Achievements since last review
openlab projects in the context of the service evolution
Highlights 2007 Q1
The process of designing/tuning the applications for RAC scalability is not easy and straightforward.

CERN developers will need recommendations and guidance in order to produce RAC scalable software.

Objectives of the programme:

- To examine real CERN cases and to study RAC scalability
- To design and develop general techniques and recommendations to improve RAC scalability
Shared disk infrastructure, all disk devices accessed from all servers

Disks

Database servers

Clients (interactive, batch)
Example of performance gain with RAC

- Commercial control application (critical for LHC and experiments)
- Archiving in Oracle
- Application without modifications: 100 “changes” per second
- CERN needs: 150,000 changes per second (x 1500)

- Iterative process, based on Oracle’s “wait interface”
- Structure = table EVENTS_HISTORY (ELEMENT_ID, VALUE…)
- Each client “measures” input and registers history with a “merge” operation in the table EVENTS_HISTORY
- 100 entries per second
- Initial state observation: database is waiting on the clients “SQL*Net message from client”
- Use of a generic library C++/DB
- Individual insert (one statement per entry)
- Update of a table which keeps “latest state” through a trigger
Changes: bulk insert to a temporary table with OCCl, then call PL/SQL to load data into history table

- From 100 to 2000 changes per second
- [awrrpt_1_5489_5490.html](awrrpt_1_5489_5490.html)
- Now top event: “db file sequential read”

<table>
<thead>
<tr>
<th>Event</th>
<th>Waits</th>
<th>Time(s)</th>
<th>Percent Total DB Time</th>
<th>Wait Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>db file sequential read</td>
<td>29,242</td>
<td>137</td>
<td>42.56</td>
<td>User I/O</td>
</tr>
<tr>
<td>enq: TX - contention</td>
<td>41</td>
<td>120</td>
<td>37.22</td>
<td>Other</td>
</tr>
<tr>
<td>CPU time</td>
<td>61</td>
<td></td>
<td>18.88</td>
<td></td>
</tr>
<tr>
<td>log file parallel write</td>
<td>1,133</td>
<td>19</td>
<td>5.81</td>
<td>System I/O</td>
</tr>
<tr>
<td>db file parallel write</td>
<td>3,951</td>
<td>12</td>
<td>3.73</td>
<td>System I/O</td>
</tr>
</tbody>
</table>
- Changes: index usage analysis and reduction, table structure changes. IOT. Replacement of merge by insert. Use of “direct path load” with ETL.
- Improvement: from 2000 changes per second to 16,000 changes per second.
- Now top event: cluster related wait event
  test5_rac_node1_8709_8710.html

<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>gc buffer busy</td>
<td>27,883</td>
<td>728</td>
<td>26</td>
<td>31.6</td>
<td>Cluster</td>
</tr>
<tr>
<td>CPU time</td>
<td>369</td>
<td></td>
<td></td>
<td>16.0</td>
<td></td>
</tr>
<tr>
<td>gc current block busy</td>
<td>6,818</td>
<td>255</td>
<td>37</td>
<td>11.1</td>
<td>Cluster</td>
</tr>
<tr>
<td>gc current grant busy</td>
<td>24,370</td>
<td>228</td>
<td>9</td>
<td>9.9</td>
<td>Cluster</td>
</tr>
<tr>
<td>gc current block 2-way</td>
<td>118,454</td>
<td>198</td>
<td>2</td>
<td>8.6</td>
<td>Cluster</td>
</tr>
</tbody>
</table>
- Changes: each “client” receives a unique number. Partitioned table. Use of “direct path load” to the partition with Extracting, Transforming and Loading
- Improvement: from 16 000 changes per second to 150 000 changes per second
- Now top event: “freezes” once upon a while

<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>row cache lock</td>
<td>813</td>
<td>665</td>
<td>818</td>
<td>27.6</td>
<td>Concurrency</td>
</tr>
<tr>
<td>gc current multi block request</td>
<td>7,218</td>
<td>155</td>
<td>22</td>
<td>6.4</td>
<td>Cluster</td>
</tr>
<tr>
<td>CPU time</td>
<td></td>
<td>123</td>
<td></td>
<td>5.1</td>
<td></td>
</tr>
<tr>
<td>log file parallel write</td>
<td>1,542</td>
<td>109</td>
<td>71</td>
<td>4.5</td>
<td>System I/O</td>
</tr>
<tr>
<td>undo segment extension</td>
<td>785,439</td>
<td>88</td>
<td>0</td>
<td>3.6</td>
<td>Configuration</td>
</tr>
</tbody>
</table>
Identification: ASM space allocation is blocking some operations

Changes: space pre-allocation, background task.

Allows to keep steady rate

Conclusion: from 100 changes per second to 150 000 changes per second

6 nodes RAC (dual CPU, 4GB RAM), 32 disks SATA with FCP link to host

4 months, re-writing of part of the application with changes interface (C++ code), changes of the database code (PL/SQL), schema change, numerous work sessions

Initial help of an Oracle consultant
Application RAC / summary

1. `insert into events_history values...`
2. `... insert into ... values (...)`
3. `bulk insert into temp table`
   - PL/SQL: `insert /*+ APPEND */ into eventh (...) partition PARTITION (1) select ... from temp...`
Objectives of the programme:

- To test following Data Guard solutions on RDBMS 10gR2:
  - Automatic Failover
  - Inter patchset SQL Apply

- To test Data Guard Automatic Failover mechanism with focus on:
  - Data size
  - Time to switch

- To deploy in production Data Guard Automatic Failover mechanism on selected CERN service, in order to reduce downtime implied by major software / hardware issues and upgrades
DataGuard automatic failover is now well understood, interaction with Oracle helped to identify how to set connect time failover / and use DB_ROLE_CHANGE event

Will be implemented in production as a core building block for database servers in 2007H1
Enterprise Manager

- New openlab subject
- Managing a large number of Oracle targets coupled with a comparatively small number of personnel
- January 2007 achievements:
  - Global report of all CERN Oracle installation which require installation of the latest CPU patch
  - Report of databases for which one or more datafiles have not been backed up in the past x days
Service evolution in the openlab context

- We have started the move of administrative applications to “AIS RAC”, applications migration started December 2006
- 10gR2 with 4 nodes (4 CPUs, 16GB per node)
- 64 bits Linux (RedHat Enterprise Linux 4 x86-64).
- No issue with RHEL4 64bits / Oracle 64 bits Linux
New hardware

- CASTOR (CERN mass storage system) uses Oracle database as central architectural component
- Multi-core systems (Woodcrest) in pre-production
- Relatively high-performance Network Attached Storages in pre-production
2007 Q1-Q2 plans

- Oracle Enterprise Manager (CERN coordinator: Chris Lambert)
- Oracle DataGuard (CERN coordinator: Anton Topurov)
- Application Design, Development and Scalability on Oracle RAC (CERN coordinator: Anton Topurov)