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

Welcome to the Oracle Upgrade Companion. This Upgrade Companion helps you upgrade from Oracle9i Release 2 (9.2) to Oracle Database 10g Release 2 (10.2), including guidance on pre-upgrade, upgrade, and post-upgrade steps. This web site is instructional only, and does not supply automation tools.

**NOTE:** This document is not a replacement for the Oracle Database 10g Upgrade Guide. It should be used as a companion document to the Oracle Database documentation.

This document is continually updated as new information becomes available. Please check back prior to your upgrade.

If advice or onsite assistance is needed during your upgrade, go to Oracle's Upgrade Management Services page for details. Oracle Upgrade Management Services is designed to simplify your unique upgrade challenges, no matter what stage you are at in the upgrade process.

The Upgrade Companion does not cover upgrading Oracle E-Business Suite or Siebel Business Applications. Please refer to your application documentation. For convenience, some common E-Business Suite and Siebel Business Application MetaLink notes are listed below. This is not a complete list.

• **Note: 601807.1** Upgrade Companion 11g

Welcome to the Oracle 11g Upgrade Companion. This Upgrade Companion helps you upgrade from Oracle9i Release 2 (9.2) or Oracle Database 10g to Oracle Database 11g, including guidance on pre-upgrade, upgrade, and post-upgrade steps. This document is instructional only, and does not supply automation tools.

**NOTE:** This document is not a replacement for the Oracle Database 11g Upgrade Guide. It should be used as a companion document to the Oracle Database documentation.

This document is continually updated as new information becomes available. Please check back prior to your upgrade.

If advice or onsite assistance is needed during your upgrade, go to Oracle's Upgrade Management Services page for details. Oracle Upgrade Management Services is designed to simplify your unique upgrade challenges, no matter what stage you are at in the upgrade process.

The Upgrade Companion does not cover upgrading Oracle E-Business Suite or Siebel Business Applications. Please refer to your application documentation. For convenience, some common E-Business Suite MetaLink notes are listed below. This is not a complete list.
Defining, implementing and managing a contingency plan is extremely important step during the upgrade process. Mission critical enterprises require a return to normal operations more quickly today than ever before. Accordingly, system availability is dependent on how well you prepare for outages. Planning and practicing for the unexpected issues helps to ensure the upgrade to the new Oracle Database 11g will be successful.

**Prepare**

Preparing the database before the upgrade begins will help reduce any unforeseen errors or circumstances that prevent the upgrade from completing, such as out of space errors. There are a number of things to consider before the upgrade begins, including fallback planning, compatible parameter setting considerations and database configuration settings.

- Execute the Pre-Upgrade Information Tool
- COMPATIBLE Initialization Parameter
- Review Non-Default Initialization Parameters
- When to Fallback
- Recommendations to Avoid Common Pitfalls
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

---

[Documentation](#)  
- Resolving High CPU Usage on Oracle Servers
- 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 10045 Trace for Tuning
- How to Identify Resource Intense 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
- DBA
- LTOU

[White Papers](#)  
- Best Practices for Load Testing System Updates
- Determining CPU Resource Usage for Linux and Unix
- Measuring Memory Resource Usage for Linux and Unix
Upgrade Paths

UPGRADE

OS Change?

Y

N

Downtime >45min?

N

Y

Export/Import

CTAS, COPY

Oracle Streams

Transportable Tablespaces

Transportable Database

SQL Apply

DBUA

ORACLE recommended

SQL> @catupgrd

CLI

ORACLE
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
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 `/`:

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

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>WERT</th>
<th>INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATUS</td>
<td></td>
<td>COMPLETED</td>
</tr>
<tr>
<td>DSTART</td>
<td>04-07-2008 12:30</td>
<td></td>
</tr>
<tr>
<td>DSTOP</td>
<td>05-07-2008 12:30</td>
<td></td>
</tr>
<tr>
<td>FLAGS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CPUSPEEDNW</td>
<td>1392.39</td>
<td></td>
</tr>
<tr>
<td>IOSEEKTIM</td>
<td>11.405</td>
<td></td>
</tr>
<tr>
<td>IOTFRSPEED</td>
<td>25595.605</td>
<td></td>
</tr>
</tbody>
</table>

...
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 10g: 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:

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

- Database performance tool - pre-10g
  - 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

How do I ensure plan stability???
Plan Stability Techniques

- Optimizer - prevent execution plan changes:
  - Classical approach:
    - Rule Based Optimizer (Please note: RBO desupport since Oracle 10g)
    - 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:

  ![Diagram showing plan acceptance](image1)

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

  ![Diagram showing plan rejection](image2)

  - Plan possibly not acceptable
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:

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)

Initial plan will be accepted if repeated and will be added to the SQL Plan Baseline

SQL MANAGEMENT BASE
Residing in SYSAUX TS. Will occupy max. 10% of SYSAUX. Weekly job will delete plans not used since 53 weeks [default].
SQL Plan Management

• Phase 2 - Selection
  • `OPTIMIZER_USE_SQL_PLAN_BASELINE=TRUE` (Default: `TRUE`)
  • Same statement parsed again but a different 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`

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

Equal or better plans can be added to the SQL Plan Baseline.

Worse plan will be kept in the Plan History and deleted automatically.

DBA schedules verification

Optimizer
SQL Plan Management

- Database Control:

<table>
<thead>
<tr>
<th>Storage</th>
<th>Database Configuration</th>
<th>Oracle Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Files</td>
<td>Memory Advisors</td>
<td>Jobs</td>
</tr>
<tr>
<td>Tablespace</td>
<td>Automatic Undo Management</td>
<td>Chains</td>
</tr>
<tr>
<td>Temporary Tablespace Groups</td>
<td>Initialization Parameters</td>
<td>Schedules</td>
</tr>
<tr>
<td>Datafile</td>
<td>Vw Database Feature Usage</td>
<td>Programs</td>
</tr>
<tr>
<td>Rollback Segments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redo Log Groups</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Archive Logs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Migrate to ASM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make Tablespace Locally Managed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics Management</th>
<th>Resource Manager</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Workload Repository</td>
<td>Getting Started</td>
<td>Users</td>
</tr>
<tr>
<td>AWR Baseline</td>
<td>Consumer Groups</td>
<td>Roles</td>
</tr>
<tr>
<td></td>
<td>Consumer Group Mappings</td>
<td>Profiles</td>
</tr>
<tr>
<td></td>
<td>Pprts</td>
<td>Audit Settings</td>
</tr>
<tr>
<td></td>
<td>Settings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Statistics</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Optimizer</th>
<th>Change Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Optimizer Statistics</td>
<td>Add Instance</td>
</tr>
<tr>
<td>SQL Plan Control</td>
<td>Delete Instance</td>
</tr>
</tbody>
</table>
SQL Plan Management

- DatabaseControl - Configuration:

Start with 5-14 weeks:
exec DBMS_SPM.CONFIGURE('plan_retention_weeks',5);
SQL Plan Management

- DatabaseControl - details:

```
SQL handle: SYS_SQL_fd1f08ed0ff7b3f98
SQL text: SELECT timezone_region FROM mgmt_targets WHERE target_name=:1 AND target_type=:2

Plan name: SYS_SQL_PLAN_ff7b3f985f59c06ad
Enabled: YES  Fixed: NO  Accepted: YES  Origin: AUTO-CAPTURE

Plan hash value: 2363372605

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost (%CPU)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td>1</td>
<td>54</td>
<td>1 (0)</td>
<td>00:00:01</td>
</tr>
<tr>
<td>1</td>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>MGMT_TARGETS</td>
<td>1</td>
<td>54</td>
<td>1 (0)</td>
<td>00:00:01</td>
</tr>
<tr>
<td>2</td>
<td>INDEX UNIQUE SCAN</td>
<td>MGMT_TARGETS_PK</td>
<td>1</td>
<td></td>
<td>0 (0)</td>
<td>00:00:01</td>
</tr>
</tbody>
</table>
```

Predicate Information (identified by operation id):

```
2 - eqsxs("TARGET_TYPE"=2 AND "TARGET_NAME"=1)
```
SQL Plan Management

• DatabaseControl - evolve:

Evolve SQL Plan Baselines

Plans that have not yet been accepted can be evolved (verified) to confirm they are suitable plan baselines.

<table>
<thead>
<tr>
<th>Name</th>
<th>SQL Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS_SQL_PLAN_6221c9c4380f747a</td>
<td>SELECT name, sql_text, SUBSTR(sql_text,1,50),categ...</td>
</tr>
</tbody>
</table>
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 10gR2

• 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 10gR2
SQL Plan Management - Upgrade 1

- Upgrade scenario 1

1. Repeatable plans will be added to the Plan Baseline upon 2nd execution.

2. Now: Different plans created with OFE=11 will be added to the Plan History for later verification.

- optimizer_features_enable=9.2.0
- optimizer_capture_sql_plan_baseline=true

- optimizer_features_enable=11.1.0
- optimizer_capture_sql_plan_baseline=false
SQL Plan Management - Upgrade 2

• Upgrade scenario 2

10.2 plans will be added to the SQL Plan Baseline

Every new better plan found in 11g will be added to the Plan History
SQL Plan Management - New Application

- New application (module) gets delivered

1. @Vendor
   - `DBMS_SPM.CREATE_STGTAB_BASELINE`
   - `DBMS_SPM.PACK_STGTAB_BASELINE`
   - Staging Table

2. @Customer
   - `DBMS_SPM.UNPACK_STGTAB_BASELINE`
   - Staging Table
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 REPOSITORY.

• 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:
    • Technical White Paper: SQL Performance Analyzer
  • Database Replay:
    • Technical White Paper: Database Replay
Real Application Testing: Tools of the Trade

<table>
<thead>
<tr>
<th>What is it?</th>
<th>SQL Performance Analyzer</th>
<th>Database Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicts SQL performance deviations before end-users can be impacted, helps assess impact of change on SQL response time</td>
<td>Replays real database workload on test system, helps assess impact of change on workload throughput</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How it works?</th>
<th>SQL Performance Analyzer</th>
<th>Database Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>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</td>
<td>Captures workloads and replays it with production characteristics including concurrency, synchronization &amp; dependencies</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>When to use?</th>
<th>SQL Performance Analyzer</th>
<th>Database Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit testing of SQL with the goal to identify the set of SQL statements with improved/regressed performance</td>
<td>Comprehensive testing of all sub-systems of the database server using real production workload</td>
<td></td>
</tr>
</tbody>
</table>

---

**ORACLE**
Real Application Testing

- Database Control:

<table>
<thead>
<tr>
<th>Database Instance: SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
</tr>
</tbody>
</table>

### Software

**Configuration**
- Collection Status
- Clone Oracle Home
- Host Configuration
- Oracle Home Inventory

**Database Software Patching**
- Patch Advisor
- View Patch Cache
- Patch Prerequisites
- Stage Patch
- Apply Patch

**Deployment Procedure Manager**
- Getting Started with Deployment Procedure Manager
- Deployment Procedures
- Procedure Completion Status
- Deployment and Provisioning Software Library

### Real Application Testing

- Database Replay
- SQL Performance Analyzer

### Support

- Support Workbench
Agenda

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

SQL Performance Analyzer
Database Replay
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

- **Production**
  - Client
  - Middle Tier
  - Oracle DB
  - Storage

- **Test**
  - Re-execute SQL
  - Make Changes / Tuning Regressions

- If adequate spare cycles available, optionally execute SQL here

* No middle & application tier setup required
SQL Performance Analyzer: Workflow

Steps

(1) Capture SQL (STS)
(2) Transport STS
(3) Execute SQL Pre-change
(4) Execute SQL Post-change
(5) Compare Perf.
(6) Reiterate
(7) Production Change / Tuning Deployment

(3) Make Change
(4) Done?

No

Yes

Tuned System
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: 10g to 11g

**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 11g 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)
    ```sql
    EXEC DBMS_SQLTUNE.CAPTURE_CURSOR_CACHE_SQLSET( . . )
    ```
  - Create STS from Top SQL in AWR / AWR Baseline
  - For 9i database enable SQL tracing:
    ```sql
    alter system set events '10046 trace name context forever, level 4';
    ```
Database Upgrade: 10g to 11g

- Step 1 in EM:

[SQL Tuning Sets]
A SQL Tuning Set is a collection of SQL Statements that can be used for tuning purposes.

[Create SQL Tuning Set: Load Methods]
Pick one of the load methods to collect and load SQL statements into the SQL tuning set.

- Incrementally capture active SQL statements over a period of time from the cursor cache.
  Specify the duration within which the SQL statements will be collected, and specify frequency over which the active SQL statements from the cursor cache will be collected repeatedly.

  - Duration 24 Hours
  - Frequency 6 Minutes

- Load SQL statements one time only
  Data Source: Cursor Cache

[Oracle]
Database Upgrade: 10g to 11g

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

- Step 3 in EM: Run SPA task

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

- **Optimizer Upgrade**
  - Test the effects of optimizer version changes on SQL Tuning Set performance.
- **Parameter Change**
  - Test and compare an initialization parameter change on SQL Tuning Set performance.
- **Guided Workflow**

**Optimizer Upgrade Simulation**
Test the effects of optimizer version changes on SQL Tuning Set performance.

- **Task Information**
  - **Task Name**: UPGRADE_10G11G
  - **SQL Tuning Set**: OOW_54
  - **Description**: Test upgrade to 11g
  - **Per-SQL Time Limit**: UNLIMITED

**Optimizer Versions**
Version 1: 10.2.0.2  Version 2: 11.1.0.6

**Evaluation**
Comparison Metric: Elapsed Time

**Schedule**
Time Zone: UTC
Immediately
Database Upgrade: 10g to 11g

- SPA results

**SQL Performance Analyzer Task Result: SYS.UPGRADE_10G11G**

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Task Owner</th>
<th>Task Description</th>
<th>SQL Tuning Set Name</th>
<th>STS Owner</th>
<th>Total SQL Statements</th>
<th>Comparison Metric</th>
<th>Buffer Gets</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPGRADE_10G11G</td>
<td>SYS</td>
<td>test upgrade to 11g</td>
<td>OOW_54G</td>
<td>SYS</td>
<td>54</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Global Statistics**

- Projected Workload Buffer Gets
  - 10g.data
  - 11g.data

- SQL Statement Count
  - Improved
  - Regressed
  - Unchanged

**Top 10 SQL Statements Based on Impact on Workload**

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>Net Impact on Workload (%)</th>
<th>Buffer Gets 10g_data</th>
<th>Buffer Gets 11g_data</th>
<th>Net Impact on SQL (%)</th>
<th>% of Workload</th>
<th>Plan Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>g4dzf4ak4rus2</td>
<td>12.000</td>
<td>92818,458,000</td>
<td>13,502,097,000</td>
<td>33.550</td>
<td>35.780</td>
<td>30.670</td>
</tr>
<tr>
<td>gfacm5jr3rz9j</td>
<td>11.990</td>
<td>6,990,541,000</td>
<td>180,401,000</td>
<td>97.420</td>
<td>12.310</td>
<td>0.410</td>
</tr>
<tr>
<td>2ny751aet2yd9</td>
<td>-0.820</td>
<td>12,973,052,000</td>
<td>13,440,825,000</td>
<td>-3.610</td>
<td>22.850</td>
<td>30.530</td>
</tr>
<tr>
<td>e2fb0ug5p7d4p</td>
<td>-0.750</td>
<td>12,740,524,000</td>
<td>13,165,998,000</td>
<td>-3.340</td>
<td>22.440</td>
<td>29.910</td>
</tr>
<tr>
<td>2wtgxbj6u2by</td>
<td>0.050</td>
<td>244,678,000</td>
<td>218,533,000</td>
<td>10.690</td>
<td>0.430</td>
<td>0.500</td>
</tr>
</tbody>
</table>

**Recommendations**

- Use the better execution plan from SQL Trial 1 by creating SQL Plan Baselines.
- Explore alternate execution plans using SQL Tuning Advisor.

---

[Oracle Logo]
Database Upgrade: 10g to 11g

- After using SQL Plan Management with a baseline

**SQL Performance Analyzer Task Result: SYS.UPGRADE_10G11G**

<table>
<thead>
<tr>
<th>Task Name</th>
<th>UPGRADE_10G11G</th>
<th>SQL Tuning Set Name</th>
<th>OOW_54G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Owner</td>
<td>SYS</td>
<td>STS Owner</td>
<td>SYS</td>
</tr>
<tr>
<td>Task Description</td>
<td>test upgrade to 11g</td>
<td>Total SQL Statements</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQL Statements With</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Errors</td>
<td></td>
</tr>
</tbody>
</table>

**Global Statistics**

- Projected Workload Buffer Gets
- SQL Statement Count

**Top 10 SQL Statements Based on Impact on Workload**

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>Net Impact on Workload (%)</th>
<th>Buffer Gets 10g_data</th>
<th>Buffer Gets post_SPM</th>
<th>Net Impact on SQL (%)</th>
<th>% of Workload 10g_data post_SPM Plan Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>g4bf4ak4rus2</td>
<td>12.010</td>
<td>20,318,458,000</td>
<td>13,501,307,000</td>
<td>33.550</td>
<td>35.780</td>
</tr>
<tr>
<td>gfacm5jr3rz9j</td>
<td>11.990</td>
<td>6,990,541,000</td>
<td>180,401,000</td>
<td>97.420</td>
<td>12.310</td>
</tr>
<tr>
<td>2wtxbj6u2by</td>
<td>0.050</td>
<td>244,678,000</td>
<td>218,533,000</td>
<td>10.690</td>
<td>0.430</td>
</tr>
</tbody>
</table>
Parameter Changes

- SPA Recommendation: SQL Tuning Advisor

**SQL Performance Analyzer Task Result: SYS.PARAM_CHANGE**

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

**Global Statistics**

- Projected Workload Buffer Gets
  - Buffer Gets: 4,500,000
  - Improvement Impact: 28%
  - Regression Impact: -3%
  - Overall Impact: 24%

- SQL Statement Count
  - Count: 25
  - Change in Buffer Gets:
    - Improved: 4,000,000
    - Regressed: 3,000,000
    - Unchanged: 1,000,000

**Recommendations**

Oracle offers two options to fix regressed SQL resulting from plan changes:

1. Use the better execution plan from SQL Trial 1 by creating SQL Plan Baselines.
2. Explore alternate execution plans using SQL Tuning Advisor.
3. (Run SQL Tuning Advisor)

**Top 10 SQL Statements Based on Impact on Workload**

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>Net Impact on Workload (%)</th>
<th>Buffer Gets</th>
<th>Net Impact on SQL (%)</th>
<th>% of Workload</th>
<th>Plan Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>73s2gy2svfrw</code></td>
<td>13.790 1,753,552.000 1,238,620.000</td>
<td>13.790 46.950 43.860 Y</td>
<td>29.370</td>
<td>46.950</td>
<td>43.860 Y</td>
</tr>
<tr>
<td><code>gg2a407mv2hsy</code></td>
<td>13.790 1,753,552.000 1,238,620.000</td>
<td>29.370 46.950 43.860 Y</td>
<td>46.950</td>
<td>43.860 Y</td>
<td></td>
</tr>
<tr>
<td><code>2wtgxbjz8u2hy</code></td>
<td>-3.050 218,621.000 332,519.000</td>
<td>-52.100 5.850 11.780 Y</td>
<td>-52.100</td>
<td>5.850</td>
<td>11.780 Y</td>
</tr>
<tr>
<td><code>fbp9za0hqk2km</code></td>
<td>-0.070 6.000 2,721.000</td>
<td>-45,250.000 0.000 0.100 Y</td>
<td>-45,250.000</td>
<td>0.000</td>
<td>0.100 Y</td>
</tr>
</tbody>
</table>
Parameter Changes

• SQL Tuning Advisor offers SQL Profiles

<table>
<thead>
<tr>
<th>SQL Tuning Results: TUNEREG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter Changes</strong></td>
</tr>
<tr>
<td><strong>• SQL Tuning Advisor offers SQL Profiles</strong></td>
</tr>
</tbody>
</table>

![SQL Tuning Results: TUNEREG](image-url)

**Recommendations**

<table>
<thead>
<tr>
<th>Select SQL Text</th>
<th>Parsing Schema</th>
<th>SQL ID</th>
<th>Statistics</th>
<th>SQL Profile</th>
<th>Index</th>
<th>Restructure SQL</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT /* my_query_14_scott <em>/ /</em> + ORDERED INDEX(t1) USE_HASH(t1) */ 'B'</td>
<td></td>
<td>t2.pg_featurevalue_0...</td>
<td>APPS</td>
<td>2wtqxbjz6u2by</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>SELECT /* my_query_4_scott <em>/ /</em> DISTINCT 'B'</td>
<td></td>
<td>t1.pg_featurevalue_47_id pg_featurevalue_47_id FROM...</td>
<td>APPS</td>
<td>fb9za0hqe2km</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELECT /* my_query_1_scott <em>/ /</em> DISTINCT 'B'</td>
<td></td>
<td>t1.pg_featurevalue_15_id pg_featurevalue_15_id FROM...</td>
<td>APPS</td>
<td>1h3c2y092ds9d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELECT /* my_query_2_scott <em>/ /</em> DISTINCT 'B'</td>
<td></td>
<td>t1.pg_featurevalue_15_id pg_featurevalue_15_id FROM...</td>
<td>APPS</td>
<td>654xs8xs5wp42</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Parameter Changes

- SPA run after SQL Profiles had been applied

---

#### SQL Performance Analyzer Task Result: SYS.PARAM_CHANGE

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Test Rule-based vs Cost-based Optimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAM_CHANGE</td>
<td>SYS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQL Tuning Set Name</th>
<th>STS Owner</th>
<th>Replay Trial 1</th>
<th>Replay Trial 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR_WORKLOAD</td>
<td>APPS</td>
<td>rule_based</td>
<td>after_tuning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total SQL Statements</th>
<th>SQL Statements With Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0</td>
</tr>
</tbody>
</table>

**Global Statistics**

**Projected Workload Buffer Gets**

- Buffer Gets:
  - rule_based: 1,753,552,000
  - after_tuning: 1,238,620,000

**SQL Statement Count**

- Improved: 29.37%
- Regressed: 46.95%
- Unchanged: 49.30%

**Top 10 SQL Statements Based on Impact on Workload**

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>Net Impact on Workload (%)</th>
<th>Buffer Gets</th>
<th>Net Impact on SQL (%)</th>
<th>% of Workload</th>
<th>Plan Changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>73s2sgy2svfrv</td>
<td>13.790</td>
<td>1,238,620,000</td>
<td>29.370</td>
<td>49.30Y</td>
<td></td>
</tr>
<tr>
<td>gq2a407my2hsv</td>
<td>13.790</td>
<td>1,238,620,000</td>
<td>29.370</td>
<td>49.30Y</td>
<td></td>
</tr>
<tr>
<td>2wtgxbj6u2by</td>
<td>5.090</td>
<td>28,343,000</td>
<td>87.040</td>
<td>1.13Y</td>
<td></td>
</tr>
</tbody>
</table>
SQL Performance Analyzer

- SPA: SQL statement details example

![Execution Statistics Table]

Problem Findings:
The performance of this SQL has regressed.

Symptom Findings:
The structure of the SQL execution plan has changed.

Information Findings:

Plan Comparison:

```
<table>
<thead>
<tr>
<th>Operation</th>
<th>Line ID</th>
<th>Object</th>
<th>Rows</th>
<th>Cost Predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td>0</td>
<td></td>
<td>1</td>
<td>2.900</td>
</tr>
<tr>
<td>HASH</td>
<td>1</td>
<td></td>
<td>1</td>
<td>2.900</td>
</tr>
<tr>
<td>VIEW</td>
<td>2</td>
<td></td>
<td>1</td>
<td>2.900</td>
</tr>
<tr>
<td>HASH</td>
<td>3</td>
<td></td>
<td>1</td>
<td>2.900</td>
</tr>
<tr>
<td>FILTER</td>
<td>4</td>
<td></td>
<td>1</td>
<td>2.781</td>
</tr>
<tr>
<td>HASH JOIN</td>
<td>5</td>
<td></td>
<td>1</td>
<td>2.646</td>
</tr>
<tr>
<td>HASH JOIN</td>
<td>6</td>
<td></td>
<td>1</td>
<td>2.646</td>
</tr>
</tbody>
</table>
```

ORACLE
SPA for a 9i workload against 11g

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

### Mapping Table

<table>
<thead>
<tr>
<th>Id</th>
<th>Owner</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>SH1</td>
<td>SAL1</td>
</tr>
<tr>
<td>124</td>
<td>SH2</td>
<td>CUST</td>
</tr>
</tbody>
</table>

Event 10046
- ORCL_ora_133674.trc
- ORCL_ora_133674.trc
- ORCL_ora_133674.trc
- ORCL_ora_133674.trc
- ...
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
## 2. Workflow in EM: Workload Capture

The Database Replay feature allows database workload to be captured on one system and replayed later on a different system. Replaying a captured workload can be useful to compare two different systems.

<table>
<thead>
<tr>
<th>Task</th>
<th>Task Name</th>
<th>Description</th>
<th>Go to Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Capture Workload</td>
<td>Choose this option to capture workload on this database.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Preprocess Captured Workload</td>
<td>Preprocessing will prepare a captured workload for replay. This must be done once for every captured workload.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Replay Workload</td>
<td>Choose this option to replay a preprocessed workload on this database.</td>
<td></td>
</tr>
</tbody>
</table>

### Overview

The following are the typical steps to perform Database Replay:

1. Capture the workload on a database. (Task 1)
2. Optionally export the AWR data. (Task 1)
3. Restore the replay database on a test system to match the capture database at the start of the workload capture.
4. Make changes (such as perform an upgrade) to the test system as needed.
5. Copy the workload to the test system.
6. Preprocess the captured workload. (Task 2)
7. Configure the test system for the replay.
8. Replay the workload on the restored database. (Task 3)

---

**Active Capture and Replay**

<table>
<thead>
<tr>
<th>Select Name</th>
<th>Type</th>
<th>Directory Object</th>
<th>Start Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>No items found</td>
</tr>
</tbody>
</table>
2. Preparation steps in EM: Workload Capture

- **Restarting the database prior to workload capture** is recommended for the best workload replay result. Consider scheduling the workload capture at a time when the database can be restarted.
- **Make sure there is enough disk space** to hold the captured workload. Consider doing a short duration workload capture and use it for estimating disk space requirement of a full workload capture.
- **Make sure you can restore the replay database** to match the capture database at the start of the workload capture. A successful workload replay depends on application transactions accessing application data identical to that on a capture system. Common ways to restore application data state include point-in-time recovery, flashback, and import/export.
3. Workload Filter:

**Database Restart Options**
A database restart prior to a workload capture is normally required to ensure a complete and accurate capture.

- TIP Not restarting could result in in-flight transactions being captured, which may adversely affect the replay of subsequent captured transactions.
- Restart the database prior to the capture. (Recommended)
- Do not restart the database prior to the capture.

**Workload Filters**
Workload filters are used to customize what workload will be captured. By default, most external client requests made to the database are captured. Refer to system documentation for more information.

- **Filter Mode**: Exclusion
- **Excluded Sessions**
  - All sessions will be captured except for those listed below.

**Filter Table**

<table>
<thead>
<tr>
<th>Filter Name</th>
<th>Type</th>
<th>Session Attribute</th>
<th>Value</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Management Service (DEFAULT)</td>
<td>Excluded</td>
<td>Program</td>
<td>OMS</td>
<td></td>
</tr>
<tr>
<td>Oracle Management Agent (DEFAULT)</td>
<td>Excluded</td>
<td>Program</td>
<td>emagent%</td>
<td></td>
</tr>
</tbody>
</table>

- TIP You may use % for wildcard in a filter value.
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:

**Capture Workload: Review**

**Database Logged In As**
- Database: SOURCE
- Logged In As: SYS

Reviewer the following settings for capturing the workload:

- Job Name: CAPTURE-SOURCE-TESLAUF01-JOB
- Capture Name: CAPTURE-SOURCE-Testlauf01
- Directory Object: TEMPORARY
- Start Time: Immediately
- Capture Duration: 0 Hours 15 Minutes

**Database Restart**

- Restart Database: No

**Workload Filters: Excluded Sessions**

<table>
<thead>
<tr>
<th>Filter Name</th>
<th>Type</th>
<th>Session Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Management Service (DEFAULT)</td>
<td>Excluded</td>
<td>Program</td>
<td>OMS</td>
</tr>
<tr>
<td>Oracle Management Agent (DEFAULT)</td>
<td>Excluded</td>
<td>Program</td>
<td>emagent%</td>
</tr>
</tbody>
</table>
Database Control - capture summary:

Confirmation

Job 'CAPTURE-ORCL-TESTLAUF_01-JOB' to capture the workload has been created successfully.

View Job

Once the capture is complete you will need to do the following prior to replaying the workload on a different system:
1. Optionally export the AWR data.
2. Restore the replay database on a test system to match the capture database at the start of the workload capture.
3. Make changes (such as perform an upgrade) to the test system as needed.
4. Copy the workload to the test system.
5. Preprocess the captured workload.

Database Replay

The Database Replay feature allows database workload to be captured on one system and replayed later on a different system. Replaying a captured workload can be useful to compare two different systems.

<table>
<thead>
<tr>
<th>Task</th>
<th>Task Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Capture Workload</td>
<td>Choose this option to capture workload on this database.</td>
</tr>
<tr>
<td>2</td>
<td>Preprocess Captured Workload</td>
<td>Preprocessing will prepare a captured workload for replay. This must be done once for every captured workload.</td>
</tr>
<tr>
<td>3</td>
<td>Replay Workload</td>
<td>Choose this option to replay a preprocessed workload on this database.</td>
</tr>
</tbody>
</table>

Overview

The following are the typical steps to perform Database Replay:
1. Capture the workload on a database. (Task 1)
2. Optionally export the AWR data. (Task 1)
3. Restore the replay database on a test system to match the capture database at the start of the workload capture.
4. Make changes (such as perform an upgrade) to the test system as needed.
5. Copy the workload to the test system.
6. Preprocess the captured workload. (Task 2)
7. Configure the test system for the replay.
8. Replay the workload on the restored database. (Task 3)
Database Replay

- Database Control - capture statistics:

<table>
<thead>
<tr>
<th>Workload Profile</th>
<th>Workload Filters</th>
</tr>
</thead>
</table>

**Average Active Sessions**

- Active Sessions vs. Time (Aug 29, 2007)

**Comparison**

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Capture</th>
<th>Percentage of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Time (hh:mm:ss)</td>
<td>00:11:19</td>
<td>00:07:59</td>
<td>70.54</td>
</tr>
<tr>
<td>Average Active Sessions</td>
<td>1.13</td>
<td>0.80</td>
<td>70.54</td>
</tr>
<tr>
<td>User Calls</td>
<td>109,251</td>
<td>102,625</td>
<td>9394</td>
</tr>
<tr>
<td>Transactions</td>
<td>289</td>
<td>54</td>
<td>13.69</td>
</tr>
<tr>
<td>Connects</td>
<td>37</td>
<td>12</td>
<td>32.43</td>
</tr>
<tr>
<td>Application Errors</td>
<td>N/A</td>
<td>51,100</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Database Replay

- Preprocessing captured workload:

<table>
<thead>
<tr>
<th>Task</th>
<th>Task Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Capture Workload</td>
<td>Choose this option to capture workload on this database.</td>
</tr>
<tr>
<td>2</td>
<td>Preprocess Captured Workload</td>
<td>Preprocessing will prepare a captured workload for replay. This must be done once for every captured workload.</td>
</tr>
<tr>
<td>3</td>
<td>Replay Workload</td>
<td>Choose this option to replay a preprocessed workload on this database.</td>
</tr>
</tbody>
</table>

Overview

The following are the typical steps to perform Database Replay:

1. Capture the workload on a database. (Task 1)
2. Optionally export the AWR data. (Task 1)
3. Restore the replay database on a test system to match the capture database at the start of the workload capture.
4. Make changes (such as perform an upgrade) to the test system as needed.
5. Copy the workload to the test system.
6. Preprocess the captured workload. (Task 2)
7. Configure the test system for the replay.
8. Replay the workload on the restored database. (Task 3)
Database Replay

- Replaying the captured workload:

<table>
<thead>
<tr>
<th>Task</th>
<th>Task Name</th>
<th>Description</th>
<th>Go to Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Capture Workload</td>
<td>Choose this option to capture workload on this database.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Preprocess Captured Workload</td>
<td>Preprocessing will prepare a captured workload for replay. This must be done once for every captured workload.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Replay Workload</td>
<td>Choose this option to replay a preprocessed workload on this database.</td>
<td></td>
</tr>
</tbody>
</table>

Overview

The following are the typical steps to perform Database Replay:

1. Capture the workload on a database. (Task 1)
2. Optionally export the AWR data. (Task 1)
3. Restore the replay database on a test system to match the capture database at the start of the workload capture.
4. Make changes (such as perform an upgrade) to the test system as needed.
5. Copy the workload to the test system.
6. Preprocess the captured workload. (Task 2)
7. Configure the test system for the replay.
8. Replay the workload on the restored database. (Task 3)
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

The database is waiting for connections from the Replay Clients. Start the Replay Clients now.

When all the Replay Clients have connected, proceed to the next step to continue the replay setup.

⚠️ This operation may take some time to complete. If you close this browser window or navigate to a different page, your place in the replay process will not be saved.
Database Replay

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

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

Workload Replay Client: Release 11.1.0.6.0 - Production on Thu Aug 30 09:02:10 2007

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

Replay Workload: Wait for Client Connections

Database: ORCL
Capture Name: CAPTURE_ORCL_TEST01
Logged In As: SYS

The database is waiting for connections from the Replay Clients. Start the Replay Clients now.

When all the Replay Clients have connected, proceed to the next step to continue the replay setup.

At least one Replay Client is connected. You may start more connections or continue to the next step.

This operation may take some time to complete. If you close this browser window or navigate to a different page, your place in the replay process will not be saved.

Client Connections

<table>
<thead>
<tr>
<th>SID</th>
<th>Host</th>
<th>OS Process ID</th>
<th>OS User Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>135</td>
<td>vmware.de.oracle.com</td>
<td>15203</td>
<td>oracle</td>
<td><a href="mailto:wrc@vmware.de.oracle.com">wrc@vmware.de.oracle.com</a> (TNS V1-V3)</td>
</tr>
</tbody>
</table>
Database Replay

- Monitor Workload Replay during replay

View Workload Replay: REPLAY_TEST02

- Status: In Progress (Stop Replay)

Summary

<table>
<thead>
<tr>
<th>Replay Name</th>
<th>REPLAY_TEST02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory Object</td>
<td>CAPTDIR</td>
</tr>
<tr>
<td>Database Name</td>
<td>ORCL</td>
</tr>
<tr>
<td>DBID</td>
<td>1159492831</td>
</tr>
<tr>
<td>Capture Name</td>
<td>CAPTURE ORCL_TEST01</td>
</tr>
<tr>
<td>Duration (hh:mm:ss)</td>
<td>00:07:32</td>
</tr>
<tr>
<td>Prepare Time</td>
<td>Aug 30, 2007 8:59:45 AM CEST</td>
</tr>
<tr>
<td>Start Time</td>
<td>Aug 30, 2007 9:03:28 AM CEST</td>
</tr>
<tr>
<td>End Time</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Workload Profile

- Network Time (hh:mm:ss): 00:00:00
- Think Time (hh:mm:ss): 00:00:00
- Clients: 1
- Clients Finished: 0

Elapsed Time Comparison

- Capture
- Replay

Assessing the Replay

The Elapsed Time Comparison chart shows how much time the replayed workload has taken to accomplish the same amount of work as captured.

When the Replay bar is shorter than the Capture bar, the replay environment is processing the workload faster than the capture environment.

The divergence table gives information about both the data and error discrepancies between the replay and capture environments, which can be used as a measure of the replay quality.
### Database Replay

- Monitor divergence stats:

#### Divergence

<table>
<thead>
<tr>
<th>Divergence</th>
<th>Number of Calls</th>
<th>Percentage of Total Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error Divergence:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session Failures Seen During Replay</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Errors No Longer Seen During Replay</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Errors Mutated During Replay</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>New Errors Seen During Replay</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>Data Divergence:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMLs with Different Number of Rows Modified</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SELECTs with Different Number of Rows Fetched</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

#### Detailed Comparison

<table>
<thead>
<tr>
<th></th>
<th>Capture</th>
<th>Replay</th>
<th>Percentage of Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (hh:mm:ss)</td>
<td>00:10:25</td>
<td>00:10:00</td>
<td>96.00</td>
</tr>
<tr>
<td>Database Time (hh:mm:ss)</td>
<td>00:17:06</td>
<td>00:15:22</td>
<td>83.86</td>
</tr>
<tr>
<td>Average Active Sessions</td>
<td>1.64</td>
<td>1.54</td>
<td>93.61</td>
</tr>
<tr>
<td>User Calls</td>
<td>18,141</td>
<td>18,133</td>
<td>99.96</td>
</tr>
</tbody>
</table>
### Database Replay

**Workload Replay result:**

**Summary**

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay Name</td>
<td>REPLAY_TEST02</td>
</tr>
<tr>
<td>Directory Object</td>
<td>CAPTDIR</td>
</tr>
<tr>
<td>Database Name</td>
<td>ORCL</td>
</tr>
<tr>
<td>DBID</td>
<td>1159492831</td>
</tr>
<tr>
<td>Replay Error Code</td>
<td>N/A</td>
</tr>
<tr>
<td>Replay Error Message</td>
<td>None</td>
</tr>
<tr>
<td>Capture Name</td>
<td>CAPTURE_ORCL_TEST01</td>
</tr>
<tr>
<td>Duration (hh:mm:ss)</td>
<td>00:10:00</td>
</tr>
<tr>
<td>Prepare Time</td>
<td>Aug 30, 2007 9:59:45 AM CEST</td>
</tr>
<tr>
<td>Start Time</td>
<td>Aug 30, 2007 9:03:28 AM CEST</td>
</tr>
</tbody>
</table>

**Workload Profile**

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Time (hh:mm:ss)</td>
<td>00:00:01</td>
</tr>
<tr>
<td>Think Time (hh:mm:ss)</td>
<td>00:00:00</td>
</tr>
<tr>
<td>Clients</td>
<td>1</td>
</tr>
<tr>
<td>Clients Finished</td>
<td>1</td>
</tr>
</tbody>
</table>

**Elapsed Time Comparison**

[Graph showing elapsed time comparison between capture and replay]

**Assessing the Replay**

- **Elapsed Time Comparison**
  - The Elapsed Time Comparison chart shows how much time the replayed workload has taken to accomplish the same amount of work as captured.
  - When the Replay bar is shorter than the Capture bar then the replay environment is processing the workload faster than the capture environment.
  - The divergence table gives information about both the data and error discrepancies between the replay and capture environments, which can be used as a measure of the replay quality.

[View Workload Replay Report]
## Database Replay

- **Workload replay reporting:**

<table>
<thead>
<tr>
<th>Workload Replay Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AWR Compare Period Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Workload Capture or Replay</td>
</tr>
<tr>
<td>Second Workload Capture or Replay</td>
</tr>
<tr>
<td>Run Report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AWR Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload Capture or Replay</td>
</tr>
<tr>
<td>Run Report</td>
</tr>
</tbody>
</table>
Additional Pieces of Info
Important Upgrade Notes at a glance

Upgrade to Oracle 10gR2
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 11g
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


ORACLE DATABASE 11g Oracle Database Upgrade

Upgrading to Oracle Database 11g provides the latest in efficient, reliable, secure data management for mission-critical on-line transaction processing applications, query-intensive data warehouses, and content management and Web2.0 applications. The right planning, preparation, and upgrade steps will make the upgrade process simpler, faster, and more predictable from start to finish.

Latest (July 2008)

- Database Rolling Upgrade Using Transient Logical Standby: Oracle Data Guard 11g (PDF) (June 2008)
- Database Rolling Upgrade Using Data Guard SQL Apply: Oracle Database 11g and 10gR2 (PDF) (April 2008)
- Database Replay (RAT) DBUA for Upgrade (PDF) (May 2008)
- Best Practices for Upgrading to Oracle Database 11g (PDF) (November 2007)
- Innovate Faster with Oracle Database 11g (PDF) (August 2007)
- Upgrading to Oracle RAC 11g on Oracle Enterprise Linux (OTN) (October 2007)

Technical Information

- Different Upgrade Methods for Upgrading your Database (Metalink)
- Complete Checklist for Manual Upgrades - 11gR1 (Metalink)
- Complete Checklist for Upgrades - 11gR1 Using DBUA (Metalink)
- Database Server Upgrade/Downgrade Compatibility Matrix (Metalink)

Support

- 11g Upgrade Companion (Metalink)
- Recommended Patches Tool (Metalink)
- Oracle Lifetime Support Policy

Previous Releases

- 10g Upgrade Companion (Metalink)
- Database Rolling Upgrade Using Transient Logical Standby - Oracle Database 10g Release 2 (PDF) (April 2008)
- Real Application Testing on earlier releases (Metalink)
- Complete Checklist for Manual Upgrades - 10gR2 (Metalink)

Questions

- Oracle MetaLink Information
OTN Upgrade Forum


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

<table>
<thead>
<tr>
<th>Thread</th>
<th>Author</th>
<th>Views</th>
<th>Replies</th>
<th>Last Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTN Database Upgrade Page</td>
<td>nsagacrmfromh...</td>
<td>19</td>
<td>1</td>
<td>Dec 8, 2008 1:55 PM</td>
</tr>
<tr>
<td>11 client installed, 10 client still exist - which client is in use?</td>
<td>nsagacrmfromh...</td>
<td>19</td>
<td>1</td>
<td>Dec 8, 2008 1:55 PM</td>
</tr>
<tr>
<td>Upgrade from 10.2.0.1.0 x64 to 10.2.0.4.0 x64</td>
<td>user613109</td>
<td>56</td>
<td>2</td>
<td>Dec 7, 2008 9:01 AM</td>
</tr>
<tr>
<td>How to Upgrade the DB through exp/imp utility</td>
<td>user10687856</td>
<td>54</td>
<td>2</td>
<td>Dec 7, 2008 8:56 AM</td>
</tr>
<tr>
<td>Using DBUA to migrate Oracle 9i to Oracle 11g</td>
<td>JuaneDejo Prudol</td>
<td>235</td>
<td>4</td>
<td>Dec 5, 2008 8:22 AM</td>
</tr>
<tr>
<td>Migrating Report from 10.7 to 11i Apps (data bases 7.6 to blast)</td>
<td>user10646223</td>
<td>65</td>
<td>1</td>
<td>Dec 5, 2008 8:31 PM</td>
</tr>
<tr>
<td>9.2.0.7 to 10.2.0.3 upgrade with DBUA do I need to change the Spfile manually</td>
<td>gagarashula</td>
<td>100</td>
<td>3</td>
<td>Dec 3, 2008 4:36 PM</td>
</tr>
<tr>
<td>trigger problem</td>
<td>user10646457</td>
<td>55</td>
<td>1</td>
<td>Dec 3, 2008 3:52 PM</td>
</tr>
<tr>
<td>upgrade oracle 10.2.3.0 patchset2 to 10.2.3.0 patchset</td>
<td>user10646457</td>
<td>55</td>
<td>1</td>
<td>Dec 3, 2008 1:20 PM</td>
</tr>
</tbody>
</table>

Search Forum

Top Users in Forum

- Mike Dietrich (130)
- Babish Khandi (65)
- ora tech (30)
- demorgan (10)
- Hemant K Chitale (10)
- SYSTAN_2 (10)
- yinghua (10)
- roy swinger (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**

<table>
<thead>
<tr>
<th>Note</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>756435.1*</td>
<td>Block change tracking on physical standby can cause incorrect backups</td>
</tr>
<tr>
<td>731599.1P*</td>
<td>Linux: Random node reboots in RAC induced by OCSSD</td>
</tr>
</tbody>
</table>

**Upgrade Issues**

<table>
<thead>
<tr>
<th>Note</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>742647.1*</td>
<td>ASM rolling upgrade to 11.1.0.7 fails in RAC environments</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Note</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7272646P</td>
<td>Linux-x86_64: ORA-27103 on startup when MEMORY_TARGET &gt; 3g</td>
</tr>
<tr>
<td>7213937</td>
<td>Various OERI using Resource Manager</td>
</tr>
<tr>
<td>7211965</td>
<td>Different default cost for BETWEEN</td>
</tr>
<tr>
<td>7113299</td>
<td>Dump [kkslnl] with fix for bug 6163785</td>
</tr>
</tbody>
</table>
Real World Experience

• White Paper available on OTN
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