Move to Oracle Database 11g – The whole Story

Mike Dietrich. Carol Tagliaferri, Roy Swonger
Database Upgrade & Utilities
ORACLE Corporation
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Parameters
- Processes
- Miscellaneous
- Tools
- Automation
- LOB & SecureFiles
- Flashback
- ADR
- Compression
Parameter COMPATIBLE

- COMPATIBLE has to be at least 10.1.0 for an 11g database
- No way back once $\geq 11.1.0$ has been enabled
  - Supported release downgrade only down to 10.1.0
  - Supported patch downgrade only down to 11.1.0
  - No `ALTER DATABASE RESET COMPATIBILITY` command anymore
Parameter COMPATIBLE

• Upgrade 9i ⇒ 11g precautions:
  • Make 100% sure that you've applied DST patches before starting the 9i database in an 11g environment
  • 9i ⇒ 11g requires COMPATIBLE ≥ 10.1
  • No possibility to have COMPATIBLE=9.2.0
  • No way back as soon as the database has been opened with a raised COMPATIBLE setting
Parameter COMPATIBLE

- DBUA raises COMPATIBLE only for 9i databases to 10.1.0
- To enable Oracle 11g features:

```
SQL> alter system
    set compatible='11.1.0' scope=spfile;
```

- Afterwards restart the database
  - New features will be enabled
  - Datafile headers will be adjusted
  - Redologfiles will be adjusted during first access
# New 11g Parameters

**New in 11g**

Oracle® Database Reference

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM_PREFERRED_READ_FAILURE_GROUPS</td>
<td>Use alternative ASM groups for datafile reads when the primary group is unavailable.</td>
</tr>
<tr>
<td>CLIENT_RESULT_CACHE_LAG</td>
<td>Lag of the client result cache.</td>
</tr>
<tr>
<td>CLIENT_RESULT_CACHE_SIZE</td>
<td>Size of the client result cache.</td>
</tr>
<tr>
<td>COMMIT_LOGGING</td>
<td>Enable or disable database commit logging.</td>
</tr>
<tr>
<td>COMMIT_WAIT</td>
<td>The amount of time Oracle will wait for a committed transaction before timing out.</td>
</tr>
<tr>
<td>CONTROL_MANAGEMENT_PACK_ACCESS</td>
<td>Access control for the control file.</td>
</tr>
<tr>
<td>DB_LOST_WRITE_PROTECT</td>
<td>Protect against data loss due to database failures.</td>
</tr>
<tr>
<td>DB_SECUREFILE</td>
<td>Security features for the database.</td>
</tr>
<tr>
<td>DB_ULTRA_SAFE</td>
<td>Additional security features for the database.</td>
</tr>
<tr>
<td>DDL_LOCK_TIMEOUT</td>
<td>The amount of time Oracle will wait for a lock on a DDL object before timing out.</td>
</tr>
<tr>
<td>DIAGNOSTIC_DEST</td>
<td>Where diagnostic files are stored.</td>
</tr>
<tr>
<td>GLOBAL_TXN_PROCESSES</td>
<td>The maximum number of global transaction processes.</td>
</tr>
<tr>
<td>JAVA_JIT_ENABLED</td>
<td>Enable or disable Just-In-Time compilation.</td>
</tr>
<tr>
<td>LDAP_DIRECTORY_SYSAUTH</td>
<td>Authentication for LDAP directories.</td>
</tr>
<tr>
<td>MEMORY_MAX_TARGET</td>
<td>The maximum amount of memory that the database can use.</td>
</tr>
<tr>
<td>MEMORY_TARGET</td>
<td>The target amount of memory that the database should use.</td>
</tr>
<tr>
<td>OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES</td>
<td>Capture SQL plan baselines.</td>
</tr>
<tr>
<td>OPTIMIZER_USE_INVISIBLE_INDEXES</td>
<td>Use invisible indexes during optimization.</td>
</tr>
<tr>
<td>OPTIMIZER_USE_PENDING_STATISTICS</td>
<td>Use pending statistics during optimization.</td>
</tr>
<tr>
<td>OPTIMIZER_USE_SQL_PLAN_BASELINES</td>
<td>Use SQL plan baselines.</td>
</tr>
<tr>
<td>PARALLEL_IO_CAP_ENABLED</td>
<td>Enable or disable parallel I/O capabilities.</td>
</tr>
<tr>
<td>PLSCOPE_SETTINGS</td>
<td>Settings for PL/SQL scope.</td>
</tr>
<tr>
<td>REDO_TRANSPORT_USER</td>
<td>Use redo transport for client nodes.</td>
</tr>
<tr>
<td>RESOURCE_MANAGER_CPU_ALLOCATION</td>
<td>Allocate CPU resources to the database.</td>
</tr>
<tr>
<td>RESULT_CACHE_MAX_RESULT</td>
<td>Maximum number of result sets in the result cache.</td>
</tr>
<tr>
<td>RESULT_CACHE_MAX_SIZE</td>
<td>Maximum size of the result cache.</td>
</tr>
<tr>
<td>RESULT_CACHE_MODE</td>
<td>The mode of the result cache.</td>
</tr>
<tr>
<td>RESULT_CACHE_REMOTE_EXPIRATION</td>
<td>The amount of time result cache entries are eligible for remote expiration.</td>
</tr>
<tr>
<td>SEC_CASE_SENSITIVE_LOGON</td>
<td>Case-sensitive logon.</td>
</tr>
<tr>
<td>SEC_MAX_FAILED_LOGIN_ATTEMPTS</td>
<td>Maximum number of failed login attempts.</td>
</tr>
<tr>
<td>SEC_PROTOCOL_ERROR_FURTHER_ACTION</td>
<td>Action to take on protocol errors.</td>
</tr>
<tr>
<td>SEC_PROTOCOL_ERROR_TRACE_ACTION</td>
<td>Trace action for protocol errors.</td>
</tr>
<tr>
<td>SEC_RETURN_SERVER_RELEASE_BANNER</td>
<td>Return server release banner.</td>
</tr>
<tr>
<td>XML_DB_EVENTS</td>
<td>Event logging for XML databases.</td>
</tr>
</tbody>
</table>
New Parameters - Selection

- **CONTROL_MANAGEMENT_PACK_ACCESS**
  - Default: DIAGNOSTIC+TUNING
  - Values: DIAGNOSTIC+TUNING
  - DIAGNOSTIC
  - NONE
  - Purpose: Controls access to several functionalities of Diagnostic and Tuning Pack
  - Example:

```
CONTROL_MANAGEMENT_PACK_ACCESS=NONE

select count(*) from v$active_session_history;
==> 0 rows
```
New Parameters - Selection

- **DIAGNOSTIC_DEST**
  - Default: $ORACLE_BASE
  - Value: OS path/directory
  - Layout: `<diagnostic_dest>/diag/rdbms/<dbname>/<instname>`
  - Purpose: Substitutes `background_dump_dest`, `user_dump_dest`, listener trace etc. and specifies the ADR home (Automatic Diagnostic Repository)
New Parameters - Selection

- **MEMORY_TARGET** and **MEMORY_MAX_TARGET**
  - Default: 0
  - Value: integer [K | M | G]
  - Purpose: Specifies the complete plus the maximum memory available to the Oracle server
New Parameters - Selection

- **OPTIMIZER_USE_INVISIBLE_INDEXES**
  - Default: FALSE
  - Values: TRUE FALSE
  - Purpose: "Invisible" indexes will be ignored by the optimizer if set to FALSE. But DMLs will be still executed to the index.
  - Motivation: Isolated testing of performance effects of an index based on a session level
  - Example:
    ```sql
    CREATE INDEX emp_ename ON emp(ename) INVISIBLE;
    
    ALTER SESSION SET
    OPTIMIZER_USE_INVISIBLE_INDEXES=TRUE;
    ```
New Parameters - Selection

- **SEC_CASE_SENSITIVE_LOGON**
  - Default: TRUE
  - Values: TRUE, FALSE
  - Purpose: Switches on/off the case sensitivity of passwords
    Attention: also valid for database links!!
  - Info:

```sql
SELECT username, password_versions FROM dba_users;
USERNAME   PASSWORD_VERSIONS
---------- -----------------
JONES      10G 11G
PRESTON    11G
BLAKE      10G
```

Created in:...

Changed in:...
New Parameters - Selection

• **SEC_CASE_SENSITIVE_LOGON**  2/2
  • Switching on the case sensitivity password mode can have several impacts:
    • Scripts with user/pw like SCOTT/TIGER won't run anymore if the user gets created or altered e.g. as SCOTT/tiger in 11g
      • Annotation: Upon upgrading the "old" mechanism is still valid
    • Check password encrypted database links
    • Can be enabled also for the password file checkings
      • `orapwd file=orapwSID password=GeHeIm ignorecase=n`
    • Very helpful:
      Check view for default passwords: `DBA_USERS_WITH_DEFPWD`
New Parameter Defaults - Selection

- **AUDIT_TRAIL**
  - Default: NONE or DB
  - Values: NONE OS XML DB DB_EXTENDED
  - Purpose: Will help auditing security relevant statements.

  Upon an upgrade **AUDIT_TRAIL** keeps its default from 9.2/10.1/10.2 (NONE). If a 11g database is created from scratch then it'll set to DB and statements get logged in **SYS.AUD$**
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Parameters
Processes
Miscellaneous
Tools
Automation
LOB & SecureFiles
Flashback
ADR
Compression
Background Processes

- Example:

```sql
select name, description from v$bgprocess, v$process
    where paddr=addr;
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARC0</td>
<td>Archival Process 0</td>
<td>MMNL</td>
<td>Manageability Monitor Pr 2</td>
</tr>
<tr>
<td>ARC1</td>
<td>Archival Process 1</td>
<td>MMON</td>
<td>Manageability Monitor Pr</td>
</tr>
<tr>
<td>ARC2</td>
<td>Archival Process 2</td>
<td>PMON</td>
<td>process cleanup</td>
</tr>
<tr>
<td>ARC3</td>
<td>Archival Process 3</td>
<td>PSP0</td>
<td>process spawner 0</td>
</tr>
<tr>
<td>CJQ0</td>
<td>Job Queue Coordinator</td>
<td>QMNC</td>
<td>AQ Coordinator</td>
</tr>
<tr>
<td>CKPT</td>
<td>checkpoint</td>
<td>RECO</td>
<td>distributed recovery</td>
</tr>
<tr>
<td>DBRM</td>
<td>Resource Manager process</td>
<td>SMCO</td>
<td>Space Manager Process</td>
</tr>
<tr>
<td>DBW0</td>
<td>db writer process 0</td>
<td>SMON</td>
<td>System Monitor Process</td>
</tr>
<tr>
<td>DIA0</td>
<td>diagnosibility process 0</td>
<td>VKTM</td>
<td>Virtual Keeper of TiMe pr</td>
</tr>
<tr>
<td>DIAG</td>
<td>diagnosibility process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSKM</td>
<td>slave DiSKMon process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FBDA</td>
<td>Flashback Data Archiver Pr.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LGWR</td>
<td>Redo etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMAN</td>
<td>Memory Manager</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# New Background Processes

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Description</th>
<th>Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACMS</td>
<td>In an RAC environment, this per-instance process (Atomic Controlfile to Memory Service) is an agent which contributes to ensuring a distributed SGA memory update is either globally committed on success or globally aborted in the event of a failure.</td>
<td>RAC</td>
</tr>
<tr>
<td>DIA0</td>
<td>Responsible for hang detection and deadlock resolution.</td>
<td>ALL</td>
</tr>
<tr>
<td>DIAG</td>
<td>Performs diagnostic dumps and executes global oradebug commands.</td>
<td>ALL</td>
</tr>
<tr>
<td>DBRM</td>
<td>The Resource Manager process is responsible for setting Resource Plans and other Resource Manager related tasks.</td>
<td>ALL</td>
</tr>
<tr>
<td>EMNC / e0xx</td>
<td>The Event Monitor Coordinator coordinates the event management and notification activity in the database which includes Streams Event Notifications, Continuous Query Notifications and Fast Application Notifications.</td>
<td>ALL with any registered event activity (optional non-fatal)</td>
</tr>
<tr>
<td>FBDA</td>
<td>The process archives historical rows for tracked tables into flashback data archives and manages the flashback archives.</td>
<td>ALL</td>
</tr>
<tr>
<td>GMON</td>
<td>Maintains disk membership in ASM disk groups</td>
<td>ASM</td>
</tr>
<tr>
<td>GTX0-j</td>
<td>Provides transparent support for XA global transactions in a RAC environment. The database autotunes the number of these processes based on the workload of XA global transactions.</td>
<td>RAC only</td>
</tr>
</tbody>
</table>
# New Background Processes

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Description</th>
<th>Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>KATE</td>
<td>Performs proxy I/O to an ASM metafile when a disk becomes offline.</td>
<td>ASM, spawned on demand.</td>
</tr>
<tr>
<td>MARK</td>
<td>Marks ASM Allocation Units as stale following a missed write to an offline disk.</td>
<td>ASM, spawned on demand.</td>
</tr>
<tr>
<td>PSP0</td>
<td>Spawns Oracle processes.</td>
<td>ALL</td>
</tr>
<tr>
<td>RMSn</td>
<td>The RAC Management Processes perform manageability tasks for RAC, e.g. creation of RAC related resources when new instances are added to the clusters</td>
<td>RAC only</td>
</tr>
<tr>
<td>RMSN</td>
<td>In a RAC environment, this process manages background slave process creation and communication on remote instances. These background slave processes perform tasks on behalf of a coordinating process running in another instance.</td>
<td>RAC only</td>
</tr>
<tr>
<td>SMCO / Wnnn</td>
<td>The space management coordinator process coordinates the execution of various space management related tasks, such as proactive space allocation and space reclamation. It dynamically spawns slave processes (Wnnn) to implement the task.</td>
<td>ALL (optional non-fatal)</td>
</tr>
<tr>
<td>VKTM</td>
<td>The Virtual Keeper of TiMe(^2) is responsible for providing a Wall-Clock time (updated every second) and Reference-Time Counter (updated every 20ms and available only when running at elevated priority).</td>
<td>ALL</td>
</tr>
</tbody>
</table>
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Parameters
Processes
Miscellaneous
Tools
Automation
LOB & SecureFiles
Flashback
ADR
**CONNECT Role**

- The CONNECT Role has been changed since $10g^{R2}$:

<table>
<thead>
<tr>
<th>GRANTEE</th>
<th>PRIVILEGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT</td>
<td>CREATE VIEW</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CREATE TABLE</td>
</tr>
<tr>
<td>CONNECT</td>
<td>ALTER SESSION</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CREATE CLUSTER</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CREATE SESSION</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CREATE SYNONYM</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CREATE SEQUENCE</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CREATE DATABASE LINK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GRANTEE</th>
<th>PRIVILEGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT</td>
<td>CREATE SESSION</td>
</tr>
</tbody>
</table>
CONNECT Role

- Find out affected database users:

```sql
SELECT grantee
FROM dba_role_privs
WHERE granted_role = 'CONNECT' and
       grantee NOT IN ('SYS', 'OUTLN', 'SYSTEM',
                       'CTXSYS', 'DBSNMP',
                       'LOGSTDBY_ADMINISTRATOR',
                       'ORDSYS', 'ORDPLUGINS',
                       'OEM_MONITOR', 'WKSYS',
                       'WKPROXY', 'WK_TEST',
                       'WKUSER', 'MDSYS',
                       'LBACSYS', 'DMSYS', 'WMSYS',
                       'EXFSYS', 'SYSMAN',
                       'MDDATA', 'XDB', 'ODM',
                       'SI_INFORMTN_SCHEMA');
```
Shared Pool calculation since 10g

- Parameter shared_pool_size is calculated differently since 10g
- Real shared pool = shared_pool_size – startup overhead

```
SQL> select * from v$sgainfo;

<table>
<thead>
<tr>
<th>NAME</th>
<th>BYTES</th>
<th>RES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed SGA Size</td>
<td>1266372</td>
<td>No</td>
</tr>
<tr>
<td>Redo Buffers</td>
<td>2924544</td>
<td>No</td>
</tr>
<tr>
<td>Buffer Cache Size</td>
<td>16777216</td>
<td>Yes</td>
</tr>
<tr>
<td>Shared Pool Size</td>
<td>83886080</td>
<td>Yes</td>
</tr>
<tr>
<td>Large Pool Size</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>Java Pool Size</td>
<td>50331648</td>
<td>Yes</td>
</tr>
<tr>
<td>Streams Pool Size</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>Granule Size</td>
<td>4194304</td>
<td>No</td>
</tr>
<tr>
<td>Maximum SGA Size</td>
<td>155189248</td>
<td>No</td>
</tr>
<tr>
<td>Startup overhead in Shared Pool</td>
<td>29360128</td>
<td>No</td>
</tr>
<tr>
<td>Free SGA Memory Available</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
```
GROUP BY results: unsorted

- Hash Group by aggregation which allows a hash algorithm to process group by statements
  - For sorts `ORDER BY` has to be used

```sql
select sum(sal), deptno from scott.emp group by deptno;
```

<table>
<thead>
<tr>
<th>SUM(SAL)</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>8750</td>
<td>10</td>
</tr>
<tr>
<td>10875</td>
<td>20</td>
</tr>
<tr>
<td>9400</td>
<td>30</td>
</tr>
</tbody>
</table>

- Description see Note:345048.1
  - `_gby_hash_aggregation_enabled=false`
- See also:
  Note:295819.1 Potential Query Tuning Related Issues
Secure-View-Merging Privilege

• To prevent optimizer issues in since 10g R2 complex view structures security:
  • `optimizer_secure_view_merging=false`
    • Otherwise it is possible for the optimizer to create different execution plans for identical statements on identical objects issued from different user schemas
  • Alternative: Grant MERGE ANY VIEW privilege to subordinate users
  • See Note:468380.1
Cascading View Issues - 10.2.0.x

- Optimizer sometimes does wrong rewrites with cascading views
  - Requirement:
    - `"_push_join_predicate"` is set to TRUE
  - Solution:
    - Patch on top of 10.2.0.4: #7445276
    - Or: `_optimizer_join_elimination_enabled=false`
  - Otherwise sometimes:
    - This Query: `select * from some_view where col1=10;`
      results in:
      - OK: `select * from (select * from base_table where col1=10);`
      - Not OK: `select * from (select * from base_table) where col1=10;`
Costed query transformations since 10g

• Observation:
  Query timing problems after upgrade to 10/11g
  • Since 10g CBO has been added costed subquery unnesting and view merging functionality.
  • Can be disabled by:
    • _optimizer_cost_based_transformation=off
CBO verifies more joins orders since 10g

- The CBO since 10g verifies more join orders to find out the least expensive one - this can sometimes lead to higher parse times
  - Remedy: `new_initial_join_orders=false`
  - Example: observation for an explain plan:
    - Set OFE=920: Number of join permutations tried: 187
    - Set OFE=10203: Number of join permutations tried: 986
PL/SQL cursors not cached ≥10.2.0.4

• Before 10.2.0.4 PL/SQL cursors will be cached automatically because of \texttt{OPEN\_CURSORS}. Since 10.2.0.4 \texttt{SESSION\_CACHED\_CURSORS} has to be defined to ensure cursor caching for PL/SQL
  • Remedy: \texttt{session\_cached\_cursors=100}
SYSAUX

- SYSAUX-Tablespace
  - SYSAUX is the new home for most repositories
  - SYSAUX can't be deleted or renamed
  - SYSAUX can't be changed to read-only
  - SYSAUX has to be created
  - Recommended size: 500 MB
  - ASSM and Locally Managed
    - Size estimation: `?/rdbms/admin/utlsyxsz.sql`

- View V$SYSAUX_OCCUPANTS shows:
  - Repositories and their owner
  - Space usage
  - Procedures for moving repositories
**SYSAUX**

- **V$SYSAUX_OCCUPANTS** - an excerpt:

<table>
<thead>
<tr>
<th>REPO</th>
<th>DESCR</th>
<th>MOVE</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>AO</td>
<td>Analytical Workspace Object Table</td>
<td>DBMS_AW.MOVE_AW</td>
<td>1.4</td>
</tr>
<tr>
<td>AUTO_TASK</td>
<td>Automated Maintenance Tasks</td>
<td></td>
<td>.3</td>
</tr>
<tr>
<td>EM_MONITORING_U</td>
<td>Enterprise Manager Monitoring User</td>
<td></td>
<td>2.4</td>
</tr>
<tr>
<td>EM</td>
<td>Enterprise Manager Repository</td>
<td>emd_maintenance</td>
<td>123.6</td>
</tr>
<tr>
<td>LOGMNR</td>
<td>LogMiner</td>
<td>SYS.DBMS_LOGMNR</td>
<td>7.6</td>
</tr>
<tr>
<td>SDO</td>
<td>Oracle Spatial</td>
<td>MDSYS.MOVE_SDO</td>
<td>0</td>
</tr>
<tr>
<td>STREAMS</td>
<td>Oracle Streams</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PL/SCOPE</td>
<td>PL/SQL Identifier Collection</td>
<td></td>
<td>.4</td>
</tr>
<tr>
<td>SQL_MANAGEMENT_</td>
<td>SQL Management Base Schema</td>
<td></td>
<td>1.6</td>
</tr>
<tr>
<td>SM/ADVISOR</td>
<td>Server Manageability - Advisor Fram</td>
<td></td>
<td>7.8</td>
</tr>
<tr>
<td>SM/AWR</td>
<td>Server Manageability - Automatic Wo</td>
<td></td>
<td>27.6</td>
</tr>
<tr>
<td>SM/OPTSTAT</td>
<td>Server Manageability - Optimizer St</td>
<td></td>
<td>3.6</td>
</tr>
<tr>
<td>SM/OTHER</td>
<td>Server Manageability - Other Compon</td>
<td></td>
<td>5.6</td>
</tr>
<tr>
<td>STATSPACK</td>
<td>Statspack Repository</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>JOB_SCHEDULER</td>
<td>Unified Job Scheduler</td>
<td></td>
<td>.4</td>
</tr>
<tr>
<td>WM</td>
<td>Workspace Manager</td>
<td>DBMS_WM.move_pr</td>
<td>7.1</td>
</tr>
</tbody>
</table>
## Summary of SYSAUX Space Estimation

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimated Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Est size of EM</td>
<td>51.1 MB</td>
</tr>
<tr>
<td>Est size of XDB</td>
<td>48.4 MB</td>
</tr>
<tr>
<td>Est size of SDO</td>
<td>21.8 MB</td>
</tr>
<tr>
<td>Est size of WM</td>
<td>6.9 MB</td>
</tr>
<tr>
<td>Est size of LOGMNR</td>
<td>5.9 MB</td>
</tr>
<tr>
<td>Est size of SM/OTHER</td>
<td>4.9 MB</td>
</tr>
<tr>
<td>Est size of SM/ADVISOR</td>
<td>4.8 MB</td>
</tr>
<tr>
<td>Est size of EXPRESSION.Filter</td>
<td>3.6 MB</td>
</tr>
<tr>
<td>Est size of EM_MONITORING_USER</td>
<td>1.6 MB</td>
</tr>
<tr>
<td>Est size of LOGSTDBY</td>
<td>0.9 MB</td>
</tr>
<tr>
<td>Est size of AO</td>
<td>0.8 MB</td>
</tr>
<tr>
<td>Est size of XSOQHIST</td>
<td>0.8 MB</td>
</tr>
<tr>
<td>Est size of ORDIM</td>
<td>0.6 MB</td>
</tr>
<tr>
<td>Est size of STREAMS</td>
<td>0.5 MB</td>
</tr>
<tr>
<td>Est size of JOB_SCHEDULER</td>
<td>0.4 MB</td>
</tr>
<tr>
<td>Est size of TSM</td>
<td>0.3 MB</td>
</tr>
<tr>
<td>Est size of Others</td>
<td>6.9 MB</td>
</tr>
<tr>
<td>Est size of SM/AWR</td>
<td>170.1 MB</td>
</tr>
<tr>
<td>Est size of SM/OPTSTAT</td>
<td>16.6 MB</td>
</tr>
</tbody>
</table>

Total Estimated SYSAUX size: 346.6 MB
Instant Client

- Universal. small footprint Oracle client
  - Works with OCI, OCCI, Pro*C, ODBC, JDBC, ODP.NET, ASP.NET, OLE DB and OO4O applications
  - No OUI Oracle Client installation necessary anymore
- Easy and simple deployment
  - Download it from OTN
  - Copy it to the target/client system(s)
  - Add its directory to \texttt{PATH/\texttt{LD\_LIBRARY\_PATH}} and set \texttt{TNS\_ADMIN}
    - For ODBC run the provided batch script
    - Patching? Just deploy the current version
- More Information on OTN:
  \url{http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html}
Instant Client

- Instant Client Download:
  - Instant Client for Microsoft Windows (32-bit)
  - Instant Client for Microsoft Windows 64-bit Itanium
  - Instant Client for Microsoft Windows (x64)
  - Instant Client for Linux x86
  - Instant Client for Linux x86-64
  - Instant Client for Linux Itanium
  - Instant Client for Linux AMD64 (32-bit and 64-bit)
  - Instant Client for Linux on Power (32-bit)
  - Instant Client for Linux on Power (64-bit)
  - Instant Client for z/Linux (32-bit and 64-bit)
  - Instant Client for Mac OS X (Intel x86) (32-bit and 64-bit)
  - Instant Client for Mac OS X (PPC)
  - Instant Client for Solaris Operating System (SPARC) (64-bit)
  - Instant Client for Solaris Operating System (SPARC) (32-bit)
  - Instant Client for Solaris x86
  - Instant Client for Solaris x86-64
  - Instant Client for HP-UX PA-RISC (64-bit)
  - Instant Client for HP-UX PA-RISC (32-bit)
  - Instant Client for AIX5L (64-bit)
  - Instant Client for AIX5L (32-bit)
  - Instant Client for HP Tru64 UNIX
  - Instant Client for HP-UX Itanium (64-bit)
  - Instant Client for HP-UX Itanium (32-bit)
Instant Client

- Instant Client Packaging

### Package Descriptions

<table>
<thead>
<tr>
<th>Instant Client Package</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>All files required to run OCI, OCCI, and JDBC-OCI applications</td>
<td>OCI, OCCI, JDBC</td>
</tr>
<tr>
<td>Basic Lite</td>
<td>Smaller version of the Basic, with only English error messages and Unicode, ASCII, and Western European character set support (10.2 only)</td>
<td>OCI, OCCI, JDBC</td>
</tr>
<tr>
<td>JDBC Supplement*</td>
<td>Additional support for XA, Internationalization, and RowSet operations under JDBC</td>
<td>JDBC</td>
</tr>
<tr>
<td>SQL<em>Plus</em></td>
<td>Additional libraries and executable for running SQL*Plus with Instant Client</td>
<td>SQL<em>Plus 10.1, SQL</em>Plus 10.2</td>
</tr>
<tr>
<td>ODBC Supplement*</td>
<td>Additional libraries for enabling ODBC applications with Instant Client (Not all platforms)</td>
<td>ODBC</td>
</tr>
<tr>
<td>SDK*</td>
<td>Additional header files and an example makefile for developing Oracle applications with Instant Client</td>
<td></td>
</tr>
<tr>
<td>ODAC*</td>
<td>Includes ODP.NET, Oracle Services for MTS, Oracle Providers for ASP.NET, Oracle Provider for OLE DB, and O04O with Oracle Instant Client</td>
<td></td>
</tr>
</tbody>
</table>

Optional packages are marked with a *.

- **Basic**: All files required to run OCI, OCCI, and JDBC-OCI applications (46 MB)
- **JDBC Supplement**: Additional support for XA, Internationalization, and RowSet operations under JDBC (1.5 MB)
- **SQL*Plus**: Additional libraries and executable for running SQL*Plus with Instant Client (0.8 MB)
- **SDK**: Additional header files and an example makefile for developing Oracle applications with Instant Client (1 MB)
- **ODBC**: Additional libraries for enabling ODBC applications (0.7 MB)
- **WRC**: Workload Replay Client used to replay workload for RAT's DB Replay Feature (6 KB)
- **Precompiler**: Additional files for "proc" binary and related files to precompile a Pro*C application and demo (0.5 MB)
## Clients & Net

- Client certification - **Note: 207303.1**

<table>
<thead>
<tr>
<th>Database Release</th>
<th>11.1.0</th>
<th>10.2.0</th>
<th>10.1.0</th>
<th>9.2.0</th>
<th>9.0.1</th>
<th>8.1.7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>11.1.0</strong></td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10.2.0</strong></td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td></td>
<td>☒</td>
</tr>
<tr>
<td><strong>10.1.0</strong></td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td><strong>9.2.0</strong></td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td><strong>9.0.1</strong></td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
<tr>
<td><strong>8.1.7</strong></td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
</tbody>
</table>

- **Certified**
- **Supported but ES**
- **Not supported anymore**
- **Never supported**
Clients & Net

- Database links
  - Only supported if matrix shows 'Supported' in both directions
  - Not supported means: Use it on your own risk
- JDBC certification
  - Note: 203849.1
- Forms/Reports certification
  - Metalink ⇒ Certify Tab ⇒ Development Tools
PL/SQL Compilation

- PL/SQL issues while upgrading from 7/8 to 10g/11g?
  - `PLSQL_V2_COMPATIBILITY = TRUE`
    PL/SQL checking methods work such as in release 8i
Upgrading your applications

• Compatibility Issues:
  • Check for reserved words
    • Appendix A, Upgrade Guide
  • Check for behaviour changes
    • Appendix A, Upgrade Guide
    • Upgrade Companion
  • Check for changes in the data dictionary
    • Appendix A, Upgrade Guide
Upgrading your applications

- Precompiler and OCI applications:
  - Upgrading the Oracle Database Server software
    - If you do not change the client environment than you won't have to precompile, compile or relink
    - Database Server version should be identical or higher than client version
  - Upgrading the Oracle Client software
    - It is recommended that Oracle client software has the same release than the server
      - Dynamic Libraries: No relink required
      - Static Libraries: Need to be relinked always
Upgrading your applications

• Precompiler and OCI applications:
  • Option 1: Leave the application unchanged
    • Don't upgrade the client installation
    • Simple and easy
  • Option 2: Precompile the application and change client
    • Recompile when you change the major release
    • You don't have to recompile for a patch release
  • Option 3: Change the application code to use 11g features
    • Most potential benefits
    • Most difficult option
    • Precompile, compile and relink
Wordsize

• Word size change 32/64-bit happens implicitly during upgrade/downgrade process
  • This will apply to patch sets, too
Conversion between XE, SE and EE

- SE ⇔ EE see Note:117048.1
  - Take a full backup
  - Run catalog.sql and catproc.sql in the EE environment

- EE ⇔ SE see Note:139642.1
  - Only Export/Import with exp/imp or expdp/imdp will be supported

- XE ⇔ SE/SEone/EE
  - http://download.oracle.com/docs/cd/B28359_01/server.111/b28300/intro.htm#BABGDCDD
  - Use the DBUA to upgrade your XE database to Seone/SE/EE
Agenda

- Preparation
- Upgrade
- News and Task List
  - Parameters
  - Processes
  - Miscellaneous
  - Tools
  - Automation
  - LOB & SecureFiles
  - Flashback
  - ADR
  - Compression
- Diagnostics & Tuning
- Performance Testing
- Best Practices
Enterprise Manager 11g

• Database Control for all database administration tasks
  
• Grid Control as an IT system administration tool
  • Supports Oracle and non-Oracle IT infrastructures
  • GC patch set 10.2.0.5 contains 11g functionality
  • GridControl 11g is not yet available
Enterprise Manager 11g

- Components
  - Database Control:
    Agent and standalone OC4J - thus no additional webserver needed
  - Grid Control:
    OMS, Management Agent and Repository Database

- Installation:
  - Database Control:
    Automatically done within the 11g database software installation
  - Grid Control:
    Separate download/DVD

- License:
  - Both products are included in any database license
  - Specific packs and plug-ins may be licensed additionally
Enterprise Manager 11g

• Database Control

Database Instance: SOURCE

Latest Data Collected From Target: Aug 21, 2007 7:55:33 PM CEST

General
- Status: Up
- Since: Aug 21, 2007 2:07:12 PM CEST
- Source: source
- Version: 11.1.0.6.0
- Host: oracleco-vweza
- Listener: LISTENER_oracleco-vweza

Host CPU
- 100%
- 75
- 50
- 25
- 0

Active Sessions
- 1.0
- 0.5
- 0.0

SQL Response Time
- 1.0
- 0.5
- 0.0

Reference collection is empty.

Diagnostic Summary
- ADDM Findings: 0
- Alert Log: No ORA- errors
- Active Incidents: 0

Space Summary
- Database Size (GB): Unavailable
- Problem Tablespace: 0
- Segment Advisor Recommendations: 0
- Policy Violations: 0
- Dump Area Used (%): 88

High Availability
- Instance Recovery Time (sec): 39
- Last Backup: n/a
- Archive Area Used (%): 88
- Flashback Database Logging: Disabled

Load: 1.00  Paging: 7.09

Maximum CPU: 1

View All Properties
Enterprise Manager 10g

- Grid Control 10.2
SQL Developer 1.5.4

• Developer tool at no extra cost
  (Installation deploys V.1.1.3 - Update available via OTN)

• Key-Features:
  • PL/SQL development and debugging
  • Extremely fast object browser
  • Excellent reporting capabilities
  • Supports Oracle 9i and above
  • Migration workbench for:
    • MS SQL-Server
    • MS Access
    • MySQL
  • No extra license costs!
SQL Developer 1.5

- Connections and object browser
APEX 3.0

- APEX = Application Express = ex-HTMLDB
  - GUI development environment
  - Easy to use - wizard driven

- APEX is available by default
  - No additional installation or license necessary
  - Current release: 3.2

- HTTP via "Embedded Gateway"

- Two configuration steps:
  - `$ORACLE_HOME/apex/apxconf.sql`
  - `ALTER USER anonymous ACCOUNT UNLOCK;`
APEX

- APEX web page: apex.oracle.com
Data Pump

• Facts
  • The „new“ export/import
  • Available and usable only for 10g and 11g
  • No backward compatibility to exp/imp
  • But syntax very similar to exp/imp
  • And much faster than exp/imp - parallelism!!!
  • It’s server based only

• Interface
  • Command line: expdp and impdp
  • Package DBMS_DATAPUMP
  • Controllable through interactive Job Control
  • API

• Documentation:
  • Database Utilities and Note:553337.1
Data Pump in Enterprise Manager

Database Instance: SOURCE

Move Row Data
- Export to Export Files
- Import from Export Files
- Import from Database
- Load Data from User Files
- Monitor Export and Import Jobs

Move Database Files
- Clone Database
- Transport Tablespaces

Streams
- Setup
- Manage

Advanced Replication
- Setup
- Manage

Related Links
- Access
- Alert Log Contents
- Blackouts
- Metric and Policy Settings
- Monitor in Memory Access Mode
- SQL Worksheet

- Advisor Central
- All Metrics
- EM SQL History
- Metric Collection Errors
- Policy Groups
- Target Properties

- Alert History
- Baseline Metric Thresholds
- Jobs
- Monitoring Configuration
- Scheduler Central
- User-Defined Metrics
Data Pump

• Step-by-step
  1) Directory in which Oracle OS user has write permissions
  2) Default DATA_PUMP_DIR already defined – may be changed in 11g: DBA_DIRECTORIES
  3) Grant non-privileged users permissions:
     grant READ,WRITE on directory DPDIR to SCOTT;
  4) Example of a full database export:
     expdp system/manager
     DUMPFILE=DATA_PUMP_DIR:full%u.dmp
     FULL=y NOLOGFILE=y
Data Pump

• Step-by-step

5) Enter the interactive job control by pressing CTRL-C:

```
Export> status

Job: SYS_EXPORT_FULL_01
  Operation: EXPORT
  Mode: FULL
  State: EXECUTING
  Bytes Processed: 0
  Current Parallelism: 1
  Job Error Count: 0
  Dump File: /oracle/u01/app/oracle/product/10.2/admin/ORCL/dpdump/full.dmp
    bytes written: 16,445,440

Worker 1 Status:
  Process Name: DWO1
  State: EXECUTING
  Object Schema: SYSMAN
  Object Name: MGMT_METRICS
  Object Type: DATABASE_EXPORT/SHEMA/TABLE/TABLE_DATA
  Completed Objects: 2
  Total Objects: 721
  Worker Parallelism: 1
```
Data Pump

- Step-by-step

6) Commands available in job control mode:

Export> help
The following commands are valid while in interactive mode.
Note: abbreviations are allowed

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD_FILE</td>
<td>Add dumpfile to dumpfile set.</td>
</tr>
<tr>
<td>CONTINUE_CLIENT</td>
<td>Return to logging mode. Job will be re-started if idle.</td>
</tr>
<tr>
<td>EXIT_CLIENT</td>
<td>Quit client session and leave job running.</td>
</tr>
<tr>
<td>FILESIZE</td>
<td>Default filesize (bytes) for subsequent ADD_FILE commands.</td>
</tr>
<tr>
<td>HELP</td>
<td>Summarize interactive commands.</td>
</tr>
<tr>
<td>KILL_JOB</td>
<td>Detach and delete job.</td>
</tr>
<tr>
<td>PARALLEL</td>
<td>Change the number of active workers for current job.</td>
</tr>
<tr>
<td></td>
<td>PARALLEL=&lt;number of workers&gt;.</td>
</tr>
<tr>
<td>REUSE_DUMPFILES</td>
<td>Overwrite destination dump file if it exists (N).</td>
</tr>
<tr>
<td>START_JOB</td>
<td>Start/resume current job.</td>
</tr>
<tr>
<td>STATUS</td>
<td>Frequency (secs) job status is to be monitored where the default (0) will show new status when available.</td>
</tr>
<tr>
<td></td>
<td>STATUS[]=interval</td>
</tr>
<tr>
<td>STOP_JOB</td>
<td>Orderly shutdown of job execution and exits the client.</td>
</tr>
<tr>
<td></td>
<td>STOP_JOB=IMMEDIATE performs an immediate shutdown of the Data Pump job.</td>
</tr>
</tbody>
</table>
Data Pump

• Step-by-step
  7) Attach to a running job and stop it:
    • Hit CTRL-C to enter the interactive job control
    • Job name:
      ```
      Export> status
      Job: SYS_EXPORT_FULL_01
          Operation: EXPORT
          Mode: FULL
      ```
    • Attach:
      ```
      $ expdp system/oracle attach=SYS_EXPORT_FULL_01
      ```
    • Stop the job:
      ```
      Export> stop_job
      Are you sure you wish to stop this job ([yes]/no): yes
      ```
Data Pump

• Behind the scenes
  • Starting data pump creates a master table in the database user schema of the user calling `expdp`
    • By default the table name is identical to the job name
  • Master table acts as the "commemoration" and is essential for the job control
  • At the end of a run of `expdp` the master table will be moved into the dump file
  • Metadata will be unloaded as XML and can be extracted easily
Data Pump Capabilities

• Some important features
  • ReMapping of table spaces, files and schemas
  • Extract all DDLs from a dump file
  • Include or exclude specific objects/users with EXCLUDE and INCLUDE - wildcard usage possible
  • Import via a network without creating a dump file
    • `impdp system/orc NETWORK_LINK=db_link1 FULL=Y`
  • Compression - dump file size reduction by 50% or more possible
    • `expdp system/orc DUMPFILE=DPDIR:full.dmp FULL=y NOLOGFILE=y COMPRESSION=all`
  • Advanced Compression Option license required
  • Encryption - encrypt dump files or data - requires Advanced Security Option
    • `expdp system/orc DUMPFILE=DPDIR:enc.dmp ENCRYPTION_PASSWORD=secret ENCRYPTION_ALGORITHM=AES128`
Agenda

Preparation

Upgrade

News and Task List

Diagnostics & Tuning

Performance Testing

Best Practices

Parameters

Processes

Miscellaneous

Tools

Automation

LOB & SecureFiles

Flashback

ADR

Compression
Automation in 11g

- Predefined Jobs: `DBA_SCHEDULER_JOBS`

```sql
SQL> select JOB_NAME,ENABLED from dba_scheduler_jobs;

<table>
<thead>
<tr>
<th>JOB_NAME</th>
<th>ENABL</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLDB_NFS_CLEANUP_JOB</td>
<td>FALSE</td>
</tr>
<tr>
<td>BSLN_MAINTAIN_STATS_JOB</td>
<td>TRUE</td>
</tr>
<tr>
<td>DRA_REEVALUATE_OPEN_FAILURES</td>
<td>TRUE</td>
</tr>
<tr>
<td>HM_CREATE_OFFLINE_DICTIONARY</td>
<td>TRUE</td>
</tr>
<tr>
<td>ORA$AUTOTASK_CLEAN</td>
<td>TRUE</td>
</tr>
<tr>
<td>AUTO_SPACE_ADVISOR_JOB</td>
<td>FALSE</td>
</tr>
<tr>
<td>GATHER_STATS_JOB</td>
<td>FALSE</td>
</tr>
<tr>
<td>FGR$AUTOPURGE_JOB</td>
<td>FALSE</td>
</tr>
<tr>
<td>PURGE_LOG</td>
<td>TRUE</td>
</tr>
<tr>
<td>RLM$SCHDNEGACITION</td>
<td>TRUE</td>
</tr>
<tr>
<td>RLM$EVTCLEANUP</td>
<td>TRUE</td>
</tr>
<tr>
<td>MGMT_STATS_CONFIG_JOB</td>
<td>TRUE</td>
</tr>
<tr>
<td>MGMT_CONFIG_JOB</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
```

Disable them in 11.1.0.6:
```sql
exec dbms_scheduler.disable('DRA_REEVALUATE_OPEN_FAILURES');
```
Automation in 11g

- Jobs – Default Maintenance Windows

<table>
<thead>
<tr>
<th>Name</th>
<th>Resource Plan</th>
<th>Enabled</th>
<th>Next Open Date</th>
<th>End Date</th>
<th>Duration (min)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEEKNIGHT_WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>FALSE</td>
<td>Aug 14, 2007 10:00:00 PM</td>
<td></td>
<td>480</td>
<td>Weeknight window for maintenance task</td>
</tr>
<tr>
<td>WEEKEND WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>FALSE</td>
<td>Aug 18, 2007 12:00:00 AM</td>
<td></td>
<td>2880</td>
<td>Weekend window for maintenance task</td>
</tr>
<tr>
<td>THURSDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 23, 2007 10:00:00 PM</td>
<td></td>
<td>240</td>
<td>Thursday window for maintenance tasks</td>
</tr>
<tr>
<td>FRIDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 24, 2007 10:00:00 PM</td>
<td></td>
<td>240</td>
<td>Friday window for maintenance tasks</td>
</tr>
<tr>
<td>SATURDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 25, 2007 6:00:00 AM</td>
<td></td>
<td>1200</td>
<td>Saturday window for maintenance tasks</td>
</tr>
<tr>
<td>SUNDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 26, 2007 6:00:00 AM</td>
<td></td>
<td>1200</td>
<td>Sunday window for maintenance tasks</td>
</tr>
<tr>
<td>MONDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 27, 2007 10:00:00 PM</td>
<td></td>
<td>240</td>
<td>Monday window for maintenance tasks</td>
</tr>
<tr>
<td>TUESDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 28, 2007 10:00:00 PM</td>
<td></td>
<td>240</td>
<td>Tuesday window for maintenance tasks</td>
</tr>
<tr>
<td>WEDNESDAY WINDOW</td>
<td>DEFAULT MAINTENANCE_PLAN</td>
<td>TRUE</td>
<td>Aug 29, 2007 10:00:00 PM</td>
<td></td>
<td>240</td>
<td>Wednesday window for maintenance tasks</td>
</tr>
</tbody>
</table>

240min = 4h ... 1200min = 20h
Automation in 11g

- Configure automatically defines maintenance jobs

**Task Settings**

- Optimizer Statistics Gathering: Enabled
- Segment Advisor: Enabled
- Automatic SQL Tuning: Enabled

**Maintenance Window Group Assignment**

<table>
<thead>
<tr>
<th>Window</th>
<th>Optimizer Statistics Gathering</th>
<th>Segment Advisor</th>
<th>Automatic SQL Tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRIDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
<tr>
<td>SATURDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
<tr>
<td>SUNDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
<tr>
<td>MONDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
<tr>
<td>TUESDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
<tr>
<td>WEDNESDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
<tr>
<td>THURSDAY WINDOW</td>
<td>Select All</td>
<td>Select All</td>
<td>Select All</td>
</tr>
</tbody>
</table>
Automation in 11g

- Segment Advisor job identifies space to free up
Mechanism: scale down tables in

- Additional storage needed
- Remove old tables
- Maintenance on indexes
Mechanism: scale down tables in:

- High Watermark

Move data

- Online
- Indexes are maintained
- Triggers won't be executed
- But: TS must be ASSM
- Exclusive Table Lock for HWM

`ALTER TABLE <tablename> SHRINK SPACE;`
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

- Parameters
- Processes
- Miscellaneous
- Tools
- Automatisms
- LOB & SecureFiles
- Flashback
- ADR
- Compression
LONG ⇔ LOB Conversion

• LONG data type is still supported, but Oracle recommends conversion to LOB types

Disadvantages of LONG:
• Maximum number of LONG columns per table : 1
• No replication possible with LONG und LONG RAW
• Attention:
  • LONG ⇔ LOB conversion is irreversible
LONG ⇔ LOB Conversion

• Conversion:
  ```sql
  ALTER TABLE long_tab MODIFY ( long_col CLOB );
  ```

• Online Table Redefinition
  • Available for LONG/LOB since 10g
  • Package: DBMS_REDEFINITION
  • Regulation:
    • LONG ⇔ CLOB
    • LONG RAW ⇔ BLOB
  • Conversion is done implicitly
    • More information: Note:251417.1 (includes a complete example)

• Advantage:
  Source table is available during the whole process
SecureFiles

- Securefiles = new LOB storage technology
  - Better performance
  - Additional features: deduplication, encryption

- Examples:

```sql
CREATE TABLE t1 ( a CLOB)
    LOB(a) STORE AS SECUREFILE (DEDUPLICATE);

CREATE TABLE t2 ( a CLOB ENCRYPT USING 'AES128')
    LOB(a) STORE AS SECUREFILE (CACHE);
```

- Tablespace must be ASSM managed

- Additional init.ora parameter

```
DB_SECUREFILE = [NEVER | PERMITTED | ALWAYS | IGNORE]
```
SecureFiles

- Securefiles are fully transparent
  - Data type still a BLOB/CLOB
  - Definition per storage clause or controlled by init Parameter
  - No functional differences
    - Even "deduplication" is fully transparent for developers

- API access:
  PL/SQL (DBMS_LOB), JDBC, .NET, PHP, ...
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Parameters
Processes
Miscellaneous
Tools
Automatisms
LOB & SecureFiles
Flashback
ADR
Compression
Flashback

- Different types of flashback and its sources
  - Flashback Query
  - Versions Query
  - Transaction Query
  - Flashback Table

- Flashback Drop

- Flashback Database

- Total Recall

- Flashback Data Archive

Enabled by default

Has to be enabled
Flashback in EM

• In Enterprise Manager:

  1. Availability

  2. View and Manage Transactions
Restore Points

- Restore point – specifies a jump label
  - Named Restore Point
    - Similar to a bookmark
    - "Can be" - but no guarantee
    - Will be recorded to the control file

  SQL> CREATE RESTORE POINT rpt;
  SQL> FLASHBACK DATABASE TO RESTORE POINT rpt;

- Guaranteed Restore Point
  - Similar to storage snapshots
  - Overrides the FLASHBACK RETENTION_TARGET
  - Attention: A guarantee restore point can stop the whole database 😞

  SQL> CREATE RESTORE POINT grpt
  GUARANTEE FLASHBACK DATABASE;
  SQL> FLASHBACK DATABASE TO RESTORE POINT grpt;
Flashback Query

- Review into the status or to the changes happened to a table
  - Flashback Query
    - Query a specific state (date, SCN ...)
  - Flashback Versions Query
    - Display changes happened in the past
  - Flashback Transaction Query
    - Detect change transactions from within the past

- Example:
  - `select * from EMP as of timestamp ...`
Flashback Query

• Example:
  - Change data:
    > `update EMP set SAL=60000 where ENAME='KING';`
  - Query data BEFORE it has changed:
    > `select * from EMP as of timestamp TO_TIMESTAMP('2005-08-01 20:30:00', 'YYYY-MM-DD HH24:MI:SS') where ENAME='KING';`
    > Result: 5000
  - Change the data back to its original status:
    > `update EMP set SAL=(select SAL from EMP as of timestamp TO_TIMESTAMP('2005-08-01 20:30:00', 'YYYY-MM-DD HH24:MI:SS') where ENAME='KING') where ENAME='KING';`
Flashback Table

- Requirements:
  - Row movement has to be switched on for the table
    - `alter table EMP enable row movement;`

- Purpose:
  - Recover a table point-in-time

- Example:
  - `flashback table EMP to timestamp to_timestamp(...);`

- Annotation:
  - Maintains indexes, takes care on constraints
  - Triggers are disabled
  - Afterwards statistics will be *stale*
Flashback Table

• Example:

  ➤ Change data:

    > delete from EMP;

  ➤ Flashback the table:

    > flashback table EMP to timestamp
       to_timestamp('01.08.2005 21:39:00',
                    'dd.mm.yyyy hh24:mi:ss');

  ➤ Table now has the state of before-delete
Flashback Drop

• Requirements:
  • Init parameter `recyclebin` is set to **ON**
  • Tablespace is locally managed and <> SYSTEM
  • Owner <> SYS, SYSTEM

• Purpose:
  • Recover deleted tables with their dependent objects

• Example:
  ```sql
  > flashback table EMP to before drop;
  ```
Flashback Drop

• How does it work?
  • `drop table`
  • Pointer to the table will be redirected:
    This includes:
    Dependent objects like indexes, constraints, trigger, nested tables, LOB segments and LOB index segments
  • Objects stay in DBA_Views but have a different name: `BIN$$globalUID$version`
  • Table can now be undropped with a single command
Flashback Drop

- Example:
  - Delete a table:
    ```
    > drop table EMP;
    ```
  - Query the `USER_RECYCLEBIN`:
    ```
    > |
    | OBJECT_NAME        | ORIG | OPER  | TYPE | CAN |
    |-------------------|------|-------|------|-----|
    | BIN$/U0DO44XueXgMFaMimgQ==$0 | EMP  | DROP  | TABLE | YES |
    ```
  - Check the table contents:
    ```
    > select * from "BIN$/U0DO44XueXgMFaMimgQ==$0"
    ```
  - Recover the table:
    ```
    > flashback table EMP to before drop;
    > flashback table "BIN$/..." to before drop;
    > flashback table EMP to before drop rename to EMP_version1;
    ```
Flashback Drop

- Sometimes there’s a misunderstanding ...
  - Tablespace won’t overflow...
Flashback **Drop**

- Search the recyclebin in SQL Developer:

  ![Diagram of SQL Developer interface showing a search for a table named T1 in the recyclebin with details such as object name and operation.](image)
Flashback Database

• How does it work?
  • **Before-Images** of changed data blocks will be written into flashback logs in intervals
  • Not every changed block will be written into the flashback logs
    - Therefore the database has to be in archivelog mode
• Example:
  ```
  > flashback database to scn 47110815;
  ```
• Annotation:
  Flashback logs have to be in the flash recovery area
Flashback Database

• Example:
  ➤ User dropped accidentally:
    > drop user SCOTT cascade;
  ➤ Flashback the whole database:
    > shutdown immediate;
    > flashback database to scn 47110815;
    > alter database open resetlogs;
Flashback Database

- (Almost) as fast as lightning ... 😊
Flashback Database

• Typical questions:
  • Q: How much flashback data will be generated?
    A: About the same amount of data as for redologs
  • Q: Are the logs cleared automatically?
    A: Yes, flashback logs get cleared automatically to reclaim space
  • Q: Can Flashback Logs be administered?
    A: No !!!
  • Q: Is this available in a Standard Edition (one) database, too?
    A: No, Flashback Database is a EE feature
  • Q: How can do performance tuning?
    A: Switch off OS file caching, use lots of disc spindles, use a small stripe-size (~128 kB)
Flashback Data Archive - Total Recall

• How does it work?
  • Background process FBDA will track data changes (only UPDATEs and DELETEs) on monitored tables in the Flashback Data Archive (asynchronous)
  • Data gets extracted from undo information
  • Default capture interval: 5min
  • Capture interval will be changed automatically upon system work load
  • Licensable option
Flashback Data Archive - Total Recall

• Requirements:
  • Tablespace for FBDA has to be ASSM managed
  • Ideally separate tablespaces get created for the flashback data archives and reside on cheaper storage for cost efficiency

• Create a flashback data archive:

```
CREATE FLASHBACK ARCHIVE fda1
  TABLESPACE fda
  RETENTION 1 year;
```

• Switch on flashback archiving for a specific object:

```
ALTER TABLE t1 FLASHBACK ARCHIVE fda1;
```
Flashback Data Archive - Total Recall

• Behind the scenes:
  • In tablespace *fda* some structures will be created:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS_FBA_DDL_COLMAP_23107</td>
<td>TABLE</td>
</tr>
<tr>
<td>SYS_FBA_HIST_23107</td>
<td>TABLE PARTITION</td>
</tr>
<tr>
<td>SYS_FBA_TCRV_23107</td>
<td>TABLE</td>
</tr>
<tr>
<td>SYS_FBA_TCRV_IDX_23107</td>
<td>INDEX</td>
</tr>
</tbody>
</table>

• 23107 is the OBJECT_ID of the table getting monitored/archived

Tracks the changes

• If the defined retention is reached the required data will be deleted automatically
• Absolutely transparent - no changes to any application necessary
• Partitioning happens automatically
Flashback Data Archive - Total Recall

• Is it possible to change data in the FDA?
  • Data will be stored such as:

<table>
<thead>
<tr>
<th>ROWID</th>
<th>STARTSCN</th>
<th>ENDSCN</th>
<th>O</th>
<th>COL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAAFplAAGAAAAAVAAA</td>
<td>1114797</td>
<td>1115141</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>AAAFplAAGAAAAAVAAB</td>
<td>1114797</td>
<td>1115141</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>AAAFplAAGAAAAAVAAC</td>
<td>1114797</td>
<td>1115141</td>
<td>I</td>
<td>23</td>
</tr>
<tr>
<td>AAAFplAAGAAAAAVAAD</td>
<td>1114797</td>
<td>1115141</td>
<td>I</td>
<td>24</td>
</tr>
<tr>
<td>AAAFplAAGAAAAAVAEE</td>
<td>1114797</td>
<td>1115141</td>
<td>I</td>
<td>25</td>
</tr>
<tr>
<td>AAAFplAAGAAAAVAAAF</td>
<td>1115141</td>
<td>1116039</td>
<td>U</td>
<td>2</td>
</tr>
</tbody>
</table>
  ...

• Data can’t be manipulated:

```
SQL> update SYS_FBA_HIST_23107 set col1=col1-1;
update SYS_FBA_HIST_23107 set col1=col1-1
* 
ERROR at line 1:
ORA-55622: DML, ALTER and CREATE UNIQUE INDEX operations are not allowed on table "SYS"."SYS_FBA_HIST_23107"
```
Flashback Data Archive - Total Recall

- How to access the historical data?

```
SELECT * FROM t1
    AS OF TIMESTAMP '27-AUG-07 2:15:00pm';

<table>
<thead>
<tr>
<th>COL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>23</td>
</tr>
</tbody>
</table>

SELECT * FROM t1
    AS OF TIMESTAMP '27-AUG-07 2:30:00pm';

<table>
<thead>
<tr>
<th>COL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>24</td>
</tr>
</tbody>
</table>
```
Flashback Data Archive - Total Recall

- Switch background process FBDA off?
  - SQL> ALTER SYSTEM SET 
    "_disable_flashback_archiver"=1 
    scope=spfile;
  - SQL> STARTUP FORCE
  - $ ps -ef | grep fbda
Agenda

1. Preparation
2. Upgrade
3. News and Task List
   - Parameters
   - Processes
   - Miscellaneous
   - Tools
   - Automatisms
4. Diagnostics & Tuning
   - LOB & SecureFiles
   - Flashback
   - ADR
   - Compression
5. Performance Testing
6. Best Practices
Automatic Diagnostic Repository

- ADR: systemwide logging and tracing directory
  - Set via `diagnostic_dest` parameter
  - Default: `ORACLE_BASE\diag`
Automatic Diagnostic Repository

- ADR: systemwide logging/tracing directory
  - "Oh ... my alert.log is now in XML format ...?"

- No panic:
  - Files are still kept also in text mode
  - There are two very comfortable interfaces:
    - *adrci*
    - DatabaseControl

- Error categories
  - Problem (Critical error like ORA-600, ORA-7445, ORA-4031 ...)
  - Incident (Singular error - if it gets raised more often it'll become a PROBLEM!!)
### V$DIAG_INFO:

```
SQL> SELECT name, value FROM v$diag_info;

<table>
<thead>
<tr>
<th>NAME</th>
<th>WERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diag Enabled</td>
<td>TRUE</td>
</tr>
<tr>
<td>ADR Base</td>
<td>c:\oracle</td>
</tr>
<tr>
<td>ADR Home</td>
<td>c:\oracle\diag\rdbms\source\source</td>
</tr>
<tr>
<td>Diag Trace</td>
<td>c:\oracle\diag\rdbms\source\source\trace</td>
</tr>
<tr>
<td>Diag Alert</td>
<td>c:\oracle\diag\rdbms\source\source\alert</td>
</tr>
<tr>
<td>Diag Incident</td>
<td>c:\oracle\diag\rdbms\source\source\incid</td>
</tr>
<tr>
<td>Diag Cdump</td>
<td>c:\oracle\diag\rdbms\source\source\cdump</td>
</tr>
<tr>
<td>Health Monitor</td>
<td>c:\oracle\diag\rdbms\source\source\hm</td>
</tr>
<tr>
<td>Default Trace File</td>
<td>c:\oracle\diag\rdbms\source\source\trace</td>
</tr>
<tr>
<td>Active Problem Count</td>
<td>1</td>
</tr>
<tr>
<td>Active Incident Count</td>
<td>7</td>
</tr>
</tbody>
</table>
```
Automatic Diagnostic Repository

- `adrci`

**Step 1: Check and set ADRCI homes**
- `adrci> show homes`
- `adrci> set homepath diag\rdbms\source\source`

**Step 2: Define the editor, e.g.:**
- `adrci> set editor uedit32`

**Step 3: Check alert.log, problems and incidents**
- `adrci> show alert -tail 50 -f`
- `adrci> show problem`
- `adrci> show incident`
Automatic Diagnostic Repository

• adrci
  • Helpful notes:
    • Note: 443529.1
      11g How To Package And Send ORA-00600/ORA-07445 Diagnostic Information To Support.
    • Note: 564269.1
      Retention Policy for ADR Incidents
      • SHORTP_POLICY:
        Used to purge informations that have a short life.
        Default: 720 => 30 days
      • LONGP_POLICY:
        To purge ADR contents that have a long life.
        Default: 8760 => 365 days
Automatic Diagnostic Repository

- DatabaseControl

[Image of the Oracle Enterprise Manager 11g interface showing the Software and Support section and the Support Workbench and Performance and Critical Error sections.]
Automatic Diagnostic Repository

Summary

<table>
<thead>
<tr>
<th>SR#</th>
<th>Active</th>
<th>Packaged</th>
<th>Number of Incidents</th>
<th>First Incident</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>7</td>
<td>August 27, 2007 2:56:35 PM CEST</td>
</tr>
</tbody>
</table>

Last Incident

- Timestamp: August 27, 2007 2:57:36 PM CEST
- Incident Source: System Generated
- Impact: 0
- Checkers Run: 0
- Checkers Findings: 0

Incorporate additional details as needed.
Agenda

- Preparation
- Upgrade
- News and Task List
  - Parameters
  - Processes
  - Miscellaneous
  - Tools
  - Automatisms
  - LOB & SecureFiles
  - Flashback
  - ADR
  - Compression
- Diagnostics & Tuning
- Performance Testing
- Best Practices
Compression

- Index compression available since 8i
- Table compression available since 9.2
  - Designated for DWH
  - Direct path and bulk INSERTs, CREATE TABLE .. AS SELECT
  - Didn't work for INSERT/UPDATE/DELETE
- Before 11g syntax:
  
  ```sql
  CREATE TABLE sales_history... COMPRESS
  ```

- New 11g syntax:
  
  ```sql
  CREATE TABLE sales_history...
  COMPRESS FOR DIRECT_LOAD OPERATIONS
  ```
Advanced Compression

- Advanced Compression in Oracle 11g works for:
  - Structured and unstructured data
  - Backups
  - Datapump exports
  - Data Guard gap resolution (11.1.0.7 => LGWR ASYNC)

- Reduces resource requirements and costs!!!
  - Storage
  - Memory

- Typically 2x to 4x compression possible

Diagram:

- DML uncompressed
- PCTFREE reached: Compression!
- Further DML uncompressed
- PCTFREE reached: Compression!

Legend:
- Overhead
- Free Space
- Uncompressed
- Compressed
‘Real World’ Compression

10 largest ERP database tables

3x Savings

Table Scans

DML Performance

< 3% overhead
Compression Advisor - Free Download


Oracle Advanced Compression

Advanced Compression, an option introduced in Oracle Database 11g Enterprise Edition, offers a comprehensive set of compression capabilities to help organizations reduce costs, while maintaining or improving performance. It significantly reduces the storage footprint of databases through compression of structured data (numbers, characters) as well as unstructured data (documents, spreadsheets, XML and other files). It provides enhanced compression for database backups and also includes network compression capabilities for faster synchronization of standby databases.

Key Benefits

- 2-4X reduction in storage across all environments such as production, standby, test, development, backup, etc.
- Improved memory efficiency as data remains compressed in memory
- Minimal or no performance impact on DML operations due to performance optimized compression technology
- Faster queries in many cases, due to improved I/O and memory efficiency
- Better network bandwidth utilization
- Transparent to applications - no application changes required
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Object Statistics
Monitoring
SQL Tuning
Tuning with the right tools ...
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
  - Object Statistics
  - Performance Monitoring
  - SQL Tuning
- Performance Testing
- Best Practices
Optimizer Statistics Gathering

- DML Monitoring is switched ON for all tables
- INSERTS, UPDATES and DELETES will be monitored
- To exclude tables lock their statistics
  
  ```
  execute
  DBMS_STATS.LOCK_TABLE_STATS('SH','TAB1');
  ```

- For statistics collection always use DBMS_STATS, never use ANALYZE anymore
Optimizer Statistics Gathering

- Default maintenance job

<table>
<thead>
<tr>
<th>Name</th>
<th>Schema</th>
<th>Type</th>
<th>Status</th>
<th>Scheduled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer Statistics Gathering</td>
<td>SYS</td>
<td>Automated Maintenance Tasks</td>
<td>Scheduled</td>
<td>Aug 21, 2007 10:00:00 PM (UTC+02:00)</td>
</tr>
<tr>
<td>Segment Advisor</td>
<td>SYS</td>
<td>Automated Maintenance Tasks</td>
<td>Scheduled</td>
<td>Aug 21, 2007 10:00:00 PM (UTC+02:00)</td>
</tr>
<tr>
<td>Automatic SQL Tuning</td>
<td>SYS</td>
<td>Automated Maintenance Tasks</td>
<td>Scheduled</td>
<td>Aug 21, 2007 10:00:00 PM (UTC+02:00)</td>
</tr>
<tr>
<td>BSIN MAINTAIN STATS JOB</td>
<td>SYS</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>Aug 26, 2007 12:00:00 AM (UTC+02:00)</td>
</tr>
<tr>
<td>DRA REEVALUATE OPEN FAILURES</td>
<td>SYS</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>MAINTENANCE WINDOW GROUP</td>
</tr>
<tr>
<td>HM CREATE OFFLINE DICTIONARY</td>
<td>SYS</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>MAINTENANCE WINDOW GROUP</td>
</tr>
<tr>
<td>MGMT_CONFIG_JOB</td>
<td>ORACLE_OCM</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>MAINTENANCE WINDOW GROUP</td>
</tr>
<tr>
<td>MGMT_STATS_CONFIG_JOB</td>
<td>ORACLE_OCM</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>Sep 1, 2007 1:01:01 AM (UTC+02:00)</td>
</tr>
<tr>
<td>ORA_AUTOTASK_CLEAN</td>
<td>SYS</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>Aug 22, 2007 3:00:00 AM (UTC+02:00)</td>
</tr>
<tr>
<td>PURGE LOG</td>
<td>SYS</td>
<td>Oracle Scheduler</td>
<td>Scheduled</td>
<td>Aug 22, 2007 3:00:00 AM (UTC+02:00)</td>
</tr>
</tbody>
</table>
Optimizer Statistics Gathering

- Configure maintenance schedule

Automated Maintenance Tasks Configuration

Task Settings

- Global Status: Enabled

Optimizer Statistics Gathering: Enabled
Segment Advisor: Disabled
Automatic SQL Tuning: Disabled

Maintenance Window Group Assignment

<table>
<thead>
<tr>
<th>Window</th>
<th>Optimizer Statistics Gathering</th>
<th>Segment Advisor</th>
<th>Automatic SQL Tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRIDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SATURDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SUNDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>MONDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>TUESDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>WEDNESDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>THURSDAY WINDOW</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Optimizer Statistics Gathering

- Configure settings and parameters

Database Instance: SOURCE > Manage Optimizer Statistics >
Global Statistics Gathering Options

- Statistics History
  - Retention Period (days) 31

- Gather Optimizer Statistics Default Options
  - Oracle recommends that you use the Gather Auto choice for the Gather Objects options when you use the Gather Optimizer Statistics process for Database and Schemas. If you choose not to use Gather Auto, the defaults for the other options are set here. Changing the options will impact the automated Optimizer Statistics Gathering task and user defined jobs.

  - Estimate Percentage
  - Degree of Parallelism
  - Granularity
  - Cursor Invalidation
  - Cascade
  - Target Object Class (Auto Job)
  - Stale Percentage 10
  - Incremental
  - Publish
  - Histograms FOR ALL COLUMNS SIZE AUTO
GATHER_STATS_JOB in 11g

• Starting the job manually:
  
  • Start:
    
    ```sql
    exec
    DBMS_AUTO_TASK_IMMEDIATE.GATHER_OPTIMIZER_STATS
    ```

  • Monitoring progress:
    
    ```sql
    SELECT job_name,state
    FROM dba_scheduler_jobs
    WHERE program_name='GATHER_STATS_PROG';
    ```

  • For more information see Note:731935.1
GATHER_STATS_JOB

Table-Monitoring ON → SMON every 3h
manually:
```
SQL> exec DBMS_STATS.FLUSH_DATABASE_MONITORING_INFO();
```

No# of DMLs in: DBAUSER_TAB_MODIFICATIONS:

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>PARTITION_NAME</th>
<th>SUBPARTITION_NAME</th>
<th>INSERTS</th>
<th>UPDATES</th>
<th>DELETE</th>
<th>TIMESTAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP</td>
<td></td>
<td></td>
<td>0</td>
<td>42</td>
<td>0</td>
<td>12.08.05</td>
</tr>
<tr>
<td>EMPTEST</td>
<td></td>
<td></td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>12.08.05</td>
</tr>
</tbody>
</table>

1. Tables without statistics: ➞ EMPTY
2. Table changed >10%: ➞ STALE

Automatic Job – once a day:

<table>
<thead>
<tr>
<th>Optimizer Statistics Gathering</th>
<th>SYS</th>
<th>Automated Maintenance Tasks</th>
</tr>
</thead>
</table>

manually:
```
SQL> exec DBMS_STATS.GATHER_SCHEMA_STATS('SCOTT');
```
GATHER_STATS_JOB

Objects:

Analysis order:

Priority

Statistics: EMPTY
Statistics: STAOLE
Statistics: OK
Optimizer without statistics

- What happens if there are no object statistics?
  - The optimizer (CBO) will estimate (at parsing time) the statistics for tables and indexes
  - Prerequisite:
    - `OPTIMIZER_DYNAMIC_SAMPLING=2` [Default since 10g]
      - Level 0 - Dynamic Sampling off (~OLTP)
      - Level 2 – Optimizer checks first 64 blocks of object
        - For descriptions of all levels: Performance Tuning Guide
  - Example:
Requirements for Diagnostics & Tuning

- **STATISTICS_LEVEL=TYPICAL** [default]
  - Alternative Settings:
    - BASIC
    - ALL
  - **TYPICAL** (recommended setting) will enable:
    - Automatic SGA Tuning
    - Automatic statistics collection
    - Active Session History
    - `_ash_enabled=false` to switch off ASH
    - DML monitoring
Diagnostics & Tuning Concept since 10g

- SGA
  - In-memory statistics
- MMON
- ADDM
  - Alerts
- AWR
  - Snapshots
  - ADDM results
- Reactive: DBA
- Proactive
Tuning Basics since 10g

- **AWR** (Automatic Workload Repository)
  - Statistics repository in the SYSAUX tablespace
  - Contains AWR snapshots (current and past periods)
  - Self managing
  - Data stored in the AWR is collected by MMON
    - Every 60 minutes and kept for 7 (8 in 11g) days [default]
    - Query AWR contents: **DBA_HIST-Views**
  - The automatic removal of certain snapshots can be suppressed for later analysis by defining them as a baseline
  - Snapshots can be created on demand at every time:
    - EXEC dbms_workload_repository.create_snapshot();
  - Size estimation:
    - @?/rdbms/admin/utlsyxsz.sql
  - Report:
    - @?/rdbms/admin/awrr rpt.sql
    - Single statements: @?/rdbms/admin/awrsqrpt.sql
Tuning Basics since 10g

- AWR
  - Change retention and interval:

[Oracle Enterprise Manager 10g interface with highlighted sections for Database Administration and Automatic Workload Repository]
Tuning Basics since 10g

- **AWR**
  - Change retention and interval:
    - Recommendation: 30-60 minutes and 20-31 days

```
begin DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(1440,120); end;
```
Tuning Basics since 10g

- Transport the AWR
  - Extract:
    ```sql
    DBMS_SWRF_INTERNAL.AWR_EXTRACT(
        dmpfile => 'awr_data.dmp', dmpdir => 'TMP_DIR',
        bid => 302, eid => 305);
    ```
  - Load in temporary schema:
    ```sql
    DBMS_SWRF_INTERNAL.AWR_LOAD(
        SCHNAME => 'AWRTEMP',
        dmpfile => 'awr_data', dmpdir => 'TMP_DIR' );
    ```
  - Move temporary data to AWR location (SYS):
    ```sql
    DBMS_SWRF_INTERNAL.MOVE_TO_AWR(
        SCHNAME => 'AWRTEMP_USER');
    ```
- Why?
  - Evaluations outside of production systems
  - Central repository
Tuning Basics since 10g

• **ASH** (Active Session History)
  • ASH is key for database diagnosis
  • Circular buffer inside the SGA
  • Fixed size: $2^{MB} \times \#CPU$s (max. 5% of the SGA or <30MB)
  • Contains information about active sessions
  • ~10% of the information gets written to the AWR by MMNL
  • View: `V$ACTIVE_SESSION_HISTORY`
  • Further info: *Note:243132.1*
  • Report:
    • `@?/rdbms/admin/ashrpt.sql`
Tuning Basics since 10g

• Define ASH reporting interval

Database Instance: V11G

Run ASH Report
Specify the time period for the report.

Start Date 3/13/09
(Example: 12/15/03)

Start Time 11:58 AM

End Date 3/13/09
(Example: 12/15/03)

End Time 12:02 PM

Generate Report
Tuning Basics since 10g

- ASH Report For V11G/v11g

<table>
<thead>
<tr>
<th>DB Name</th>
<th>DB Id</th>
<th>Instance</th>
<th>Inst num</th>
<th>Release</th>
<th>RAC</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>V11G</td>
<td>2840163171</td>
<td>v11g</td>
<td>1</td>
<td>11.1.0.7.0</td>
<td>NO</td>
<td>VMTKEXP1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPUs</th>
<th>SGA Size</th>
<th>Buffer Cache</th>
<th>Shared Pool</th>
<th>ASH Buffer Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>511M (100%)</td>
<td>156M (30.5%)</td>
<td>168M (32.9%)</td>
<td>2.0M (0.4%)</td>
</tr>
</tbody>
</table>

**ASH Report**

- **Sample Time**
  - Analysis End Time: 13-Mar-09 12:02:13
  - Elapsed Time: 4.0 (mins)
  - Sample Count: 91
  - Average Active Sessions: 0.38
  - Avg. Active Session per CPU: 0.38
  - Report Target: None specified

- **Data Source**
  - V$ACTIVE_SESSION_HISTORY

---

**ORACLE**
Tuning Basics - STATSPACK

- **STATSPACK** is still available in 10g and 11g
  - Use either AWR/ASH (Diagnostic Pack license required) or STATSPACK
  - See Note:394937.1 (STATSPACK Guide)
Tuning Basics since 10g

- **ADDM** (Automatic Database Diagnostic Monitor)
  - Knowledge base for performance diagnostics
  - Analyze performance snapshots and give recommendations
  - Runs proactively every hour (upon AWR snapshot) or can be invoked manually:
    ```sql
    SQL> @$ORACLE_HOME/rdbms/admin/addmrpt.sql
    SQL> @$ORACLE_HOME/rdbms/admin/addmrpti.sql (RAC)
    ```
Tuning Basics since 10g

• ADDM findings – example:

<table>
<thead>
<tr>
<th>Database Time (minutes)</th>
<th>Period Start Time</th>
<th>Period Duration (minutes)</th>
<th>Average Active Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.7</td>
<td>May 8, 2006 5:19:35 PM CEST</td>
<td>70.5</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Task Owner: SYS
Task Name: TASK_1326

Finding: Individual database segments responsible for significant user I/O wait were found.
Impact (minutes): 1.6
Impact (%): 11.5

Recommendations

- Investigate application logic involving I/O on TABLE "TPCC.STOCK" with object id 53361.
- Run "Segment Advisor" on TABLE "TPCC.STOCK" with object id 53361.

Rationale: The I/O usage statistics for the object are: 6 full object scans, 51273 physical reads, 1 physical writes and 0 direct reads.

Findings Path

Expand All | Collapse All

Findings

- Individual database segments responsible for significant user I/O wait were found.
- Wait class 'User I/O' was consuming significant database time.

Impact (%): 11.5
Additional Information

Impact (%): 35.6
Additional Information
Memory Tuning - SGA & PGA

OLTP

Streams Pool

Buffer Cache

Large Pool

SQL Cache

Java Pool

sort

Batch

Streams Pool

Buffer Cache

Large Pool

SQL Cache

Java Pool

sort

SGA

PGA

ORACLE
Automatic Shared Memory Management

• To enable ASMM SGA_TARGET (or MEMORY_TARGET) and STATISTICS_LEVEL=TYPICAL must be set

SGA_MAX_SIZE=...
Automatic Shared Memory Management

- Statistics collection driven by MMAN in the background

Diagram:
- MMAN coordinates size of each single SGA component
- Background: SGA Memory Broker
- Monitors all components and coordinates resize operations
DB Management Pack Enhancements 11g

- **Diagnostic Pack**
  - ADDM for RAC
  - AWR Baselines and Adaptive Thresholds
  - Transportable AWR

- **Tuning Pack**
  - Automatic SQL Tuning Advisor
  - Partition Advisor
  - Real-time SQL Monitoring
  - SQL Plan Management and Plan Evolution
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
  - Object Statistics
  - Performance Monitoring
  - SQL Tuning
- Performance Testing
- Best Practices
DB Management Pack Enhancements

• Diagnostic Pack
  • ADDM for RAC
  • AWR Baselines and Adaptive Thresholds
  • Transportable AWR

• Tuning Pack
  • Real-time SQL Monitoring
  • Automatic SQL Tuning Advisor
  • Partition Advisor
SQL Real-Time Monitoring

- Dedicated statistics collected for a single execution of a SQL statement when its execution becomes high-load
  - Target all parallel queries, parallel DML or parallel DDL
  - Target serial execution that exceed 5 sec of CPU or I/O time
- Global SQL level statistics are collected: `V$SQL_MONITOR`
- Plan level statistics are collected (#rows, memory, temp space, start/end date): `V$SQL_PLAN_MONITOR`
- Statistics are updated quasi real-time while the query executes
- Statistics for completed executions are retained for at least 5 minutes
- On by default
SQL Real-Time Monitoring Report

- `DBMS_SQLTUNE.REPORT_SQL_MONITOR()` allows to display monitoring information
  - Format: xml, text and html
  - Join with ASH data
- SQL Real-time monitoring with GUI integrated in 11.1.0.7

Additional Monitoring Links
Top Sessions and Top SQL data from ASH can be found on the Top Activity page.

- Top Activity
- Top Consumers
- Duplicate SQL
- Blocking Sessions
- Hang Analysis
- Instance Locks
- Instance Activity
- Search Sessions
- Search SQL
- Snapshots
- AMR Baselines
- SQL Tuning Sets
- SQL Performance Analyzer
- SQL Monitoring

Database Instance: V11G > Monitored SQL Executions

<table>
<thead>
<tr>
<th>Status</th>
<th>Duration</th>
<th>SQL ID</th>
<th>Session</th>
<th>Parallel</th>
<th>Database Time</th>
<th>ID</th>
<th>Start</th>
<th>Ended</th>
<th>SQL Text</th>
</tr>
</thead>
</table>
| ✔      | 0.0s     | 12345678 | 88      |          | 0.7s          | 2155| 12:40:06 AM | 11:40:14 AM | `select /*+ use_nil(6)...

ORACLE Enterprise Manager 11g Database Control

Logged In As SYS
SQL Real-Time Monitoring Report

- Example for an **active** and long running PQ statement:

```
SELECT /*+ parallel(2) */ count(*) FROM test2.test3 test1 WHERE test1.obj# = 123456789
```

### Global Information: EXECUTING

<table>
<thead>
<tr>
<th>Instance ID</th>
<th>Session ID</th>
<th>SQL ID</th>
<th>SQL Execution ID</th>
<th>Plan Hash Value</th>
<th>Execution Started</th>
<th>First Refresh Time</th>
<th>Last Refresh Time</th>
<th>Fetch Cells</th>
</tr>
</thead>
</table>

### Parallel Execution Details (OPP=2)

<table>
<thead>
<tr>
<th>PX Coordinator</th>
<th>Type</th>
<th>Server#</th>
<th>Buffer Gets</th>
<th>IO Count</th>
<th>Database Time</th>
<th>Wait Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>Set 1</td>
<td>1</td>
<td>3262 10 (6.3%)</td>
<td>2078 10 (43%)</td>
<td>D (0.1%)</td>
<td>31 (31%)</td>
</tr>
<tr>
<td>p001</td>
<td>Set 1</td>
<td>2</td>
<td>322 10 (6.2%)</td>
<td>2078 10 (43%)</td>
<td>D (0.2%)</td>
<td>21 (21%)</td>
</tr>
<tr>
<td>p002</td>
<td>Set 2</td>
<td>1</td>
<td>2078 10 (43%)</td>
<td>2078 10 (43%)</td>
<td>D (0.2%)</td>
<td>21 (21%)</td>
</tr>
<tr>
<td>p001</td>
<td>Set 2</td>
<td>2</td>
<td>2078 10 (43%)</td>
<td>2078 10 (43%)</td>
<td>D (0.2%)</td>
<td>21 (21%)</td>
</tr>
</tbody>
</table>

### SQL Plan Monitoring Details:

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Estimated Rows</th>
<th>Cost</th>
<th>Active Period (s)</th>
<th>Starts</th>
<th>Actual Rows</th>
<th>Memory</th>
<th>Temp</th>
<th>CPU Activity</th>
<th>Wait Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT</td>
<td></td>
<td>3000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SORT AGGREGATE</td>
<td>500</td>
<td>1</td>
<td>0</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PX COORDINATOR</td>
<td>TQ10802</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PX SEND QC (RANDOM)</td>
<td>TQ10802</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SORT AGGREGATE</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MERGE JOIN</td>
<td>1750K</td>
<td>30043</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SORT JOIN</td>
<td>1750K</td>
<td>17321</td>
<td>2</td>
<td>2</td>
<td>1230K</td>
<td>25156K</td>
<td>33%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PX RECEIVE</td>
<td>1750K</td>
<td>5655</td>
<td>3</td>
<td>1915K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>PX SEND HASH</td>
<td>TQ10802</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PX BLOCK ITERATOR</td>
<td>1750K</td>
<td>5585</td>
<td>2</td>
<td>1915K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TABLE ACCESS FULL</td>
<td>TESTMON</td>
<td>1750K</td>
<td>5585</td>
<td>20</td>
<td>1915K</td>
<td></td>
<td>8.3%</td>
<td>25%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>SORT JOIN</td>
<td>1750K</td>
<td>17321</td>
<td>2</td>
<td>0</td>
<td>1512K</td>
<td>27256K</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>PX RECEIVE</td>
<td>1750K</td>
<td>5655</td>
<td>2</td>
<td>1915K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PX SEND HASH</td>
<td>TQ10802</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>PX BLOCK ITERATOR</td>
<td>1750K</td>
<td>5585</td>
<td>2</td>
<td>1915K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TABLE ACCESS FULL</td>
<td>TESTMON</td>
<td>1750K</td>
<td>5585</td>
<td>26</td>
<td>1915K</td>
<td></td>
<td>8.3%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*Oracle*
Manual SQL Tuning Challenges

• Complex - requires expertise in several domains
  • SQL optimization: adjust the execution plan
  • Access design: provide fast data access
  • SQL design: use appropriate SQL constructs

• Time consuming
  • Each SQL statement is unique
  • Potentially large number of statements to tune

• Never ending task
  • SQL workload always evolving
  • Plan regressions can happen
SQL Tuning in Oracle Database 10g

Workload → AWR → ADDM

Invoke Advisor → SQL Tuning Candidates

Evaluate Recommendations → Implement

DBA

Generate Recommendations

Some meaningful automation, but the DBA is still required
SQL Tuning Automation in 11g

- Fully automatic SQL Tuning task
  - Runs automatically:
    - Maintenance Window, CPU resource controlled, on/off switch
    - Identifies, ranks and tunes candidate SQL
    - Leverages SQL Tuning Advisor

- Candidate SQL automatically chosen
  - Excluded: parallel queries, DML/DDL, recursive, ad-hoc (infrequent)

- Tests and (optionally) implements SQL profiles
  - Only implements significantly improved plans (3x)
  - Time budgeted to avoid run-away plans

- DBA can still invoke the advisor manually for reactive tuning, like in 10g
SQL Tuning Automation in 11g

- Configure Automatic SQL Tuning
### Automatic SQL Tuning Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Time Spent Per SQL During Tuning (sec)</td>
<td>1200</td>
</tr>
<tr>
<td>Automatic Implementation of SQL Profiles</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum SQL Profiles Implemented Per Execution</td>
<td>20</td>
</tr>
<tr>
<td>Maximum SQL Profiles Implemented (Overall)</td>
<td>10000</td>
</tr>
</tbody>
</table>

**TIP** You need to login as SYS to make the change.
Automatic SQL Tuning in 11g

1. Workload
2. Implement SQL Profiles
3. Test SQL Profiles
4. Generate Recommendations
5. Choose Candidate SQL
6. SQL Tuning Candidates
7. It’s Automatic!
8. View Reports / Control Process
9. AWR
10. DBA
11. one week
Result Summary

Automatic SQL Tuning Result Summary
The Automatic SQL Tuning runs during system maintenance windows as an automated maintenance task, searching for ways to improve the execution plans of high-load SQL statements.

Task Status
Automatic SQL Tuning (SYS_AUTO_SQL_TUNING_TASK) is currently Enabled [Configure]
Automatic Implementation of SQL Profiles is currently Enabled

Task Activity Summary
The activity summary graph shows the benefit of the task activities on the systems high-load SQL. Only profiles that significantly improve SQL performance were implemented.

Time Period: All [Go] View Report

Begin Date: Jan 28, 2007 6:00:01 AM (UTC-08:00)  End Date: Feb 8, 2007 10:03:10 AM (UTC-08:00)

Overall Task Statistics
Executions: 10  Candidate SQL: 668  Distinct SQL Examined: 106

SQL Examined Status

Breakdown by Finding Type
Number of SQL

- SQL Profile
- Index
- Statistics
- Restructure SQL

Not implemented Implemented
SQL Tuning - step-by-step

1. Monitoring
2. Identify statement(s) requiring tuning
3. Use tuning advisors
   - SQL Access Advisor:
     - Indexes
     - Materialized Views
     - Indexes on Materialized Views
     - Requires a workload
   - SQL Tuning Advisor
     - Optimizer in Tuning Mode
     - Create profiles for the statements
     - Can operate on a single SQL statement
4. Implement recommendations
SQL Tuning Advisor

• Can be used in EM or on CLI (via DBMS_SQLTUNE)
• 2 Different tuning modes are available:
  • “Comprehensive mode” runs through all possible analysis options
  • “Limited mode” will not generate a SQL Profile
• SQL Profiles contain information that lead to improved execution plans without changing the application code
  • Use different optimizer settings
  • Correct wrong/missing statistics
  • Correct wrong estimates
  • SQL Profiles are persistent
  • SQL Profiles don’t change the original SQL statement
  • SQL Profiles can be transported within SQL Tuning Sets (STS)
  • SQL Profiles can be tested and verified without any risk
SQL Tuning - Sources

Automatic selection

AWR → ADDM → High-load SQL

Manual choice

SQL Sources

AWR → Cursor Cache → User-defined

Filter / Rank → SQL Tuning Set (STS)

SQL Tuning Advisor
**SQL Tuning**

- Example: Results „Tuning Advisor“ ➔ SQL Profile

Recommendations for SQL ID:40yqk9cdgkgk

Only one recommendation should be implemented.

**SQL Text**

```sql
SELECT /*+ use_n(c) ordered */ COUNT(*) FROM sh sales s, sh customers c WHERE c.cust_id=s.cust_id AND cust_first_name='Dina'
```

**Select Recommendation**

<table>
<thead>
<tr>
<th>Select Type</th>
<th>Findings</th>
<th>Recommendations</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Profile</td>
<td>A potentially better execution plan was found for this statement.</td>
<td>Consider accepting the recommended SQL profile.</td>
<td>99.77</td>
</tr>
<tr>
<td>Index</td>
<td>The execution plan of this statement can be improved by creating one or more indices.</td>
<td>Consider running the Access Advisor to improve the physical schema design or creating the recommended index. SH CUSTOMERS(&quot;CUST_FIRST_NAME&quot;) SH SALES(&quot;CUST_ID&quot;)</td>
<td>59.66</td>
</tr>
</tbody>
</table>

**Implement**

- [Original Explain Plan (Annotated)](URL)

- [Implement Plan](URL)

- [Compare Explain Plans](URL)
#### SQL Tuning

- **Example: Compare original to new explain plan**

**Compare Explain Plans**

**Original Explain Plan (Annotated)**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Line ID</th>
<th>Object</th>
<th>Object Type</th>
<th>Order</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost</th>
<th>Time</th>
<th>CPU Cost</th>
<th>I/O Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td>0</td>
<td></td>
<td></td>
<td>9</td>
<td>0.017</td>
<td></td>
<td>919,732</td>
<td>11,037</td>
<td>8,498,774,016</td>
<td>919,271</td>
</tr>
<tr>
<td>SORT AGGREGATE</td>
<td>1</td>
<td></td>
<td></td>
<td>8</td>
<td>0.017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td>2</td>
<td></td>
<td></td>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td>3</td>
<td></td>
<td></td>
<td>5</td>
<td>15.552</td>
<td>919,732</td>
<td>11,037</td>
<td>8,498,774,016</td>
<td>919,271</td>
<td></td>
</tr>
<tr>
<td>PARTITION RANGE ALL</td>
<td>4</td>
<td></td>
<td></td>
<td>3</td>
<td>4,486,538</td>
<td></td>
<td>428</td>
<td>6</td>
<td>7,341,376</td>
<td>428</td>
</tr>
<tr>
<td>BITMAP CONVERSION TO ROWIDS</td>
<td>5</td>
<td></td>
<td></td>
<td>2</td>
<td>4,486,538</td>
<td></td>
<td>428</td>
<td>6</td>
<td>7,341,376</td>
<td>428</td>
</tr>
<tr>
<td>BITMAP INDEX FAST FULL SCAN</td>
<td>6</td>
<td>SH_SALES_CUSTOMER_PK</td>
<td>INDEX (BITMAP)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX UNIQUE SCAN</td>
<td>7</td>
<td>SH_CUSTOMERS_PK</td>
<td>INDEX (UNIQUE)</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>0</td>
<td>SH_CUSTOMERS</td>
<td>TABLE</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**New Explain Plan With SQL Profile**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Line ID</th>
<th>Object</th>
<th>Object Type</th>
<th>Order</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost</th>
<th>Time</th>
<th>CPU Cost</th>
<th>I/O Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td>0</td>
<td></td>
<td></td>
<td>7</td>
<td>0.017</td>
<td></td>
<td>839</td>
<td>11</td>
<td>131,239,648</td>
<td>832</td>
</tr>
<tr>
<td>SORT AGGREGATE</td>
<td>1</td>
<td></td>
<td></td>
<td>6</td>
<td>0.017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HASH JOIN</td>
<td>2</td>
<td></td>
<td></td>
<td>5</td>
<td>15.552</td>
<td>839</td>
<td>11</td>
<td>131,239,648</td>
<td>832</td>
<td></td>
</tr>
<tr>
<td>TABLE ACCESS FULL</td>
<td>3</td>
<td>SH_CUSTOMERS</td>
<td>TABLE</td>
<td>1</td>
<td>1.371</td>
<td>405</td>
<td>5</td>
<td>22,792,460</td>
<td>404</td>
<td></td>
</tr>
<tr>
<td>PARTITION RANGE ALL</td>
<td>4</td>
<td></td>
<td></td>
<td>4</td>
<td>4,486,538</td>
<td></td>
<td>423</td>
<td>6</td>
<td>7,341,375</td>
<td>428</td>
</tr>
<tr>
<td>BITMAP CONVERSION TO ROWIDS</td>
<td>5</td>
<td></td>
<td></td>
<td>3</td>
<td>4,486,538</td>
<td></td>
<td>423</td>
<td>6</td>
<td>7,341,375</td>
<td>428</td>
</tr>
<tr>
<td>BITMAP INDEX FAST FULL SCAN</td>
<td>6</td>
<td>SH_SALES_CUSTOMER_PK</td>
<td>INDEX (BITMAP)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SQL Tuning with DBMS_SQLTUNE

• 4 steps to get an SQL Profile
  • `DBMS_SQLTUNE.CREATE_TUNING_TASK`
  • `DBMS_SQLTUNE.EXECUTE_TUNING_TASK`
  • `DBMS_SQLTUNE.REPORT_TUNING_TASK`
  • `DBMS_SQLTUNE.ACCEPT_SQL_PROFILE`
# SQL Tuning - DBMS_SQLTUNE

## Tuning Task Management
- CANCEL_TUNING_TASK
- CREATE_TUNING_TASK
- DROP_TUNING_TASK
- EXECUTE_TUNING_TASK
- IMPLEMENT_TUNING_TASK
- INTERRUPT_TUNING_TASK
- REPORT_AUTO_TUNING_TASK
- REPORT_TUNING_TASK
- RESUME_TUNING_TASK
- SCRIPT_TUNING_TASK
- RESET_TUNING_TASK
- SET_TUNING_TASK_PARAMETER

## SQL Profile Management
- ACCEPT_SQL_PROFILE
- ALTER_SQL_PROFILE
- DROP_SQL_PROFILE

## Select SQL Statements
- CAPTURE_CURSOR_CACHE_SQLSET
- SELECT_CURSOR_CACHE
- REPORT_SQL_MONITOR
- SELECT_WORKLOAD_REPOSITORY
- SQLTEXT_TO_SIGNATURE

## Staging Table Management
- CREATE_STGTAB_SQLPROF
- CREATE_STGTAB_SQLSET
- PACK_STGTAB_SQLPROF
- PACK_STGTAB_SQLSET
- REMAP_STGTAB_SQLPROF
- REMAP_STGTAB_SQLSET
- UNPACK_STGTAB_SQLPROF
- UNPACK_STGTAB_SQLSET

## SQL Tuning Set Management
- ADD_SQLSET_REFERENCE
- CREATE_SQLSET
- DELETE_SQLSET
- DROP_SQLSET
- LOAD_SQLSET
- REMOVE_SQLSET_REFERENCE
- SELECT_SQLSET
- UPDATE_SQLSET
SQL Profile containing literals - not binds

- SQL Profiles can handle statements containing literals (instead of binds) as well:
  - Since 11.1.0.6 possible in EM:
  - In 10.2 only possible on command line:

```sql
exec :p_name:=dbms_sqltune.accept_sql_profile
  (task_name=>'XT',name=>'XT_PROFILE', FORCE_MATCH=>TRUE);

SQL> select name, status, force_matching, sql_text
    from dba_sql_profiles;

NAME    STATUS   FOR SQL_TEXT
------- -------- ---- --------------------------------------------
MY_PROF ENABLED  YES select /*+ use_nl(c) ordered */ count(*)
                      from sh.sales s, sh.customers c
                      where c.cust_id = s.cust_id
                      and CUST_FIRST_NAME = 'Mike'
```
SQL Profile - evaluation

- SQL Profiles should be evaluated before making them available to every user:

  ```sql
  exec
  :p_name:=dbms_sqltune.accept_sql_profile
  task_name=>'XT',name=>'XT_PROFILE',
  category=>'TEST_ENV', FORCE_MATCH=>TRUE)
  ```

  ```sql
  alter session set SQLTUNE_CATEGORY='TEST_ENV';
  ```

- Now evaluate the statement's profile in a limited user context
- If verification went fine, make it accessible to everybody

  ```sql
  exec
  dbms_sqltune.alter_sql_profile
  (name=>'XT_PROFILE',
   attribute_name=>'CATEGORY',value=>'DEFAULT')
  ```
SQL Profiling Effectiveness

- Workload of a big market research customer in EMEA - 73 high load queries identified requiring tuning
- Query execution time Before vs. After

**Before ...**

![Before Graph]

**... After**

![After Graph]
SQL Profiling Effectiveness

- Workload of a big market research customer in EMEA - 73 high load queries identified requiring tuning
- Manual tuning compared to automatic tuning of these 73 complex DWH statements

<table>
<thead>
<tr>
<th></th>
<th>Average Response Time</th>
<th>Worst Response Time</th>
<th>Cumulative Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Tuning</td>
<td>817s</td>
<td>5751s</td>
<td>58821s</td>
</tr>
<tr>
<td>Manual Tuning</td>
<td>30s</td>
<td>275s</td>
<td>2131s</td>
</tr>
<tr>
<td>Automatic Tuning (SQL Pr)</td>
<td>13s</td>
<td>59s</td>
<td>929s</td>
</tr>
</tbody>
</table>
SQL Profiling Time

- All together the optimization for all 73 statements took approximately 1½h !!!
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Challenges
- SQL Plan Management
- Database Replay
- SQL Performance Analyzer
Challenges

• >90% of so-called upgrade problems aren't really upgrade problems but performance issues after the upgrade
  • It's important to use sufficient test scenarios and methods
  • Typical problem areas:
    • Optimizer - execution plans
      • Queries are slow
      • Reports or batches take longer to complete
    • Increased resource requirements
    • Code path changes
Challenges

• **Optimizer - prevent execution plan changes:**
  • Classical approach:
    • Rule Based Optimizer ([RBO desupport since Oracle 10g - Note:189702.1](#))
    • Hints
    • Stored Outlines
    • Rewriting SQL statements
      • `optimizer_features_enabled=n.n.n`
    • Change specific optimizer parameters
    • Import and fix object and systems statistics
  • Modern, efficient and better resource consumption:
    • SQL Plan Management
    • SQL Profiling
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Challenges
- SQL Plan Management
- Database Replay
- SQL Performance Analyzer
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:

```
| SQL | Parse | GB | HJ | Execute | Plan acceptable |
```

- Now some conditions get changed (statistics, upgrade, parameters)
  - A new plan will be created - is it better or worse???

```
| SQL | Parse | GB | NL | Execute | 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
  - Consist of accepted execution plans
  - Contains only plans for statements being parsed/executed more than once
- Only known, 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 SYSAUX
  • Selection
    • Only accepted plans will be used
    • New execution plans will be recorded to 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
  - Schematically - `OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=TRUE`

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
  • Same statement parsed again but a **different** plan will be created

  ![Diagram showing SQL Plan Management process]

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

  - `OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=FALSE`
SQL Plan Management

• Phase 2 - Selection
  • Schematically: `OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=FALSE`

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

Plan known and acceptable
SQL Plan Management

• Phase 3 - Evolution
  • Schematically:

- Worse plan will be kept in the Plan History
- Equal or better plans can be added to the SQL Plan Baseline

DBA schedules verification

Optimizer
# SQL Plan Management

## Database Control:

- **Database Configuration**
  - Memory Advisors
  - Automatic Undo Management
  - Initialization Parameters
  - View Database Feature Usage

- **Resource Manager**
  - Getting Started
  - Consumer Groups
  - Consumer Group Mappings
  - Plans
  - Settings
  - Statistics

- **Oracle Scheduler**
  - Jobs
  - Chains
  - Schedules
  - Programs
  - Job Classes
  - Windows
  - Window Groups
  - Global Attributes
  - Automated Maintenance Tasks

- **Security**
  - Users
  - Roles
  - Profiles
  - Audit Settings
  - Transparent Data Encryption
  - Virtual Private Database Policies
  - Application Contexts

- **Query Optimizer**
  - Manage Optimizer Statistics
  - SQL Plan Control

- **Change Database**
  - Add Instance
  - Delete Instance
SQL Plan Management

- DatabaseControl - Configuration:

  Start with 5-14 weeks:
  exec DBMS_SPM.CONFIGURE('plan_retention_weeks',5);
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 the 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 11g
SQL Plan Management - Upgrade 1

- Upgrade scenario 1

**OPTIMIZER_FEATURES_ENABLE=9.2.0**
**OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=TRUE**

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=11.1.0**
**OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=FALSE**
SQL Plan Management - Upgrade 2

- Upgrade scenario 2

1. Staging Table

   exp imp expdp impdp DB-Link...

2. 10.2 plans will be added to the SQL Plan Baseline

   Every new and better plan will be stored in the Plan History
SQL Plan Management - New Application

• New application (module) gets delivered

1 @Vendor

2 @Customer
SQL Plan Management - New Application

- New application (module) gets delivered - and it's possible to deliver the right execution plans, too:

  - @Software Vendor:
    - Create a staging table using `DBMS_SPM.CREATE_STGTAB_BASELINE`
    - Pack the required baselines into the staging table using `DBMS_SPM.PACK_STGTAB_BASELINE`
    - Export the staging table into a dump file using Data Pump or Export and transport it to the target system

  - @Customer:
    - Import the dump file into the target database
    - Unpack the SQL Plan Baselines from the staging table into the SQL Management Base of the target system
    - `DBMS_SPM.UNPACK_STGTAB_BASELINE`
SQL Plan Management - Outlines

• Migration of Stored Outlines to SQL Plan Management:
  • Tuning-Pack:
    Record execution plans in 10.2 and transport them (see Upgrade Case 1)
  • No Tuning-Pack license:
    • Upgrade your database with Outlines to 11g
    • Set CAPTURE_SQL_PLAN_BASELINES=TRUE
    • Make sure your run your outlined queries 2x
      • Plans will be recorded to the Baseline
    • Switch CAPTURE_SQL_PLAN_BASELINES=FALSE
    • Delete the Stored Outlines
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Challenges
- SQL Plan Management
- Database Replay
- SQL Performance Analyzer
Real Application Testing

• Goal:
  • Record and replay a real workload to see how the new system performs
  • Find regressions and changing plans before the upgrade

• Licensable database pack "Real Application Testing"
  ⇒ Available since Oracle Database 11.1.0.6
  ⇒ Available with patch set 10.2.0.4
  ⇒ Available as single patch for 9.2.0.8 and 10.2.0.2/3
  ⇒ For patch numbers please see Note:560977.1
Real Application Testing

- Real Application Testing consists of:
  - Database Replay
    - Package `DBMS_WORKLOAD_CAPTURE`
    - Package `DBMS_WORKLOAD_REPLAY`
      - Can be used for upgrades from:
        - 9.2.0.8 and 10.2.0.2/3/4 to 11.1.0.x
  - SQL Performance Analyzer (SPA)
    - Package `DBMS_SQLPA`
      - Can be used for upgrades from:
        - 9.2.0.8 and 10.2.0.2/3/4 to 11.1.0.x
        - 9.2.0.8 to 10.2.0.2/3/4
        - 10.2.0.2/3 to 10.2.0.4
  - SQL Tuning Sets (STS)
    - Package `DBMS_SQLTUNE`
Real Application Testing

• White Paper:
  • Database Replay:

• SQL Performance Analyzer:

• OTN:
  • Command line examples for Replay and SPA:
    • http://www.oracle.com/technology/products/manageability/database/tools/db_replay_cli.zip
    • http://www.oracle.com/technology/products/manageability/database/tools/spa_scripts.zip
Real Application Testing

- Database Control:

  - Database Instance: SOURCE

    - 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
  
    - Support
      - Support Workbench

  - Real Application Testing
    - Database Replay
    - SQL Performance Analyzer
Agenda

1. Preparation
2. Upgrade
3. News and Task List
4. Diagnostics & Tuning
   - Challenges
   - SQL Plan Management
   - Database Replay
   - SQL Performance Analyzer
5. Performance Testing
6. Best Practices
Database Replay

- Replay actual production database workload in test environment
- 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
Database Replay

• Workflow:
Restrictions

• Capture won't record (currently):
  • Direct Path Loads
  • MTS Requests
  • Oracle Streams and Advanced Replication
  • Non-PL/SQL based AQ
  • Flashback queries
  • Non-SQL based object access
  • Distributed transaction
  • Remote **DESCRIBE** and **COMMITs**
Database Replay

• Step-By-Step
Database Replay

1. Create a copy of the database to replay the workload
   • RMAN Duplicate
   • Snapshot Standby
     • Possibly create a guaranteed restore point
Database Replay

2. Some guidelines for Workload Capture

- Ideally restart prod database
  - Why? All transactions can be closed correctly
- RAC: Just one instance up and running, then start Capture and now start the other instances
- Start Capture before users logon to the database
  - Therefore start database in RESTRICTED mode
  - As soon as Capture begins it will switch to UNRESTRICTED
2. Workflow in EM: Workload Capture

Database Instance: SOURCE >

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>
<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>[Link]</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>[Link]</td>
</tr>
<tr>
<td>3</td>
<td>Replay Workload</td>
<td>Choose this option to replay a preprocessed workload on this database.</td>
<td>[Link]</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)

View Workload Capture History

Active Capture and Replay

Select Name | Type | Directory Object | Start Time |
------------|------|------------------|------------|
No items found |
Database Replay

2. Preparation steps in EM: Workload Capture

The following prerequisites should be met before proceeding to capturing the workload to avoid potential problems.

1. It is highly recommended to meet and acknowledge each of the following prerequisites.

<table>
<thead>
<tr>
<th>Prerequisite</th>
<th>Acknowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
3. Define Workload Filter:

- **Default:**
  All user sessions will be recorded

- **Inclusion** filter:
  Only specified user session will be recorded

- **Exclusion** filter:
  All but specified user sessions will be recorded
  - For instance exclude "OMS" and "emagent%"

- Both filters can't be combined

- Package:

  ```sql
  DBMS_WORKLOAD_CAPTURE.ADD_FILTER(fname=>'myFilter',
                                     fattribute=>'USER',
                                     fvalue=>'SCOTT');
  DBMS_WORKLOAD_CAPTURE.DELETE_FILTER(fname=>'myFilter');
  ```
Database Replay

• Start Capture in line mode:

  BEGIN
  DBMS_WORKLOAD_CAPTURE.START_CAPTURE
  (name => 'hammerora_single_01',
   dir => 'TESTING',
   default_action => 'EXCLUDE');
  END;
 /

• Stop Capture:

  BEGIN
  DBMS_WORKLOAD_CAPTURE.FINISH_CAPTURE();
  END;
 /

Database Replay

- Database Control - capture statistics:

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

**Average Active Sessions**

![Graph showing active sessions over time]

**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>93.94</td>
</tr>
<tr>
<td>Transactions</td>
<td>289</td>
<td>54</td>
<td>18.69</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

- Export AWR in line mode:
  
  ```sql
  BEGIN
    DBMS_WORKLOAD_CAPTURE.EXPORT_AWR
    (capture_id => 5);
  END;
  /
  ```
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</td>
<td>Preprocessing will prepare a captured workload for replay. This must be</td>
</tr>
<tr>
<td></td>
<td>Workload</td>
<td>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

• Preprocessing in line mode:
  • BEGIN
    DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE
    (capture_dir => 'TESTING');
  END;
/

• Restore database from backup
Database Replay

- Import AWR snap

```sql
DECLARE db_id number;
BEGIN
  db_id := DBMS_WORKLOAD_CAPTURE.IMPORT_AWR(capture_id => 5, staging_schema => 'TPCC');
END;
/
```
Database Replay

- Workload Replay: replay parameters

Some replay parameters can be modified to change the behavior of the replay. Refer to system documentation for more information.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>synchronization</td>
<td>This parameter determines if synchronization will be used during workload replay. If this parameter is set to TRUE, the COMMIT order in the captured workload will be preserved during replay and all replay actions will be executed only after all dependent COMMIT actions have completed. The default value is TRUE.</td>
<td>TRUE</td>
</tr>
<tr>
<td>connect_time_scale</td>
<td>This parameter scales the elapsed time from when the workload capture started to when the session connects with the specified value and is interpreted as a % value. The default value is 100.</td>
<td>100 %</td>
</tr>
<tr>
<td>think_time_scale</td>
<td>This parameter scales the elapsed time between two successive user calls from the same session and is interpreted as a % value. Setting this parameter to 0 will send user calls to the database as fast as possible during replay. The default value is 100.</td>
<td>100 %</td>
</tr>
<tr>
<td>think_time_auto_correct</td>
<td>This parameter reduces the think time if workload replay goes slower than workload capture. If this parameter is set to TRUE, the system will correct the think time (based on the think_time_scale parameter) between calls when user calls take longer to complete during replay than during capture. The default value is TRUE.</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
Database Replay

• Initialize and parameterize workload replay clients (wrc):
  • BEGIN
    
    DBMS_WORKLOAD_REPLAY.INITIALIZE_REPLAY
    (replay_name => 'play_hammerora_01',
     replay_dir => 'TESTING');

    DBMS_WORKLOAD_REPLAY_PREPARE_REPLAY
    (synchronization => TRUE);

    END;
    /
  •
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 2007

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

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

- Start Workload Replay in command line mode:

```sql
BEGIN
    DBMS_WORKLOAD_REPLAY.START_REPLAY ();
END;
/
```
**Database Replay**

- Monitor divergence stats:

<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>83.96</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:

<table>
<thead>
<tr>
<th>Summary</th>
<th>Replay Name</th>
<th>Capture Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REPLY_TEST02</td>
<td>CAPTURE_ORCL_TEST01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Directory Object</th>
<th>Duration (hh:mm:ss)</th>
<th>Prepare Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPTDIR</td>
<td>00:10:00</td>
<td>Aug 30, 2007 8:59:45 AM CEST</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Database Name</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Workload Profile</th>
<th>Connection Mappings</th>
<th>Replay Parameters</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elapsed Time Comparison</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture</td>
<td>Replay</td>
</tr>
<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>
</tbody>
</table>

Clients: 1
Clients Finished: 1

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 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.
Database Replay

- Workload replay reporting in CLI:

```sql
DECLARE
cap_id NUMBER;
rep_id NUMBER;
rep_rpt CLOB;
BEGIN
  cap_id := DBMS_WORKLOAD_REPLAY.GET_REPLAY_INFO
             (dir => 'TESTING');

  /* Get the latest replay for that capture */
  SELECT max(id) INTO rep_id
  FROM   dba_workload_replays WHERE capture_id=cap_id;

  rep_rpt := DBMS_WORKLOAD_REPLAY.REPORT
             (replay_id => rep_id,
              format => DBMS_WORKLOAD_REPLAY.TYPE_TEXT);
END;
/
```
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Challenges
- SQL Plan Management
- Database Replay
- SQL Performance Analyzer
**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

**Production**

**Test**

- Make Change
- Comparison
- Production Change / Tuning Deployment
- Done? (Yes/No)
- Tuned System
SPA for a 9i workload against 11g

- Upgrade scenario from 9i directly to 11g

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
...

See Note:455889.1 - SPA example
SPA for 10.2.0.2 => 10.2.0.4

- Setup

![Diagram showing the process of upgrading a database from 10.2.0.2 to 10.2.0.4, including capturing SQLs into a STS, upgrading the database, and analyzing the SQLs using SQL Performance Analyzer.](image-url)
SPA for a 9i workload against 11g

- Enable statistics timing:
  - `alter system set timed_statistics=true;`

- Enable tracing on 9i:
  - `alter system set events '10046 trace name context forever, level 4';`
SPA for a 9i workload against 11g

- Create the mapping table in 9i:
  ```sql
  create table MAPPING_TABLE as
  select object_id id, owner,
         substr(object_name, 1, 30) name
  from   dba_objects
  where  object_type NOT IN
         ('CONSUMER GROUP', 'EVALUATION CONTEXT', 'FUNCTION',
          'INDEXTYPE', 'JAVA CLASS', 'JAVA DATA', 'JAVA
          RESOURCE', 'LIBRARY', 'LOB', 'OPERATOR', 'PACKAGE',
          'PACKAGE BODY', 'PROCEDURE', 'QUEUE', 'RESOURCE
          PLAN', 'SYNONYM', 'TRIGGER', 'TYPE', 'TYPE BODY')
  union all
  select user_id id, username owner, null name
  from   dba_users;
  ```

- Disable tracing in 9i afterwards:
  ```sql
  alter system set events '10046 trace name context forever off';
  ```

- Export mapping table with `exp`
SPA for a 9i workload against 11g

- Create a directory object in 11g:
  - `create or replace DIRECTORY 'SPA_DIR' as '/tmp/spa';`

- Install SPA one-off patch (see Note:562899.1) or use 11.1.0.7
- Transport event-10046 trace files to the 11g system
- Import mapping table
SPA for a 9i workload against 11g

- Create a SQL Tuning Set (STS) in 11g:

```sql
declare
    mycur     DBMS_SQLTUNE.SQLSET_CURSOR;
begin
    DBMS_SQLTUNE.CREATE_SQLSET('SPA_9iWKLD');
    open mycur for
        select value(p) from
            table(DBMS_SQLTUNE.SELECT_SQL_TRACE(
                directory => 'SPA_DIR',
                file_name => '%ora%',
                mapping_table_name => 'MAPPING_TABLE',
                select_mode => DBMS_SQLTUNE.SINGLE_EXECUTION)) p;
    DBMS_SQLTUNE.LOAD_SQLSET('SPA_9iWKLD', mycur);
    close mycur;
end;
/
```

- Monitor progress in DBA_SQLSET
SPA for a 9i workload against 11g

• Create a SPA task in 11g:

```sql
var tname varchar2(200);
execute
  tname := DBMS_SQLPA.CREATE_ANALYSIS_TASK(
    sqlset_name => 'SPA_9iWKLD',
    task_name   => 'SPA_TASK_9i_11g',
    description => 'Test 9i to 11g upgrade');
print tname
```
SPA for a 9i workload against 11g

- Parameterize the SPA task

  ```sql
  begin
  DBMS_SQLPA.SET_ANALYSIS_TASK_PARAMETER(
    task_name => 'SPA_TASK_9i_11g',
    parameter => 'WORKLOAD_IMPACT_THRESHOLD',
    value => 0);
  DBMS_SQLPA.SET_ANALYSIS_TASK_PARAMETER(
    task_name => 'SPA_TASK_9i_11g',
    parameter => 'SQL_IMPACT_THRESHOLD',
    value => 5);
  end;
  /
  ```

- To filter "noise" use "SQL_IMPACT_THRESHOLD"=5 (means only regressed SQL 5% above this threshold will be shown)
SPA for a 9i workload against 11g

- Establish "Before Change" trial (from collected 9i info)
  ```sql
  begin
    DBMS_SQLPA.EXECUTE_ANALYSIS_TASK(
      task_name => 'SPA_TASK_9i_11g',
      execution_name => 'SPA_RUN1_9i',
      execution_type => 'CONVERT SQLSET',
      execution_desc => '9i run generated from STS');
  end;
/
  ```

- Run 11g execution
  ```sql
  begin
    DBMS_SQLPA.EXECUTE_ANALYSIS_TASK(
      task_name => 'SPA_TASK_9i_11g',
      execution_name => 'SPA_RUN2_11g',
      execution_type => 'TEST EXECUTE',
      execution_desc => 'Test now against 11g');
  end;
/
  ```
SPA for a 9i workload against 11g

• Compare results

execute
    DBMS_SQLPA.EXECUTE_ANALYSIS_TASK(
        task_name => 'SPA_TASK_9i_11g',
        execution_name => 'COMPARE_9i_11g_CPU',
        execution_type => 'COMPARE PERFORMANCE',
        execution_params => DBMS_ADVISOR.ARGLIST(
            'COMPARISON_METRIC', 'CPU_TIME',
            'EXECUTION_NAME1', 'SPA_RUN1_9i',
            'EXECUTION_NAME2', 'SPA_RUN2_11g'),
        execution_desc => 'Compare 9i vs. 11g on CPU_TIME');

• Comparison metrics options:
  • CPU_TIME
  • BUFFER_GETS
  • PARSE_TIME
  • ELAPSED_TIME
  • USER_IO_TIME
  • DISK_READS
  • DIRECT_WRITES
  • OPTIMIZER_COST

} Compare performance for at least these two metric
SPA for a 9i workload against 11g

- Generate **summary** report

```sql
set heading off
set long 1000000000
set longchunksize 10000
set echo off;
set linesize 1000;
spool /tmp/spa_9i_11g_cpu_summary.html

select
  xmltype(DBMS_SQLPA.REPORT_ANALYSIS_TASK(
    'SPA_TASK_9i_11g',     /* task_name */
    'html',                /* type */
    'typical',             /* level */
    'summary',             /* section */
    null,                  /* object_id */
    100,                   /* top_sql */
    'COMPARE_9i_11g_CPU').getclobval(2,2) /* execution_name */
from dual;
spool off
```
SPA for a 9i workload against 11g

- Generate **regressed SQL** report

```sql
set heading off
set long 1000000000
set longchunksize 10000
set echo off;
set linesize 1000;

spool /tmp/spa_9i_11g_cpu_regressed.html

select xmltype(DBMS_SQLPA.REPORT_ANALYSIS_TASK(
    'SPA_TASK_9i_11g',                     /* task_name */
    'html',                                /* type */
    'regressed',                           /* level */
    'all',                                 /* section */
    null,                                  /* object_id */
    null,                                  /* top_sql */
    'COMPARE_9i_11g_CPU')).getclobval(2,2) /* execution_name */
from dual;

spool off
```
SPA for a 9i workload against 11g

- Generate **changed plans** report

```sql
set heading off
set long 1000000000
set longchunksize 10000
set echo off;
set linesize 1000;
spool /tmp/spa_9i_11g_changed_plans.html
select xmltype(DBMS_SQLPA.REPORT_ANALYSIS_TASK('SPA_TASK_9i_11g', 'html', 'changed_plans', 'all', null, null, 'COMPARE_9i_11g_CPU')).getclobval(2,2) from dual;
spool off
```
Real World Experience: **SPA**

- Regressed report in detail:

<table>
<thead>
<tr>
<th>SQL Category</th>
<th>SQL Count</th>
<th>Plan Change Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>3552</td>
<td>1593</td>
</tr>
<tr>
<td>Improved</td>
<td>938</td>
<td>150</td>
</tr>
<tr>
<td>Regressed</td>
<td>99</td>
<td>48</td>
</tr>
<tr>
<td>Unchanged</td>
<td>1756</td>
<td>1138</td>
</tr>
<tr>
<td>With Errors</td>
<td>1026</td>
<td>0</td>
</tr>
</tbody>
</table>

### SQL Statements Sorted by their Absolute Value of Change Impact on the Workload

<table>
<thead>
<tr>
<th>object_id</th>
<th>sql_id</th>
<th>Impact on Workload</th>
<th>Impact on SQL</th>
<th>% Workload Before</th>
<th>% Workload After</th>
<th>Plan Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>7277</td>
<td>a304c0gqaq0g40f3</td>
<td>-0.4%</td>
<td>20</td>
<td>220</td>
<td>-1000%</td>
<td>.04%</td>
</tr>
<tr>
<td>8404</td>
<td>726i50g0i10i0m</td>
<td>-0.26%</td>
<td>20</td>
<td>150</td>
<td>-650%</td>
<td>.04%</td>
</tr>
<tr>
<td>7282</td>
<td>860c00v3v27c40c</td>
<td>-0.22%</td>
<td>360</td>
<td>470</td>
<td>-30,56%</td>
<td>.73%</td>
</tr>
<tr>
<td>7241</td>
<td>acnh0m3i1u08x</td>
<td>-0.2%</td>
<td>520</td>
<td>620</td>
<td>-19,23%</td>
<td>1.05%</td>
</tr>
<tr>
<td>10654</td>
<td>99ny0cudik24e</td>
<td>-0.16%</td>
<td>120</td>
<td>200</td>
<td>-200%</td>
<td>.00%</td>
</tr>
<tr>
<td>10726</td>
<td>0140y0c0d0m4w</td>
<td>-0.1%</td>
<td>100</td>
<td>150</td>
<td>-50%</td>
<td>.2%</td>
</tr>
<tr>
<td>7432</td>
<td>bssxd9l0d4f9d</td>
<td>-0.04%</td>
<td>10</td>
<td>30</td>
<td>-200%</td>
<td>.02%</td>
</tr>
<tr>
<td>8191</td>
<td>77mp08w0w58q50</td>
<td>-0.04%</td>
<td>10</td>
<td>30</td>
<td>-200%</td>
<td>.02%</td>
</tr>
<tr>
<td>10076</td>
<td>7a3e571hy2f0w4</td>
<td>-0.04%</td>
<td>70</td>
<td>90</td>
<td>-28,57%</td>
<td>.14%</td>
</tr>
<tr>
<td>7122</td>
<td>acp13lyv30tdc</td>
<td>-0.02%</td>
<td>0</td>
<td>10</td>
<td>-1000%</td>
<td>0%</td>
</tr>
<tr>
<td>7122</td>
<td>adc1m0h0v6g0k</td>
<td>-0.02%</td>
<td>0</td>
<td>10</td>
<td>-1000%</td>
<td>0%</td>
</tr>
<tr>
<td>7493</td>
<td>agc0h0yam0k1</td>
<td>-0.02%</td>
<td>0</td>
<td>10</td>
<td>-1000%</td>
<td>0%</td>
</tr>
<tr>
<td>7493</td>
<td>e9n0d4s4c0bk</td>
<td>-0.02%</td>
<td>0</td>
<td>10</td>
<td>-1000%</td>
<td>0%</td>
</tr>
<tr>
<td>7221</td>
<td>bsgd0grgs0wsw0w</td>
<td>-0.02%</td>
<td>0</td>
<td>10</td>
<td>-1000%</td>
<td>0%</td>
</tr>
</tbody>
</table>

**Oracle**
Real World Experience: SPA

• Regressed report in detail:

**Report Details:** Statements Sorted by their Absolute Value of Change Impact on the Workload

<table>
<thead>
<tr>
<th>SQL Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object ID  : 7277</td>
</tr>
<tr>
<td>Schema Name: LURG200C</td>
</tr>
<tr>
<td><strong>SQL ID</strong> : a304c099xxf3</td>
</tr>
<tr>
<td>Execution Frequency : 1</td>
</tr>
<tr>
<td>SQL Text:</td>
</tr>
<tr>
<td>select a,b,c from (select ware a,kommentar b, p.nachname c from BUCHUNGEN b, PERSON p where zeit &gt; '20080710000000' and aid = 349905 and kommentar like 'show_user%' and b.ware = p.id and p.status != 'I' order by zet desc) where rownum &lt; 90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Execution Statistics:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stat Name</strong></td>
</tr>
<tr>
<td>elapsed_time</td>
</tr>
<tr>
<td>parse_time</td>
</tr>
<tr>
<td>cpu_time</td>
</tr>
<tr>
<td>buffer_gets</td>
</tr>
<tr>
<td>cost</td>
</tr>
<tr>
<td>reads</td>
</tr>
<tr>
<td>writes</td>
</tr>
<tr>
<td>rows</td>
</tr>
</tbody>
</table>

**Findings (3):**

1. Die Performance dieser SQL-Anweisung wurde vermindert.
2. Die Struktur des SQL-Ausführungsplans wurde geändert.
3. Die Anzahl von zurückgegebenen Zeilen in Ausführung 'CONV_SPA_TASK_KLAUS' unterscheidet sich von Ausführung 'EXEC_SPA_TASK_KLAUS'.
Real World Experience: SPA

- Regressed report in detail:

**Execution Plan Before Change:**

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>COUNT STOPKEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VIEW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SORT ORDER BY STOPKEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NESTED LOOPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>BUCHUNGEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>INDEX RANGE SCAN</td>
<td>I_EID_BUCHUNGEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>PERSON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>INDEX UNIQUE SCAN</td>
<td>SYS_C0010236</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Execution Plan After Change:**

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+1</td>
<td>COUNT STOPKEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VIEW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+3</td>
<td>SORT ORDER BY STOPKEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NESTED LOOPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>NESTED LOOPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+6</td>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>BUCHUNGEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+7</td>
<td>INDEX RANGE SCAN</td>
<td>I_EID_BUCHUNGEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+8</td>
<td>INDEX UNIQUE SCAN</td>
<td>SYS_C0012673</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+9</td>
<td>TABLE ACCESS BY INDEX ROWID</td>
<td>PERSON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Empty because source db was an Oracle 9i database.
SPA for a 9i workload against 11g

- Compare results
  - Be cautious in interpreting trial results
  - Analysis based on “single_execution” of SQL in 9i does not account for multiple bind variables and plans
  - Environmental traps: SPA is agnostic of the environment used for pre- and post-change trials
  - Data changes during SQL tracing
  - Bias caused by SQL tracing in 9i
  - Reports may be overly optimistic or pessimistic due to above reasons

- Recommended approach is to investigate as follows:
  - Summary of change impact by CPU_TIME and BUFFER_GETS
  - Regressed SQL based on CPU_TIME and BUFFER_GETS
  - All SQL with changed plans
Capturing a SQL Workload for SPA

• Capture from cursor cache
  • Create a SQL Tuning Set (STS)
    • BEGIN
      DBMS_SQLTUNE.CREATE_SQLSET(
        sqlset_name => 'SPA_STS',
        description => 'Upgrade test STS for SPA');
    END;
    /
  • Capture SQL directly from the cursor cache into STS
    • EXEC
      DBMS_SQLTUNE.CAPTURE_CURSOR_CACHE_SQLSET(
        sqlset_name => 'SPA_STS',
        time_limit => 15*60,
        repeat_interval => 10,
        capture_mode => dbms_sqltune.MODE_ACCUMULATE_STATS);
Capturing a SQL Workload for SPA

• Filter statements from a STS:
  • EXEC
    
    DBMS_SQLTUNE.DELETE_SQLSET(
        sqlset_name => 'SPA_STS',
        basic_filter => 'parsing_schema_name=''MDSYS''');

  • This will delete all statements belonging to MDSYS from the STS
  • BASIC_FILTER will be treated like a WHERE clause for this
    DELETE action

• Proceed with creation of a SPA analysis task:
  • EXECUTE
    
    DBMS_SQLPA.CREATE_ANALYSIS_TASK(
        task_name => 'SPA_TASK_CURSOR_CACHE',
        description => 'SPA with statements from CC',
        sqlset_name => 'SPA_STS');
Parameter Changes

- SPA Recommendation: SQL Tuning Advisor

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

- **Task Name**: PARAM_CHANGE
- **Task Owner**: SYS
- **Test Rule-based vs Cost-based Optimizer**: rule-based

- **SQL Tuning Set Name**: HR_WORKLOAD
- **STSOwner**: APPS

- **Total SQL Statements**: 50
- **SQL Statements With Errors**: 0
- **Replay Trial 1**: rule_based
- **Replay Trial 2**: cost_based
- **Comparison Metric**: Buffer Gets

**Global Statistics**

- **Projected Workload Buffer Gets**
  - Rule-based: 4,500,000
  - Cost-based: 1,500,000

- **SQL Statement Count**
  - Improved: 25
  - Regressed: 10
  - Unchanged: 15

- **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.
    - Explore alternate execution plans using 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>↑ 73s2svy2svfrw</td>
<td>13.790</td>
<td>1,753,552,000</td>
<td>1,238,620,000</td>
<td>29.370</td>
<td>46.950</td>
</tr>
<tr>
<td>↑ g2a407mv2hsy</td>
<td>13.790</td>
<td>1,753,552,000</td>
<td>1,238,620,000</td>
<td>29.370</td>
<td>46.950</td>
</tr>
<tr>
<td>2w2tgvzhs6u2by</td>
<td>-3.050</td>
<td>218,621,000</td>
<td>332,519,000</td>
<td>-52.100</td>
<td>5.850</td>
</tr>
<tr>
<td>↓ fbp9za0hql2km</td>
<td>-0.070</td>
<td>6,000</td>
<td>2,721,000</td>
<td>-45,250,000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
## Parameter Changes

- SQL Tuning Advisor offers SQL Profiles

### SQL Tuning Results: TUNEREG

<table>
<thead>
<tr>
<th>Select SQL Text</th>
<th>Parsing Schema</th>
<th>SQL ID</th>
<th>Statistics</th>
<th>SQL Profile Index</th>
<th>Restructure SQL</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT /* my_query_1_scott <em>/ /</em>+ ORDERED INDEX(t1 USE_HASH(t1) /*+ 'B'</td>
<td></td>
<td>t2.pg_featurevalue_0...</td>
<td>APPS</td>
<td>2wtgxbjg6u2by</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 FRO...</td>
<td>APPS</td>
<td>fbp9za0hqlk2km</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 FRO...</td>
<td>APPS</td>
<td>1h3c2y092ds9d</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 FRO...</td>
<td>APPS</td>
<td>654xs8xs5wp42</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Fallback Strategies
Guidelines
Real World Experiences
Fallback Strategy

• First make clear:
  • If anything unforeseen happens and you'll have to step back, will you be allowed to lose data (i.e. changes done to the data in the system after the upgrade): YES or NO?
    • If **YES**: restore a backup, flashback (since 10g)
    • If **NO**: export/import, downgrade
Fallback Strategy - Backup

- Restore a backup
  - Complete online backup (RMAN)
  - Please verify:
    - Where is your backup located? Tapes, HD, ...
    - Does the restore work?
    - How long will it take?
      - Check the priority of your restoration jobs especially in Virtual Tape Drives?
    - How long will the recovery take?
Fallback Strategy - Backup

• No possibility to take a complete backup??
  • Put all data tablespaces into read-only mode
  • Shutdown the database immediate
  • Copy SYSTEM, TEMP, UNDO and SYSAUX datafiles and controlfiles/redologs
  • Startup database again

• If something fails during the upgrade:
  • Shutdown the database
  • Copy SYSTEM, TEMP, UNDO, SYSAUX, controlfile and redologs from the backup location
  • Startup the database in the old environment
Fallback Strategy - Flashback

- Flashback Database
  - Considered to be much faster than restore/recover but requires to not increase COMPATIBLE
  - Works beginning from 10.2
    - Create a GUARANTEED RESTORE POINT
    - Upgrade your database
    - In case of failure flashback to the restore point
    - In case of success delete the restore point

```
SQL> CREATE RESTORE POINT grpt
    2 GUARANTEE FLASHBACK DATABASE;
```

```
SQL> FLASHBACK DATABASE TO RESTORE POINT grpt;
```
Fallback Strategy: exp/imp

- Downgrade with exp/imp to 9.2.0.x
  - Note:158845.1
    - Prepare an empty database for the import just in case you'll have to step back
  - Then:
    - Run the appropriate `?/rdbms/admin/catexp.sql` to create the 9.2 export views in the upgraded database
    - Use "old" 9.2 exp for the export
    - Use "old" 9.2 imp for the import
  - Annotation: See Note:550740.1
    - Connect to 11g database with SYSDBA
    - CREATE OR REPLACE VIEW exu9defpswitches (
      compflgs, nlslensem ) AS
      SELECT a.value, b.value
      FROM sys.v$parameter a, sys.v$parameter b
      WHERE a.name = 'plsql_code_type' AND b.name = 'nls_length_semantics'
    /
    - Re-start exp version 9iR2 to extract from 11g
Fallback Strategy: catdwgrd.sql

- Downgrade with `catdwgrd.sql` to 10.1.0.x
  - **Note:** 443890.1
    - Downgrade to the release you've upgraded from
      - 10.1.0.5
      - 10.2.0.3/4
    - Only possible if `COMPATIBLE hasn't been raised`!!!
      - Please note:
        A downgrade will only be possible to the release you've upgraded from - so if a patch set has been applied always apply it before the upgrade starts - otherwise you'll only be able to downgrade to the release you've patched
Fallback Strategy: `catdwgrd.sql`

- Downgrade with `catdwgrd.sql` to 10g
  - Task in 11g environment:
    ```sql
    SQL> SPOOL /tmp/downgrade.log
    SQL> STARTUP DOWNGRADE
    SQL> @catdwgrd.sql
    SQL> SPOOL OFF
    ```
  - Afterwards tasks in 10g environment:
    ```sql
    SQL> STARTUP UPGRADE
    SQL> SPOOL /tmp/reload.log
    SQL> @catrelod.sql
    -- The catrelod.sql script reloads the appropriate version of
    -- all of the database components in the downgraded database.
    SQL> SPOOL OFF
    ```

- Please note: additional steps are required if EM repository resides in the database - please see chapter 6 *Downgrading a Database* in the Oracle 11g Upgrade Guide.
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Fallback Strategies
Guidelines
Real World Experiences
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!!!
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: 738538.1 Known Issues specific to 11.1.0.7
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: 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
Agenda

- Preparation
- Upgrade
- News and Task List
- Diagnostics & Tuning
- Performance Testing
- Best Practices

Fallback Strategies
Guidelines
Real World Experiences
400 databases from 9i to 11g

- Customer facts & figures
  - One of the most important international retailing companies
  - 2,221 outlets in 31 countries in Europe, Africa and Asia
  - ~280,000 employees
  - Revenue in 2007: € 64 billion
400 databases from 9i to 11g

• Tech facts:
  • Oracle 9.2.0.8
  • 400 databases
    • IBM AIX5L servers - each running 3 databases
    • Each database hosting 10 retail stores
  • Nightly OLTP batches
    • 25 GB redo per batch
    • Extremely tight timing constraints
  • Experience in the past when upgrading from 8.1.7 to 9.2:
    • Optimizer issues
    • Application changes were necessary
400 databases from 9i to 11g

• Goal:
  • Upgrade directly from Oracle 9i to Oracle Database 11g
  • Leverage new 11g features for performance optimization
    • Automatic SQL Profiling
  • Use SQL Performance Analyzer
    • Detection of plan changes
  • Use Database Replay
    • Functional Tests
  • Automate all steps for reusability
400 databases from 9i to 11g

- Setup:
  - IBM P670 - 16 CPUs - 32 GB RAM
  - EMC Storage DMX2000
  - 3 production database copies (Oracle 9.2.0.8)
  - Recorded nightly batch run - can be replayed anytime
    - Reference run time of batch run: 1:45:03
Real World Experience: Upgrade

- Upgrade directly from Oracle 9.2.0.8 to 11.1.0.6
  - Upgrade has been done with catupgrd.sql as part of a shell script starting a backup and doing additional sanity operations
  - Result:
    - Extremely easy
    - Just a few patches were necessary - no show-stoppers
    - Fully scriptable
    - Production rollout will be with 11.1.0.7

<table>
<thead>
<tr>
<th>Component</th>
<th>Status</th>
<th>Version</th>
<th>HH:MM:SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Server</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>03:06:48</td>
</tr>
<tr>
<td>JServer JAVA Virtual Machine</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:15:43</td>
</tr>
<tr>
<td>Oracle Workspace Manager</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:01:38</td>
</tr>
<tr>
<td>Oracle XDK</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:35:35</td>
</tr>
<tr>
<td>Oracle XML Database</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:04:28</td>
</tr>
<tr>
<td>Oracle Database Java Packages</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:00:33</td>
</tr>
<tr>
<td>Oracle Multimedia</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:05:49</td>
</tr>
<tr>
<td>Spatial</td>
<td>VALID</td>
<td>11.1.0.6.0</td>
<td>00:08:53</td>
</tr>
<tr>
<td>Gathering Statistics</td>
<td></td>
<td></td>
<td>00:08:18</td>
</tr>
</tbody>
</table>

Total Upgrade Time: 04:27:49
Test run 1: **Use all 9i parameters**

- Database successfully upgraded to 11g
- No adjustments had been done except for the required changes like `diagnostic_dest`
- Result:

![Bar chart showing comparison between Reference Run - Oracle 9i (01:45:03) and All 9i Parameters - Oracle 11g (02:42:31)]

- Conclusion:
  - While upgrade went very smoothly some tuning is clearly needed
Test run 2: **Use 11g defaults**

- Database successfully upgraded to 11g
- Oracle Database 11g default parameters have been set
- Result:

  ![Bar chart](chart.png)

- Conclusion:
  - It is recommended to remove relics from previous releases and start with the 11g defaults
Test run 3: Create system statistics

• System statistics get created for a workload period
  • exec DBMS STATS.GATHER_SYSTEM_STATS('start');
  • exec DBMS_STATS.GATHER_SYSTEM_STATS('stop');
  • See view AUX_STATS$

• Result:

• Conclusion:
  • Valid system statistics will help the optimizer to find better execution plans
Test run 4: `init.ora` optimization with SPA

- Using the SQL Performance Analyzer for 9i $\Rightarrow$ 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
...
Test run 4: init.ora optimization with SPA

- Using SPA reports to find the ideal init.ora settings:

<table>
<thead>
<tr>
<th>Parameters / Reports</th>
<th>CPU_TIME</th>
<th>BUFFER_GETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimizer_features_enable = 9.2.0</td>
<td>![CPU_TIME Table]</td>
<td>![BUFFER_GETS Table]</td>
</tr>
<tr>
<td>optimizer_features_enable = 11.1.0.6</td>
<td>![CPU_TIME Table]</td>
<td>![BUFFER_GETS Table]</td>
</tr>
<tr>
<td>optimizer_index_cost_adj = 75</td>
<td>![CPU_TIME Table]</td>
<td>![BUFFER_GETS Table]</td>
</tr>
<tr>
<td>optimizer_mode = first_rows_10</td>
<td>![CPU_TIME Table]</td>
<td>![BUFFER_GETS Table]</td>
</tr>
</tbody>
</table>
Test run 4: init.ora optimization with SPA

- Using SPA to optimize the init.ora
- Result:

- Conclusion:
  - SPA is a great tool to detect plan regressions before the upgrade
Test run 5: PL/SQL Native Compilation

- PL/SQL native compilation parameters:
  - plsql_code_type=native
  - plsql_optimization_level=3
  - Recompilation: exec DBMSUTILITY.COMPILE_SCHEMA('<username>');

- Result:

- Conclusion:
  - Native compilation lead to a slightly better overall completion timing
Test run 6: **SQL Profiling**

- Automatic SQL Tuning switched on
  - Upon every AWR snapshot the database will evaluate high-load SQL from the snapshot as possible candidates for the SQL Tuning Advisor
  - Goal is to create SQL Profiles automatically
Test run 6: SQL Profiling

- Automatic SQL Tuning switched on
  - Performance boost with 6 SQL Profiles - no DBA intervention necessary
  - No changes to the applications were required

- Result:

- Conclusion:
  - SQL Profiling speeds up SQL but doesn't change the application!!!
Real World Experience: Database Replay

- Using Database Replay for 9i $\Rightarrow$ 11g

- Conclusion: No application changes were necessary
Real World Experience: Results

• Upgrade directly from Oracle 9.2.0.8 to 11.1.0.7
  • Works extremely well and easier than expected
• SQL Performance Analyzer
  • Helps to detect plan changes right before the upgrade
  • Does a great job to optimize the init.ora
• Database Replay
  • Ensures that all application statements complete error-free
• SQL Profiling
  • Speed up queries and reports

• 50% less testing effort for upgrades to the new release
• Biggest surprise: No application changes necessary
• Production rollout has been started in May 2009
Real World Experience

- White Paper available on OTN

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

*An Oracle White Paper*  
*October 2008*
“Moving directly from Oracle 9i to Oracle Database 11g with Real Application Clusters, ASM and Data Guard was much easier and smoother than we’d expected.”

Dr. Lars Köller
Departmental Manager
Computing Center
Bielefeld University
Project 2: Bielefeld University

- Bielefeld University
  - 18000 students
  - 1500 employees
  - Initiative of Excellence
  - Bielefeld University will lead the way for 23 universities in Germany

- Setup before the upgrade:
  - Sun Solaris - Sun hardware
  - 10 databases - all Oracle 9.2.0.8
  - Highly critical applications

- Goals:
  - Move from single instance to RAC with ASM and Data Guard
  - Ensure very good performance
  - Run different Oracle versions in the same RAC environment
Project 2: Bielefeld University

- 5 phases:
  - Setup phase
  - Test phase
  - Performance phase
  - High Availability phase
  - Rollout phase
Project 2: Setup phase

• Phase 1 - Setup:
  • 2-node Sun Solaris cluster with shared storage
  • 3x Oracle 11.1.0.6 homes
    • 1 for the Oracle Clusterware
    • 1 for ASM
    • 1 as Database Home
  • 1x Oracle 10.2.0.4 home
  • 1 powerful Solaris machine has been set up as well to handle the standby
  • Identical test systems setup as well
  • Monitoring with Oracle Grid Control 10.2.0.4. and Grid Control 10.2.0.4 agents
Project 2: Test phase

• Phase 2 - Test:
  • All current database were successfully either upgraded or migrated (UTF8 changes involved) to their target releases
  • Mission-critical applications such as the University Class Schedule, the Central User Management System and the web presence received careful testing with internal test tools
    • Only a few changes were required because of the character set conversion
Project 2: Performance phase

- Phase 3 - Performance:
  - Using the SQL Performance Analyzer (SPA) to detect plan regressions and to perfect performance:

  Initial run – no tuning:

<table>
<thead>
<tr>
<th>SQL Category</th>
<th>SQL Count</th>
<th>Plan Change Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved</td>
<td>625</td>
<td>384</td>
</tr>
<tr>
<td>Regressed</td>
<td>147</td>
<td>74</td>
</tr>
<tr>
<td>Unchanged</td>
<td>1742</td>
<td>1132</td>
</tr>
</tbody>
</table>

  Second run – init.ora tuning:

<table>
<thead>
<tr>
<th>SQL Category</th>
<th>SQL Count</th>
<th>Plan Change Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved</td>
<td>658</td>
<td>410</td>
</tr>
<tr>
<td>Regressed</td>
<td>99</td>
<td>45</td>
</tr>
<tr>
<td>Unchanged</td>
<td>1756</td>
<td>1138</td>
</tr>
</tbody>
</table>

  Third run: final changes:

<table>
<thead>
<tr>
<th>SQL Category</th>
<th>SQL Count</th>
<th>Plan Change Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved</td>
<td>642</td>
<td>394</td>
</tr>
<tr>
<td>Regressed</td>
<td>94</td>
<td>48</td>
</tr>
<tr>
<td>Unchanged</td>
<td>1777</td>
<td>1151</td>
</tr>
</tbody>
</table>
Project 2: Performance phase

- Phase 3 - Performance:
  - Remedy the remaining 94 regressed statements with SQL Tuning Advisor
    - 34 SQL Profiles had been found and implemented
    - Some speed up SQL statement's performance by 99%
Project 2: High Availability phase

• Phase 4 - High Availability:
  • Standby database has been built up with RMAN automatically
  • Since Oracle 11g RMAN is able to build up the standby from the current production database without taking a backup and restores the standby through network transfer
  • Target: *Rolling Upgrade with a Transient Standby* will be used to apply patch set 11.1.0.7 with only 2 minutes of downtime

  *http://www.oracle.com/technology/deploy/availability/pdf/maa_wp_11g_transientlogicalrollingupgrade.pdf*
Project 2: Rollout phase

• Phase 5 - Rollout:
  • Go-live date: 12-September 2008
  • Rollout finished mid-October 2008
  • Already highest load on the system during start of the winter semester
    • 30% more logins
    • Response times decreased by 50%
  • Rolling Upgrade with a Transient Standby done:
    • Friday, 13th of February 2009
Project 2: Transient Standby

1. Guaranteed Restore Point
2. Logminer Build
3. KEEP IDENTITY
4. Upgrade with DBUA
5. SWITCHOVER
6. FLASHBACK DATABASE TO:
7. Guaranteed Restore Point
8. New $OH
9. Upgrade via Logshipping:
10. SWITCHOVER
Project 3: Upgrading 50 DWHs unattended

- International Oracle Partner
  - ~50 DWH solutions mostly installed at large TelCos
  - Database sizes vary between 8 TB and 12 TB
  - Application monitors network performance and bottlenecks
  - Up to 200,000 partitions in the database

- Goal:
  - Upgrade all databases from 10.2.0.2 to 11.1.0.7
  - Platform is RH Linux 64bit
  - Installation, configuration and upgrade: unattended
Project 3: Upgrading 50 DWHs unattended

- Silent installation:
  - Record a response file with OUI:
    ```
    ./runInstaller -record -destinationFile /tmp/inst.rsp
    ```
  - Customize the response file and use it for an unattended installation:
    ```
    ./runInstaller -silent -responseFile /tmp/inst.rsp
    ```

- Silent patch installation:
  - Install patch set with OUI in record mode and use the response file for a silent installation of the patch set

- Silent listener configuration:
  - Modify the delivered NETCA response file
  - Run: `./netca /silent /responseFile netca.rsp`
Project 3: Upgrading 50 DWHs unattended

- Silent upgrade with DBUA:
  - `dbua -silent -sid dwh`
  - `oracleHome /opt/oracle/product/RDBMS10g`
  - `diagnosticDest /opt/oracle/diag`
  - `sysDBAUserName sys`
  - `sysDBAPassword manager`
  - `recompile_invalid_objects true`
  - `degree_of_parallelism 4`
  - `emConfiguration LOCAL`
    - `dbsnmpPassword manager`
    - `sysmanPassword manager`
Project 3: Upgrading 50 DWHs unattended

- First successful production upgrade to 11.1.0.7 done: 22th of November 2008
Summary

- Upgrade to Oracle Database 11g is easy
- Stable database release
- Great features
- Let's go ... :-)

- Any questions, need help for the upgrade to 11g??
  - Mike.Dietrich@oracle.com
  - Carol.Tagliaferri@oracle.com
  - Roy.Swonger@oracle.com