Tuning All Layers Of E-Business Suite – Performance Topics

Isam Alyousfi, Senior Director
Lester Gutierrez, Senior Architect, EBS Applications Performance
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Objectives

• Overview of Performance Methodology for Optimizing E-business suite

• Outline best practices for Tuning E-Business layers
  – Tuning the Applications Tier
  – Tuning the Concurrent Manager
  – Tuning the Database Tier
  – Tuning the Applications
R12 Apps Basic Layers / Products

Browser

JVM1
- App1
- App2

JVM2
- App3
- App2

OC4J

mod_oc4j

mod_proxy

mod_plsql

mod_osso

Oracle Forms

Concurrent Processor, Manager

Ext. procs

Repository

RAC Database + db Listener

OID

oc4j_security

IdM

OAM

LDAP
Performance Methodology for Optimizing of E-business suite

• How to Approach a Performance Issue
• Where is the time going? Get the right diagnostics
  – Where?
  – What?
  – Why?
How to Approach a Performance Issue

1. DEFINE the problem clearly

1. GATHER the right data to analyze the issue

1. Identify the ROOT CAUSE of the problem, possibly gather additional data

1. Search for a KNOWN SOLUTION or workaround that addresses the root cause of the problem

1. If it is a product issue, PASS ON the right information to support/dev through the regular channels

1. Try to identify a TEMPORARY WORKAROUND to alleviate the issue while you get a product fix
Define the problem

• Get a clear understanding of the performance issue

What
Where
When
Why
How
Define the problem

Browser

Application (in MT or in DB)

Database

Sequence Diagram

Time spent in database
Executing task

Time spent in middle tier processing
Gather data

• Start with Best Practices: (note: 1121043.1)

• SQL Tuning
  – Trace files
  – SQLT output (note: 215187.1)
  – Trace Analyzer (note: 224270.1)
  – AWR Report (note: 748642.1)
  – AWR SQL Report (awrsqrpt.sql)
  – 11g SQL Monitoring
  – SQL Tuning Advisor

• PL/SQL Tuning
  – Product logs
  – PL/SQL Profiler (note: 808005.1)

• Reports Tracing
  – note: 111311.1

• Database Tuning
  – AWR Report (note: 748642.1)
  – ADDM report (note: 250655.1)
  – Automated Session History (ASH) Report
  – LTOM output (note: 352363.1)

• Forms Tuning
  – Forms Tracing (note: 373548.1)
  – FRD Log (note: 445166.1)
  – Generic note: 438652.1

• Middletier Tuning
  – JVM Logs
  – JVM Sizing (note: 362851.1)
  – JDBC Tuning (note: 278868.1)

• OS
  – OSWatcher (note: 301137.1)
Root Cause Analysis

• Where is the time going?

• Who is consuming time?

• Why?
Where is the time going? Get the right diagnostics

• Identify the components that do the processing
  – Refer to the architecture
    • which tier(s) play a part?
    • which components within a tier come into play?
    • 80% of issues are DB related
Root Cause Analysis: Check the basics

• Version levels
• Required Configuration
  – Init.ora: Notes 396009.1 (R12) & 216205.1 (11i)
• Latest patch levels
• Recommended Patch lists
  – Note “Recommended Performance Patches for the Oracle E-Business Suite”, 244040.1
  – Also check for product specific notes
• Available Fixes
• Best Practices in Performance for EBS
  – http://blogs.oracle.com/stevenChan/
Where is the time going? Get the right diagnostics

• If non-database (client side) processing occurs within the flow that was traced, you may see wait events like
  – SQL*Net Message%
  – TCP Socket%
  – PLSQL Lock Timer

• Other useful data in identifying where time is spent
  – Raw sql trace file
  – Product logs
  – App server logs such as http logs
  – Client process cpu consumption from OS (for example ps)
Tuning the Application Tier
Tuning the Applications Tier

• Forms
  – To reduce load on the database server, ensure users are optimally utilizing the professional Forms interfaces.
    – Avoid Blind queries
    – Provide selective criteria in Find windows and LOVs
  – Minimize network traffic and form open times
    – Avoid opening and closing forms across transactions
    – Combine forms from multiple-products onto a single menu
  – Generate complete SQL trace and Forms Runtime Diagnostics (FRD) to debug performance issues
Where is the time going? Get the right diagnostics

- **Client Process CPU Consumption**
  - Help -> Oracle Applications
  - Forms and Database process id & CPU Time
Tuning the Applications Tier

OC4J/JVM

• Use one JVM per 2 CPUs
• ~100 concurrent users per JVM
Tuning the Applications Tier

OC4J

• Response Time/CPU Usage
• OutOfMemoryErrors
Tuning the Applications Tier

Response Time/CPU Issues

• First, rule out any SQL issues.
  – Note 357597.1 on how to enable SQL trace for OA Framework applications

• If there are no database-related issues, then
  – you need to analyze the JVM

• Techniques you can use:
  – Thread dumps – Send to Oracle Support for analysis
  – GC logs: Configure JVM sizing
  – Profilers - Send to Oracle Support for analysis
Tuning the Applications Tier

• GC tuning
  – Total available memory is the most important factor affecting GC performance.
  – Review the frequency of collections, especially major collections (i.e. Full GC)
  – Enable verbose GC to tune heap sizes based on the GC traffic
  – Start with: -Xms1024M and -Xmx1024M
  – Setting -Xms and -Xmx to the same value avoid the need to memory allocation during runtime
  – If full GCs are too frequent, consider increasing Xms and Xmx
Tuning the Applications Tier

• Common causes of OutOfMemoryError
  – Memory Leak – Some objects not cleaned up
    • Strong reached references, connection objects, etc.
  – Memory Hemorrhage –
    • Some kind of infinite loop
    • Reading large data sets consuming vast amounts of memory in a short period of time
    • Connection leaks leading to vast object allocations
    • SQL Statements/caching may lead to excess memory used in JDBC libraries
  – Sizing Issues – Heap is undersized for expected load
Tuning the Applications Tier

• Symptoms of memory leak
  – Heap usage increases steadily, over a relative longer period of time (e.g. several hours or days)
  – When the amount of memory leak is large enough, you will start seeing continuous Full GC in the GC log
  – Different users may encounter OOM on totally unrelated flows
  – Generate Heap Dump files when OutOfMemoryError occurs
    • Help Oracle Support identify the root cause of the leaks
OA Fwk : JVM Process ID

Client Process CPU Consumption

- OA FWK – JVM Process ID
- The “About” link is at the bottom of each page
  - Requires FND_DIAGNOSTICS profile set to

```
System Property          Value                                    
APPLRGSF                 /u01/appmgr/visbde3comn/rgf/VISBDE3_bde-linux3
APPL_TOP                 /u01/appmgr/visbde3appl
APPS_JDBC_DRIVER_TYPE    THIN                                    
BNEDBCFILE               /u01/appmgr/visbde3appl/fnd/11.5.0/secure/VISBDE3_bde-linux3
CLIENT_PROCESSID         29334                                    
COMMON_TOP               /u01/appmgr/visbde3comn
DBFILE                   /u01/appmgr/visbde3appl/fnd/11.5.0/secure/VISBDE3_bde-linux3
DBLOCATION               /u01/appmgr/visbde3appl/fnd/11.5.0/secure/VISBDE3_bde-linux3
DB_HOST                  bde-linux4_us.oracle.com
```

Process ID of the JVM handling your session
Tuning the Concurrent Manager
Tuning the Concurrent Manager

– To maximize throughput for jobs which spawn parallel workers (i.e. Auto Invoice, Payroll), consider reducing the sleep time of the Conflict Resolution Manager (CRM)
  • Default is 60s, consider 5 or 10 seconds
– Increase the cache size (number of requests cached) to at least twice the number of target processes
– If the CM and Database are co-located, use IPC or bequeath connection
– Consider dedicating certain concurrent managers to process either short or long running programs to avoid queue backup
– Purge the FND tables on a regular basis using the “Purge Concurrent Request and Manager Data” program
Tuning the Concurrent Manager

• Workload Management
  – Trim the fat. Review and eliminate concurrent jobs that are not required and/or are not being used by your users.
  – Avoid enabling an excessive number of standard or specialized managers
  – A common guideline is between 1-2 target process per CPU, but this needs to be balanced with resources required by online activity
Tuning the Concurrent Manager

• Workload Management
  – Use specialization rules and work shifts to bind specific jobs to specific time windows
  • Helps avoid scheduling resource intensive batch requests during peak activity
  – Reschedule some programs to run when the concurrent managers have excess capacity and
  – Add more queues if all queues are running at maximum capacity only if resources permit.
Tuning the Concurrent Manager

• Transaction Managers - TMs
  – Used for synchronous online processing (ex: Inventory Transactions)
  – Ensure enough TMs exist to service the request load
    • Set the profile “Concurrent: Wait for Available TM” to 1 (second).
  – Set the sleep time on the TMs to a high number (e.g. 10 minutes)
    • Avoids constant polls to check for shutdown requests
Tuning the Client & Network
Tuning the Client

- Minimize the browser memory footprint
  - Minimize toolbars, browser themes, extensions, plug-ins, helper applications, and phishing controls
  - Do not minimize E-Business Suite components (but minimize other apps)
  - If there is a choice, use OAF/HTML (smaller memory profile)
- Upgrading RAM is generally more important than CPU
- Always keep menus small (customize if needed)
  - Menus are both memory and network intensive
- Oracle E-Business Suite 11i and 12 PC Client Performance
Tuning the Network

Routing Issues
Lost Packets due to device saturation

DB and MT must be co-located on a local private network

Traffic from MT to Client:
Consider using Local proxy in local offices (global single instance), (note 153287.1)
Tuning the Database
Tuning the Database Tier – Configuration

• Ensure mandatory init.ora parameters are set correctly
  – MOS Docs 396009.1 & 216205.1 (R12 & 11i respectively)
  – Use AWR advisories to optimize SGA/PGA sizing

• Required Patches & Fixes/Workarounds for Known Issues
  – Recommended Performance Patches -MOS Doc 244040.1
  – Required Patching – Interoperability Notes
  • 11gR1 Certified with 11i and R12 – MOS Docs 452783.1 & 735276.1
  • 11gR2 Certified with 11i – MOS Doc 881505.1
  • 10gR2 Uptake patch bundle 8362683, MOS Doc 791049.1
Tuning the Database Tier – Configuration

• Use key Database features
  – Auto Memory Management
    • Exception – Linux HugePages – MOS Doc 744769.1
  – System Managed Undo
  – Auto Segment Space Management
  – PL/SQL Native Compilation

• Convert to the OATM Tablespace Model
  – Consolidates # of Apps tablespaces from 400 to ~12
    • MOS Doc 248857.1
  – Best practices: LMT, ASSM, groups objects by size/type/access pattern
  – Migration utility can be used to migrate a schema at a time
  – Use AUTOALLOCATE extent sizes
Tuning the Database Tier – Disk IO

• I/O Optimization
  – Configure & verify direct IO is used if using file system datafiles
    • Improves performance/scalability by simulating raw devices, avoiding
      file system cache traffic
  – Ensure your IO subsystem can handle your peak IO load & IOPS
    • RAID, S.A.M.E. configs with sufficient spindles
    • Measure with ORION tool or in 11g
      DBMSRESOURCE_MANAGER.CALIBRATE_IO – MOS Doc 727062.1
  – Reduce the IO workload
    • Tune expensive SQL
    • Trim the workload of unnecessary tasks
    • Do 3X less IO on average with 11g Advanced Compression Option
Tuning the Database Tier – Statistics Gathering

• Use only FNDSTATS or the Gather Schema / Table Statistics Concurrent Programs
  – Do NOT USE the analyze or dbms_stats command directly. It is not supported, and can result in sub-optimal plans

• Use the GATHER_AUTO option to gather incrementally

• Analyze all schemas at 10%, then specific objects at x%
  – Due to data skew, some tables benefit from higher sampling %
  – For 11g, can use value of zero (AUTO_SAMPLE_SIZE)

• Disable the 10g/11g automatic job to gather stats
Tuning the Database Tier – Statistics Gathering

- Consider locking stats for very volatile tables once a representative set is gathered
  - Some Interface tables
  - Other examples: ap_selected_invoices, wsh_pr_workers
  - Use FND_STATS.LOAD_XCLUD_TAB to skip the table

- Do not gather statistics excessively on entire schemas or the entire database such as nightly or weekly

- FND_STATS now supports Database 11g Extended Stats. Patch 9470196. Also, in patch set 9962815
Tuning the Database Tier – Performance Management

• Establish baselines for different workloads
  – Use AWR baselines & diff reports
    • `DBMS_WORKLOAD_REPOSITORY` or Enterprise Manager
    • OS metrics with tools such as OS Watcher – MOS Doc 301137.1
  – Timings/Traces for critical flows & concurrent programs

• Use 11g’s SQL Performance Analyzer (SPA) & SQL Plan Management features (SPM)
  – Change management performance impact
  – Stabilize execution plans as needed & quickly restore “good” plans while analysis and permanent solutions are found
Tuning the Database Tier - AWR

- Regularly Review Automatic Workload Repository data
- Review Top 5 Timed Events section to determine utilization & bottlenecks and review relevant detailed sections for these

### Top 5 Timed Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Waits</th>
<th>Time(s)</th>
<th>Avg Wait(ms)</th>
<th>% Total Call Time</th>
<th>Wait Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>latch free</td>
<td>31,342,490</td>
<td>613,374</td>
<td>20</td>
<td>58.0</td>
<td>Other</td>
</tr>
<tr>
<td>CPU time</td>
<td>386,590</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>db file sequential read</td>
<td>13,474,726</td>
<td>37,319</td>
<td>3</td>
<td>3.5</td>
<td>User I/O</td>
</tr>
<tr>
<td>latch: cache buffers chains</td>
<td>2,270,036</td>
<td>12,802</td>
<td>6</td>
<td>1.2</td>
<td>Concurrency</td>
</tr>
<tr>
<td>db file scattered read</td>
<td>434,296</td>
<td>6,151</td>
<td>14</td>
<td>.6</td>
<td>User I/O</td>
</tr>
</tbody>
</table>
Tuning the Database Tier - AWR

- **For latch related waits**
  - Often due to non-sharable SQL, sub-optimal SQL which performs full table or full index scans, dynamic object creation/removal, etc.
  - Review the latch Statistics section to determine the hot latches
  - Trace some waiter and holder sessions to determine actual cause

- **For I/O related waits**
  - Review SQL sections by Logical/Physical reads and the Segment Statistics sections by I/O

- **For enqueue related or buffer busy waits**
  - Review the following sections: enqueue, segment statistics by buffer busy waits, row lock & ITL waits
Tuning the Database Tier - AWR

• Link AWR with ASH data
  – Map sessions & SQL to main bottlenecks
  – Query Active Session History (ASH) data
  – Enable tracing for a few of the sessions with heavy waits determine the actual cause and SQL statements.
  – Review the Advisory sections in AWR to fine tune SGA & PGA
  – Correlate top SQL in AWR to business flows
  – Produce and review SQL AWR reports for top SQLs (awrsqrpt.sql)
  – Monitor OS Stats (included in 11g AWR)
    • CPU and Memory Utilization & Paging, I/O statistics & response times
Tuning the Database Tier – DB Console

```sql
select item, item_id, item_description, inventory_item_id, item_identifier_type, decode
  (item_identifier_type, 'INT', 'I', 'CUST', '2', item_identifier_type) item_identifier_type_meaning, inventory_item_id, Address,
  cust_address, item_definition_level from ce_items_y where (UPPER(item) LIKE '3 AND (ITEM LIKE '4 OR ITEM LIKE '5
  OR ITEM LIKE '6 OR ITEM LIKE '7) AND ((sold_to_org_id = 8 OR sold_to_org_id is null) and (item_status = 'ACTIVE' OR
  item_status IS NULL) and (cross_ref_status = 'ACTIVE' or cross_ref_status IS NULL)) order by item
```

**Execution Plan**

Data Source: Cursor Cache
Plan Hash Value: 2245809145
Capture Time: Dec 8, 2004 11:28:18 AM
Parsing Schema: APPS

<table>
<thead>
<tr>
<th>Operation</th>
<th>Object</th>
<th>Order</th>
<th>Number of Rows</th>
<th>Cost (seconds)</th>
<th>CPU Cost</th>
<th>IO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td></td>
<td>48</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORT ORDER BY</td>
<td></td>
<td>47</td>
<td>31.793</td>
<td>27</td>
<td>1</td>
<td>7414759</td>
</tr>
<tr>
<td>VIEW</td>
<td>VIEW</td>
<td>46</td>
<td>31.793</td>
<td>28</td>
<td>1</td>
<td>3833386</td>
</tr>
<tr>
<td>UNION-ALL</td>
<td></td>
<td>452</td>
<td>10</td>
<td>11</td>
<td>1</td>
<td>127676</td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>30</td>
<td>1.0322</td>
<td>15</td>
<td>1</td>
<td>166450</td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>27</td>
<td>1.0263</td>
<td>12</td>
<td>1</td>
<td>142969</td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td></td>
<td>24</td>
<td>1.0263</td>
<td>11</td>
<td>1</td>
<td>127676</td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>21</td>
<td>1.018</td>
<td>10</td>
<td>1</td>
<td>112153</td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>18</td>
<td>1.0113</td>
<td>9</td>
<td>1</td>
<td>95430</td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td></td>
<td>15</td>
<td>1.0113</td>
<td>8</td>
<td>1</td>
<td>80017</td>
</tr>
</tbody>
</table>
Tuning the Database Tier –
Common corrective actions to maximize scalability

• If bottleneck is IO related...
  (db file sequential read, db file scattered read, log file sync, free buffer waits, etc...)
  – Tune Top SQL is always the first step
  – Maximize Memory availability, allocate generously to buffer cache, review AWR advisories & monitor swapping & paging
  – Use ASM or alternatively use the SAME methodology for db files
  – IO Sub-system:
    • RAID 10 still preferred most often for high end requirements  Ideal avg. response times of < 10ms.
  – Check for excessive/redundant indexing
  – Configure async IO, use quick IO technologies
Tuning the Database Tier –
Common corrective actions to maximize scalability

- If the bottleneck is concurrency related (enq %, latch%, buffer busy%, etc)

  - Use global hash-partitioned indexes for hot leaf blocks - Identify via AWR "Top logical IOs by Segment"
  - Some of these waits are commonly caused by bad SQL execution plans
    - i.e.: latch: cache buffer chains, %buffer busy waits, read by other session
  - Make sure to use ASSM and OATM tablespace model.
  - Increase INITRANS to alleviate ITL contention
  - Increase sequence caches
  - Work with support/consulting to evaluate if table/index partitioning will help
  - If the program allows, more frequent commits (batch size) can help by reducing the CR (consistent read) work required
Tuning the Database Tier –
Common corrective actions to maximize scalability

• If the bottleneck is RAC related ...
  \( gc \%, \text{global}\% \text{, etc...} \)
  
  – First do all the steps mentioned previously...
  – Check interconnect configuration -- Use Jumbo frames.
  – Do not use parallel query/dml across nodes
    • Use instance_groups & parallel_instance_groups settings
  – Review Parallel Concurrent Processing & Application Affinity config
    • 12.1.3 introduced option to define node affinity at the program level –
      Concurrent -> Program -> Define
  – Perform RAC focused AWR analysis …
Tuning the Database Tier –
Common corrective actions to maximize scalability

• RAC AWR Analysis
  1. Verify SQL execution is optimized – 10046 trace/SQL Monitoring
  2. Collect AWR data from all instances
     - `awrgrpt.sql` and `awrgdrpt.sql` (cluster-wide base and diff reports)
  1. Verify IO, CPU utilization and Interconnect messaging performance are within acceptable limits
  2. Drill down on cluster wait events + “SQL order by Cluster Waits”
  3. Correlate SQL executions on all instances accessing/modifying top segments in Segment Statistics by
     • Global Cache Buffer Busy & Buffer Busy Waits
     • CR/Current Blocks Received
     • Logical Reads
Tuning the Database Tier – TKPROF: Good, Bad & Ugly

• To get full value from tracing…

  – Raw trace must be
    • A complete, non-truncated trace file
      *** DUMP FILE SIZE IS LIMITED TO 12345 BYTES***
      • From before the user action began To the point the DB session ends**
  – Tkprof output MUST have
    • Runtime Execution Plans and Row Counts & Row Source Stats
    • Run Time Execution Plan = “Row Source Operation” != “Explain Plan”
  – Best practices, detailed how-to steps & recommended tracing patches documented
    • MOS Doc 1121043.1
      - Also review Appendix B – Getting In-Memory partial runtime stats
## Tuning the Database Tier – Good TKPROF

<table>
<thead>
<tr>
<th>call</th>
<th>count</th>
<th>cpu</th>
<th>elapsed</th>
<th>disk</th>
<th>query</th>
<th>current</th>
<th>rows</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>----------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>Parse</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Execute</td>
<td>7674758</td>
<td>780.79</td>
<td>728.70</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fetch</td>
<td>7674758</td>
<td>6328.78</td>
<td>6757.80</td>
<td>225987</td>
<td>153846532</td>
<td>0</td>
<td>7674758</td>
</tr>
<tr>
<td></td>
<td></td>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>----------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>total</td>
<td>15349517</td>
<td>7009.58</td>
<td>7486.52</td>
<td>225987</td>
<td>153846533</td>
<td>0</td>
<td>7674758</td>
</tr>
</tbody>
</table>

Misses in library cache during parse: 1
Misses in library cache during execute: 1
Optimizer mode: ALL_ROWS
Parsing user id: 173 (recursive depth: 2)

Rows | Row Source Operation
|------|---------------------------------------------------
| 7674758| SORT AGGREGATE (cr=153846532 pr=225987 pw=0 time=7174772779 us)
| 7675089| NESTED LOOPS (cr=153846532 pr=225987 pw=0 time=6919378560 us)
| 7675089| SORT UNIQUE (cr=123146176 pr=162809 pw=0 time=6083080435 us)
| 55878643| TABLE ACCESS BY INDEX ROWID AR_TRX_COGS_GT (cr=123146176 pr=162809 pw=0 time=5577872837 us)
| 136390387| INDEX RANGE SCAN AR_TRX_COGS_N1 (cr=23439707 pr=37706 pw=0 time=2136416980 us) (object id..)
| 7675089| TABLE ACCESS BY INDEX ROWID RA_CUSTOMER_TRX_LINES_ALL (cr=30700356 pr=63178 pw=0 time=539908369 us)
| 7675089| INDEX UNIQUE SCAN RA_CUSTOMER_TRX_LINES_U1 (cr=23025267 pr=2820 pw=0 time=196155435 us) (object ..)
Tuning the Database Tier

• Why upgrade EBS to Oracle Database 11g
  – Internal tests show 10-20% improvement in EBS batch & reporting workloads
  – Significant RAC Performance improvements
  – Oracle internal’s production EBS system on 11gR1;
    • 11gR2 scheduled for the summer
  – Lower cost with Advanced compression, better/automated diagnostics, lower-risk upgrade with SPM/RAT/SPA …
  – Best Practices/Tips white papers
    • twp_upgrading_10g_to_11g_what_to_expect_from_optimizer.pdf
    • http://www.oracle.com/apps_benchmark
11g SQL Tracing Enhancements

- New tracing parameter PLAN_STAT
  - first_execution | all_executions | adaptive*
- Connection pool problem to get row source stats is solved
- Key tracing patches* listed on MOS Doc 1121043.1
- Turn trace on for just 1 or a list of sql_id’s
- With real-time row source stats monitoring from v$sql_plan_statistics – no more waiting hours for a complete trace to get the right trace data

- Expanded event syntax:
  - ALTER SESSION SET EVENTS 'sql_trace [sql:a5ks9fhw2v9s1|56bs32ukywdsq] ...
  - sql_trace wait=true | false, bind=true | false, planstat=never | first_execution | all_executions | adaptive level = 1| 4| 8| 12 | 16
11g Automatic SQL Monitoring & Tuning

**Overview**
- **SQL ID**: 344h50pvdmnp
- **Parallel**: 16
- **Execution Started**: Sun Aug 29, 2009 12:42:09 PM
- **Last Refresh Time**: Sun Aug 29, 2009 12:50:29 PM
- **Execution ID**: 15777216
- **User**: DBTEST
- **Fetch Cells**: 0

**Time & Wait Statistics**
- **Duration**: 7.3m
- **Database Time**: 1.9h
- **PL/SQL & Java**: 0.0s
- **Wait Activity %**: 100

**IO Statistics**
- **Buffer Gets**: 71M
- **IO Requests**: 1479K
- **IO Bytes**: 45665
- **Cell Offload Efficiency**: -44.93%

**Details**

<table>
<thead>
<tr>
<th>Plan Hash Value</th>
<th>Name</th>
<th>Est...</th>
<th>Cost T...</th>
<th>Exec...</th>
<th>Act...</th>
<th>Mem...</th>
<th>Te...</th>
<th>I... R...</th>
<th>Call...</th>
<th>CPU Activity %</th>
<th>Wait Activity %</th>
</tr>
</thead>
<tbody>
<tr>
<td>3236879576</td>
<td>CREATE TABLE STATEMENT</td>
<td></td>
<td></td>
<td>17</td>
<td>1667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PX COORDINATOR</td>
<td></td>
<td></td>
<td>17</td>
<td>1667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PX SEND QC (RANDOM)</td>
<td>:TQ10000</td>
<td>724M</td>
<td>151K</td>
<td>16</td>
<td>1667</td>
<td>442MB</td>
<td>1021K</td>
<td>79</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOAD AS SELECT</td>
<td></td>
<td></td>
<td>15</td>
<td>1667</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PX BLOCK ITERATOR</td>
<td>724M</td>
<td>151K</td>
<td>16</td>
<td>750M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TABLE ACCESS STORAGE FULL</td>
<td>ROLLUP_TBL_H</td>
<td>724M</td>
<td>151K</td>
<td>3072</td>
<td>750M</td>
<td>457K</td>
<td>11</td>
<td>21</td>
<td>2.39</td>
<td></td>
</tr>
</tbody>
</table>
11g Performance Enhancements

• Optimizer improvements
  – SPM (SQL Plan Management)– quick relief while permanent solution arrives
  – Better density estimates for skewed data
  – Faster, more accurate statistics with AUTO_SAMPLE_SIZE
  – Extended statistics - address predicate correlation
  – Adaptive cursor sharing – to address bind peeking issues

• Some others…
  – Interval partitioning, Additional partitioning schemes
  – Text indexing performance enhancements
  – RAC scales better & AWR is more RAC aware
  – Many others..
11g Advanced Compression

- Data compression for all data types
  - Structured, Unstructured, Backup, Network
- Reduces storage consumption by 2 to 4 times
- Improves read performance
- Enhances memory, buffer cache utilization
- Complete application transparency
- Benefits diverse application workloads
11g Advanced Compression

- **Oracle Database 11g** extends table compression for OLTP data
  - *Support for conventional DML Operations (INSERT, UPDATE)*
  - New algorithm significantly reduces write overhead
    - Batched compression minimizes impact on OLTP transactions
  - No impact on reads
    - Reads data directly in its compressed format
    - Many reads actually see improved performance due to fewer I/Os and enhanced memory efficiency
11g Advanced Compression – Index Compression

- Compress multi-column indexes to reduce the storage overhead of repeated values
- Stores more keys per leaf block
- Less I/O, better performance of range scans
11g Advanced Compression - Compression Advisor

- Estimate Storage Savings for your case
  - Available in 11g Release 2 or ...
  - Available on OTN * for previous releases
  - Shows projected compression ratio for uncompressed tables
  - Reports actual compression ratio for compressed tables (11g Only)

```
SQL> set serveroutput on
SQL> execute dbms_compression.getratio(ownername=>'SH',tabname=>'SALES',sampling_percent=>10
Sampling table: SH.SALES
Sampling percentage: 10%
Expected Compression ratio with Advanced Compression Option: 2.96
PL/SQL procedure successfully completed.
SQL> 
```

11g Advanced Compression with E-Business Suite

• Oracle’s Internal E-Business Suite Production system deployed ACO in ‘09
  – Average overall storage savings - 3x
    • Table compression 4x
    • Index compression 2x
    • LOB compression 2.3x
  – 231TB of realized storage savings primary, standby & test systems
    • Additional benefits also accrued in Dev clones and backup

• Payroll, Order-2-Cash, AP/AR batch flows, Self-Service flows run without regression, Queries involving full table scans show speedup
11g Upgrade Best Practices for EBS

• White paper: Upgrade to Oracle 11g - Performance Best Practices

• Special attention to
  – Testing of 11g or 10g workloads with Real Application Testing
  – Managing SQL Execution Plans and minimizing performance degradation with 11g SQL Plan Management
Tuning the Applications
Tuning the Applications

• MyOracleSuport note “Recommended Performance Patches for the Oracle E-Business Suite”, 244040.1
  – Recommended performance patches for all the modules and tech. stack components are consolidated in this note

• Apply latest Release Update Packs
  • Functionality, fixes, and updates for all Release 12 functional applications and the technology stack
  • Several product family RUPs included are also available individually
Tuning the Applications

• Apply latest Release Update Packs
  – Release 12.1.3: Patch 9239090
    • MOS Doc: 1080973.1
  – Release 12.1: Patch 7303030
    • MOS Doc: 752619.1
  – Release 12.0 - RUP6: Patch 6728000
    • MOS Doc: 743368.1

• Review EBS Benchmark Publications
Tuning the Applications

• Performance Best Practices
  – Workflow
    • For background engines via Concurrent Manager, set the ‘Process Stuck’ parameter to ‘No’
      – Start a separate background engine to handle stuck processes with a low frequency –i.e. once or twice a day
    • Use deferred activities to improve online response times for flows such as Scheduling, PO Document Approval, etc.
    • Frequently purge runtime data (Conc Program: FNDWFPR)
    • Run $FND_TOP/sql/wffngen.sql to translate the activity function calls into static calls
    • Disable retention on Workflow Queues
      – DBMS_AQADM.ALTER_QUEUE(queue_name=>:b1,retention_time=>0);
    • For high volume batch processing in RAC use ITEM_TYPE to Node affinity
Tuning the Applications

• Performance Best Practices
  – **Purge & Archive**
    • Implement a regular purge process including transient data as well as obsolete transactional data
    • Purge old data before performing upgrades
    • Many documented standard programs available
      – ~260 purge programs in R12
  • Purge Portal
    – Purge programs can be configured, initiated and monitored
      • Set the execution frequency as well view history of purge programs
    – R12: programs tagged with the “Purge” program type
Tuning the Applications

• Runtime Performance Testing Tips
  – Use Automated, scripted tools
    • EBS Test Started Kits (Winrunner/QTP)
      – Bundled QA based automated scripts for EBS testing - Patch 8408886
    • Oracle Applications Testing Suite (Accelerators for EBS)
      – Web and Forms based flows
  – Complement with user participation tests and batch load tests with frequent and critical jobs
  – References
References
References

- R12.1 documentation roadmap (790942.1)
- “Oracle E-Business Suite Release 12.1 Info center” (806593.1)
- Database preparation guidelines for R12.1 upgrade (761570.1)
- Patching FAQs (459156.1, 225165.1)
- Using staged or shared APPL_TOP and distributed AD (734025.1, 384248.1, 236469.1)
- OAM “Patch Wizard” overview and FAQ (976188.1, 976688.1)
- AD Command Line Options for Release R12 (1078973.1)
- EBS 12.1.3 Data Model Comparison Report (1290886.1)
- Recommended Performance Fixes (244040.1)
- R12 Upgrade Sizing & Best Practices (399362.1)
SUMMARY

DO’s..
• Read and follow official documentation and have a project plan
• Engage Support early on to validate your upgrade plan
• Identify and execute tasks you can do today
• Test ! Test ! Test !
• Keep track of patches applied in test upgrades
• Be smart about using the right tools and explore published downtime reduction techniques
• Optimize patch runs to suite your H/W

DON’Ts..
• Ignore including relevant functional SMEs in the planning process from the beginning
• Skip failed or long running jobs as a regular practice
• Drop indexes
• Run adpatch in serial mode
• Perform workarounds without approval from support/DEV
• Insufficient time and/or # of rounds of upgrade and critical functional flow testing
Oracle Products Available Online

Oracle Store

Buy Oracle license and support online today at oracle.com/store
Hardware and Software

Engineered to Work Together
San Francisco 2011
October 2–6
Register and Exhibit Now! oracle.com/openworld