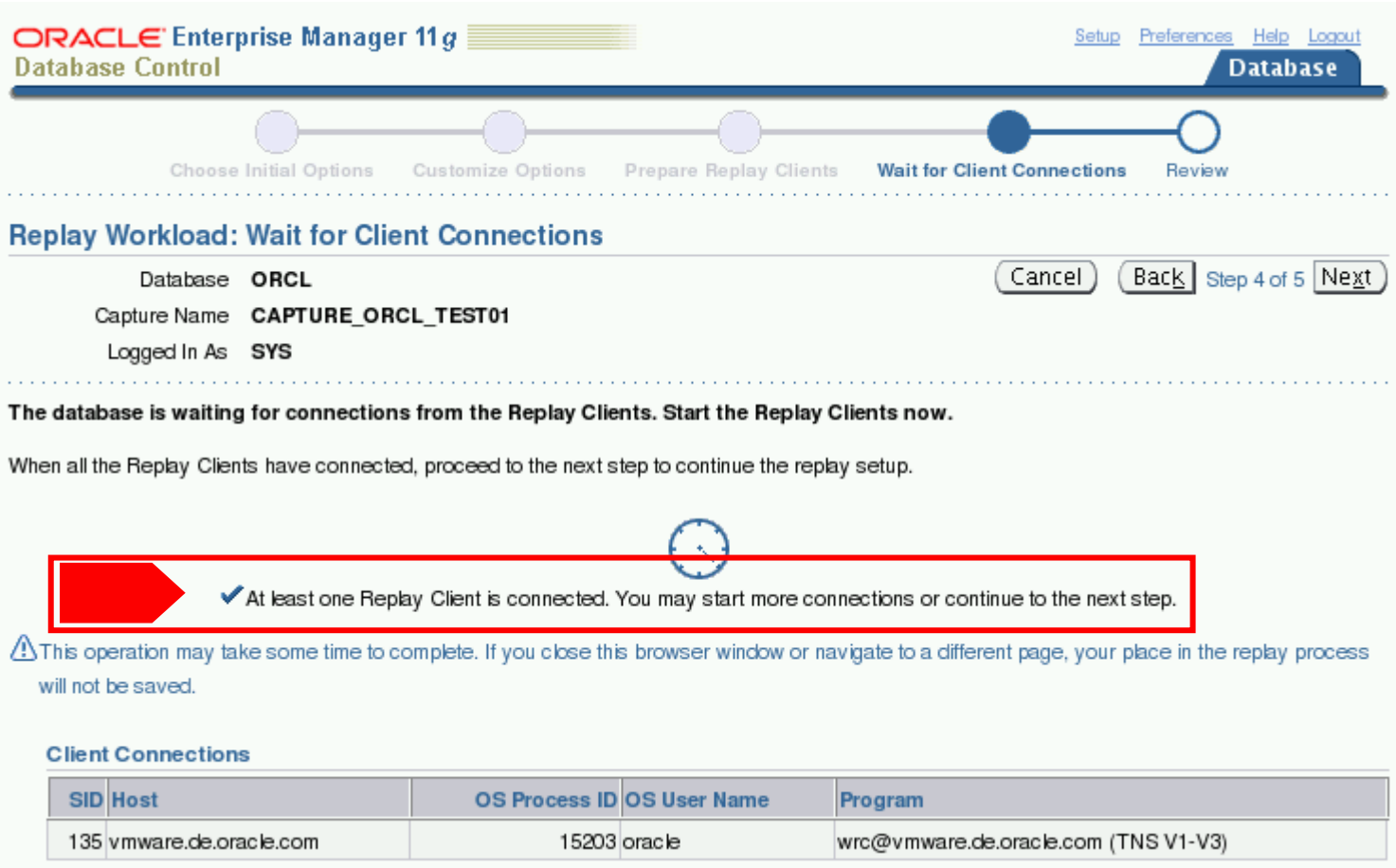
Database Replay

- Workload Replay Clients: Calibrate
  - **`$> wrc mode=calibrate replaydir=/tmp/testing`**
- Workload Replay Clients: Start
  - **`$> wrc system/oracle mode=replay`**

```
$ wrc system/oracle@orcl mode=replay

Workload Replay Client: Release 11.1.0.6.0 - Production on Thu Aug 30 09:02:10 2
007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.


Wait for the replay to start (09:02:10)
```

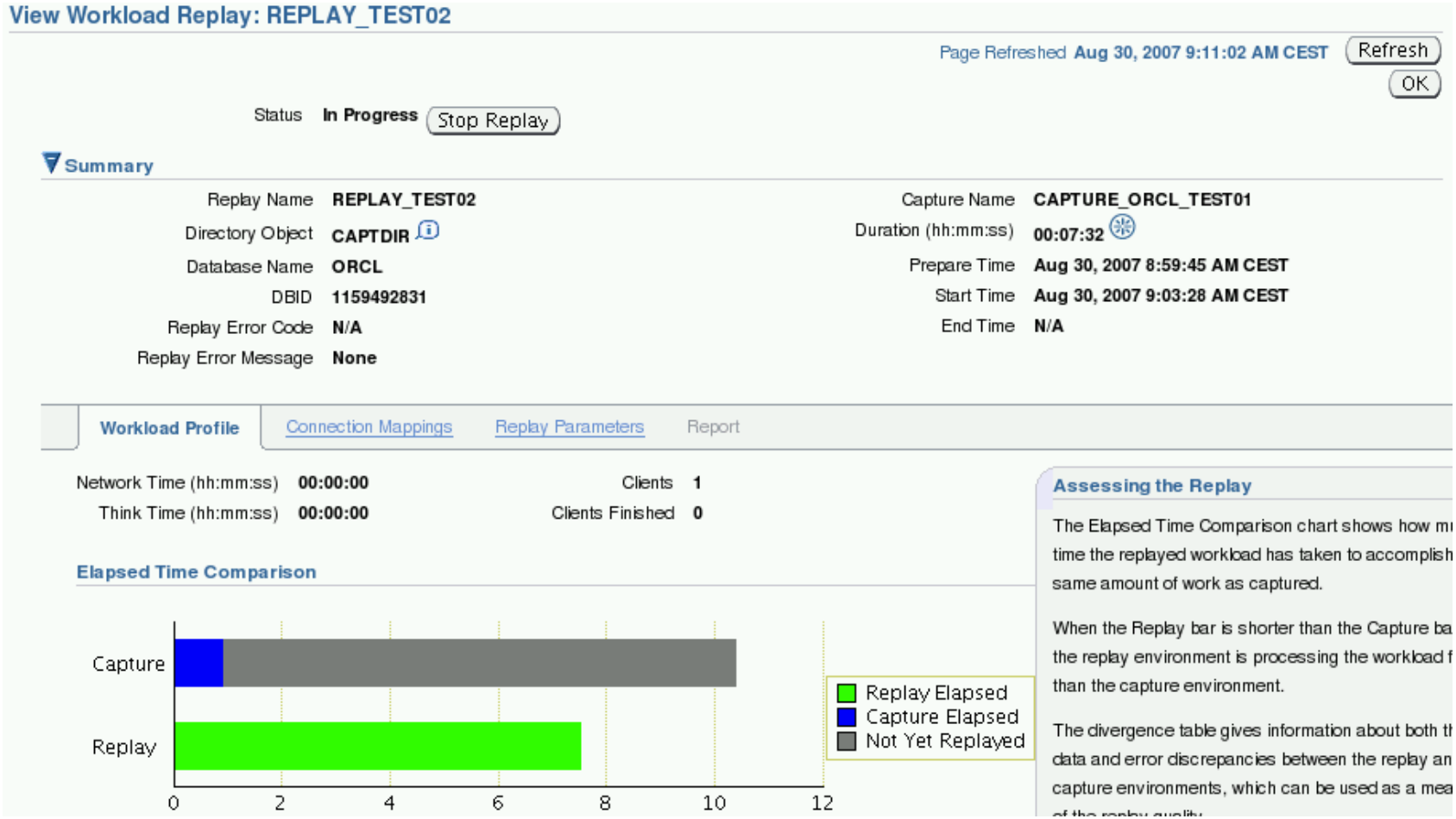
Database Replay

- Workload Replay: WRC client(s) connected



Database Replay

- Monitor Workload Replay during replay



Database Replay

- Monitor divergence stats:

Divergence

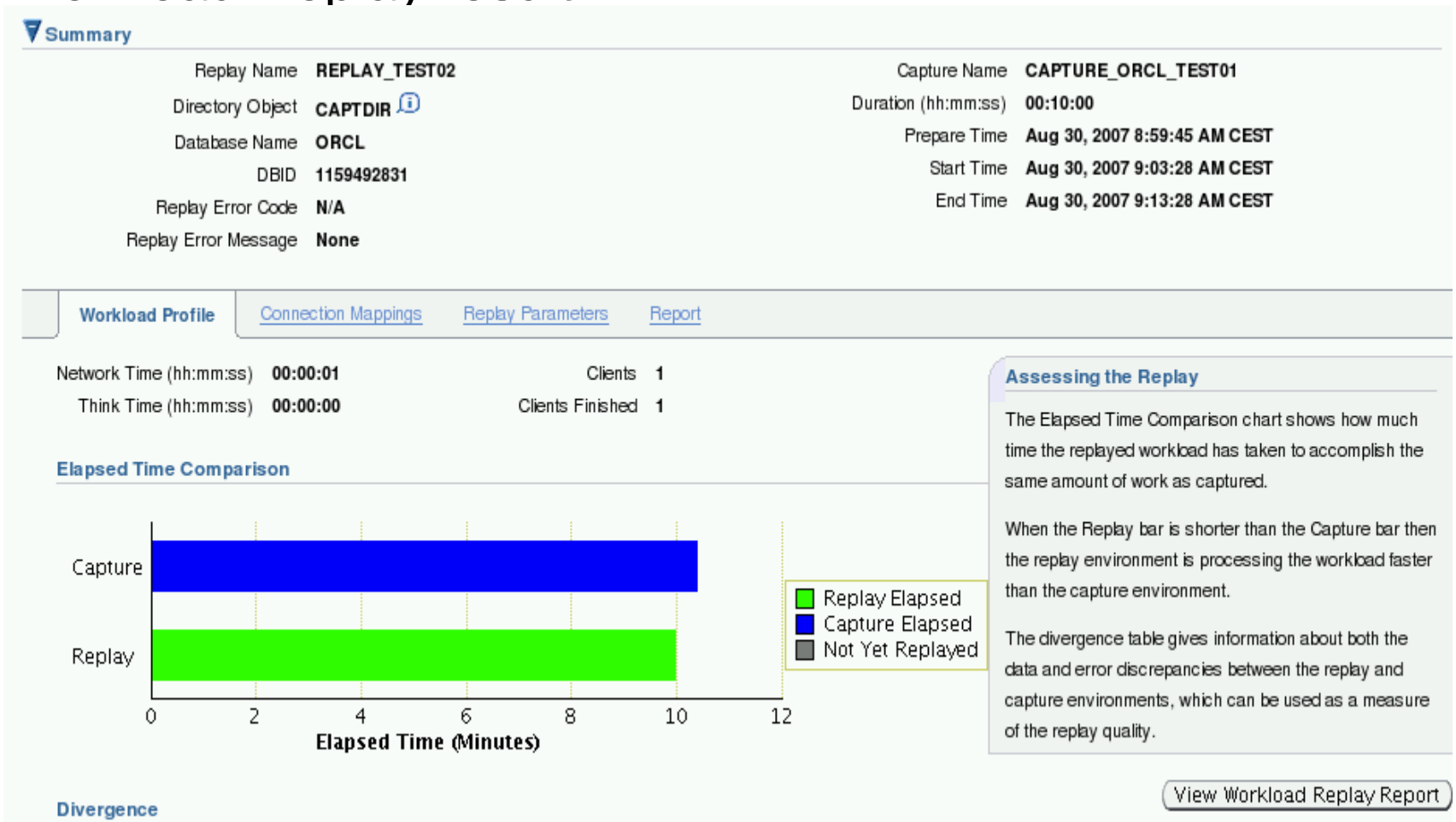
	Number of Calls	Percentage of Total Calls
Error Divergence:		
Session Failures Seen During Replay	0	0.00
Errors No Longer Seen During Replay	0	0.00
Errors Mutated During Replay	0	0.00
New Errors Seen During Replay	0	0.00
Data Divergence:		
DMLs with Different Number of Rows Modified	0	0.00
SELECTs with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

Detailed Comparison

	Capture	Replay	Percentage of Capture
Duration (hh:mm:ss)	00:10:25	00:10:00	96.00
Database Time (hh:mm:ss)	00:17:06	00:15:22	89.86
Average Active Sessions	1.64	1.54	93.61
User Calls	18,141	18,133	99.96

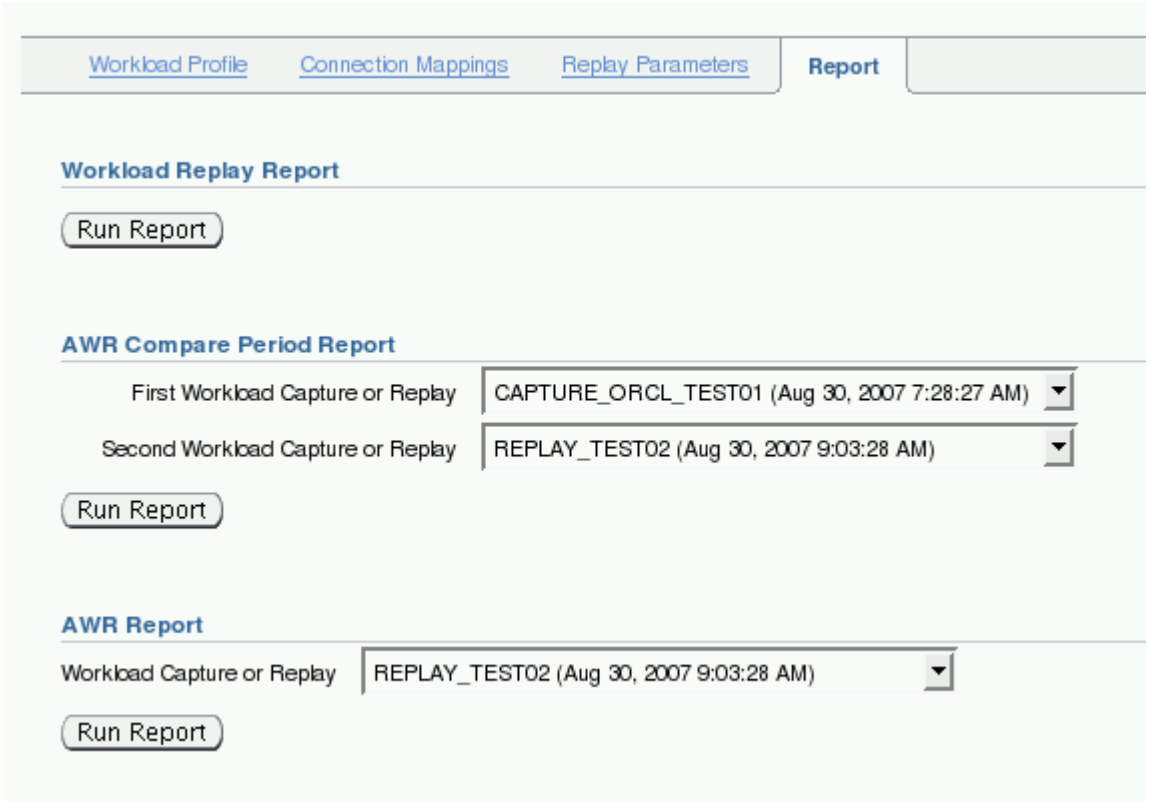
Database Replay

- Workload Replay result:



Database Replay

- Workload replay reporting:



Additional Pieces of Info

Important Upgrade Notes at a glance

Upgrade to Oracle 10*gR2*

Note: 466181.1 Upgrade Companion 10gR2
Note: 316889.1 Complete Checklist for Manual Upgrade to 10gR2
Note: 555579.1 Known Issues 10.2.0.4

Upgrade to Oracle 11*g*

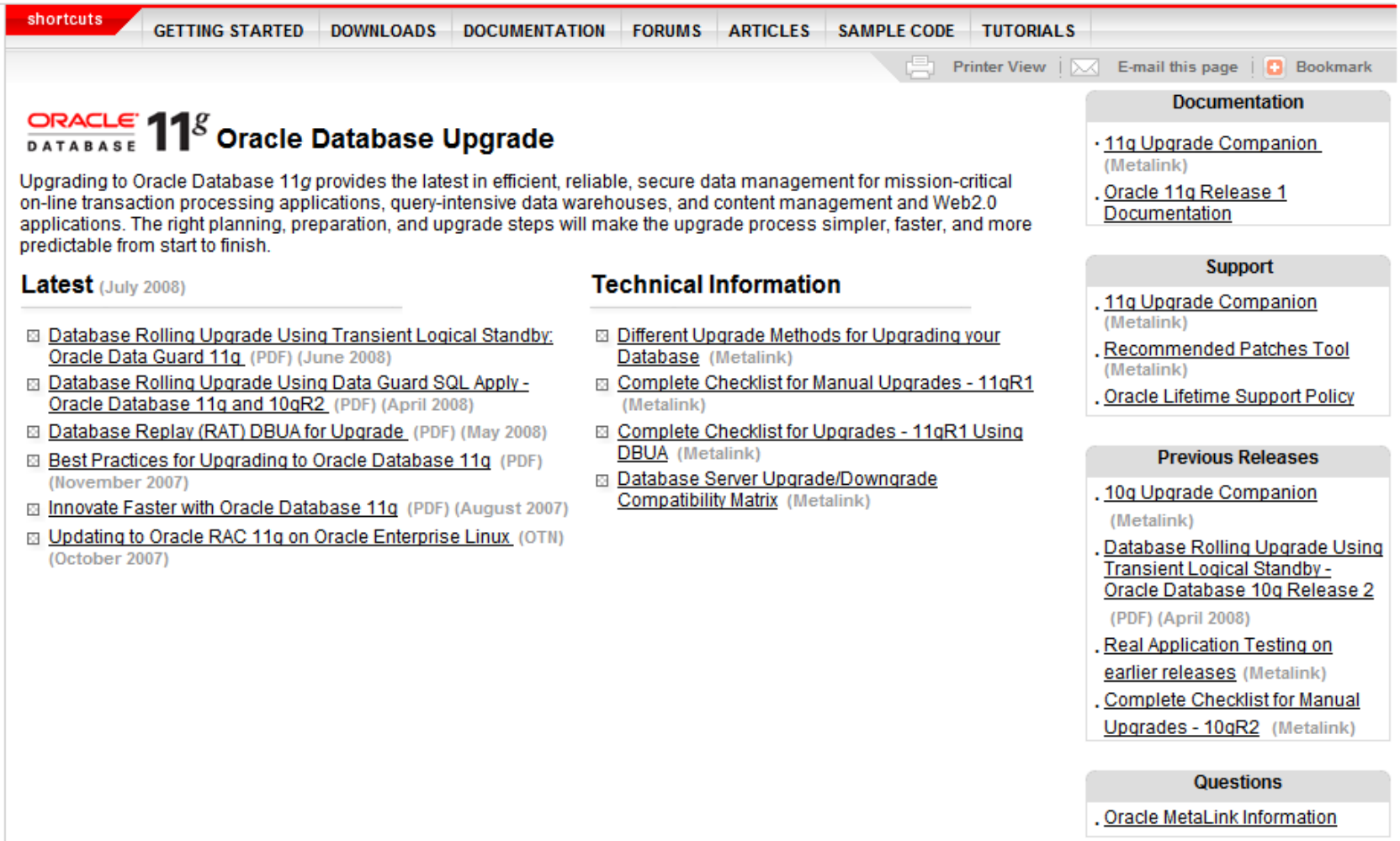
Note: 601807.1 Upgrade Companion 11g
Note: 429825.1 Complete Checklist for Manual Upgrades to 11g
Note: 454506.1 Known Issues and Alerts 11.1.0.6

Performance Testing

Note: 560977.1 Real Application Testing available for earlier releases
Note: 562899.1 Using SQL Performance Analyzer for upgd. 9.2 to 10.2
Note: 394937.1 Statspack Guide
Note: 295819.1 Potential Query Tuning Related Issues
Note: 345048.1 GROUP BY does not sort if you don't use ORDER BY
Upgrading to Oracle Database 10g: What to expect from the Optimizer

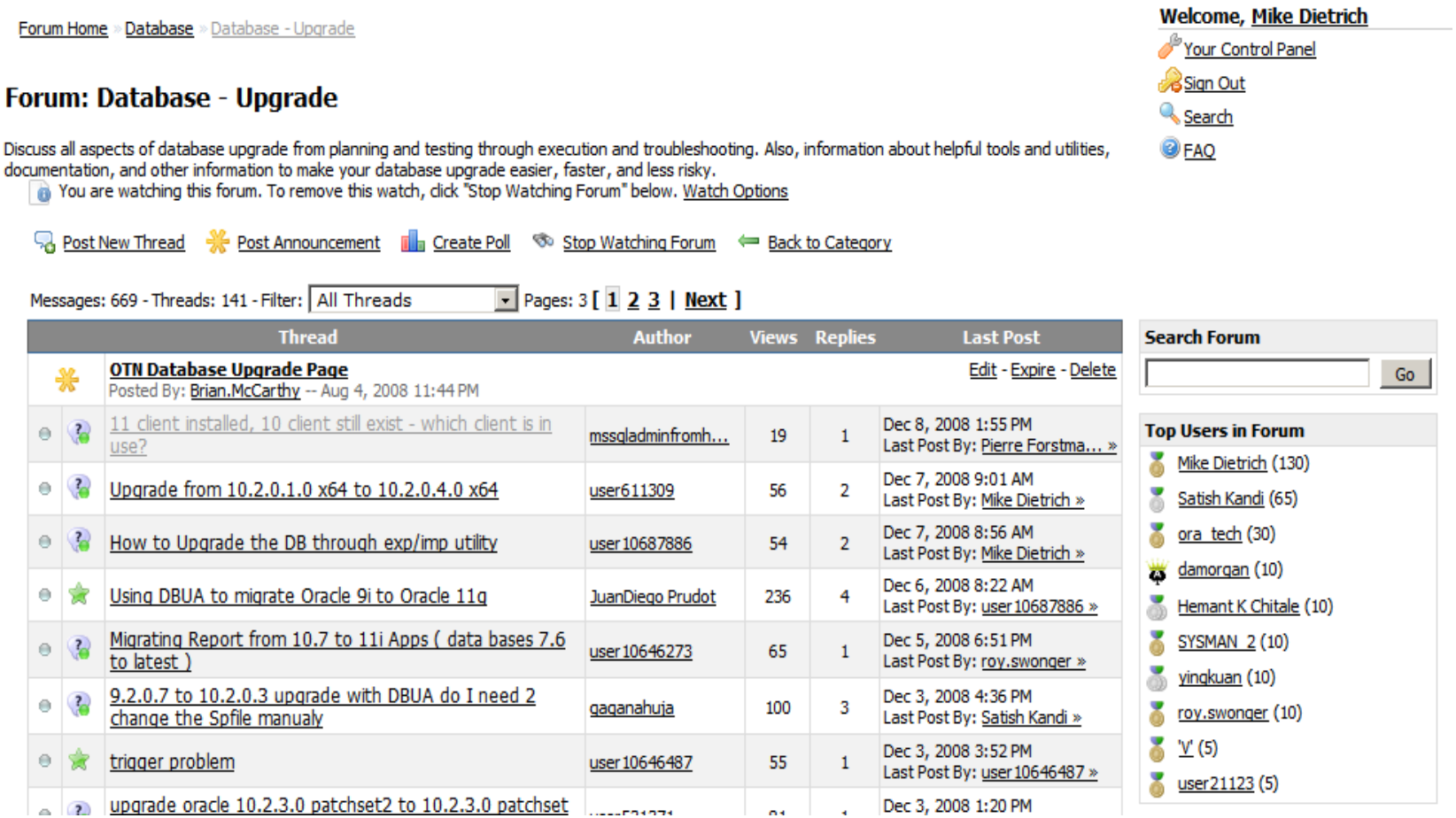
Upgrade Webpage on OTN

- http://www.oracle.com/technology/products/database/oracle11g/upgrade/index.html



OTN Upgrade Forum

- http://forums.oracle.com/forums/forum.jspa?forumID=583&start=0



Forum Home » Database » Database - Upgrade

Welcome, Mike Dietrich
Your Control Panel
Sign Out
Search
FAQ

Forum: Database - Upgrade

Discuss all aspects of database upgrade from planning and testing through execution and troubleshooting. Also, information about helpful tools and utilities, documentation, and other information to make your database upgrade easier, faster, and less risky.

You are watching this forum. To remove this watch, click "Stop Watching Forum" below. Watch Options

Post New Thread | Post Announcement | Create Poll | Stop Watching Forum | Back to Category

Messages: 669 - Threads: 141 - Filter: All Threads Pages: 3 [1 2 3 | Next]

Thread	Author	Views	Replies	Last Post
OTN Database Upgrade Page Posted By: Brian.McCarthy -- Aug 4, 2008 11:44 PM				Edit - Expire - Delete
11 client installed, 10 client still exist - which client is in use?	mssqladminfromh...	19	1	Dec 8, 2008 1:55 PM Last Post By: Pierre Forstma... »
Upgrade from 10.2.0.1.0 x64 to 10.2.0.4.0 x64	user611309	56	2	Dec 7, 2008 9:01 AM Last Post By: Mike Dietrich »
How to Upgrade the DB through exp/imp utility	user10687886	54	2	Dec 7, 2008 8:56 AM Last Post By: Mike Dietrich »
Using DBUA to migrate Oracle 9i to Oracle 11g	JuanDiego Prudot	236	4	Dec 6, 2008 8:22 AM Last Post By: user10687886 »
Migrating Report from 10.7 to 11i Apps (data bases 7.6 to latest)	user10646273	65	1	Dec 5, 2008 6:51 PM Last Post By: roy.swonger »
9.2.0.7 to 10.2.0.3 upgrade with DBUA do I need 2 change the Spfile manualy	gaganahuja	100	3	Dec 3, 2008 4:36 PM Last Post By: Satish Kandi »
trigger problem	user10646487	55	1	Dec 3, 2008 3:52 PM Last Post By: user10646487 »
upgrade oracle 10.2.3.0 patchset2 to 10.2.3.0 patchset				Dec 3, 2008 1:20 PM

Search Forum
Go

Top Users in Forum
Mike Dietrich (130)
Satish Kandi (65)
ora_tech (30)
damorgan (10)
Hemant K Chitale (10)
SYSMAN_2 (10)
yingkuan (10)
roy.swonger (10)
'V' (5)
user21123 (5)

Latest Patchset 11.1.0.7.0

Patch Number 6890831

Available for all major platforms

Upgrade Information / Alerts

- Note 738538.1 Known Issues specific to the 11.1.0.7 Patch Set

11.1.0.7 Alerts / Issues

This section lists alerts and important issues relevant to 11.1.0.7 .

General Alerts / Issues

Note:756435.1*	Block change tracking on physical standby can cause incorrect backups
Note:731599.1P*	Linux: Random node reboots in RAC induced by OCSSD

Upgrade Issues

Note:742647.1*	ASM rolling upgrade to 11.1.0.7 fails in RAC environments

Issues introduced in 11.1.0.7

This section lists bugs **introduced** in 11.1.0.7 (if any). Such issues may be either serious or trivial but the aim is to list them all to help customers assess the risk of applying the Patch Set on top of 11.1.0.6

7272646P	Linux-x86_64: ORA-27103 on startup when MEMORY_TARGET > 3g
7213937	Various OERI using Resource Manager
7211965	Different default cost for BETWEEN
7113299	Dump [kkslmtl] with fix for bug 6163785

Real World Experience

- White Paper available on OTN
  - http://www.oracle.com/technology/products/database/oracle11g/upgrade/presentations/9i_to_11g_real_world_customer_experience.pdf


Upgrading from Oracle 9i to Oracle Database 11g:
A Real World Customer Experience

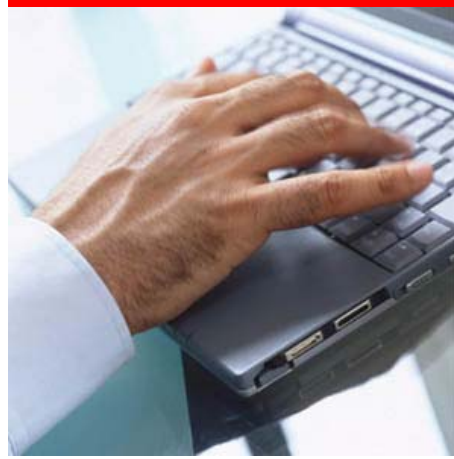
An Oracle White Paper
October 2008

Guidelines

- Never ever change too many system components at once!
- Document all changes detailed and clearly into a change log!
- Always (!!!) use real world data for testing!
- Don't underestimate the test efforts. Reserve enough time and resources for testing.
- ALWAYS collect as much performance data BEFORE the upgrade will be started!!
- Create a fallback strategy!
- PLEASE test your fallback strategy - does it really work??

- Please remember:
  Upgrade has never been easier - but you still have to test!!!

Agenda



- Recap
- Challenges & Best Practices
- AWR & STATSPACK
- SQL Plan Management
- Real Application Testing
- Q&A


ORACLE®