Tuning the Oracle E-Business Suite Environment

Isam Alyousfi – Senior Director, Applications Performance
Lester Gutierrez – Senior Architect, 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.
Session Agenda

- Architecture & Techstack
- Tuning the Applications Tier
- Tuning the Concurrent Manager
- Tuning the Client Tier & Network
- Tuning the Database Tier
- Tuning the Applications
- Upgrade Performance Tips
Oracle E-Business Suite R12 Architecture

Client

Java

Application

ORACLE
FUSION MIDDLEWARE APPLICATION SERVER

10g

OC4J
JSP
BC4J
UIX
BI Publisher
Forms

Database

ORACLE
DATABASE

10gR2
or
11g

Global Single Data Model

RAC & ASM

HTTP/S
Application & SQL Management
- Space Management
- Backup & Recovery Management
- Storage Management
- System Resource Management
- Data Management

Procedural stmts.
- SQL Statement Executor
- PL/SQL Executor
- Procedural Statements

PL/SQL engine
- CPU
- SGA
- Memory
- SQL engine

Buffer Cache
- Concurrency

SQL engine
- SQL Statements
- Procedural stmts.
- SQL Statements

100,000+ PL/SQL packages
40,000+ Tables
600,000+ SQL statements

200+ Products
FND, GL, AP, AR, OM, SC, ...

JAVA
- JDBC
- OJSP
- Forms
- BI Publisher
- OC4J
- Forms
- XML
- SOA
- BPEL

JDK
- Forms
- BI Publisher
- OC4JOA Framework
- XML

Web Listener
- Servlet Engine
- JSP
- BC4J
- UIX
- Reports
- Forms

Application & SQL Statement Executor

Concurrent manager
- Concurrent Manager

Application Server
- Application Server

Web Listener
- Servlet Engine
- JSP
- BC4J
- UIX
- Reports
- Forms

Application Server
- Application Server
Tuning the Applications Tier
Tuning the Applications Tier

• Upgrade to the latest certified technology stack
• Metalink Note 380482.1: Oracle E-Business Suite Release 12 Technology Stack Documentation Roadmap
• OA Framework Applications
  – Recommended Patches for Applications: note 275880.1 (Framework Roadmap)
• Oracle E-Business Suite Recommended Performance Patches
  – Metalink Note: 244040.1
• Technology Blog for latest certifications and releases:
  – http://blogs.oracle.com/stevenChan/
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
Tuning the Applications Tier

OC4J/JVM

- Use one JVM per 2 CPUs
- ~100 concurrent users per JVM
- JVM correct mode: use -server.
Tuning the Applications Tier

- Response Time/CPU Usage
- OutOfMemoryErrors
- EM Monitoring
- JDK 6 Jconsole/Visual VM
Tuning the Applications Tier

Response Time/CPU Issues

- Users complain about response time
  - configure Apache to log the time it takes to service a request
    - Edit: $ORA_CONFIG_HOME/10.1.3/Apache/Apache/conf/httpd.conf
      - LogFormat "%h %T
    - Logs: $LOG_HOME/ora/10.1.3/Apache/access_log*
**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
  - 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:** `-Xms512M` and `-Xmx512M`
  - 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

• GC tuning
  – Total available memory is the most important factor affecting GC performance.
  – Bigger heaps => Full GC will take longer
  – Longer GCs => users may experience pauses
  – Increase the number of JVMs used may provide the ability to handle more users with faster GC times.
    • Each JVM has a smaller Xmx,Xms memory footprint so GCs will be faster!
  – Also try to change the JVM collectors used and track the performance with Jconsole
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. serveral 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
  – Getting Heap Dump files when OutOfMemoryError occurs
    • Help Oracle Support identify the root cause of the leaks
Tuning the Applications Tier

- **JDBC Connection Identification**
  - Allows you to map the JDBC session from V$SESSION to a particular JVM process
  - Set automatically

<table>
<thead>
<tr>
<th>SID</th>
<th>MACHINE</th>
<th>PROCESS</th>
<th>MODULE</th>
<th>LOGON</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>aptier1.us.oracle.com</td>
<td>28806</td>
<td>JDBC Thin Client</td>
<td>10/22/06 17:26:43</td>
</tr>
<tr>
<td>44</td>
<td>aptier1.us.oracle.com</td>
<td>28807</td>
<td>JDBC Thin Client</td>
<td>10/22/06 17:29:17</td>
</tr>
</tbody>
</table>

aptier1> ps -ef | grep 28806

apps **28806** 28561  0 17:26:39 pts/20 0:00 /bin/sh ./java.sh
apps **28807** 28806 53 17:26:40 pts/20 8:55 /jdk1.4.2/bin/.../sparc/native_threads/ java
Tuning the Applications Tier

- JDBC Connections
  - Leaked connections will automatically be reclaimed and closed!
  - Connection leak will be logged in FND_LOG_MESSAGES
    - Allows a single view of all leaked connections for all JVMs
    - No need to monitor per JVM
Tuning the Applications Tier

- **Web Applications**
  - Ensure the users are trained to use the Logout or Home global buttons when completing their transactions.
    - Do not train them to use the browser close (“x”) link.
  - Logging out gracefully releases the memory and corresponding resources (i.e. connections, etc.)
    - Releases process resources sooner
Tuning the Applications Tier

• Pool Monitor can be used to monitor the Framework Applications as well as the JVM utilization
  – Application Module Pool
  – Memory Utilization
  – JVM Properties
  – Invoke with
    • Login ➔ Diagnostics ➔ Show Pool Monitor
Tuning the Applications Tier

- Diagnostics ➔ Show Pool Monitor
Tuning the Applications Tier
**Tuning the Applications Tier**

- **EM Monitoring (AD4J)**
  - Allows the ability to monitor multiple JVMs on different hosts
  - Monitor production machines in real-time with very low overhead < 1%
  - Generate and compare live heap dumps in production
  - No server restarts, no application instrumentation
  - Monitor transactions from AS to the Oracle DB.
  - View resource bottlenecks (DB, I/O, CPU, locks, thread hangs)
Tuning the Applications Tier

- AD4J
Tuning the Applications Tier

- JDK 6 – Jconsole
  - Allows heap dumps once connected to a JVM pid
  - Goto java.sun.management->HotSpotDiagnostic->Operations->dumpHeap
    - P0 : absolute file name (directory + filename)
    - P1: leave to true so that only reachable objects are dumped
- Use any binary hprof heap analyzer to analyze the heap dump and see what objects take up space
- JDK 6u7 and above has VisualVM
  - Does thread/heap dumps, profiling, graphs
Some Patches to Consider

- Num    Subject
- 8537237 PERFORMANCE ISSUE WITH TREE RENDERING IN IE
- 8514839 ADS12.1:CORE : POSSIBLE LEAKS IN MEMORY
- 8444976 EXCESSIVE/REDUNDANT JVM MEMORY USAGE IN ORACLEPREPAREDSTATEMENT
- 8363349 SLOW RESPONSE WHEN OPENING A SUBCONFIGURATION
- 8231536 W5C TOO FREQUENT JVM FULL GC'S REQUIRING JVM TO BE KILLED - SPINOFF
- 7646325 LINES BUDGET TOOK OVER 20 MINUTES TO DOWNLOAD INTO EXCEL
- 8840725 forward port of 7130147 to 12.0.6 (MDS patch 7326283).
- 7174340 enables restrictions for 1 session per named user (details on Metalink note#304209.1)
Tuning the Concurrent Manager
Tuning the Concurrent Manager

• General Tips
  – 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 Concurrent Manager

• Concurrent Reports
  – Ensure users provide selective parameters to the Reports
  – Review the Report output:
    • Number of rows & pages of the report.
    • Is it output being used by the business?
  – SQL Trace can be generated for slow reports
  – Reports (Tracing)
    • If the SQL trace for the report does not account for the elapsed time of the request, generate a Reports trace
    • Enabling Reports Trace
      – Use the Debug Options page at request submission
Tuning the Concurrent Manager

- SQL Tracing & PL/SQL Profiling
  - Program Level
    - Enable the trace/profiling option in the concurrent program definition form
  - Request Level
    - Use the Debug Options button in the Request Submission Form
    - The profile “Concurrent: Allow Debugging” should be set to ‘Yes’
Tuning the Concurrent Manager

- In RAC deployment, it is recommended to use Parallel Concurrent Processing (PCP)
- Maximize utilization of node capacity
- Use application affinity to distribute load
- Provide automated failover of workload in case of failure
Tuning the Client & Network
Tuning the Client

- White paper available
  - *Oracle E-Business Suite 11i and 12 PC Client Performance*
    - Recommendations to extend the life of low-spec clients
    - Answers the question: Is CPU or Memory more important?
  - Focuses on Windows XP across a range of clients
    - Describes how to measure client memory
    - Reviews OAF/HTML and form opening times
    - Network traffic analysis
    - Recommendations for browser optimization
    - Provides a holistic methodology for your own testing
**Tuning the Client**

- **R12 Client Performance Tips**
  - Minimize the browser memory footprint
    - Minimize toolbars, browser themes, extensions, plug-ins, helper applications, and phishing controls
    - The leanest technologies: IE6 with JInitiator (11i) JRE (R12)
    - 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
Tuning the Network

• Ideally DB and Application tiers should be co-located on an isolated private virtual GB network
  – Latency should ideally be ~1-2ms per round-trip
  – Users will notice performance drops with > 3ms
  – Can also impact concurrent programs
Tuning the Network

- E-Business Suite traffic tends to have large or small packets, with little in the middle
- Use Ping command to determine the latency of requests
  - Check that the priority of ping (ICMP) traffic is not reduced
  - Use default 32-byte packet to check
- Use a full size packet to highlight network problems
  - Maximum Transmission Unit (MTU) normally 1500 bytes for IP traffic
  - 28 bytes added by network card so ping using 1472 bytes
  - Check that the packet can be sent unfragmented using the –f flag (depends on platform)
Network Performance

C:\>ping 10.10.10.1
Pinging 10.10.10.1 with 32 bytes of data:
Reply from 10.10.10.1: bytes=32 time=227ms TTL=243
Reply from 10.10.10.1: bytes=32 time=192ms TTL=243
Reply from 10.10.10.1: bytes=32 time=193ms TTL=243
Reply from 10.10.10.1: bytes=32 time=191ms TTL=243
Ping statistics for 10.10.10.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 191ms, Maximum = 227ms, Average = 200ms

C:\>ping -l 1472 10.10.10.1
Pinging 10.10.10.1 with 1472 bytes of data:
Reply from 10.10.10.1: bytes=1472 time=360ms TTL=243
Reply from 10.10.10.1: bytes=1472 time=323ms TTL=243
Reply from 10.10.10.1: bytes=1472 time=260ms TTL=243
Reply from 10.10.10.1: bytes=1472 time=265ms TTL=243
Ping statistics for 10.10.10.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 260ms, Maximum = 360ms, Average = 302ms

- **Small packet**
  - If this fails: ICMP is blocked
  - If time is very high: ICMP traffic priority has been reduced or network problems

- **Large packet**
  - If the time is not much larger than the 32-byte packet then network is good
  - If the time is much larger then network problems
Network Performance

C:\$ping -l 1472 -f 10.10.10.1
Ping 10.10.10.1 with 1472 bytes of data:
Reply from 10.10.10.1: bytes=1472 time=330ms TTL=243
Reply from 10.10.10.1: bytes=1472 time=268ms TTL=243
Reply from 10.10.10.1: bytes=1472 time=252ms TTL=243
Reply from 10.10.10.1: bytes=1472 time=311ms TTL=243
Ping statistics for 10.10.10.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 252ms, Maximum = 330ms, **Average = 290ms**

C:\$ping -l 1473 -f 10.10.10.1
Ping 10.10.10.1 with 1473 bytes of data:
**Packet needs to be fragmented but DF set.**
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Ping statistics for 10.10.10.1:
   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

- **-f = Do not fragment the packet**
- **1472 + 28 = 1500 bytes**
- **1473 + 28 = 1501 bytes**
- **Cannot be sent as it would have to be broken into two smaller packets**
- **Useful to spot fragmentation and network segment problems**
Tuning the Network

• Application Server Web Cache
  – Can be used to reduce network traffic between the client and middle-tiers for OAF/HTML pages
    • Compression
    • Caching of images, style sheets, and Java script
  – Web Cache 10.1.2.2 is certified with R12
  – Refer to MetaLink note 380486.1
Tuning the Network

- Application Server Web Cache
  - Compression reduces E-Business Suite page sizes & improves response times considerably
Tuning the Database Tier
Tuning the Database Tier

- **Init.Ora**
  - Refer to the MetaLink document 396009.1 "Database Initialization Parameters for Oracle Applications" for R12.
    - Ensure mandatory parameters are set correctly
  - Optimally configure the buffer cache and shared pool as per the workload and the number of users
    - Undersized buffer cache >> excessive physical I/O
    - Undersized shared pool >> lib-cache / shared pool latch contention due to reloads, space shortage
Tuning the Database Tier

• **I/O Optimization**
  – Consider using Quick I/O or equivalent if using file system based files for the DB files
    • Quick I/O improves performance/scalability by simulating raw devices, avoiding file system cache traffic for DB files
  – Ensure your IO subsystem can handle your peak IO load & IOPS requirements - RAID, S.A.M.E. configs with sufficient spindles still recommended for high end systems

• **Use key performance features**
  – Auto Memory Management
  – System Managed Undo
  – Auto Segment Space Management
  – PL/SQL Native Compilation
Tuning the Database Tier

**AWR**
- Review Automatic Workload Repository (AWR) data
- Establish AWR baselines for different workloads and use AWR diff reports
- 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></td>
<td>386,590</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 continued**
  - For latch related waits
    - Latch contention is often a symptom due to a legitimate problem such as 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 & SQL statements
  - 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

• 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.
    • For CM, OA and Forms, Trace file identifier is automatically set to the Applications user name: prod11i_ora_2738_BASMITH.trc
  – Review the Advisory statistics sections in AWR to fine tune your memory pools
    • Buffer cache, Shared Pool, PGA Target, Java, Streams
  – 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
**Top Activity**

Drag the shaded box to change the time period for the detail section below.

**Detail for Selected 5 Minute Interval**

Start Time: Feb 3, 2008 4:15:18 PM PST

<table>
<thead>
<tr>
<th>Activity (%)</th>
<th>SQL ID</th>
<th>SQL Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.57</td>
<td>639wl1hqmef1</td>
<td>SELECT</td>
</tr>
<tr>
<td>10.01</td>
<td>7w9twxxv0v2qj</td>
<td>SELECT</td>
</tr>
<tr>
<td>5.40</td>
<td>5pjx5g6xmnn7</td>
<td>SELECT</td>
</tr>
<tr>
<td>4.33</td>
<td>bgdk8zs4o6fx</td>
<td>SELECT</td>
</tr>
<tr>
<td>4.13</td>
<td>3g6c13809b5jn</td>
<td>SELECT</td>
</tr>
<tr>
<td>3.94</td>
<td>gg6arddkhggg</td>
<td>PUSQL</td>
</tr>
<tr>
<td>3.65</td>
<td>d8b3zhkdq5pv</td>
<td>INSERT</td>
</tr>
<tr>
<td>2.51</td>
<td>36phq0q7w3yv</td>
<td>SELECT</td>
</tr>
<tr>
<td>2.51</td>
<td>2spqjy2b0d8</td>
<td>SELECT</td>
</tr>
<tr>
<td>2.31</td>
<td>cbr3azcp055n</td>
<td>SELECT</td>
</tr>
</tbody>
</table>

**Top Sessions**

<table>
<thead>
<tr>
<th>Activity (%)</th>
<th>Session ID</th>
<th>User Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.92</td>
<td>5160</td>
<td>APPS</td>
<td>sqlplus@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>7975</td>
<td>APPS</td>
<td>sqlplus@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>1211</td>
<td>APPS</td>
<td>NODEE_-_STANDARD@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>5699</td>
<td>APPS</td>
<td>sqlplus@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>5682</td>
<td>APPS</td>
<td>NODEE_-_STANDARD@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>6361</td>
<td>APPS</td>
<td>NODEE_-_STANDARD@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>3653</td>
<td>APPS</td>
<td>sqlplus@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>1234</td>
<td>APPS</td>
<td>sqlplus@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>2465</td>
<td>APPS</td>
<td>ALWAYS2@ants881 (TNS V1-V3)</td>
</tr>
<tr>
<td>1.92</td>
<td>7009</td>
<td>APPS</td>
<td>NODEE_-_LONG_RUNNING@ants881 (TNS V1-V3)</td>
</tr>
</tbody>
</table>

Total Sample Count: 12,096
Tuning the Database Tier

• Gathering Statistics
  – Do not gather statistics excessively on entire schemas or the entire database such as nightly or weekly
  – Use the GATHER_AUTO option to gather incrementally on what has changed significantly
  – Use only FND_STATS or the Gather Schema and Gather Table Statistics Concurrent Programs
    • Do NOT USE the analyze or dbms_stats command directly. It is not supported, and results in sub-optimal plans
  – Due to data skew, some products/tables benefit from higher sampling %
    • Analyze all schemas at 10%, then specific objects at x%
  – Disable the 10g/11g automatic job to gather stats
Tuning the Database Tier

• Gathering Statistics
  • Consider locking stats for very volatile tables once a representative set is gathered
    • Some Interface tables
    • Other examples: ap_selected_invoices, wsh_pr_workers
  • FND_STATS patch 6079369 recommended for using GATHER_SCHEMA_STATS with locked stats
  • Review the table and index statistics for the objects which appear in the top SQL section of AWR
    • fnd_stats.verify_stats
Tuning the Database Tier

- **Convert to the OATM Tablespace Model**
  - Consolidates number of Apps tablespaces from 400 to ~12 by type of data and access pattern
  - Uniform extents, locally managed, ASSM
    - No more fragmentation!
  - Migration utility can be used to migrate a schema at a time
    - MetaLink note 248857.1
    - ARU # is 3942506
  - Sample migration timings: (complete migration)
    - 1.4 Terabyte customer DB
      - Ran in 24 hrs, new size = 850 Gb
Tuning the Database Tier

- SQL Issue Triage notes
  - 10046 trace with waits is still essential
  - Complement with the Sqlt xplain utility (use XTRACT option)
    - Metalink note 215187.1
- Execution plans review with DBMS_XPLAN
  - DISPLAY_CURSOR – Gets “runtime plan” from v$sql_plan
  - DISPLAY_AWR – Gets “runtime plans” from DBA_HIST_SQL_PLAN

```sql
SQL> select * from table(dbms_xplan.display_awr('7qs7fx89194u1'))
SELECT PROFILE_OPTION_VALUE FROM FND_PROFILE_OPTION_VALUES WHERE PROFILE_OPTION_ID = :B4
AND APPLICATION_ID = :B3 AND LEVEL_ID = :B2 AND LEVEL_VALUE = :B1 AND ... 
Plan hash value: 1299660510
```

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost (%CPU)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td>3 (100)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>INDEX RANGE SCAN</td>
<td>FND_PROFILE_OPTION_VALUES_N2</td>
<td>1</td>
<td>28</td>
<td>3 (0)</td>
<td>00:00:01</td>
</tr>
</tbody>
</table>
Tuning the Database Tier

- Compare Execution plan Estimates to Actuals
  - **DISPLAY_CURSOR with Runtime Statistics** – Gets “runtime plan” from v$sql_plan and run statistics from v$sql_plan_statistics - ALLSTATS

SQL> select /*+ gather_plan_statistics */ count(*) from hz_parties where party_type='PERSON';
COUNT(*)
--------
17181756

SQL> select * from table(dbms_xplan.display_cursor(null,null,'ALLSTATS'));

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Starts</th>
<th>E-Rows</th>
<th>A-Rows</th>
<th>A-Time</th>
<th>Buffers</th>
<th>Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SORT AGGREGATE</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>00:04:06.05</td>
<td>1693K</td>
<td>1693K</td>
</tr>
<tr>
<td>* 2</td>
<td>TABLE ACCESS FULL</td>
<td>HZ_PARTIES</td>
<td>1</td>
<td>17M</td>
<td>17M</td>
<td>00:03:09.06</td>
<td>1693K</td>
<td>1693K</td>
</tr>
</tbody>
</table>

Predicate Information (identified by operation id):

2 - filter("PARTY_TYPE"='PERSON')

- E-Rows = Estimated Rows: analogous to regular plan_table #’s
- A-Rows = Actual rows: like the row source statistics in a tkprof output
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:
    - HW 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
Common corrective actions to maximize scalability

• If the bottleneck is concurrency related ... 
  (enq%, latch%, buffer busy waits, etc )
  – Use 10g global hash-partitioned indexes to reduce common index concurrency issues. Start with PK/UK/sequence driven indexes, and identify others via AWR "Top logical IOs by Segment"
  – These waits could again be caused by bad SQL execution plans for example :latch: cache buffer chains is many times seen as a result of this..so again, this should be covered first.
  – Make sure to use ASSM and OATM tablespace model.
  – Increase INITRANS to alleviate ITL contention
  – Increase sequence caches
  – Work with support and consulting to evaluate if further table/index partitioning will help your specific situation.
  – If the program allows, more frequent commits can help by reducing the CR (consistent read) work required
Common corrective actions to maximize scalability

• If the bottleneck is RAC related ...
  
  *(gc%, global%, etc...)*
  
  – First do all the steps mentioned previously...
  – Check interconnect configuration -- make sure private network is being used (gv$cluster_interconnects) and use ping/traceroute to validate optimal latency. Use Jumbo frames.
  – Do not use parallel query/dml accross nodes (instance_groups & parallel_instance_groups settings)
  – Use Parallel Concurrent Processing, using CM rules to minimize the execution of high volume/high frequency DMLs on the same objects concurrently across nodes.
  – This would include constraining child requests of a single parent to run in 1 node, as well as grouping heavy dml programs by node based on Application or DML activity to large transaction tables
Tuning the Database Tier

• Go to 11g
  – Recommended performance patches by DB version updated in Metalink 244040.1
  – 11gR1 Certified with 11i and R12 - Metalink 452783.1 & 735276.1
  – 11gR2 Certified with 11i - Metalink Note 881505.1
  – If still on 10gR2 - Uptake patch bundle 8362683, Metalink 791049.1

• Why upgrade EBS to 11g?
  – Internal volume tests show 10-20% improvement in EBS batch & reporting workloads
  – Initial issues in 11.1.0.6 shaken out in 11.1.0.7 / 11.2
  – Oracle internal already on 11g for one primary production system and all dev systems, GSI scheduled this fall
  – Several customers live, and lot’s more upgrading
  – Lower cost with compression, better/automated diagnostics, lower-risk upgrade with SPM/RAT/SPA …
Tuning the Database Tier

- **My favorite 11g performance features for EBS**
  - Advanced Compression
  - Automatic SQL Monitoring & Tuning
  - SQL Tracing Enhancements
  - Optimizer Improvements
    - SQL Plan Management
    - Extended Statistics
    - Adaptive Cursor Sharing
  - Better RAC performance & diagnostics
- **Upgrade without the pain**
  - Best Practices/Tips white papers
    - twp_upgrading_10g_to_11g_what_to_expect_from_optimizer.pdf
    - http://www.oracle.com/apps_benchmark
11g Advanced Compression Option with EBS

- **2X - 4X** Compression rates observed
- Oracle’s internal systems using this & production deployment is underway with ~ 5 TB of savings after compression of largest 100 tables.
- Payroll, Order, 2-Cash, AP/AR, batch flows, Self-Service flows tested and many others without regression.
- Query improvements seen, specially for unselective sqls & fts.
- Table block write penalty generally overcome by:
  - Read intensive OLTP Read/Write ratios
  - Index maintenance and other transaction processing components
  - Improved Buffer Caching efficiency
  - General reduced physical IO

Uncompressed Block

<table>
<thead>
<tr>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 • John Doe 2 • Jane Doe 3 • John Smith 4 • Jane 5 • Jack Smith</td>
</tr>
<tr>
<td>Free Space</td>
</tr>
</tbody>
</table>

Compressed Block

<table>
<thead>
<tr>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 • John Doe 2 • Jane Doe 3 • John Smith 4 • Jane 5 • Jack Smith</td>
</tr>
<tr>
<td>Free Space</td>
</tr>
</tbody>
</table>

Local Symbol Table

More Data Per Block

Oracle's internal systems using this & production deployment is underway with ~ 5 TB of savings after compression of largest 100 tables.
11g Advanced Compression Option with EBS

• Performance Recommendations
  – Get latest fixes -12 must-have fixes on 11.1.0.7 identified (Note: 244040.1)
  – Put compressed tables on ASSM tablespaces
  – Increase INITRANS, PCTFREE for compressed OLTP tables with heavy UPDATE activity
  – Start with large tables & with column count < 255
  – Skip setting compression as a tablespace default attribute
  – Table level DATE based partitioning can also be used to compress inactive data only

• White paper with further detailed guidelines for EBS underway
Almost as good as SQL trace in terms of diagnostic data, but always on for expensive SQL

Better than trace in terms of data presentation, ease of analysis

Intelligent, pragmatic tuning tests result in recommendations proven to work for you
11g SQL Tracing Enhancements

- PLAN_STAT – first_execution | all_executions
- Connection pool problem to get row source stats is solved
- Pursuing tkprof enhancement to aggregate row source stats for all_executions case
- Expect up to double trace size
- Turn trace on for just 1 or a list of sql_id’s..very cool, combined with real-time row source stats monitoring from v$sql_plan_stats – no more waiting hours for a complete trace to get the right trace data
- Timing issues fixed in bugs 7522002 and 7168259
- 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_executionsss|level = 1| 4| 8| 12 | 16
11g Optimizer Enhancements ..

• **Optimizer improvements**
  - SPM – SQL Plan Management – quick relief while permanent solution arrives
  - Better density estimates for skewed data
  - 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..
Tuning the Applications
Tuning the Applications

• MetaLink 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
    – Release 12.1 : Patch 7303030
      • Metalink Note 752619.1
    – Release 12.0 - RUP6: Patch 6728000
      • Metalink Note : 743368.1
Tuning the Applications

- Recommended Performance Fixes
  - **Create Accounting - XLA**
    - Improvements on the output file generation:
      - 7341297:R12.XLA.A
    - Core create accounting batch
      - 8319065:R12.XLA.A
      - 8250875:R12.XLA.A
    - Accounts Payables related accounting
      - XLA-AP : 8403738
  - Metalink Note 791049.1 - Optimizing Create Accounting using multiple workers
Tuning the Applications

- Recommended Performance Fixes
  - **Accounts Receivables**
    - July ‘09 CPC patch 7828946:R12.AR.A
      - Metalink Note ID: 862091.1

- With above fixes, AR batch/online benchmark set new performance records
  - Autoinvoice > 8 million /hr
  - Autorec/Autorem/Clearance each at over 550K/hr
  - Create Accounting ~ 3 million per hr
  - 10K online users - over 100K trxs per hr
  - RAC
Tuning the Applications

• Recommended Performance Fixes
  – Payables
    • Upgrade Scripts -
      – 7653100:R12.FIN_PF.A (appdstln.sql)
      – 7598653:R12.AP.A (apidstln.sql)
      – 7665691:R12.AP.A (aptppupg.sql)
  – Autoselect
    – 8637005
  – Invoice Validation -
    – 7866960:R12.AP.A
    – 7902867:R12.AP.A
  – Consolidated - Latest AP CPC -
    – 7665691:R12.AP.A + Patches 8263883, 7902867
Tuning the Applications

• Recommended Performance Fixes
  – iPayment – IBY
    • Build payment batch flow had issues with large batches
      – R12.IBY.A (12.0.4)- 8215621, 8309822, 8328642
      – R12.IBY.B (12.1.1) - 8328642
      – R12.CE. A - 8277703
  – Payables Payment Manager UI Performance
    – 7505803:R12.IBY.A
  – Funds Disbursement Dashboard
    – R12.IBY.A (12.0.2/4)- 7481665 , 8585497
    – R12.IBY.B (12.1.1) - 8585497
  – Record Print Status
    – R12.IBY.A (12.0.4) - 8805690
    – R12.IBY.B (12.1.1) - 8542151
Tuning the Applications

• Recommended Performance Fixes
  – **Incentive Compensation – CN**
    • Optimized Incentive Comp batch flow to scale on a RAC cluster of small, commodity servers with undirected load
  • Sales Credit Allocation & TIL Programs
    – R12.CN.A - 7524578, 7609836, 7672898, 835415, 8354130
  • Calculation
    – R12.CN.A - 7330382, 7487179, 7487171
    – Includes new trx summarization feature

• Achieved volume of ~ 1 million sales transactions in nightly batch window
Tuning the Applications

• **Recommended Performance Fixes**
  - **Time & Labor**
    • OTL was optimized to scale on RAC
      - global hash partitioned indexes
      - sequence cache changes
      - index consolidation
      - code optimization

• **Recommended Patches**
  - 8278235 - Payroll - NEED OBJECT CHANGES FOR OTL/PAYROLL PERF ON RAC ENABLED SYSTEMS
  - 7359347 - OTL ON RAC-- SEVERE SLOWDOWN ISSUE DURING TIMECARD ENTRY
**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)
      - Fix : 7538770
    - Run `$FND_TOP/sql/wffngen.sql` to translate the activity function calls into static calls
    - Other Recommended Fixes : , 5676227, 8931855
    - Disable retention on Workflow Queues
      - `DBMS_AQADM.ALTER_QUEUE(queue_name=>:b1, retention_time=>0);`
Tuning the Applications

- Performance Best Practices
  - **Order Management**
    - R12 Performance Fixes: Metalink Note 849060.1
    - Use the *streamlined* workflows
      - Removes unnecessary activities and sub-processes
      - Improves performance and of WF data volume significantly
      - Refer to MetaLink note 130511.1
    - Enable AutoScheduling & use Deferred tax & credit check
    - For online order entry, use the Quick Sales Order Form
    - Ensure that debugging is disabled:
      - OM: Debug Level, set to 0 for OFF
      - QP: Debug Mode, set to Request Viewer Off
      - WSH: Debug Enabled, set to No
      - WSH: Debug Level, set to 0 for OFF
Tuning the Applications

- Performance Best Practices
  - Payroll
    - Tune the number of threads, chunk size and the buffer sizes (pay_action_parameters):
      - THREADS=25 *
      - CHUNK_SIZE = 20
      - BAL BUFFER SIZE=550
      - EE BUFFER SIZE =550
      - RR BUFFER SIZE =550
      - COST BUFFER SIZE=550
    - Set threads to ~1.5 times the number of data server processors *
    - Set LOW_VOLUME=N to enable optimal plans
Tuning the Applications

• Performance Best Practices
  – Accounts Receivable
    • Profile Option: AR: AutoInvoice Gather Statistics
      – Set to "NO" once representative stats have been gathered for RA_INTERFACE* tables
    • Index trx flexfields for headers & lines used in validation
      – interface_line_attributeN columns
  • Parallelize batch flow
    – By supplied parameters where possible
      • Example: Autoinvoice
    – Manually by data sets if needed
      • Example: Rev Rec, AutoReceipts/Remittance, Auto-accounting (R12)
Tuning the Applications

• Performance Best Practices
  – Logging
    • FND: Debug Log Enabled - Set it to ‘Yes’
    • FND: Debug Log Level - Set to Unexpected (Level=6)
    • Query fnd_log_messages to determine the logging traffic
    • If the profiles are set as recommended above, you should not see many messages logged
      – If the query shows a large number of rows being created, there could be exceptions and errors which need to be investigated
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

- Accessing the Purge Portal
  - System Administrator > Oracle Applications Manager > Purging/Critical Activities
Upgrade Performance Tips
Upgrade Performance Tips

- See Metalink Note 403339.1 for supported combinations of Apps and RDBMS for R12 upgrade
- Pre
  - Convert to OATM
  - Upgrade to 10gR2
  - Gather stats - gather_auto
  - Run pre-upgrade processes documented to reduce downtime
  - Assign post upgrade jobs to specialized CM queue
- During
  - Add "extension plsql_no compile yes" line in u4440000.drv file to optimize plsql creation and compilation
  - Test & tune batch size, # of workers and init.ora upgrade settings
    - Suggested values and reference configuration & sizing documented in upgrade manual
Upgrade Performance Tips

- Rule of thumb / Sensible testing starting points:
  - 10K batch size
  - # of AD Workers = 1.5 - 2 X # CPUs in DB host
  - Increase parallel_max_servers = 2X #CPUs
  - Increase job_queue_processes = # of CPUs
  - Increase db_cache_size, pga_aggregate_target and redo log_buffer
- Adjust above settings constrained by factors such as memory and CPU utilization (scale down if at 100%) & I/O response times (scale down if over 20 ms)
- Recommended Performance Fixes:
  - AD-Parallel improved scalability : 8917381
  - Forms/reports generation regression with 11g : 8557019
## Key Metalink References

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Initialization Parameters and Configuration for Oracle Applications Release 11i</td>
<td>216205.1</td>
</tr>
<tr>
<td>Guidelines to setup the JVM in Apps Ebusiness Suite</td>
<td>362851.1</td>
</tr>
<tr>
<td>Database Initialization Parameters for Oracle Applications Release 12</td>
<td>396009.1</td>
</tr>
<tr>
<td>Oracle E-Business Suite Recommended Performance Patches</td>
<td>244040.1</td>
</tr>
<tr>
<td>Upgrading Developer 6i with Oracle Applications 11i</td>
<td>125767.1</td>
</tr>
<tr>
<td>Upgrading Forms and Reports 10g in Oracle Applications Release 12</td>
<td>437878.1</td>
</tr>
<tr>
<td>Using AutoConfig to Manage System Configurations with Oracle Applications 11i</td>
<td>165195.1</td>
</tr>
<tr>
<td>Using AutoConfig to Manage System Configurations in Oracle E-Business Suite R12</td>
<td>387859.1</td>
</tr>
<tr>
<td>Overview of Using Java with Oracle E-Business Suite Release 11i</td>
<td>300482.1</td>
</tr>
<tr>
<td>Upgrading to J2SE 1.4.2 with Oracle Applications 11i</td>
<td>246105.1</td>
</tr>
<tr>
<td>Using J2SE Version 5.0 with Oracle E-Business Suite 11i</td>
<td>304099.1</td>
</tr>
<tr>
<td>Latest 11i Applications Recommended Patch List</td>
<td>123456.1</td>
</tr>
<tr>
<td>Oracle Application Framework Release 11i Documentation Roadmap</td>
<td>275880.1</td>
</tr>
</tbody>
</table>
# Key Metalink References

<table>
<thead>
<tr>
<th>Title</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementing and Using the JSP Precompiler</td>
<td>215268.1</td>
</tr>
<tr>
<td>AOL/J JDBC Connection Pool White Paper</td>
<td>278868.1</td>
</tr>
<tr>
<td>Oracle Applications Tablespace Model Release 11i - Tablespace Migration Utility</td>
<td>248857.1</td>
</tr>
<tr>
<td>Oracle Applications Release 11i with Oracle 10g Release 2 (10.2.0)</td>
<td>362203.1</td>
</tr>
<tr>
<td>Installing and Configuring Web Cache 10.1.2 and Oracle E-Business Suite 11i</td>
<td>306653.1</td>
</tr>
<tr>
<td>Installing and Configuring Web Cache 10g and Oracle E-Business Suite 12</td>
<td>380486.1</td>
</tr>
<tr>
<td>Oracle 10gR2 Database Preparation Guidelines for an E-Business Suite Release 12 Upgrade</td>
<td>403339.1</td>
</tr>
</tbody>
</table>
TakeAway

• Know Your system
• Utilize tools available
• Plan Hardware for peak load
• Plan to run heavy batches during light system activities
• Use DB monitoring tools (EM)
• Use MT monitoring tools (EM, AD4J)
• Plan CM scheduling carefully
• Manage statistics
• Purge/Archive old data
• Test, Test, Test
For More Information

search.oracle.com

or

oracle.com
ORACLE IS THE INFORMATION COMPANY