Tuning All Layers of the Oracle E-Business Suite

Isam Alyousfi
Senior Director
Applications Performance Group
Oracle Corporation

Lester Gutiérrez
Senior Architect
Applications Performance Group
Oracle Corporation
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
- Scalability Tips
R12 Technology Stack

Client
- User Interface

Application logic
- Web Listener
  - OC4J
  - JSP
  - BC4J
  - UIX
  - Reports
  - Forms

Database
- Database logic

Database
- 10g

Application Server
- ORACLE

R12 Technology Stack
- 10g

Technology Stack
- Client
- Application
- Database
Tuning the Applications Tier
Tuning the Applications Tier

- Upgrade to the latest certified technology stack
  - OC4J: OracleAS 10g 10.1.3.3.0, Metalink Note 454811.1
  - Forms: OracleAS 10.1.2.2, Metalink Note 437878.1
  - ATG: RUP 12.0.4 (Patch 6272680)
- Upgrade to the latest JDK
  - Metalink Note 418664.1 for using Java with R12
  - Metalink Note 300482.1 for the latest certifications

<table>
<thead>
<tr>
<th>Apps Version</th>
<th>JDK Version</th>
<th>Metalink Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>R12</td>
<td>1.5 u8 or higher</td>
<td>384249.1</td>
</tr>
<tr>
<td>R12</td>
<td>1.6 update 2</td>
<td>455492.1</td>
</tr>
</tbody>
</table>
**Tuning the Applications Tier**

- **Forms**
  - Deploy with socket mode for internal users:
    - R12: Refer to Note 384241.1.
  - Enable Forms Dead Client Detection
    - Value specified in minutes: FORMS_TIMEOUT=10
    - Terminates fwebmx processes for dead clients.
  - Enable Forms Abnormal Termination Handler
    - Do not set FORMS_CATCHTERM
  - Disable Cancel Query
    - Cancel Query increases middle-tier CPU as well as DB CPU
    - Refer to MetaLink Note 138159.1 on how to enable and tune Cancel query related parameters
    - To Disable Cancel Query
      - Set the Profile “FND: Enable Cancel Query” to ‘No’
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
  - No more than one JVM/CPU
  - No more than 100 concurrent users per JVM
- JVM correct mode: use -server.
Tuning the Applications Tier

- Response Time/CPU Usage
- OutOfMemoryErrors
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 => GC will take longer
  - Longer GCs => users may experience pauses
  - 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
  • Memory Hemorrhage –
    • Some kind of infinite loop
    • Reading large data sets consuming vast amounts of memory in a short period of time
  • 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

• OA Framework Applications
  • Recommended Patches for Applications: note 275880.1 (Framework Roadmap)
  • Upgrade to the latest JDBC Drivers
    • 10.2.0.3 JDBC Drivers are bundled with ATG RUP3 6077669 and above
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 \rightarrow Diagnostics \rightarrow Show Pool Monitor
Tuning the Applications Tier

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

- R12 Application Server Control
  - Included in Apps Server 10.1.3.1
  - Allows you to monitor JVM usage through a web interface
    - Number of sessions
    - Heap Usage
    - Number of active threads
    - Thread deadlocks
  - Access via the URL: http://<site>:<port>/ascontrol
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 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

```bash
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
C:\> ping -l 1472 -f 10.10.10.1
Pinging 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
Pinging 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
    - A poorly sized buffer cache results in excessive physical I/O
    - A poorly sized shared pool results in library cache and shared pool latch contention due to reloads and lack of space
Tuning the Database Tier

• I/O
  • Enable asynchronous I/O
  • Consider using Quick I/O or equivalent if using file system based files for the DB files
    • Quick I/O improves performance and scalability by simulating raw devices, and avoiding file system cache traffic for the DB files
  • Use locally managed temp files (uniform) for the temporary tablespace
  • Review Automatic Workload Repository (AWR) data
Tuning the Database Tier

- AWR
  - Review Automatic Workload Repository (AWR) data

WORKLOAD REPOSITORY report for

<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>GSCU</td>
<td>317772662</td>
<td>gscu</td>
<td>3</td>
<td>10.2.0.3.0</td>
<td>YES</td>
<td>rgt073</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Snap ID</th>
<th>Snap Time</th>
<th>Sessions</th>
<th>Cursors/Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin Snap:</td>
<td>17886</td>
<td>22-Feb-07 18:00:41</td>
<td>178</td>
</tr>
<tr>
<td>End Snap:</td>
<td>17891</td>
<td>22-Feb-07 23:00:42</td>
<td>179</td>
</tr>
<tr>
<td>Elapsed:</td>
<td>300.02 (mins)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB Time:</td>
<td>17,623.71 (mins)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Report Summary

Cache Sizes

<table>
<thead>
<tr>
<th></th>
<th>Begin</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Cache:</td>
<td>8,192M</td>
<td>8,192M</td>
</tr>
<tr>
<td>Shared Pool Size:</td>
<td>6,144M</td>
<td>6,144M</td>
</tr>
<tr>
<td>Std Block Size:</td>
<td>8K</td>
<td></td>
</tr>
<tr>
<td>Log Buffer:</td>
<td>8,448K</td>
<td></td>
</tr>
</tbody>
</table>
Tuning the Database Tier

- AWR
  - Review the 10g Time Model statistics & Wait Class summary to establish where the time is going and the nature of the major bottlenecks

<table>
<thead>
<tr>
<th>Statistic Name</th>
<th>Time (s)</th>
<th>% of DB Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>sql execute elapsed time</td>
<td>1,057,029.04</td>
<td>60.06</td>
</tr>
<tr>
<td>DB CPU</td>
<td>306,590.45</td>
<td>16.50</td>
</tr>
<tr>
<td>sequence bad elapsed time</td>
<td>4,571.39</td>
<td>0.43</td>
</tr>
<tr>
<td>PL/SQL execution elapsed time</td>
<td>87.64</td>
<td>0.01</td>
</tr>
<tr>
<td>parse time elapsed</td>
<td>66.03</td>
<td>0.01</td>
</tr>
<tr>
<td>connection management call elapsed</td>
<td>87.05</td>
<td>0.00</td>
</tr>
<tr>
<td>hard parse elapsed time</td>
<td>32.91</td>
<td>0.00</td>
</tr>
<tr>
<td>PL/SQL compilation elapsed time</td>
<td>20.72</td>
<td>0.00</td>
</tr>
<tr>
<td>failed parse elapsed time</td>
<td>1.85</td>
<td>0.00</td>
</tr>
<tr>
<td>hard parse (sharing criteria) elapsed</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>repeated bind elapsed time</td>
<td>0.07</td>
<td>0.00</td>
</tr>
<tr>
<td>hard parse (bind mismatch) elapsed</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>DB time</td>
<td>1,057,422.54</td>
<td></td>
</tr>
<tr>
<td>background elapsed time</td>
<td>29,313.43</td>
<td></td>
</tr>
<tr>
<td>background cpu time</td>
<td>20,687.97</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wait Class</th>
<th>Waits</th>
<th>% Time-outs</th>
<th>Total Wait Time (s)</th>
<th>Avg wait (ms)</th>
<th>Waits /txn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>33,444,820</td>
<td>5.87</td>
<td>613,897</td>
<td>13</td>
<td>1,102.12</td>
</tr>
<tr>
<td>User I/O</td>
<td>16,734,496</td>
<td>0.00</td>
<td>46,445</td>
<td>3</td>
<td>551.46</td>
</tr>
<tr>
<td>Concurrency</td>
<td>5,403,582</td>
<td>0.01</td>
<td>13,890</td>
<td>3</td>
<td>177.97</td>
</tr>
<tr>
<td>System I/O</td>
<td>2,504,203</td>
<td>0.00</td>
<td>10,065</td>
<td>4</td>
<td>82.52</td>
</tr>
<tr>
<td>Configuration</td>
<td>700,938</td>
<td>15.00</td>
<td>6,881</td>
<td>13</td>
<td>23.39</td>
</tr>
<tr>
<td>Commit</td>
<td>19,062</td>
<td>0.00</td>
<td>2,104</td>
<td>110</td>
<td>0.53</td>
</tr>
<tr>
<td>Cluster</td>
<td>5,163</td>
<td>0.60</td>
<td>26</td>
<td>5</td>
<td>0.17</td>
</tr>
<tr>
<td>Application</td>
<td>244</td>
<td>3.28</td>
<td>18</td>
<td>73</td>
<td>0.01</td>
</tr>
<tr>
<td>Network</td>
<td>68,829</td>
<td>0.00</td>
<td>2</td>
<td>0</td>
<td>2.30</td>
</tr>
</tbody>
</table>

ORACLE
Tuning the Database Tier

- AWR
  - Review Top 5 Timed Events section in the AWR reports to determine utilization & bottlenecks

<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</td>
<td>2,270,036</td>
<td>12,802</td>
<td>6</td>
<td>1.2</td>
<td>Concurrency</td>
</tr>
<tr>
<td>chains</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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

- 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 of the AWR report 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

- 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: `prodlll_i_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
- Monitor OS Stats
  - CPU and Memory Utilization & Paging, I/O statistics & response times
**Tuning the Database Tier**

- Enable the Auto memory manager
  - `Pga_Aggregate_Target` - set to tune the PGA for the entire instance
  - Automatically tunes hash area and sort area sizes
  - Returns unused memory to the OS
- Enable System Managed Undo
  - No more ORA-1555s!
  - Set `undo_retention` to the length of the longest running request
- Utilize Auto Segment Space Management
  - No need to manually set Freelist/freelist groups. A bitmap is used to automatically scale the segment related parameters
- Enable PL/SQL Native Compilation
  - Improves PL/SQL execution performance
  - Improves scalability by removing a lot of stress off of the shared pool
Tuning the Database Tier

• Be sure to leverage what 10g offers in terms of performance & manageability features
  • DB Console – management portal, part of 10g Grid Control
  • What happened ? - AWR & ASH
  • What to do ? - ADDM & SQL Tuning Advisor
  • Runtime exec plan data
    • v$sql_plan, v$sql_bind_capture & v$sql_plan_statistics
  • SQL Identification - V$SQL (PROGRAM_ID, PROGRAM_LINE#)
  • Global-hash partitioning indexes
  • Online segment shrink – HWM Fix
  • Many more…
DB Console

SQL Details: f0xdmd0h7xg54

```
SELECT item_id, item_description, inventory_item_id, item_identifier_type, decode
  (item_identifier_type, 'INT', 1, 'CUST', 2, item_identifier_type) item_identifier_type Meaning, inventory_item_id, Address,
cost_address, item_definition_level FROM oe_items_v 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 = 0 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          Current Statistics          Execution History          Tuning History

Data Source: Cursor Cache  Plan Hash Value: 2245903145  Module: OEXOEORD
Parsing Schema: APPS

<table>
<thead>
<tr>
<th>Operation</th>
<th>Object</th>
<th>Object Type</th>
<th>Order</th>
<th>Number of Rows</th>
<th>KB Cost</th>
<th>Time (seconds)</th>
<th>CPU Cost</th>
<th>IO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORT ORDER BY</td>
<td></td>
<td>VIEW</td>
<td>47</td>
<td>31.793</td>
<td>27</td>
<td>17414759</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>UNION-ALL</td>
<td></td>
<td>VIEW</td>
<td>45</td>
<td>31.793</td>
<td>26</td>
<td>1833366</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>VIEW</td>
<td>30</td>
<td>10.322</td>
<td>15</td>
<td>166460</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>VIEW</td>
<td>27</td>
<td>10.263</td>
<td>12</td>
<td>142969</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td></td>
<td>VIEW</td>
<td>24</td>
<td>10.196</td>
<td>11</td>
<td>127576</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>VIEW</td>
<td>21</td>
<td>1.015</td>
<td>10</td>
<td>112153</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS OUTER</td>
<td></td>
<td>VIEW</td>
<td>18</td>
<td>1.0113</td>
<td>9</td>
<td>95400</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td></td>
<td>VIEW</td>
<td>15</td>
<td>1.0104</td>
<td>8</td>
<td>80007</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

ORACLE
Tuning the Database Tier

• Gathering Statistics
  • Do not gather statistics excessively on entire schemas or the entire database such as nightly or weekly
  • Do not gather statistics on permanent objects during peak intervals
    • Gathering statistics invalidates cursors
      • Unless you use the ‘No Invalidate’ option
  • Gathering statistics requires dictionary and object level locks
  • Plans are not likely to change if the data distribution has not changed
Tuning the Database Tier

• Gathering Statistics
  • For tables which are growing at a rapid rate, gather statistics only on those tables
  • 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 benefit from higher sampling %
    • Analyze all schemas at 10%, then specific objects at x%
  • Review the table and index statistics for the objects which appear in the top SQL section of AWR
Tuning the Database Tier

- Gathering Schema Statistics
  - Auto Gather option
    - Gather Statistics only on tables which have changed
    - Change threshold is user definable (10% default)
    - Utilizes the Table Monitoring feature
  - Auto List option
    - Lists the objects which have changed
  - Maintains history of Statistics Collection
Tuning the Database Tier

- Gather Schema Statistics
Tuning the Database Tier

- Verifying the Statistics

```sql
APPS@GSIFIN12> exec fnd_stats.verify_stats('RECEIVABLE','HZ_PARTIES');
```

<table>
<thead>
<tr>
<th>Table</th>
<th>HZ_PARTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>last analyzed</td>
<td>03-07-2007</td>
</tr>
<tr>
<td>sample_size</td>
<td>3929124</td>
</tr>
<tr>
<td>num_rows</td>
<td>3929124</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index name</th>
<th>last analyzed</th>
<th>num_rows</th>
<th>LB</th>
<th>DK</th>
<th>LB/key</th>
<th>DB/key</th>
<th>CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HZ_PARTIES_N1</td>
<td>03-07-2007</td>
<td>39715080</td>
<td>296390</td>
<td>11801124</td>
<td>1</td>
<td>2</td>
<td>34043210</td>
</tr>
<tr>
<td>HZ_PARTIES_N10</td>
<td>03-07-2007</td>
<td>212533</td>
<td>955</td>
<td>181904</td>
<td>1</td>
<td>1</td>
<td>202858</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>HZ_PARTIES_N2</td>
<td>03-07-2007</td>
<td>17359940</td>
<td>69250</td>
<td>527100</td>
<td>1</td>
<td>30</td>
<td>16045910</td>
</tr>
<tr>
<td>HZ_PARTIES_N3</td>
<td>03-07-2007</td>
<td>15502780</td>
<td>53720</td>
<td>205570</td>
<td>1</td>
<td>69</td>
<td>14325460</td>
</tr>
<tr>
<td>HZ_PARTIES_N4</td>
<td>03-07-2007</td>
<td>22260680</td>
<td>128580</td>
<td>3223752</td>
<td>1</td>
<td>6</td>
<td>21253800</td>
</tr>
<tr>
<td>HZ_PARTIES_N5</td>
<td>03-07-2007</td>
<td>40528170</td>
<td>304000</td>
<td>9971951</td>
<td>1</td>
<td>3</td>
<td>35584690</td>
</tr>
<tr>
<td>HZ_PARTIES_U1</td>
<td>03-07-2007</td>
<td>40168990</td>
<td>104890</td>
<td>40168990</td>
<td>1</td>
<td>1</td>
<td>26564210</td>
</tr>
<tr>
<td>HZ_PARTIES_U2</td>
<td>03-07-2007</td>
<td>38083790</td>
<td>122320</td>
<td>38083790</td>
<td>1</td>
<td>1</td>
<td>26754860</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Histogram Stats

<table>
<thead>
<tr>
<th>Schema</th>
<th>Table Name</th>
<th>Status</th>
<th>last analyzed</th>
<th>Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECEIVABLE</td>
<td>HZ_PARTIES</td>
<td>present</td>
<td>07-03-2007 06:09</td>
<td>PARTY_TYPE</td>
</tr>
</tbody>
</table>
### Tuning the Database Tier

- **Verifying the Statistics – Column Statistics**

```
APPS@GSIFIN12> exec fnd_stats.verify_stats('RECEIVABLE','HZ_PARTIES');
```

<table>
<thead>
<tr>
<th>Column</th>
<th>Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESS1</td>
<td>sample_size 1345914 num_distinct 1296029 num_nulls 25832100 density 0.00000771 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ADDRESS2</td>
<td>sample_size 369532 num_distinct 204489 num_nulls 35595920 density 0.0000489 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ADDRESS3</td>
<td>sample_size 58283 num_distinct 40694 num_nulls 38708410 density 0.00024573 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ADDRESS4</td>
<td>sample_size 17910 num_distinct 11528 num_nulls 39112140 density 0.000086745 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ANALYSIS_FY</td>
<td>sample_size 483 num_distinct 8 num_nulls 39286410 density 0.125 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>APPLICATION_ID</td>
<td>sample_size 1535656 num_distinct 21 num_nulls 23934680 density 0.047619047 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ATTRIBUTE1</td>
<td>sample_size 64617 num_distinct 15451 num_nulls 38645070 density 0.0006472 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ATTRIBUTE10</td>
<td>sample_size 60293 num_distinct 3884 num_nulls 38688310 density 0.00257466 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ATTRIBUTE11</td>
<td>sample_size 39706 num_distinct 4891 num_nulls 38894180 density 0.00204457 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ATTRIBUTE12</td>
<td>sample_size 69453 num_distinct 3108 num_nulls 38596710 density 0.0032175 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ATTRIBUTE13</td>
<td>sample_size 48736 num_distinct 2266 num_nulls 38803880 density 0.00441306 last analyzed 03-07-2007 06:09</td>
</tr>
<tr>
<td>ATTRIBUTE14</td>
<td>sample_size 37220 num_distinct 19753 num_nulls 38919040 density 0.00050625 last analyzed 03-07-2007 06:09</td>
</tr>
</tbody>
</table>
```
Tuning the Database Tier

- Convert to the OATM Tablespace Model for the E-Business Suite:
  - Consolidates the number of Apps tablespaces from 400 to ~12
  - Ideal for customers migrating to RAC or migrating to raw devices
  - Uniform extents, locally managed
    - No more fragmentation!
  - Migration script can be used to migrate a schema at a time
Tuning the Database Tier

- OATM Tablespace Model for the E-Business Suite:
  - Objects are classified by object type and access patterns:
    - Transaction tables
    - Transaction indexes
    - Reference/SEED
    - Interface
    - Summary Management
    - Archive
    - NoLogging
    - Queue
    - Media
    - Temporary
    - Undo
    - System

- Total = 12 Tablespaces
Tuning the Database Tier

• Applications Tablespace Migration Utility
  • MetaLink note 248857.1
  • ARU # is 3942506

• Migration timings: (complete migration)
  • 1.4 Terabyte customer DB
    • 24 hours
    • New size of DB after migration (850 GB).
  • 180 GB customer DB
    • 4 hours
    • New size of DB after migration (120 GB).
Tuning the Database Tier

• There are several optimizer related patches that need to be applied on top of 10.2.0.(2/3). Refer to note 244040.1 for an updated list
• Test, test, test…
### Tuning the Database Tier

- **Execution plans 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')); -- provided sql_id

PLAN_TABLE_OUTPUT

SQL_ID 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 PROFILE_OPTION_VALUE IS NOT NULL

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>0</td>
<td>0</td>
<td>3 (100)</td>
<td>00:00:01</td>
</tr>
<tr>
<td>1</td>
<td>INDEX RANGE SCAN</td>
<td>FND_PROFILEOPTIONVALUES_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

```
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>0:04:06:05</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>0:03:09:06</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
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
  - Release 12 RUP4: Patch 6435000
    - functionality, fixes, and updates for all Release 12 functional applications and the technology stack
  - Release 12 RUP5
    - 6836355 for Financials
    - 6610000 for HRMS
      - Techstack only version also available - (Patch 6077669)
Tuning the Applications

- **Workflow**
  - When starting 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 and facilitate asynchronous processing for flows such as Scheduling, PO Document Approval, etc.
  - Establish a regular frequency for purging Workflow runtime data (FNDWFPR - Purge Obsolete Workflow Runtime Data)
Tuning the Applications

- **Workflow**
  - Run the script ($FND_TOP/sql/wffngen.sql) to translate the activity function calls into static calls. This improves performance of Workflow processes by 25%.
  - **Disable retention on Workflow Queues**

```
SQL> select owner, name, queue_table, retention from dba_queues
    where owner='APPLSYS' and retention != 0;
SQL> exec
DBMS_AQADM.ALTER_QUEUE(queue_name=>:b1,retention_time=>0);
```
Tuning the Applications

- Order Management
  - 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
  - Use the *streamlined* workflows
    - Removes unnecessary activities and sub-processes
    - Improves performance and amount of WF data significantly
    - Refer to MetaLink note 130511.1
Tuning the Applications

• Order Management
  • For online order entry, use the Quick Sales Order Form
    • Provides a checkbox “deferred pricing” which can be used to defer price calculation until the order is saved
    • Also improves overall speed of order entry from a functional / navigational perspective
  • Enable AutoScheduling
  • Use deferred activities
    • Tax Calculation
    • Credit Check
Tuning the Applications

• 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

- Financials
  - 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**

- **Logging**
  - FND: Debug Log Enabled
    - Set it to ‘Yes’
  - FND: Debug Log Level
    - Must be 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

• Profiles
  • Set the frequently referenced profiles at the User level to avoid additional level checks. The check hierarchy is as follows:
    • Session
    • User
    • Organization
    • Server
    • Responsibility
    • Application
    • Site
  • Example: Set the profile “INV: Debug Trace” to “No” at the user level
Tuning the Applications

• 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” concurrent program type
Tuning the Applications

- Accessing the Purge Portal
  - System Administrator > Oracle Applications Manager > Purging/Critical Activities
### Tuning the Applications

**Applications Dashboard**

- Applications Dashboard: [Link](#)
- Critical Activities: [Link](#)
- Diagnostics: [Link](#)
- Business Rules: [Link](#)
- Security: [Link](#)
- Software Updates: [Link](#)

**Critical Activities**

- Last Updated: 25-10-2009 01:27:54
- Update Frequency: [Link](#)
- Modify Monitored Program List: [Link](#)

<table>
<thead>
<tr>
<th>Focus Program Name</th>
<th>Request ID</th>
<th>Last Run Date</th>
<th>Oracle Recommended Frequency</th>
<th>On Schedule (Oracle Recommended)</th>
<th>Overtime Frequency</th>
<th>On Schedule (Overtime)</th>
<th>Success Rate</th>
<th>Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Applications Manager Data</td>
<td>S230542</td>
<td>25-10-2006 19:00:32</td>
<td>✓</td>
<td>Oracle Recommended</td>
<td>✓</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purge Log and Clob System Data</td>
<td>S230542</td>
<td>25-10-2006 19:00:32</td>
<td>✓</td>
<td>Oracle Recommended</td>
<td>✓</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purge Oracle Workflow Runtime Data</td>
<td>S230542</td>
<td>25-10-2006 19:00:32</td>
<td>✓</td>
<td>Oracle Recommended</td>
<td>✓</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purge System Audit Data</td>
<td>S230542</td>
<td>25-10-2006 19:00:32</td>
<td>✓</td>
<td>Oracle Recommended</td>
<td>✓</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Oracle Applications Manager Data**

- [Link](#)
- [Link](#)
- [Link](#)

**Oracle Applications Manager**

- [Link](#)
- [Link](#)
- [Link](#)
Tuning the Applications

• System Administration
  • Oracle Applications Manager and the Grid Control Plug-in for Oracle E-Business Suite offer an integrated set of features for EBS system management.
  • Grid Control plug-in is also referred to as “Application Management Pack for Oracle E-Business Suite” and requires OEM version 10gR2
Tuning the Applications

- System Administration – Grid Control EBS Plug-in
  - Central monitoring & alerting
  - Cloning automation for EBS
  - Tight Integration with OAM
  - EBS specific topology views
  - Improved self discovery when using Autoconfig
    - Capturing EBS configuration, patching and setup details in the OEM repository
Tuning the Applications
Tuning the Applications

• System Administration – Applications Manager
  • Better RAC support - views by instance
  • Improved Alert support
    • Concurrent requests runtime/waits alerts
    • Alert ‘flood control’ feature
    • XML format support for alerts for integration mechanisms
  • Schedule diagnostics tests in batch & view results in dashboard
  • Enhanced patch impact analysis
    • Includes registered customized files
    • Group patches into set for analysis
  • Continued support for key administration features
    • CM, WF and Forms administration & monitoring
    • Configuration & Patch Management
Tuning the Applications
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)
High End Scalability
Database Tips for EBS Apps
DB Scalability for Apps

• Process has not changed
  • Test, measure, identify bottlenecks, take corrective action
  • Trace & AWR comparison of single vs. multiple thread run is key

• Tools & features do constantly change and improve
  • AWR instance and sql reports, ASH, SQL Trace better with each release
  • OEM DB & App Control, plus diagnostic packs
  • Real Application Testing works with 11g & 10gR1-R2/9iR2
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 with. 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.
  • Do not use parallel query/dml across 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
# 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>
QUESTIONS
ANSWERS
JVM Response Time/CPU Issues

R12 httpd.conf example: (Edit the LogFormat referred in the CustomLog directive)

LogFormat "%h %l %u %t "%r" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined

LogFormat "%h %T %l %u %t "%r" %>s %b" common

LogFormat "%{Referer}i -> %U" referer

LogFormat "%{User-agent}i" agent

Add the %T option

# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here. Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#

CustomLog "|/d2/apps/r12xb19/apps/tech_st/10.1.3/Apache/Apache/bin/rotatelogs
/d2/apps/r12xb19/inst/apps/r12xb19_ap610perf/logs/ora/10.1.3/Apache/access_log 86400" common
Response Time/CPU Issues

• Access log output example:

148.87.19.51 1 - - [01/May/2007:11:38:49 -0700] "GET /OA_HTML/AppsLocalLogin.jsp HTTP/1.1" 200 5986
148.87.19.51 0 - - [01/May/2007:11:38:53 -0700] "POST /OA_HTML/fndvald.jsp HTTP/1.1" 302 259
148.87.19.51 5 - - [01/May/2007:11:38:58 -0700] "GET /OA_HTML/OA.jsp?OAFunc=OAHOMEPAGE HTTP/1.1" 200 41609
148.87.19.51 5 - - [01/May/2007:11:39:15 -0700] "GET /OA_HTML/OA.jsp?OAFunc=OAHOMEPAGE&akRegionApplicationId=0&navResp
Id=20420&navRespAppId=1&navSecGrpId=0&transactionid=36910577&oapc =2 HTTP/1.1" 200 94506

• The second field is the response time in second (since %T was added as the second field in the previous examples)

• Note if the entry is written when the request is complete, so if it is hanging, you will not find an entry
Thread Dumps

- A Thread Dump shows the stack of each thread in the JVM
- A series of thread dumps taken in succession could be used sometimes to identify the expensive methods.
- A thread dump also shows if a thread is blocked, and if there are deadlocks
- To get thread dumps on UNIX, issue `kill -QUIT <jvm pid>`
- For Sun JDK 1.5 or above, you can also try `jstack <pid>`. jstack will run in a separate process and attaches to the target JVM to get the stack
- With HP, you can use gdb to get java stack as well. See HP’s JVM diagnostics guide for details
Thread Dumps

• For Sun and HP, the thread dump will be written to the stdout file
  • For 11.5:
    • $IAS_ORACLE_HOME/Apache/Jserv/logs/jvm/OACoreGroup.*.stdout
  • For R12:
    • $LOG_HOME/ora/10.1.3/opmn/OC4J~oacore~default_group_*
• For IBM, by default, each thread dump is written to a different file, the file location is listed in the stdout file shown above
  JVMDUMP006I Processing Dump Event "user", detail "" - Please Wait.
  JVMDUMP007I JVM Requesting Java Dump using
  '/d2/apps/r12xb19/apps/tech_st/10.1.3/j2ee/home/javacore.20070925.180320.20709.txt

  JVMDUMP010I Java Dump written to
  /d2/apps/r12xb19/apps/tech_st/10.1.3/j2ee/home/javacore.20070925.180320.20709.txt
  JVMDUMP013I Processed Dump Event "user", detail "".

• To find process id for the oacore JVM, look for the property
  CLIENT_PROCESSID in the “About” page
Thread Dumps

• Once a user is logged in, his/her requests will be served by the same JVM

• The “About” link is at the bottom of each page (you need to have FND_DIAGNOSTICS profile set to Y)

Process ID of the JVM handling your session
Thread Dumps

Full thread dump:

"Thread-32361" daemon prio=1 tid=0x90787f40 nid=0x542e runnable [0x91e4f000..0x91e5086c]
at java.net.SocketInputStream.socketRead(Native Method)
at java.net.SocketInputStream.read(SocketInputStream.java:85)
...
"Thread-27325" daemon prio=1 tid=0x87cfa08 nid=0x4313 waiting on monitor [0x90dff000..0x90dff86c]
at java.lang.Thread.sleep(Native Method)
at oracle.apps.jtf.base.session.Monitor.run(Monitor.java:50)
...
"Thread-12" daemon prio=1 tid=0x8099058 nid=0x595c waiting on monitor [0x8f1ff000..0x8f1ff86c]
at java.lang.Object.wait(Native Method)
at oracle.apps.fnd.common.Pool.run(Pool.java:1873)
at java.lang.Thread.run(Thread.java:479)
...
Thread Dumps

Thread dump showing a live lock wait:

"Thread-167473" daemon prio=10 tid=000b8f70 nid=171653 lwp_id=7886850
   runnable [0x37eff000..0x37efe4f0]
       at java.net.SocketInputStream.socketRead0(Native Method)
       at java.net.SocketInputStream.read(SocketInputStream.java:134)
   . . .
       at oracle.jdbc.driver.OraclePreparedStatement.executeUpdate(OraclePreparedStatement.java:656)
       - locked <46f6ca40> (a oracle.jdbc.driver.OracleCallableStatement)
       - locked <46d64270> (a oracle.jdbc.driver.OracleConnection)
   . . .
"Thread-1" daemon prio=10 tid=000b6000 nid=14 lwp_id=2015294 waiting for
   monitor entry [0x39a77000..0x39a764f0]
       at oracle.jdbc.driver.OracleConnection.isClosed(OracleConnection.java:1554)
       - waiting to lock <46d64270> (a oracle.jdbc.driver.OracleConnection)
       at oracle.apps.fnd.common.AppsContext.getJDBCConnection(AppsContext.java:2260)
   . . .
Thread Dumps

- IBM’s thread dump output
eProf for HP JVM

- HP provides eprof which can be used to capture execution profile data for the JVM
- eProf options can be found at http://www.hp.com/products1/unix/java/infolibrary/prog_guide/xeprof.html?jumpid=reg_R1002_USEN
- The eprof output can be opened in HPJmeter
- HPJmeter has views to see top methods by elapsed time, CPU or call count, as well as tree views
eProf for HP JVM – Call Tree View
eProf for HP JVM – Method View

```
Method Call Count (partial data)

475,725 oracle.adf.mds.JSPMDSContextImpl.getCustomizationContext()
240,680 java.lang.String.equals(java.lang.Object)
232,008 oracle.adf.mds.JSPMDSContextImpl.getLanguage()
224,167 oracle.adf.mds.CustContextImpl.getKey()
224,095 oracle.adf.mds.CustContextImpl.getUserKey()
224,080 oracle.adf.mds.internal.SnAPSHOT.getSNAPSHOT(oracle.adf.mds.MDSContext,oracle.adf.mds.internal)
215,984 oracle.cabo.u.AttributeKey.equals(java.lang.Object)
212,919 oracle.adf.mds.JSPMDSContextImpl.getMetadataManager()
166,992 oracle.adf.mds.internal.MElementImpl.getAttributeExpression(oracle.adf.mds.MDSContext,oracle.adf.mds.internal)
166,992 oracle.adf.mds.internal.MetadataManagerBase.getAttributeValueInternal(oracle.adf.mds.MDSContext,oracle.adf.mds.internal)
166,597 oracle.adf.mds.internal.SnAPSHOT.getSNAPSHOTRaw(oracle.adf.mds.MDSContext,oracle.adf.mds.internal)
144,322 java.lang.String.hashCode()
```
Monitor JVM Garbage Collection

- Use `-verbose:gc` option to monitor JVM garbage collection and heap usage
  - For Sun’s JVM:
    - Output goes to stdout
    - Adding `-XX:+PrintGCTimeStamps` option will add a timestamp when garbage collection occurs.
  - For IBM’s JVM:
    - Output goes to stderr by default
    - Use `-Xverbosegclog:/path/to/file_name.gc`
    - Timestamp is automatic
    - Additional detail use `-Xtgc:*`
  - For HP’s JVM:
    - Output goes to stdout by default
    - `-Xverbosegc:file=/path/to/file_name.gc`
    - Timestamp is automatic
    - The HP-only `-Xverbosegc` option generates additional GC information that is used by HPJMeter. It is preferable to use `-Xverbosegc` instead of `-verbose:gc`. 
Monitor JVM Garbage Collection

- Location of GC logs
  - For 11.5:
    - $IAS_ORACLE_HOME/Apache/Jserv/logs/jvm/OACoreGroup.*.stdout
  - For R12:
    - $LOG_HOME/ora/10.1.3/opmn/OC4J~oacore~default_group_*
- For JDK 1.5.0_08 or above, GC timestamp is relative to JVM start
- Prior to JDK 1.5.0_08, GC timestamp is relative to first GC (ie, first GC will always have timestamp 0.000)
- If heap usage is steadily increasing, then there may be a memory leak
Monitor JVM Garbage Collection

• Sample output for Sun’s JVM:

```
0.000: [GC 143357K->34512K(514048K), 1.5519252 secs]
103.925: [GC 177872K->44238K(514048K), 0.7564096 secs]
124.894: [GC 187598K->51968K(514048K), 0.5778231 secs]
687.205: [Full GC 366626K->230896K(514048K), 7.3117923 secs]
```

Minor GC is usually fast

Full GC is much slower

Heap Usage before GC

Heap Usage after GC

Current Heap Capacity

Time taken for GC

GC begin time
Monitor JVM Garbage Collection

- You can use `–XX:+PrintGCDetails` to get more information about the usage of each generation at GC time:

  6.037: [Full GC [PSYoungGen: 1718K→0K(12480K)] [PSOldGen: 3488K→5178K(113856K)] 5207K→5178K(126336K) [PSPermGen: 10241K→10241K(20736K)], 0.0691040 secs]

  9.187: [GC [PSYoungGen: 423K→80K(12480K)] 5602K→5258K(126336K), 0.0011820 secs]

  9.188: [Full GC [PSYoungGen: 80K→0K(12480K)] [PSOldGen: 5178K→5179K(113856K)] 5258K→5179K(126336K) [PSPermGen: 10242K→10242K(23296K)], 0.0629940 secs]

- You can also use `jstat` (from JDK 1.5) to monitor GC and heap size information

- For example, this gets the heap and GC info for pid 5539 every 5s:

  ```
  (sto) tmp- /local/java/jdk1.5.0/bin/jstat -gc 5539 5s
  S0C   S1C   S0U   S1U   EC   EU       OC      OU       PC
  PU    YGC    YGCT    FGC    FGCT    GCT
  1728.0 1728.0  0.0   0.0 10752.0 160.2 113856.0  5076.7  20224.0
  10248.5  6   0.028   5   0.338   0.366
  1728.0 1728.0  0.0   0.0 10752.0 243.3 113856.0  5076.7  20224.0
  10250.6  6   0.028   5   0.338   0.366
  ```
Monitor JVM Garbage Collection

- Sample output from IBM’s JVM

```xml
<?xml version="1.0" ?>
<verbosegc version="200704_19">
  <af type="tenured" id="1" timestamp="Mon Sep 10 17:39:38 2007" intervalms="0.000">
    <minimum requested_bytes="1024000" />
    <time exclusiveaccessms="0.013" />
    <tenured freebytes="509288" totalbytes="4194304" percent="12" >
      <soa freebytes="299880" totalbytes="3984896" percent="7" />
      <loa freebytes="209408" totalbytes="209408" percent="100" />
    </tenured>
  </af>
  <gc type="global" id="1" totalid="1" intervalms="0.000">
    <refs_cleared soft="0" threshold="32" weak="0" phantom="0" />
    <finalization objectsqueued="1" />
    <timesms mark="1.517" sweep="0.132" compact="0.000" total="1.687" />
    <tenured freebytes="3782776" totalbytes="4194304" percent="90" >
      <soa freebytes="3531384" totalbytes="3942912" percent="89" />
      <loa freebytes="251392" totalbytes="251392" percent="100" />
    </tenured>
  </gc>
  <tenured freebytes="2758776" totalbytes="4194304" percent="65" >
    <soa freebytes="2507384" totalbytes="3942912" percent="63" />
    <loa freebytes="251392" totalbytes="251392" percent="100" />
  </tenured>
  <time totalms="1.846" />
</af>
```
Monitor JVM Garbage Collection

- Sample output from HP’s JVM
- This can be read by JMeter

```
<GC: 1 0 38.823276 1 8208 31 191168512 8208 191168512 0 16681128 47710208 0 0 573243392
   21605544 21605544 104857600 0.302841 0.302841 >
<GC: 1 0 711.377044 2 24 2 191168512 24 191168512 16681128 29504560 47710208 0 0 573243392
   26548456 26548456 104857600 0.401796 0.401796 >
<GC: 1 0 1025.698704 3 6016 31 191168512 6016 191168512 29504560 22986176 47710208 0
   15723792 573243392 30380528 30380528 104857600 0.544584 0.544584 >
<GC: 1 0 1534.280575 4 2097168 2 190055472 2097168 191168512 22986176 32373648 47710208
   15723792 15723792 573243392 40264616 40264616 104857600 0.403723 0.403723 >
<GC: 1 0 2589.000505 5 736 31 191168512 736 191168512 32373648 23695240 47710208 15723792
   30169616 573243392 40526768 40526768 104857600 0.408634 0.408634 >
<GC: 1 0 3063.722734 6 24 3 191168512 24 191168512 23695240 30431752 47710208 30169616
   30169616 573243392 41117920 41117920 104857600 0.283276 0.283276 >
<GC: 1 0 5986.510420 7 48 31 191168512 48 191168512 30431752 16796232 47710208 30169616
   39440232 573243392 41358304 41358304 104857600 0.251891 0.251891 >
```