




ORACLE®

RAC Performance Experts Reveal All

Barb Lundhild RAC Product Management
Michael Zoll RAC Development, Performance

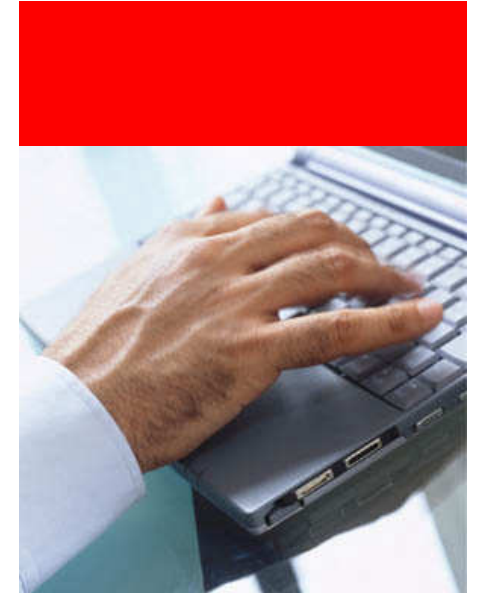


The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

Practical RAC Performance Analysis Revealed

- RAC Fundamentals and Infrastructure
- Common Problems and Symptoms
- Application and Database Design
- Diagnostics and Problem Determination
- Summary: Practical Performance Analysis
- Appendix





OBJECTIVE

- Realize that RAC performance does not requires “Black Magic”
- General system and SQL analysis and tuning experience is practically sufficient for RAC
- Problems can be identified with a minimum of metrics and effort
- Diagnostics framework and Advisories are efficient

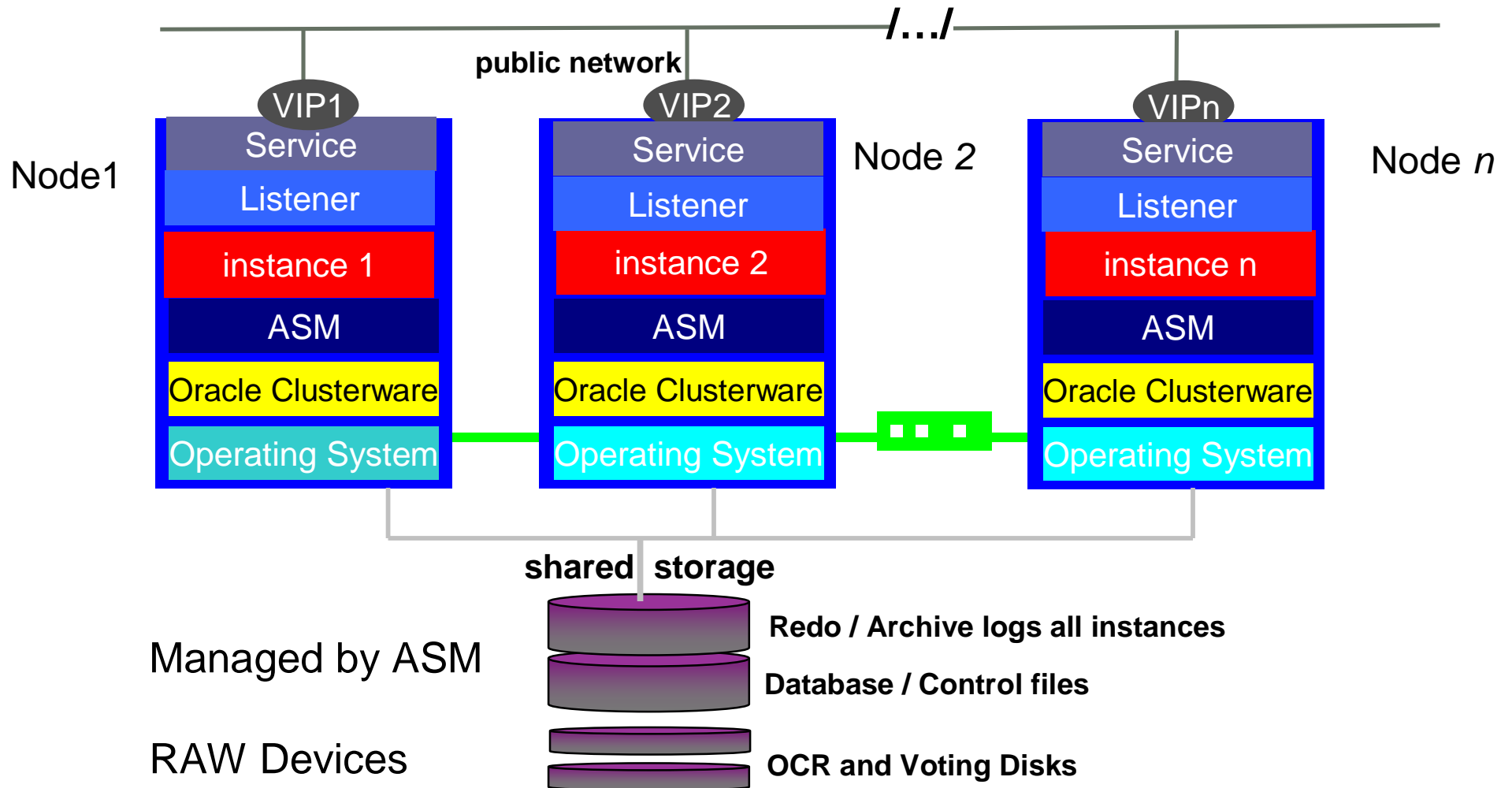


RAC Fundamentals and Infrastructure



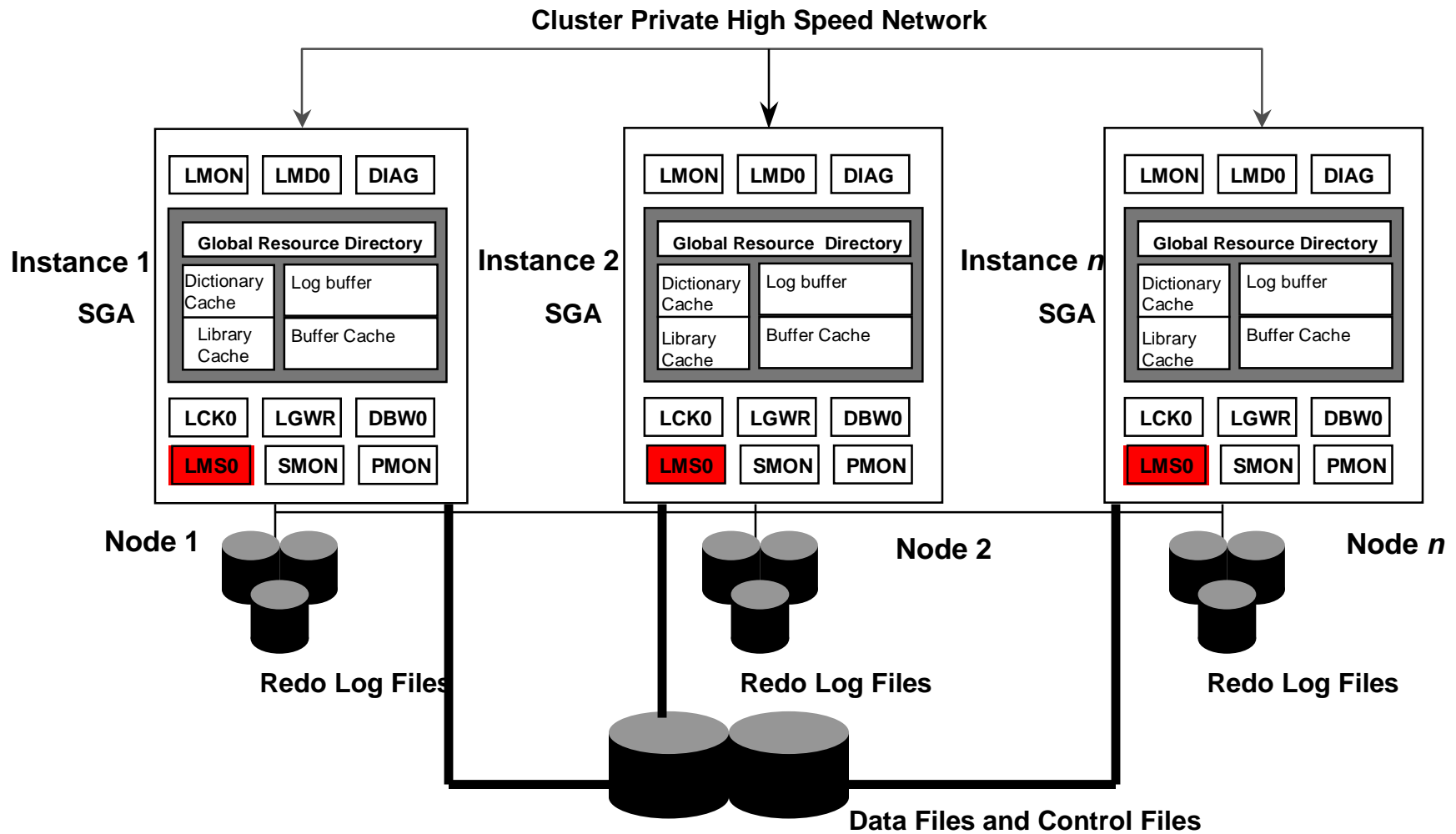


RAC 10g Architecture





Under the Covers

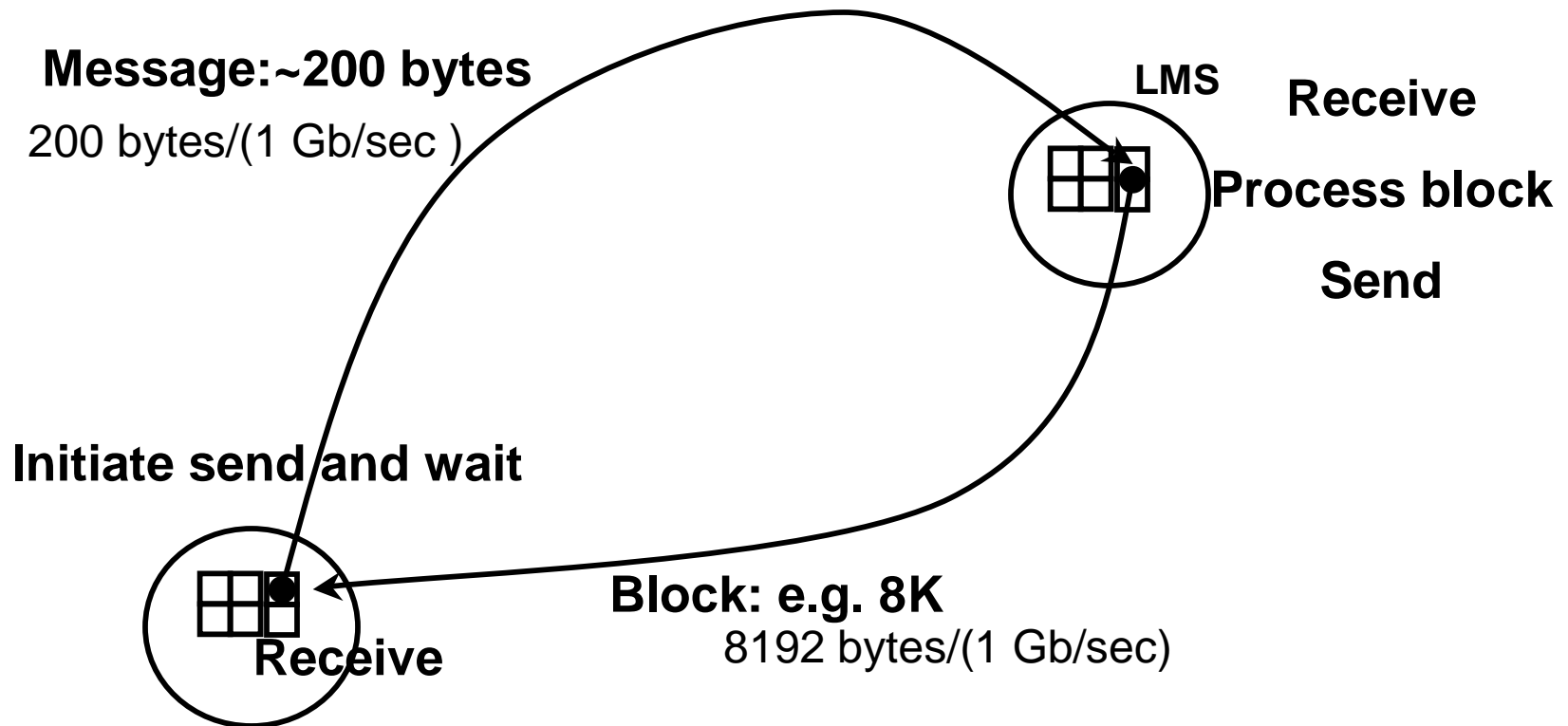




Global Cache Service (GCS)

- Guarantees cache coherency
- Manages caching of shared data via Cache Fusion
- Minimizes access time to data which is not in local cache and would otherwise be read from disk or rolled back
- Implements fast direct memory access over high-speed interconnects for all data blocks and types
- Uses an efficient and scalable messaging protocol

Interconnect and IPC processing



Total access time: e.g. ~360 microseconds (UDP over GBE)

Network propagation delay ("wire time") is a minor factor for roundtrip time
(approx.: 6% , vs. 52% in OS and network stack)



Block Access Cost

Cost determined by

- Message Propagation Delay
- IPC CPU
- Operating system scheduling
- Block server process load
- Interconnect stability



Block Access Latency

- Defined as roundtrip time
- Latency variation (and CPU cost) correlates with
 - processing time in Oracle and OS kernel
 - db_block_size
 - interconnect saturation
 - load on node (CPU starvation)
- ~300 microseconds is lowest measured with UDP over Gigabit Ethernet and 2K blocks
- ~ 120 microseconds is lowest measured with RDS over Infiniband and 2K blocks



Fundamentals: Minimum Latency (*), UDP/GBE and RDS/IB

Block size RT (ms)	2K	4K	8K	16K
UDP/GE	0.30	0.31	0.36	0.46
RDS/IB	0.12	0.13	0.16	0.20

(*) roundtrip, blocks are not “busy” i.e. no log flush, defer etc.; latency distributions can vary based on load and access distribution

AWR and Statspack reports would report averages as if they were normally distributed, in reality the network latencies show tail off to one side



Infrastructure: Private Interconnect

- Network between the nodes of a RAC cluster MUST be private
- Supported links: GbE, IB (IPoIB: 10.2)
- Supported transport protocols: UDP, RDS (10.2.0.3)
- Use multiple or dual-ported NICs for redundancy and increase bandwidth with NIC bonding
- Large (Jumbo) Frames for GbE recommended



Infrastructure: Interconnect Bandwidth

- Bandwidth requirements depend on
 - CPU power per cluster node
 - Application-driven data access frequency
 - Number of nodes and size of the working set
 - Data distribution between PQ slaves
- Typical utilization approx. 10-30% in OLTP
 - 10000-12000 8K blocks per sec to saturate 1 x Gb Ethernet (75-80% of theoretical bandwidth)
- Multiple NICs generally not required for performance and scalability



Infrastructure: IPC configuration

- Settings:
 - Socket receive buffers (256 KB – 1MB)
 - Negotiated top bit rate and full duplex mode
 - NIC ring buffers
 - Ethernet flow control settings
 - CPU(s) receiving network interrupts
- Verify your setup:
 - CVU does checking
 - Load testing eliminates potential for problems



Infrastructure: Operating System

- Block access latencies increase when CPU(s) busy and run queues are long
- Immediate LMS scheduling is critical for predictable block access latencies when CPU > 80% busy
- Fewer and busier LMS processes may be more efficient. i.e. monitor their CPU utilization
- Real Time or fixed priority for LMS is supported
 - Implemented by default with 10.2



Infrastructure: IO capacity

- Disk storage is shared by all nodes, i.e the aggregate IO rate is important
- Log file IO latency can be important for block transfers
- Parallel Execution across cluster nodes requires a well-scalable IO subsystem
 - Disk configuration needs to be responsive and scalable
 - Test with dd or Orion



For More Information on Capacity Planning

- S281269 Oracle Real Application Clusters: Sizing and Capacity Planning Then and Now
4:00 PM Wednesday 304 South



Common Problems and Symptoms





Misconfigured or Faulty Interconnect Can Cause:

- Dropped packets/fragments
- Buffer overflows
- Packet reassembly failures or timeouts
- Ethernet Flow control kicks in
- TX/RX errors

“lost blocks” at the RDBMS level, responsible for
64% of escalations



“Lost Blocks”: NIC Receive Errors

Db_block_size = 8K

```
ifconfig -a:
```

```
eth0 Link encap:Ethernet  HWaddr 00:0B:DB:4B:A2:04  
      inet addr:130.35.25.110  Bcast:130.35.27.255  Mask:255.255.252.0  
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
      RX packets:21721236  errors:135  dropped:0  overruns:0  frame:95  
      TX packets:273120  errors:0  dropped:0  overruns:0  carrier:0  
      ...
```



“Lost Blocks”: IP Packet Reassembly Failures

```
netstat -s
```

```
Ip:
```

```
84884742 total packets received
```

```
...
```

```
1201 fragments dropped after timeout
```

```
...
```

```
3384 packet reassembles failed
```

Finding a Problem with the Interconnect or IPC

Top 5 Timed Events				Avg %Total	
~~~~~				wait Call	
Event	Waits	Time(s)	(ms)	Time	Wait Class
log file sync	286,038	49,872	174	41.7	Commit
gc buffer busy	177,315	29,021	164	24.3	Cluster
gc cr block busy	110,348	5,703	52	4.8	Cluster
<b>gc cr block lost</b>	<b>4,272</b>	<b>4,953</b>	<b>1159</b>	<b>4.1</b>	<b>Cluster</b>
cr request retry	6,316	4,668	739	3.9	Other

*Should never be here*



# Impact of IO capacity issues or bad SQL execution on RAC

- Log flush IO delays can cause “busy” buffers
- “Bad” queries on one node can saturate the link
- IO is issued from ALL nodes to shared storage ( beware of one-node “myopia” )

Cluster-wide impact of IO or query plan issues responsible for 23% of escalations

# Cluster-Wide IO Impact

## Node 1

Top 5 Timed Events			Avg	%Total
~~~~~			wait	Call
Event	Waits	Time(s)(ms)	Time	Time
-----	-----	-----	-----	-----
log file sync	286,038	49,872	174	41.7
gc buffer busy	177,315	29,021	164	24.3
gc cr block busy	110,348	5,703	52	4.8

Node 2

Load Profile

~~~~~	Per Second
	-----
Redo size:	40,982.21
Logical reads:	81,652.41
Physical reads:	51,193.37

# IO and bad SQL problem fixed

```
Top 5 Timed Events
~~~~~
```

Event	Waits	Time (s)	Avg wait (ms)	%Total Call Time	Wait Class
CPU time	4,580	65.4			
log file sync	276,281	1,501	5	21.4	Commit
log file parallel write	298,045	923	3	13.2	System I/O
gc current block 3-way	605,628	631	1	9.0	Cluster
gc cr block 3-way	514,218	533	1	7.6	Cluster

# CPU Saturation or Memory Depletion

Top 5 Timed Events ~~~~~				Avg %Total wait Call	
Event	Waits	Time(s)	(ms)	Time	Wait Class
-----	-----	-----	-----	-----	-----
db file sequential read	1,312,840	21,590	16	21.8	User I/O
gc current block <b>congested</b>	275,004	21,054	<b>77</b>	21.3	Cluster
gc cr grant <b>congested</b>	177,044	13,495	<b>76</b>	13.6	Cluster
gc current block 2-way	1,192,113	9,931	8	10.0	Cluster
gc cr block <b>congested</b>	85,975	8,917	<b>104</b>	9.0	Cluster

*“Congested”: LMS could not de-queue messages fast enough  
Cause : Long run queues and paging on the cluster nodes*



# Health Check

Look for:

- High impact of “lost blocks” , e.g.

`gc cr block lost` 1159 ms

- IO capacity saturation , e.g.

`gc cr block busy` 52 ms

- Overload and memory depletion, e.g

`gc current block congested` 14 ms

*All events with these tags are potential issue, if their % of db time is significant.*

*Compare with the lowest measured latency*

*( target , c.f. SESSION HISTORY reports or SESSION HISTOGRAM view )*



# Application and Database Design





# General Principles

- No fundamentally different design and coding practices for RAC
- Badly tuned SQL and schema will not run better
- Serializing contention makes applications less scalable
- Standard SQL and schema tuning solves > 80% of performance problems



# Scalability Pitfalls

- Serializing contention on a small set of data/index blocks
  - monotonically increasing key
  - frequent updates of small cached tables
  - segment without ASSM or Free List Group (FLG)
- Full table scans
- Frequent hard parsing
- Concurrent DDL ( e.g. truncate/drop )



# Index Block Contention: Optimal Design

- Monotonically increasing sequence numbers
  - Randomize or cache
  - Large ORACLE sequence number caches
- Hash or range partitioning
  - Local indexes



# Data Block Contention: Optimal Design

- Small tables with high row density and frequent updates and reads can become “globally hot” with serialization e.g.
  - Queue tables
  - session/job status tables
  - last trade lookup tables
- Higher PCTFREE for table reduces # of rows per block



# Large Contiguous Scans

- Query Tuning
- Use parallel execution
  - Intra- or inter instance parallelism
  - Direct reads
  - GCS messaging minimal



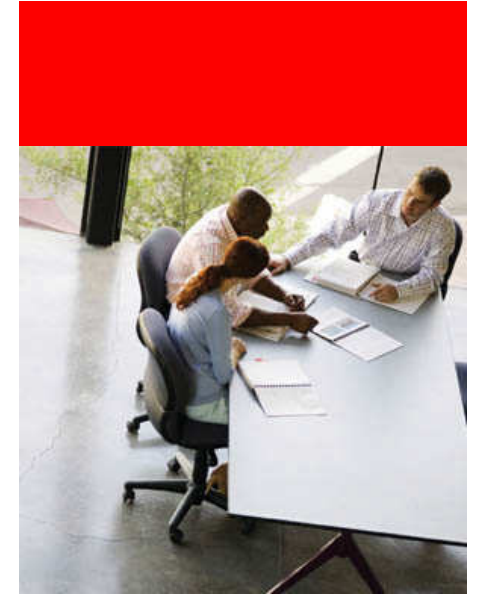
# Health Check

Look for:

- Indexes with right-growing characteristics
  - Eliminate indexes which are not needed
- Frequent updated and reads of “small” tables
  - “small”=fits into a single buffer cache
- SQL which scans large amount of data
  - Bad execution plan
  - More efficient when parallelized



# Diagnostics and Problem Determination





# Performance Checks and Diagnosis

- Traditionally done via AWR or Statspack reports
- “Time-based” paradigm, i.e. identify which events consume the highest proportion of the database time
- Global cache ( “gc” ) events are typical for RAC
- Drill-down to SQL and Segment Statistics



# Event Statistics to Drive Analysis

- Global cache (“gc” ) events and statistics
  - Indicate that Oracle searches the cache hierarchy to find data fast
  - as “normal” as an IO ( e.g. db file sequential read )
- GC events tagged as “busy” or “congested” consuming a significant amount of database time should be investigated
  - At first, assume a load or IO problem on one or several of the cluster nodes



# Global Cache Event Semantics

All Global Cache Events will follow the following format:

GC ...

- CR, current
  - Buffer requests and received for read or write
- block, grant
  - Received block or grant to read from disk
- 2-way, 3-way
  - Immediate response to remote request after N-hops
- busy
  - Block or grant was held up because of contention
- congested
  - Block or grant was delayed because LMS was busy or could not get the CPU

# “Normal” Global Cache Access Statistics

Event	Waits	Time(s)	Avg (ms)	%Total Call Time	Wait Class
CPU time	4,580	65.4			
log file sync	276,281	1,501	5	21.4	Commit
log file parallel write	298,045	923	3	13.2	System I/O
gc current block 3-way	605,628	631	1	9.0	Cluster
gc cr block 3-way	514,218	533	1	7.6	Cluster

*Reads from remote cache instead of disk*

*Avg latency is 1 ms or less*

# “Abnormal” Global Cache Statistics

Top 5 Timed Events			Avg %Total		
~~~~~			wait	Call	
Event	Waits	Time(s)	(ms)	Time	Wait Class
log file sync	286,038	49,872	174	41.7	Commit
<b>gc buffer busy</b>	177,315	29,021	<b>164</b>	24.3	<b>Cluster</b>
<b>gc cr block busy</b>	110,348	5,703	<b>52</b>	4.8	<b>Cluster</b>

*“busy” indicates contention*

*Avg time is too high*



# Checklist for the Skeptical Performance Analyst ( AWR based )

- Check where most of the time in the database is spend (“Top 5” )
- Check whether gc events are “busy”, “congested”
- Check the avg wait time
- Drill down
  - SQL with highest cluster wait time
  - Segment Statistics with highest block transfers

# Drill-down: An IO capacity problem

## Top 5 Timed Events

Avg %Total  
wait Call

Event	Waits	Time(s)	(ms)	Time	Wait Class
db file scattered read	3,747,683	368,301	98	33.3	User I/O
gc buffer busy	3,376,228	233,632	69	21.1	Cluster
db file parallel read	1,552,284	225,218	145	20.4	User I/O
gc cr multi block request	35,588,800	101,888	3	9.2	Cluster
read by other session	1,263,599	82,915	66	7.5	User I/O

*Symptom of Full Table Scans*

*IO contention*



# Drill-down: SQL Statements

*“Culprit”: Query that overwhelms IO subsystem on one node*

Physical Reads	Executions	per Exec	% Total
-----	-----	-----	-----
<b>182,977,469</b>	<b>1,055</b>	<b>173,438.4</b>	99.3

SELECT SHELL FROM ES_SHELL WHERE MSG_ID = :msg_id ORDER BY ORDER_NO ASC

*The same query reads from the interconnect:*

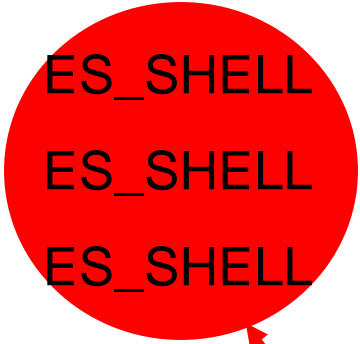
Cluster	CWT % of	CPU	
Wait Time (s)	Elapsd Tim	Time(s)	Executions
-----	-----	-----	-----
<b>341,080.54</b>	<b>31.2</b>	<b>17,495.38</b>	1,055

SELECT SHELL FROM ES_SHELL WHERE MSG_ID = :msg_id ORDER BY ORDER_NO ASC



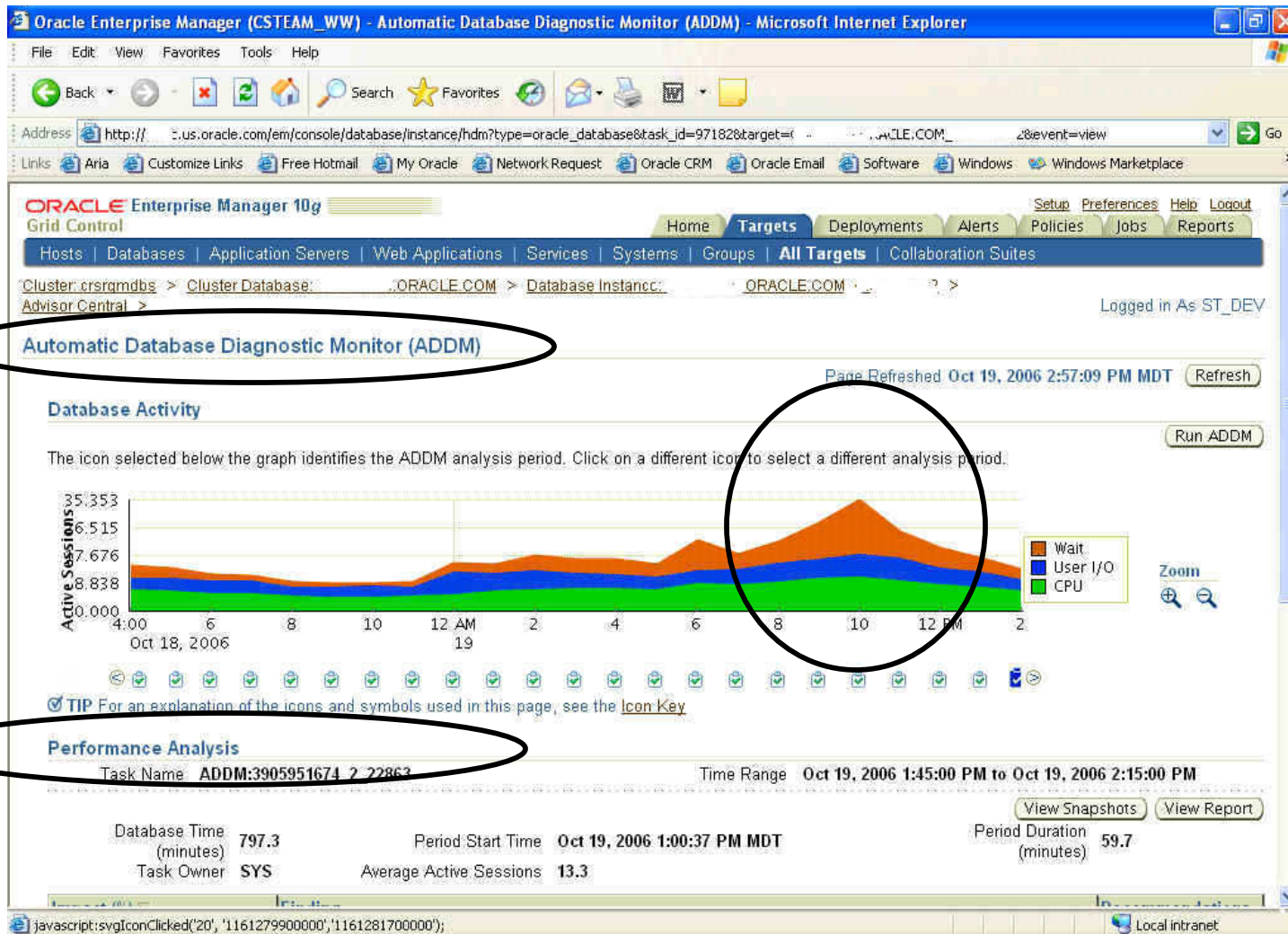
# Drill-Down: Top Segments

Tablespace Name	Object Name	Subobject Name	Obj. Type	GC Buffer Busy	% of Capture
ESSMLTBL	ES_SHELL	SYS_P537	TABLE	311,966	9.91
ESSMLTBL	ES_SHELL	SYS_P538	TABLE	277,035	8.80
ESSMLTBL	ES_SHELL	SYS_P527	TABLE	239,294	7.60
...					



**Apart from being the table with the highest IO demand it was the table with the highest number of block transfers AND global serialization**

# ... and now for something different: Automated Performance Analysis



# Impact of RAC Findings

Impact

Oracle Enterprise Manager - Automatic Database Diagnostic Monitor (ADDM) - Microsoft Internet Explorer

Address: http://...us.oracle.com/em/console/database/instance/hdm?type=oracle_database&task_id=97226&target=... ORACLE.COM_ ?event=view

Oct 18, 2006 19

**Performance Analysis**

Task Name: **ADDM:3905951674_1_22863** Time Range: Oct 19, 2006 1:45:00 PM to Oct 19, 2006 2:15:00 PM

Database Time (minutes): 751 Period Start Time: Oct 19, 2006 1:00:31 PM MDT Period Duration (minutes): 59.9

Task Owner: SYS Average Active Sessions: 12.5

Impact (%)	Finding	Recommendations
51.4	PL/SQL execution consumed significant database time.	1 SQL Tuning
35.2	Time spent on the CPU by the instance was responsible for a substantial part of database time.	2 SQL Tuning
29.4	SQL statements consuming significant database time were found.	5 SQL Tuning
14.8	Wait class "User I/O" was consuming significant database time.	
14.3	SQL statements responsible for significant inter-instance messaging were found.	5 SQL Tuning
12.9	Contention on the high watermark (HWM) enqueue was consuming significant database time.	5 Schema
4.4	Read and write contention on database blocks was consuming significant database time.	2 Schema
6.1	Waits on event "log file sync" while performing COMMIT and ROLLBACK operations were consuming significant database time.	1 Host Configuration
4.1	Wait class "Other" was consuming significant database time.	
3.7	Higher than expected latency of the cluster interconnect was responsible for significant database time on this instance.	1 Host Configuration

Informational Findings

Home | Targets | Deployments | Alerts | Policies | Jobs | Reports | Setup | Preferences | Help | Logout

Copyright © 1996, 2006, Oracle. All rights reserved.

Local intranet

# Automated Findings and Actions: Interconnect

**Performance Finding Details**

Database Time (minutes): 751      Period Start Time: Oct 19, 2006 1:00:31 PM MDT      Period Duration (minutes): 59.9  
Task Owner: SYS      Task Name: ADDM:3905951674_1_22863      Average Active Sessions: 12.5

**Finding** (circled): Higher than expected latency of the cluster interconnect was responsible for significant database time on this instance.

Impact (minutes): 28      Impact (%): 3.7

**Recommendations**

Details	Category	Benefit (%)
Hide	Host Configuration	3.7

**Action** (circled): Check the configuration of the cluster interconnect. Check OS setup like adapter setting, firmware and driver release. Check that the OS's socket receive buffers are large enough to store an entire multiblock read. The value of parameter "db_file_multiblock_read_count" may be decreased as a workaround. Investigate cause of high network interconnect latency between database instances. Oracle's recommended solution is to use a high speed dedicated network.

**Rationale** The instance was consuming 14883 kilo bits per second of interconnect bandwidth.

**Findings Path**

Findings	Impact (%)	Additional Information
Higher than expected latency of the cluster interconnect was responsible for significant database time on this instance.	3.7	
Inter-instance messaging was consuming significant database time on this instance.	16.6	
Wait class "Cluster" was consuming significant database time.	16.6	

# Automated Findings and Actions: Block Contention

The screenshot displays the Oracle Enterprise Manager interface for Performance Finding Details. The browser title is "Oracle Enterprise Manager - Performance Finding Details - Microsoft Internet Explorer". The address bar shows the URL: <http://...us.oracle.com/em/console/database/instance/hdm?event=findingDetails&findingID=6&target=...>. The page is titled "Performance Finding Details" and is part of the "Automatic Database Diagnostic Monitor (ADDM)" section.

**Finding** (circled in black):

- Database Time (minutes): 751
- Period Start Time: Oct 19, 2006 1:00:31 PM MDT
- Period Duration (minutes): 59.9
- Task Owner: SYS
- Task Name: ADDM:3905951674_1_22863
- Average Active Sessions: 12.5
- Finding: Read and write contention on database blocks was consuming significant database time.
- Impact (minutes): 70.3
- Impact (%): 9.4

**Action** (circled in black):

**Recommendations**

Details	Category	Benefit (%)
Hide	Schema	0.3
Action	Consider rebuilding the TABLE "SCHEM1.PAGES_EXT_HEADER" with object id 22939 using a higher value for PCTFREE.	
Database Object	SCHEM1.PAGES_EXT_HEADER	
Show	Schema	0.3

**Findings Path**

Findings	Impact (%)	Additional Information
Read and write contention on database blocks was consuming significant database time.	9.4	
Inter-instance messaging was consuming significant database time on this instance.	16.6	
Wait class "Cluster" was consuming significant database time.	16.6	

Navigation links: Home | Targets | Deployments | Alerts | Policies | Jobs | Reports | Setup | Preferences | Help | Logout

Copyright © 1996, 2006, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

# Automated Findings and Actions: SQL

Impact (minutes) 107.5  
Impact (%) 14.3

**Recommendations**

Schedule SQL Tuning Advisor

Select All | Select None | Show All Details | Hide All Details

Select Details	Category	Benefit (%)
<input type="checkbox"/>	Hide SQL Tuning	8.6
<b>Action</b> Tune the PL/SQL block with SQL_ID "203jw45ug461q". Refer to the "Tuning PL/SQL Applications" chapter of Oracle's "PL/SQL User's Guide and Reference" SQL Text BEGIN ES_MESSAGE_API.insert_message(p_mid,p_but... SQL ID 203jw45ug461q		
Rationale SQL statement with SQL_ID "203jw45ug461q" was executed 223794 times and had an average elapsed time of 0.1 seconds. Rationale Average time spent in Cluster wait events per execution was 0.013 seconds.		
<input checked="" type="checkbox"/>	Hide SQL Tuning	4.7
<b>Action</b> Run SQL Tuning Advisor on the SQL statement with SQL_ID "62q3a0304j9kt". <a href="#">Run Advisor Now</a> SQL Text INSERT INTO ES_BODY (MSG_ID, BODY) SELECT :B2_BODY FROM ES_BODY WHERE MSG_ID = ... SQL ID 62q3a0304j9kt		
Rationale SQL statement with SQL_ID "62q3a0304j9kt" was executed 4600 times and had an average elapsed time of 0.46 seconds. Rationale Average time spent in Cluster wait events per execution was 0.01 seconds.		
<input type="checkbox"/>	Show SQL Tuning	4.7
<input checked="" type="checkbox"/>	Show SQL Tuning	3.2
<input checked="" type="checkbox"/>	Show SQL Tuning	2.8

**Findings Path**

Expand All | Collapse All

Findings	Impact (%)	Additional Information
SQL statements responsible for significant inter-instance messaging were found	14.3	
Wait class "Cluster" was consuming significant database time.	16.6	

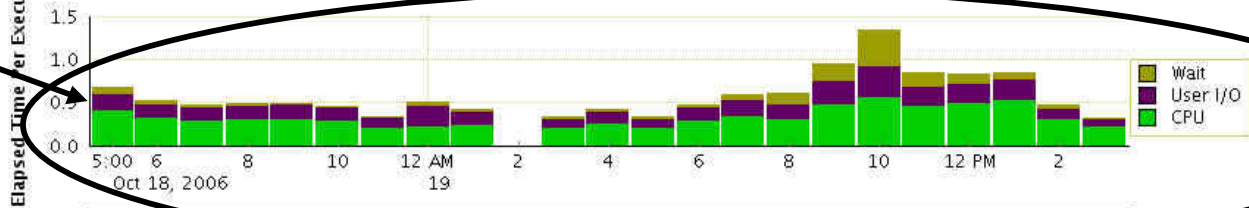
Action

# Automated SQL Drill-Down

SQL Text

```
INSERT INTO ES_BODY (MSG_ID, BODY) SELECT :B2 , BODY FROM ES_BODY WHERE MSG_ID = :B1
```

Per SQL  
Statistics  
Over Time



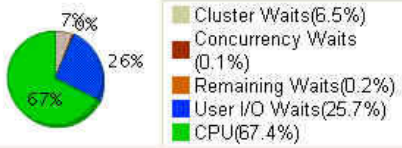
### Snapshot Details

Snapshot Time Oct 19, 2006 2:00:26 PM to Oct 19, 2006 3:00:24 PM

### General

Module **essmo@rgmum107 (TNS V1-V3)**  
Action  
Parsing Schema **ES_MAIL**

### Activity By Waits



### Activity By Time

Elapsed Time (sec) **1,347.63**  
CPU Time (sec) **908.53**  
Wait Time (sec) **439.10**

### Elapsed Time Breakdown

SQL Time (sec) **1,347.63**  
PL/SQL Time (sec) **0.00**  
Java Time (sec) **0.00**



# Summary: Practical Performance Analysis





# Diagnostics Flow

- Start with simple validations :
  - Private Interconnect used ?
  - Lost blocks and failures ?
  - Load and load distribution issues ?
- Check avg latencies, busy, congested events and their significance
- Check OS statistics ( CPU, disk , virtual memory )
- Identify SQL and Segments

**MOST OF THE TIME, A PERFORMANCE PROBLEM IS NOT  
A RAC PROBLEM**



# Actions

- Interconnect issues must be fixed first
- If IO wait time is dominant , fix IO issues
  - At this point, performance may already be good
- Fix “bad” plans
- Fix serialization
- Fix schema



# Checklist for Practical Performance Analysis

- ADDM provides RAC performance analysis of significant metrics and statistics
  - ADDM findings should always be studied first
  - It provides detailed findings for SQL, segments and blocks
- AWR for detailed statistics and historical performance analysis
  - Export statistics repository long-term
- ASH provides finer-grained session specific data
  - Catches variation in snapshot data
  - Stored in AWR repository
  - Used by ADDM



## Recommendations

- Most relevant data for analysis can be derived from the wait events
- ***Always use EM and ADDM reports for performance health checks and analysis***
- ASH can be used for session-based analysis of variation
- Export AWR repository regularly to save all of the above



## For More Information


<http://search.oracle.com>

**REAL APPLICATION CLUSTERS**



or

[otn.oracle.com/rac](http://otn.oracle.com/rac)



ORACLE®

