Real Life RAC Performance Tuning

Arup Nanda
Lead DBA
Starwood Hotels
Who am I

- Oracle DBA for 13 years and counting
- Working on OPS from 1999
- Implemented and supported 10g RAC in 83 sites since 2004
- Speak at conferences, write papers, books
Why This Session

- I get emails like this:
  - *We are facing performance issues in RAC. What should I do next?*

- Real Life Advice
  - Common Issues (with Wait Events)
  - Dispelling Myths
  - Formulate a Plan of Attack
  - Real Life Case Study

- proligence.com/downloads.html
Our RAC Implementation

- **Oracle 10g RAC in March 2004**
  - Itanium Platform running HP/UX
  - Oracle 10.1.0.2
- Result: Failed ☹
- **Second Attempt: Dec 2004 – 10.1.0.3**
- Result: Failed Again ☹
- **Third Attempt: March 2005 – 10.1.0.4**
- Result: Success! 😊
Challenges

- **Technology**
  - Lone ranger
  - A lot of “mystery” and disconnected “facts”!

- **People**
  - Building a team that could not only deliver; but also sustain the delivered parts
  - Each day we learned something new

- In today’s session: real performance issues we faced and how we resolved them, along with wait events.
Why “RAC” Performance?

- All tuning concepts in single instance applied to RAC as well
- RAC has other complexities
  - More than 1 buffer cache
  - Multiple caches – library cache, row cache
  - Interconnect
  - Pinging
  - Global Locking
Why RAC Perf Tuning

- We want to make sure we identify the right problem and go after it
- ..... not just a problem
Areas of Concern in RAC

- More than 1 buffer cache
- Multiple caches – library cache, row cache
- Interconnect
- Global Locking
Cache Issues

- Two Caches, requires synchronization
- What that means:
  - A changed block in one instance, when requested by another, should be sent across via a “bridge”
  - This bridge is the Interconnect
Interconnect Performance

- Interconnect must be on a private LAN
- Port aggregation to increase throughput
  - APA on HPUX
- If using Gigabit over Ethernet, use Jumbo Frames
Checking Interconnect Used

- **Identify the interconnect used**
  
  ```
  $ oifcfg getif
  lan902  172.17.1.0  global
  cluster_interconnect
  lan901  10.28.188.0  global  public
  ```

- **Is lan902 the bonded interface? If not, then set it**
  
  ```
  $ oifcfg setif ...
  ```
Pop Quiz

- If I have a very fast interconnect, I can perform the same work in multiple node RAC as a single server with faster CPUs. True/False?

- Since cache fusion is now write-write, a fast interconnect will compensate for a slower IO subsystem. True/False?
Cache Coherence Times

- The time is a sum of time for:
  - Finding the block in the cache
  - Identifying the master
  - Get the block in the interconnect
  - Transfer speed of the interconnect
  - Latency of the interconnect
  - Receive the block by the remote instance
  - Create the consistent image for the user
So it all boils down to:

- **Block Access Cost**
  - more blocks -> more the time
  - Parallel Query

- **Lock Management Cost**
  - More coordination -> more time
  - Implicit Cache Checks – Sequence Numbers

- **Interconnect Cost**
  - Latency
  - Speed
  - more data to transfer -> more the time
Hard Lessons

- In RAC, problem symptoms may not indicate the correct problem!
- Example:
  - When the CPU is too busy to receive or send packets via UDP, the packets fail and the Clusterware thinks the node is down and evicts it.
OS Troubleshooting

- **OS utilities to troubleshoot CPU issues**
  - top
  - glance

- **OS Utilities to troubleshoot process issues:**
  - truss
  - strace
  - dbx
  - pstack
Reducing Latency

- A factor of technology
- TCP is the most latent
- UDP is better (over Ethernet)
- Proprietary protocols are usually better
  - HyperFabric by HP
  - Reliable Datagram (RDP)
  - Direct Memory Channel
- Infiniband
  - UDP over Infiniband
  - RDP over Infiniband
Start with AWR

Top 5 Timed Events

<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>3,754,273</td>
<td>30,966</td>
<td>70.67</td>
<td>User I/O</td>
</tr>
<tr>
<td>CPU time</td>
<td></td>
<td>8,320</td>
<td>18.99</td>
<td></td>
</tr>
<tr>
<td>db file parallel read</td>
<td>64,468</td>
<td>1,456</td>
<td>3.32</td>
<td>User I/O</td>
</tr>
<tr>
<td>gc cr grant 2-way</td>
<td>1,470,759</td>
<td>984</td>
<td>2.25</td>
<td>Cluster</td>
</tr>
<tr>
<td>read by other session</td>
<td>79,807</td>
<td>486</td>
<td>1.11</td>
<td>User I/O</td>
</tr>
</tbody>
</table>
gc current|cr grant 2-way

Instance 1

Session

LMS

Instance 2

Log Buffer

LGWR

Database

© Arup Nanda
gc current|cr block 2-way

Instance 1

Session

LMS

Instance 2

Log Buffer

LGWR

Database

log file sync

gc current block 2-way
gc current|cr block 3-way

- Instance 1: Requestor
- Instance 2: Master
- Instance 3: Blocker

Session
RAC related Stats

RAC Statistics

<table>
<thead>
<tr>
<th>Number of Instances:</th>
<th>Begin</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

Global Cache Load Profile

<table>
<thead>
<tr>
<th></th>
<th>Per Second</th>
<th>Per Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Cache blocks received:</td>
<td>293.67</td>
<td>11.19</td>
</tr>
<tr>
<td>Global Cache blocks served:</td>
<td>271.61</td>
<td>10.35</td>
</tr>
<tr>
<td>GCS/GES messages received:</td>
<td>2,655.12</td>
<td>101.17</td>
</tr>
<tr>
<td>GCS/GES messages sent:</td>
<td>2,515.61</td>
<td>95.86</td>
</tr>
<tr>
<td>DBWR Fusion writes:</td>
<td>11.10</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Global Cache Efficiency Percentages (Target local+remote 100%)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer access - local cache %:</td>
<td>95.89</td>
</tr>
<tr>
<td>Buffer access - remote cache %:</td>
<td>0.73</td>
</tr>
<tr>
<td>Buffer access - disk %:</td>
<td>3.38</td>
</tr>
</tbody>
</table>
RAC Stats contd.

Global Cache and Enqueue Services - Workload Characteristics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg global enqueue get time (ms)</td>
<td>0.3</td>
</tr>
<tr>
<td>Avg global cache cr block receive time (ms)</td>
<td>1.1</td>
</tr>
<tr>
<td>Avg global cache current block receive time (ms)</td>
<td>1.3</td>
</tr>
<tr>
<td>Avg global cache cr block build time (ms)</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg global cache cr block send time (ms)</td>
<td>0.1</td>
</tr>
<tr>
<td>Global cache log flushes for cr blocks served %</td>
<td>1.5</td>
</tr>
<tr>
<td>Avg global cache cr block flush time (ms)</td>
<td>3.6</td>
</tr>
<tr>
<td>Avg global cache current block pin time (ms)</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg global cache current block send time (ms)</td>
<td>0.1</td>
</tr>
<tr>
<td>Global cache log flushes for current blocks served %</td>
<td>0.1</td>
</tr>
<tr>
<td>Avg global cache current block flush time (ms)</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Global Cache and Enqueue Services - Messaging Statistics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg message sent queue time (ms)</td>
<td>0.1</td>
</tr>
<tr>
<td>Avg message sent queue time on ksxp (ms)</td>
<td>0.6</td>
</tr>
<tr>
<td>Avg message received queue time (ms)</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg GCS message process time (ms)</td>
<td>0.0</td>
</tr>
<tr>
<td>Avg GES message process time (ms)</td>
<td>0.0</td>
</tr>
<tr>
<td>% of direct sent messages</td>
<td>48.38</td>
</tr>
<tr>
<td>% of indirect sent messages</td>
<td>49.81</td>
</tr>
<tr>
<td>% of flow controlled messages</td>
<td>1.81</td>
</tr>
<tr>
<td>Event</td>
<td>Waits</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>db file sequential read</td>
<td>3,754,273</td>
</tr>
<tr>
<td>db file parallel read</td>
<td>64,468</td>
</tr>
<tr>
<td>gc cr grant 2-way</td>
<td>1,470,759</td>
</tr>
<tr>
<td>read by other session</td>
<td>79,807</td>
</tr>
<tr>
<td>db file parallel write</td>
<td>216,065</td>
</tr>
<tr>
<td>gc buffer busy</td>
<td>87,088</td>
</tr>
<tr>
<td>enq: US - contention</td>
<td>1,280,682</td>
</tr>
<tr>
<td>gc current block 2-way</td>
<td>432,631</td>
</tr>
<tr>
<td>RFS dispatch</td>
<td>8,192</td>
</tr>
<tr>
<td>RFS write</td>
<td>8,192</td>
</tr>
<tr>
<td>gc cr block 2-way</td>
<td>392,748</td>
</tr>
<tr>
<td>gc current grant 2-way</td>
<td>418,405</td>
</tr>
<tr>
<td>db file scattered read</td>
<td>49,224</td>
</tr>
<tr>
<td>log file sync</td>
<td>85,247</td>
</tr>
<tr>
<td>SQL*Net more data from client</td>
<td>69,137</td>
</tr>
<tr>
<td>log file sequential read</td>
<td>8,199</td>
</tr>
<tr>
<td>log file parallel write</td>
<td>179,193</td>
</tr>
<tr>
<td>PX Deq Credit; send bld</td>
<td>9,213</td>
</tr>
<tr>
<td>gc cr multi block request</td>
<td>261,317</td>
</tr>
<tr>
<td>DFS lock handle</td>
<td>245,617</td>
</tr>
<tr>
<td>gc cr block busy</td>
<td>15,005</td>
</tr>
<tr>
<td>control file sequential read</td>
<td>61,821</td>
</tr>
<tr>
<td>control file parallel write</td>
<td>27,898</td>
</tr>
<tr>
<td>latch: KCL gc element parent latch</td>
<td>2,660</td>
</tr>
<tr>
<td>enq: TX - index contention</td>
<td>2,057</td>
</tr>
<tr>
<td>gc current grant busy</td>
<td>39,048</td>
</tr>
<tr>
<td>latch: cache buffers chains</td>
<td>2,486</td>
</tr>
<tr>
<td>row cache lock</td>
<td>12,780</td>
</tr>
<tr>
<td>gc current block busy</td>
<td>1,137</td>
</tr>
<tr>
<td>ges inquiry response</td>
<td>19,650</td>
</tr>
<tr>
<td>gc cr grant congested</td>
<td>1,814</td>
</tr>
<tr>
<td>latch: cache buffers lru chain</td>
<td>1,275</td>
</tr>
<tr>
<td>CGS wait for IPC msg</td>
<td>467,085</td>
</tr>
<tr>
<td>gcs log flush sync</td>
<td>7,317</td>
</tr>
</tbody>
</table>
Other GC Block Waits

- gc current/cr block lost
  - Lost blocks due to Interconnect or CPU
- gc current/cr block busy
  - The consistent read request was delayed, most likely an I/O bottleneck
- gc current/cr block congested
  - Long run queues and/or paging due to memory deficiency.
Hung or Slow?

- **Check** `V$SESSION` for `WAIT_TIME`
  - If 0, then it’s not waiting; it’s hung
- **When hung:**
  - Take a systemstate dump from all nodes
  - Wait some time
  - Take another systemstate dump
  - Check change in values. If unchanged, then system is hung
Chart a Plan

- Rule out the obvious
- Start with AWR Report
- Start with Top-5 Waits
- See if they have any significant waits
- ... especially RAC related
- Go on to RAC Statistics
- Base your solution based on the wait event
Rule out the obvious

- Is interconnect private?
- Is interconnect on UDP?
- Do you see high CPU?
- Do you see a lot of IO bottleneck?
- How about memory?
- Are the apps spread over evenly?
- Do you see lost blocks?
Make Simple Fixes

- Strongly consider RAID 0+1
- Highest possible number of I/O paths
- Use fastest interconnect possible
- Use private collision free domain for I/C
- Cache and NOORDER sequences
The interconnect configuration and inter-node communication will influence the performance of cluster databases. The tables below show network interfaces on all hosts and network interfaces currently in use by cluster databases. It is important that cluster databases are configured to use a private interconnect for message and block transfers.

Private Interconnect Transfer Rate (MB/Sec): 0.419
Transfer rate on the private network in the last 5 minutes.

### Interfaces by Hosts

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Subnet</th>
<th>Interface Type</th>
<th>Total I/O Rate (MB/Sec) (Last 5 Minutes)</th>
<th>Total Error Rate (%) (Last 5 Minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ndsstg</td>
<td>Cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hndssdb1</td>
<td>Host</td>
<td>172.17.1.0</td>
<td>Private</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>lan902</td>
<td>Interface</td>
<td>172.17.1.0</td>
<td>Private</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>hndssdb2</td>
<td>Host</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lan902</td>
<td>Interface</td>
<td>172.17.1.0</td>
<td>Private</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>hndssdb3</td>
<td>Host</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Interfaces in Use by Cluster Databases

© Arup Nanda
Buffer Busy

**Cause**
- Instance wants to bring something from disk to the buffer cache
- Delay, due to space not available
- Delay, b’coz the source buffer is not ready
- Delay, I/O is slow
- Delay, b’coz redo log is being flushed

**In summary**
- Log buffer flush -> gc buffer busy
One major issue in RAC is parallel query that goes across many nodes.
Restricting PQ

- **Define Instance Groups**

  Specify in `init.ora`

  - `prodb1.instance_groups='pqgroup1'`
  - `prodb2.instance_groups='pqgroup2'`

- **Specify Instance Groups in Session**

  SQL> alter session set parallel_instance_group = 'pqgroup1';
Forcing PQ on both Nodes

- Define a common Instance Group
  ```
  prodb1.instance_groups='pqgroup1'
  prodb1.instance_groups='pq2nodes'
  prodb2.instance_groups='pqgroup2'
  prodb2.instance_groups='pq2nodes'
  ```

- Specify Instance Groups in Session
  ```
  SQL> alter session set parallel_instance_group = 'pq2nodes';
  ```
Vital Cache Fusion Views

- **gv$cache_transfer**: Monitor blocks transferred by object
- **gv$class_cache_transfer**: Monitor block transfer by class
- **gv$file_cache_transfer**: Monitor the blocks transferred per file
- **gv$temp_cache_transfer**: Monitor the transfer of temporary tablespace blocks
“Hot” Tables

- Tables, e.g. Rate Plans
  - Small
  - Compact blocks
  - High updates
  - High reads

- Symptoms
  - gc buffer busy waits

- Solution
  - Less rows per block
  - High PCTFREE, INITRANS,
  - ALTER TABLE ... MINIMIZE RECORDS_PER_BLOCK
Hot Sequences

- **Symptoms:**
  - High waits on Sequence Number latch
  - High waits on SEQ$ table

- **Solution:**
  - Increase the cache
  - Make it NOORDER

- Especially AUDSESS$ sequence in SYS, used in Auditing
Read Only? Say So.

- Reading table data from other instances
  create “gc *” contentions
- Suggestion:
  - Move Read Only tables to a single tablespace
  - Make this tablespace Read Only

```
SQL> alter tablespace ROD read only;
```
Partitioning

- Partitioning creates several segments for the same table (or index)
- => more resources
- => less contention
Monotonically Increasing Index

- **Problem:**
  - “Reservation ID”, a sequence generated key
  - Index is heavy on one side

- **Symptoms**
  - Buffer busy waits
  - Index block splits

- **Solutions:**
  - Reverse key indexes
  - Hash partitioned index (even if the table is not partitioned) 10gR2
In RAC, Library Cache is global
So, parsing cost is worse than non-RAC
Solutions:
- Minimize table alters, drops, creates, truncates
- Use PL/SQL stored programs, not unnamed blocks
Log Files

- In 10g R2, the log files are in a single location:
- `$CRS_HOME/log/<Host>/...
  - racg
  - crsd
  - cssd
  - evmd
  - client
  - cssd/oclsmone`
- `$ORACLE_HOME/racg/dump`
Case Study

Top 5 Timed Events

<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>4,137,096</td>
<td>50,323</td>
<td>24.47</td>
<td>User I/O</td>
</tr>
<tr>
<td>gc domain validation</td>
<td>16,456</td>
<td>30,784</td>
<td>14.97</td>
<td>Cluster</td>
</tr>
<tr>
<td>gc buffer busy</td>
<td>148,267</td>
<td>26,707</td>
<td>12.99</td>
<td>Cluster</td>
</tr>
<tr>
<td>gc cr failure</td>
<td>18,799</td>
<td>22,914</td>
<td>11.14</td>
<td>Cluster</td>
</tr>
<tr>
<td>CPU time</td>
<td></td>
<td>17,609</td>
<td>8.56</td>
<td></td>
</tr>
</tbody>
</table>

© Arup Nanda
Diagnosis

- `ifconfig -a` shows no congestion or dropped packets
- Top shows 1% idle time on node 2
- Top processes
  - LMS and LMD
- And, several Netbackup processes
Further Diagnosis

- **SQL:**
  ```sql
  select * from v$instance_cache_transfer
  where class = 'data block'
  and instance = 1;
  ```

- **Output:**
  ```plaintext
<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>CLASS</th>
<th>CR_BLOCK</th>
<th>CR_BUSY</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------</td>
<td>-----------------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>CR_CONGESTED</td>
<td>CURRENT_BLOCK</td>
<td>CURRENT_BUSY</td>
<td>CURRENT_CONGESTED</td>
</tr>
<tr>
<td>1 data block</td>
<td>162478682</td>
<td>5097149</td>
<td></td>
</tr>
<tr>
<td>477721</td>
<td>347917908</td>
<td>2950144</td>
<td>16320267</td>
</tr>
<tr>
<td>See increases</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **After sometime:**
  ```plaintext
<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>CLASS</th>
<th>CR_BLOCK</th>
<th>CR_BUSY</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------</td>
<td>-----------------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>CR_CONGESTED</td>
<td>CURRENT_BLOCK</td>
<td>CURRENT_BUSY</td>
<td>CURRENT_CONGESTED</td>
</tr>
<tr>
<td>1 data block</td>
<td>162480580</td>
<td>5097185</td>
<td></td>
</tr>
<tr>
<td>477722</td>
<td>347923719</td>
<td>2950376</td>
<td>16320269</td>
</tr>
</tbody>
</table>
  ```
Diagnosis:
- CPU starvation by LMS/D processes caused GC waits.

Solution:
- Killed the Netbackup processes
- LMD and LMS got the CPU
Increasing Interconnect Speed

- Faster Hardware
  - Gigabit Ethernet; not Fast
  - Infiniband, even if IP over IB

- NIC settings
  - Duplex Mode
  - Highest Top Bit Rate

- TCP Settings
  - Flow Control Settings
  - Network Interrupts for CPU
  - Socket Receive Buffer

- LAN Planning
  - Private LANs
  - Collision Domains
High Speed Interconnects

- Oracle will support RDS over Infiniband
- On 10 Gig Ethernet as well
In summary: Planning

- Adequate CPU, Network, Memory
- Sequences – cache, noorder
- Tablespaces read only
- Un-compact small hot tables
- Keep undo and redo on fastest disks
- Avoid full table scans of large tables
- Avoid DDLs and unnamed PL/SQL blocks
In summary: Diagnosis

- Start with AWR
- Identify symptoms and assign causes
- Don’t get fooled by “gc” waits as interconnect issues
- Find the correlation between “dropped” packets in network, CPU issues from sar and “gc buffer lost” in sysstat reports.
Thank You!

Download from:
proligence.com/downloads.html