

ORACLE®



**ORACLE
OPEN
WORLD**

Your. Open. World.

**Practical Performance Management
for Oracle RAC**

Barb Lundhild RAC Product Management
Michael Zoll RAC Development, Performance

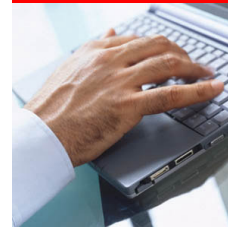
ORACLE

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.

ORACLE

Agenda

- Oracle RAC Fundamentals and Infrastructure
- Common Problems and Symptoms
- Application and Database Design
- Diagnostics and Problem Determination
- Appendix



ORACLE

OBJECTIVE

- Convey a few simple and fundamental concepts of Oracle RAC performance
- Provide some simple sizing hints
- State that general system and SQL analysis and tuning experience is practically sufficient for Oracle RAC
- Note that problems can be identified with a minimum of metrics and effort
- Builds on similar presentation from OOW 2007

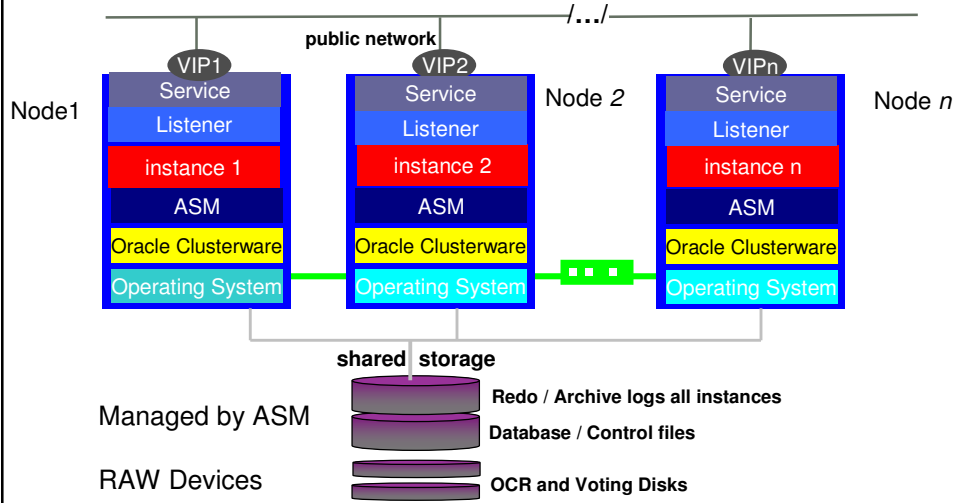
ORACLE

RAC Fundamentals and Infrastructure



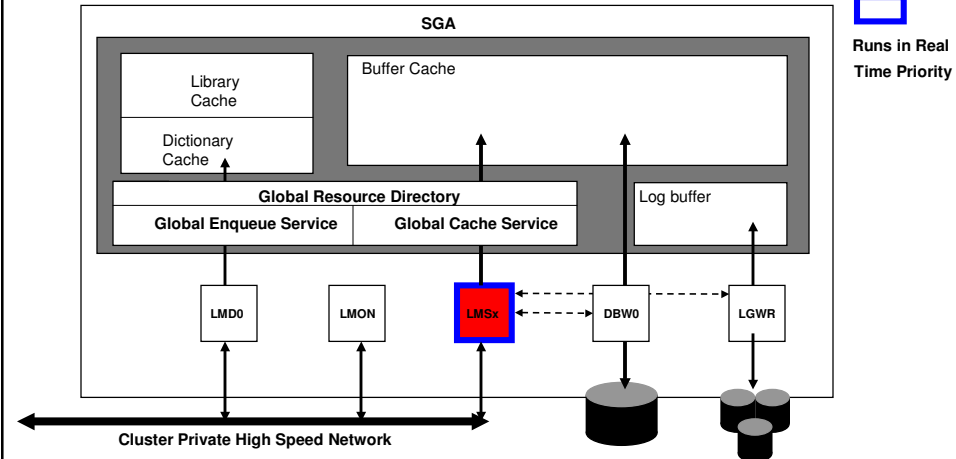
ORACLE

Oracle RAC Architecture



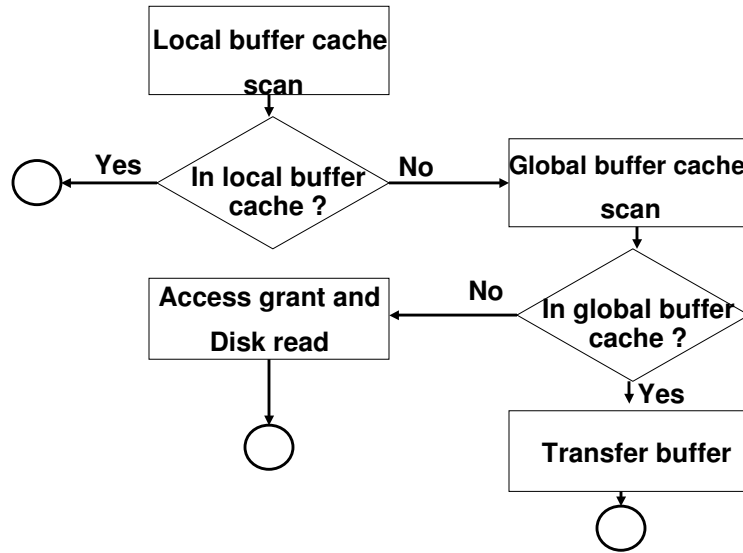
ORACLE

Global Cache and Global Enqueue Service: Functions and Processes



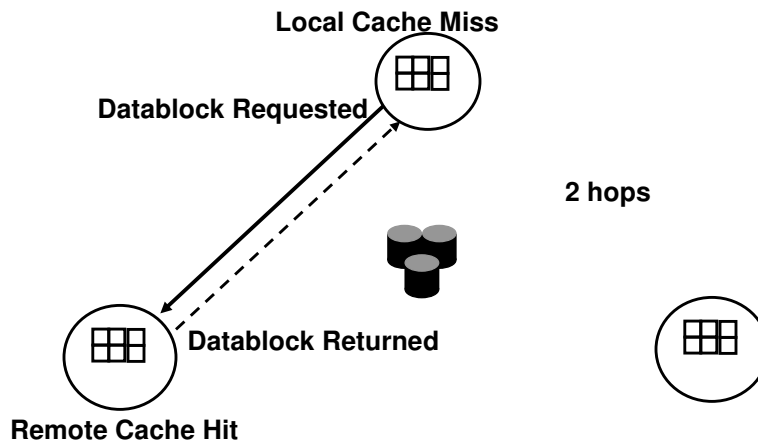
ORACLE

Global Buffer Cache



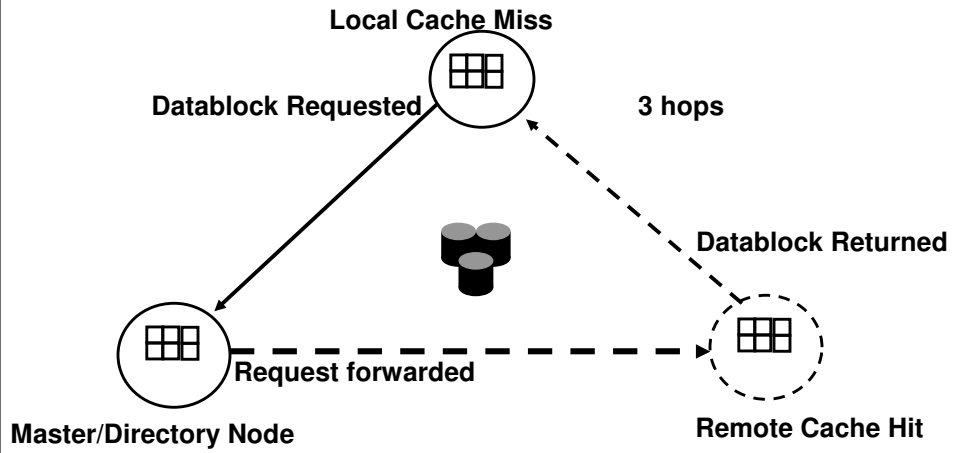
ORACLE

Data in Remote Cache: 2-way



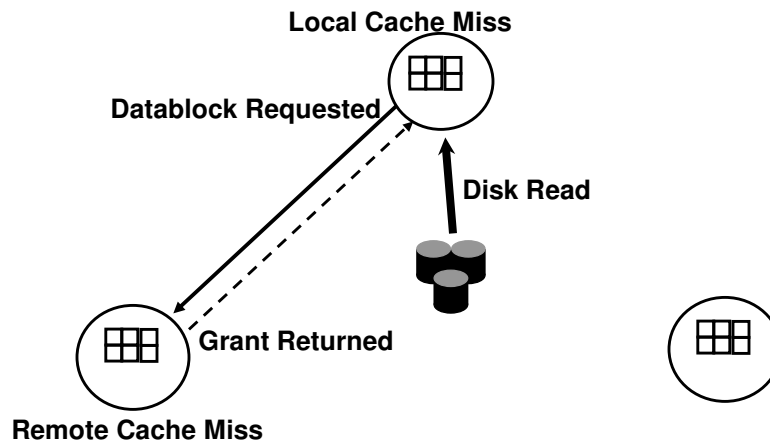
ORACLE

Data in Remote Cache: 3-way



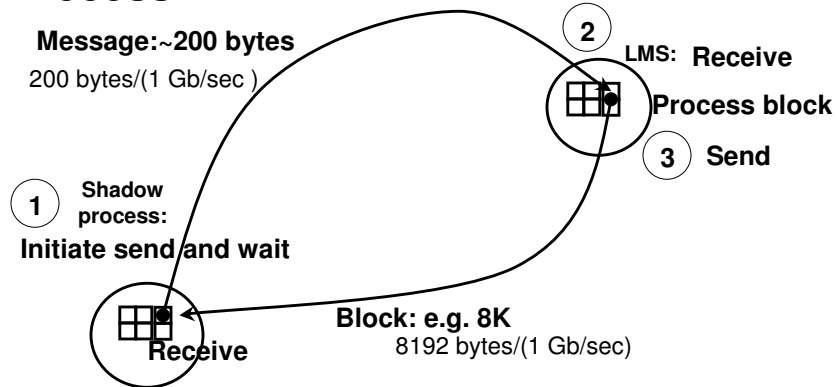
ORACLE

Data On Disk



ORACLE

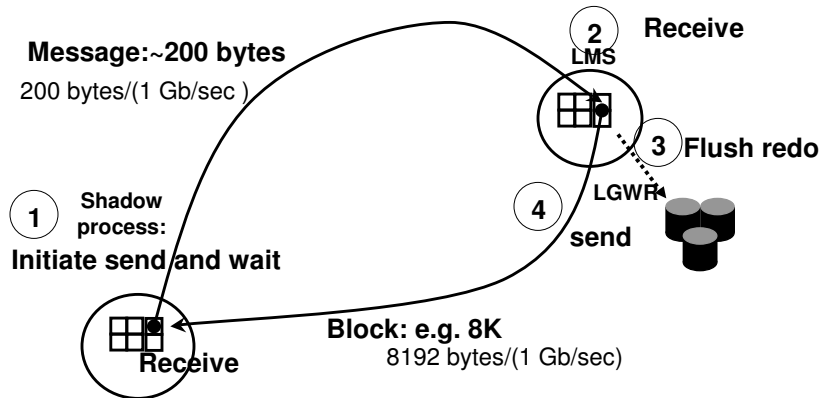
Performance Model: Immediate Block Access



Total access time: e.g. ~360 microseconds (UDP over GBE)
 - Network propagation delay :minor factor for roundtrip time if sufficient bandwidth
 - Message processing: CPU and scheduling in Oracle and OS

ORACLE

Performance Model: Block Access with contention (“busy block”)



Total access time: e.g. ~360 microseconds transfer(UDP over GBE) +
 time for log flush and context switch , e.g. 3ms = 3.36 ms

ORACLE

Fundamentals: Minimum Latency (*), UDP/GbE and RDS/IB (2-way/3-way)

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, AWR and Statspack reports would report averages as if they were normally distributed, the session wait history which is included in Statspack in 10.2 and AWR in 11g will show the actual quantiles

The minimum values in this table are the optimal values for 2-way and 3-way block transfers, but can be assumed to be the expected values (I.E. 10ms for a 2-way block would be very high)

ORACLE

Sizing and Configuration Guidelines



ORACLE

Private Interconnect

- Network between the nodes of a RAC cluster MUST be private
- Bandwidth requirements depend on several factors
 - #of CPUs and CPU speed per node
 - # of nodes
 - Type of Workload: OLTP or DSS
- Large (Jumbo) Frames for GbE recommended
 - Avoids fragmentation and reassembly
- Interconnect bandwidth should be tested with non-Oracle utilities (e.g. iPerf)
 - No packet loss at 75% - 80% of bandwidth utilization

ORACLE

Interconnect Bandwidth

- Prediction based on upper bound heuristics
 - Large data messages ("blocks") received can range from 1-8% of logical reads/sec
 - Small message rate approx.
 $((\text{Total physical reads per sec}) * (N-1)/N) * 4 + \text{blocks per sec} * 3$
- EXAMPLE for bandwidth calculation
 - **Predicted database load per node**
 - Physical reads per node: 10000/sec
 - Logical reads per node: 100000/sec
 - **Estimated messages per node**
 - 1000 – 8000 **blocks/sec** * 2 (receive and send)
 - 33000 – 54000 **messages/sec**
 - 35000 – 70000 **total messages/sec**
 - **Estimated bandwidth utilization**
 - 22.6MB/sec – 74.8 MB/sec

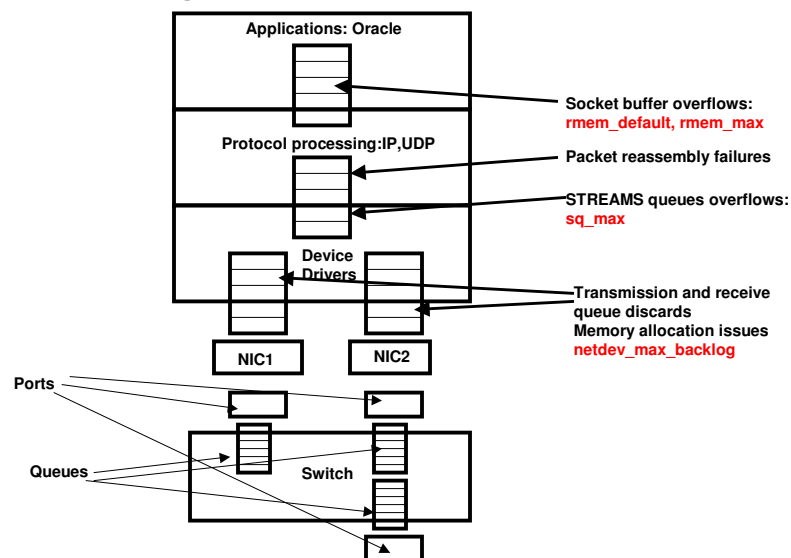
ORACLE

Interconnect Bandwidth

- Generally, 1Gb/sec sufficient for performance and scalability **in OLTP**.
 - Typical link utilization approx. 10-30% **in OLTP**
 - NIC bonding and load balancing recommended
 - 10GbE possible, but not much experience with RAC
- DSS/DW systems should be designed with > 1Gb/sec capacity
- Buffer overflows, congested links and flow control can have severe consequences for performance
- A sizing approach with rules of thumb is described in
 - Project MegaGrid: Capacity Planning for Large Commodity Clusters (<http://otn.oracle.com/rac>)

ORACLE

IPC configuration



ORACLE

Configuration Hints

- On systems with a large number of CPUs or NUMA systems,
 - Direct NIC interrupts to specific CPUs
 - Bind LMS to those CPUs (or NUMA boards)
- Important Link, NIC and IPC Settings
 - Negotiated top bit rate and full duplex mode
 - UDP receive and send buffers
- CVU does checking

ORACLE

Block Server (LMSx) Processes

- Block access latencies can increase when CPU(s) are busy and run queues are long
- Fewer and busier LMS processes may be more efficient.
 - monitor their CPU utilization
 - the default is good for most requirements
 - LMS processes usually top CPU consumers
 - 70-80% busy per LMS is ok
- Higher priority for LMS is default
 - The implementation is platform-specific

ORACLE

Block Server (LMSx) Processes

- Oracle RAC 10g (10.2.0.4)

# of CPUs	# of LMS
<=2	1
<=12	2
>=12	#cpu/4

- Oracle RAC 11g

# of CPUs	# of LMS
<=4	1
<=16	2
>=16	#cpu/4

- An additional lms for every 32 CPUs

- gcs_server_processes overrides defaults
- Defaults are good for most requirements

ORACLE

Block Server (LMSx) Processes

- SCHED_RR on most platforms
 - Real time priority
 - SCHED_NOAGE = 178 on HP/UX (10.2.0.4)
- High priority can cause starvation when the number of LMS processes exceeds the number of available CPUs
 - Multiple databases per cluster
 - ASM instance
- `_os_sched_high_priority = 0`
 - Turns off SCHED_RR , I.E. real-time priority for LMS

ORACLE

Application and Database Design for Scalability in RAC



ORACLE

General Principles

- No fundamentally different design and coding practices for RAC
- BUT: flaws in execution or design have higher impact in RAC
 - Performance and scalability in RAC will be more sensitive to bad plans or bad schema design
- Standard SQL and schema tuning solves > 80% of performance problems

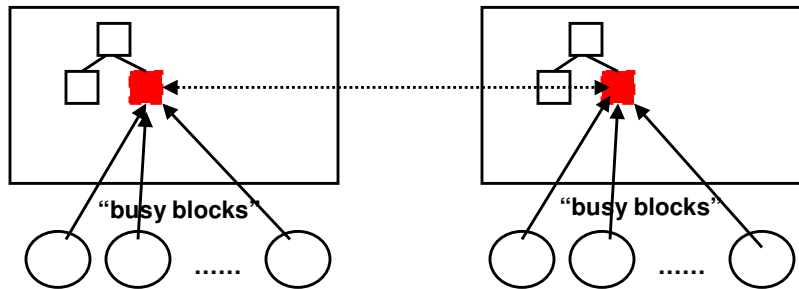
ORACLE

Scaling and Workloads

- DML intensive
 - Scale well, if care is taken to avoid bottlenecks
- Read-intensive
 - Scale well

ORACLE

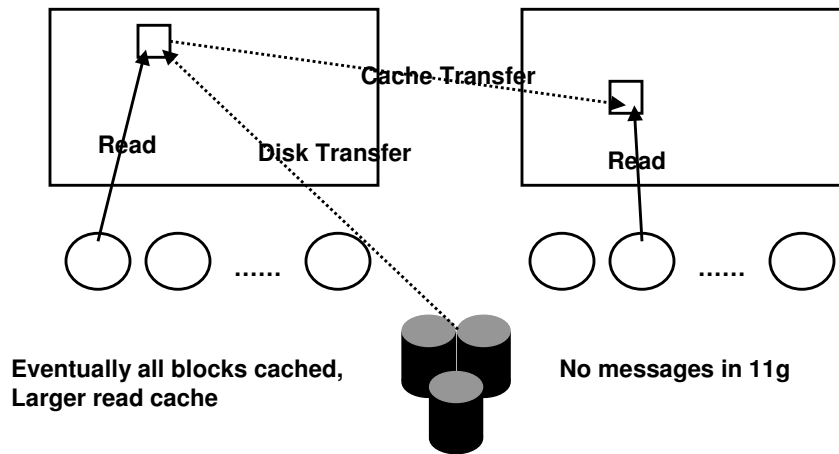
DML Contention and Serialization



INSERT ... WHERE Key = sequence
Other DML/modification intensive operations on small set of
(cached) blocks

ORACLE

Read-intensive



ORACLE

Scalability: Hot Spots

- Serialization on frequently modified blocks has higher cost in RAC
 - INSERT into indexes with right growing characteristics
 - UPDATES of data in the same block
 - Segments without ASSM or Freelist Groups
- A small set of blocks is concurrently modified by all instances
 - Processes queue for "busy" buffers
 - Redo may have to be written before sending data
- Application Data Partitioning (Range, Hash) and routing per partitioning key in application server tier
 - For high performance

ORACLE

Best Practices for Predictable Scalability

- Automatic Segment Space Management required
- Indexes with right-growing characteristics have monotonically increasing keys E.G. DATE columns or keys generated by sequence numbers
 - Cache sequence numbers per instance
 - Hash or range partitioned table with LOCAL indexes
- Frequent updates of “small” and compact tables
 - “small”=fits into a single buffer cache
 - Reduce block size (2K) and row density of blocks (PCTFREE 99)
- Application Modules with such characteristics could also be directed to particular nodes
 - Use cluster managed services as early as possible

ORACLE

Scalability: CPU cost for remote block copies

- A small set of blocks is randomly modified and frequently read
 - UPDATE followed by frequent SELECT
 - INSERTS into index with non-sequential keys
- Traffic across the interconnect higher
- CPU cost for messaging increases but latency low
- Scales as long as CPU is available
- In 11.1, CPU cost is further reduced by data compression and aggregation of message

ORACLE

Migration to RAC and RAC scalability

- Rough database design review points out bottlenecks

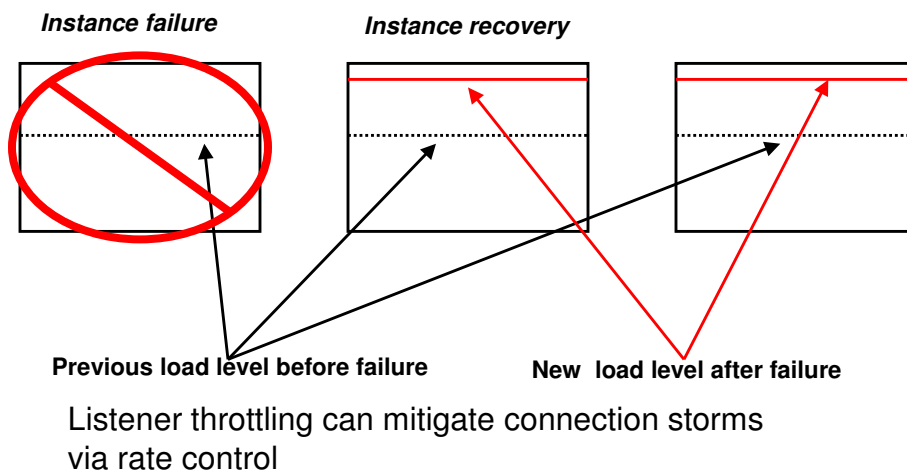
Look for:

- Indexes with right-growing characteristics
- Monotonic, ordered sequences
- Frequently updated and read “small” tables with high row density
- Single instance AWR or Statspacks report
 - “buffer busy” waits indicate contention, may not scale in RAC
 - Index contention waits in segment statistics can point to hot spots in index
 - Row lock waits in segment statistics identify design that may not scale

ORACLE

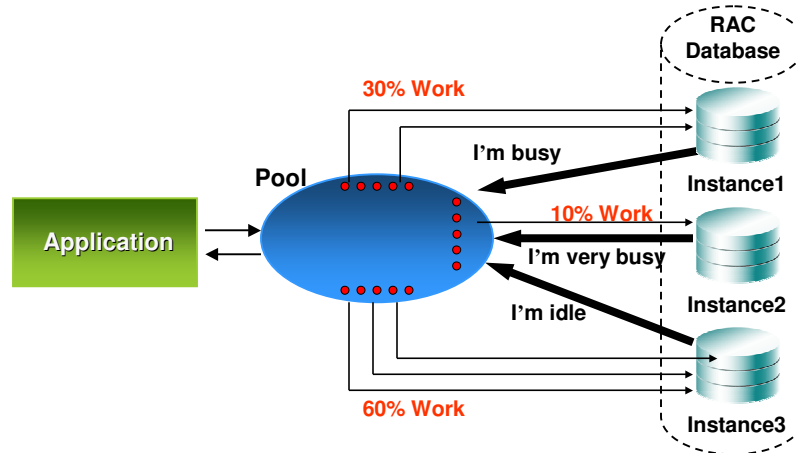
Load Control

Ensure capacity when failures



ORACLE

Leverage Connection Pools Oracle UCP



ORACLE

Performance Health Checks: Metrics and Method using AWR



ORACLE

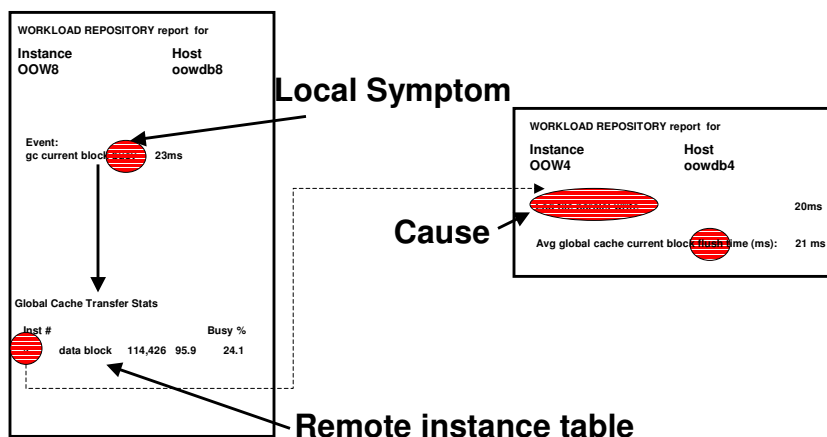
Quick Primer of Global Cache Events

- Time-based events for global cache (gc)
 - Gc cr/current block 2-way/3-way
 - Single immediate block get via interconnect
 - Gc cr/current block busy
 - Single block get via interconnect delayed by log flush
 - Gc cr multi block request
 - Contiguous read via interconnect
 - Gc buffer busy
 - Waiting behind one of the foregoing events
 - Gc cr/current grant 2-way
 - Access granted to data, followed by disk read
 - Gc cr/current block lost
 - Message was dropped
- Often related to other events in terms of causality
 - E.g. gc current block busy -> log file sync/log file parallel write

ORACLE

Global View of Metrics

Note: Cause and effect are distributed



ORACLE

Example: Review AWR

Top 5 Timed Events

	Waits	Time (s)	AVG (ms)	%Time
db file sequential read	2,627,295	21,808	8	43.2 %
CPU time		9,156		18.2
gc current block 3-way	3,289,371	4,019	1	8.0
gc buffer busy	373,777	3,272	9	6.5
gc current block 2-way	3,982,284	3,192	1	6.3

} 20.8%

Data received from other instances:
Faster than IO

ORACLE

Example: Performance Health Check

- IO time for random IOs (43% of wait time)
 - Avg read time: ok -> NO ACTION
- Global Cache (20% of wait time)
 - Avg block access time : ok -> NO ACTION
 - Some serialization, but not a big problem -> INVESTIGATE

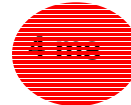
ORACLE

Example: Investigate Serialization

gc buffer busy 373,777 3,272 9 ms

Waits for

gc current block busy 125,595 2,931



Not OK!

Global Cache Transfer Stats

Inst	Block	Blocks	%	%
No	Class	Received	Immed	Busy
4	data block	114,426	95.9	4.1
7	data bloc	162,630	76.6	23.4

Log file IO

Avg global cache current block flush time (ms): 3.7

ORACLE

Example: Analysis

- Serialization is likely to be due to log flush time on another node
 - 3-4 ms for log file IO
 - Some data is modified frequently and changes may not have been logged before data blocks are “pinged” out of cache
- Reason for serialization (“busy”) is normally
 - Block transfers which require a log flush by the server
 - Immediate block transfers which take a long time
- A queue of waiters forms while the transfer is in progress
- Analogous to buffer waits while disk IO is in progress

ORACLE

Example: Segment Statistics

Segments by Global Cache Buffer Busy

ES_BILLING TABLE 97.41 %

Segments by Current Blocks Received

ES_BILLING TABLE 85.81 %

ANALYSIS: TABLE ES_BILLING is frequently read and modified on all nodes. The majority of global cache accesses and serialization can be attributed to this .

ORACLE

Performance Health Check: Method

- Check events with highest % of wait time ("Top 5")
- Check avg access times
 - If higher than normal, look for reasons
- Check for serialization ("busy" events)
 - Find instance(s) which transfer highest % of busy blocks
 - Check avg log flush time and log sync time on those instances
 - Find tables and indexes with highest serialization
- Run advisories on the analyzed snapshots

ORACLE

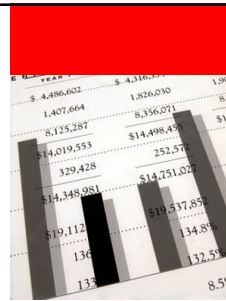
Common Problems and Symptoms



ORACLE

Common Problems and Symptoms

- Interconnect or Switch Problems
- Slow or bottlenecked disks
- System load and scheduling
- Unexpectedly high latencies



ORACLE

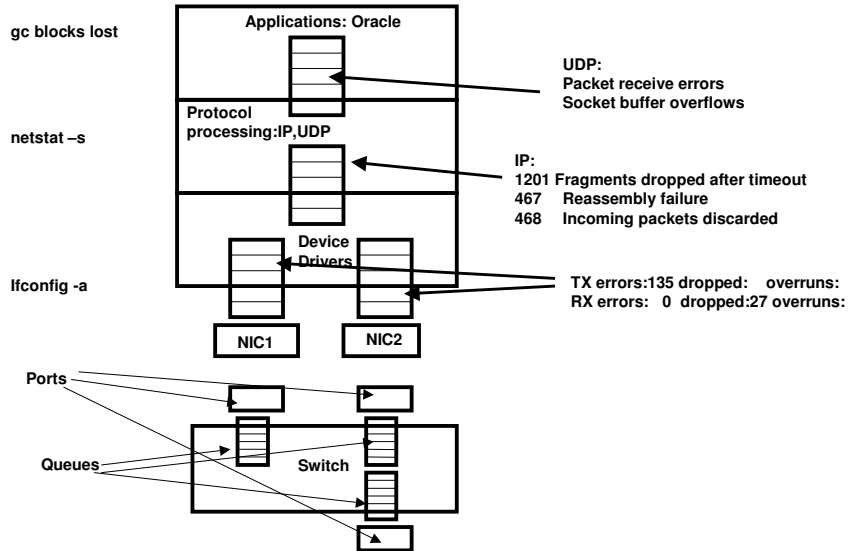
Symptoms of an Interconnect Problem

Event	Waits	Time(s)	Avg wait (ms)	%Total Call 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
gc cr block lost	4,272	4,953	1159	4.1	Cluster
cr request retry	6,316	4,668	739	3.9	Other

*Should never be here
Always a severe performance problem*

ORACLE

Interconnect or IPC problems



ORACLE

Probable Causes of Interconnect Problems

- Network Congestion
 - discarded packets/fragments
 - Packet reassembly failures or timeouts
 - Buffer overflows
 - Flow control
- Checksum errors or corrupted headers

```
$netstat -s
```

```
Ip:
```

```
84884742 total packets received
```

```
1201 fragments dropped after timeout
```

```
3384 packet reassembles failed
```

ORACLE

Probable Causes of Interconnect Problems : NIC or driver errors

- Checksum errors or corrupted frames
- Memory allocation , Queues full

```
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:27 overruns:0 carrier:0
```

```
...
```

ORACLE

Global Cache Lost Block

- Almost always a IPC configuration (buffering and flow control) or downstream (switch etc.) network problem
 - Downstream problems are hard to debug
- Get OS data immediately: ifconfig, netstat
- Detection Time in 10.2.0.4 and 11g reduced
 - 500ms (around 5 secs in 10.2.0.3)
 - can be lowered if necessary
- *Cr request retry* event related to lost blocks
 - It is highly likely to see it when gc cr blocks lost show up

ORACLE

Cluster-Wide Disk I/O Impact

Node 1

Top 5 Timed Events		Avg %Total		
Event	Waits	Time (s)	wait (ms)	Call 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

CAUSE:
Expensive Query in Node 2
Causes IO bottleneck
Node 2

1. IO on disk group containing redo logs is slow
2. Block shipping for frequently modified blocks is delayed by log flush IO
3. Serialization builds up

Load Profile	Per Second
Redo size:	40,982.21
Logical reads:	81,652.41
Physical reads:	51,193.37

ORACLE

Node 2: An IO capacity problem

Top 5 Timed Events		Avg %Total			
Event	Waits	Time(s)	wait (ms)	Call 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

Disk I/O contention

- Disk IO slow due to rogue query
- Slow concurrent scans causes buffer cache contention
- Multiple symptoms for the same cause

ORACLE

CPU Saturation or Long Run Queues

Top 5 Timed Events		Avg %Total			
Event	Waits	Time(s)	wait (ms)	Call Time	Wait Class
db file sequential read	1,312,840	21,590	16	21.8	User I/O
gc current block congested	275,004	21,054	77	21.3	Cluster
gc cr grant congested	177,044	13,495	76	13.6	Cluster

ORACLE

High Latencies

Event	Waits	Time (s)	AVG (ms)	% Call Time
<code>gc cr block 2-way</code>	317,062	5,767	18	19.0
<code>gc current block 2-way</code>	201,663	4,063	20	13.4

Unexpected: To see > 1 ms (AVG ms should be around 1 ms)

- Check network configuration (private ? bandwidth ?)
- Check for high CPU consumption
 - Runaway or spinning processes

ORACLE

Hints and Tips

Look for:

- Unexpected Events
- Unexpected “Hints” (“busy”, “congested”)
- Unexpected high avg
- Maintain “global” perspective
 - Symptom and cause are often not on the same instance
- Execute top down and assume simplest case:
 - The top wait is usually the suspect
 - Gc events look scary but are often secondary effects.

ORACLE

Transient Problems and Hangs



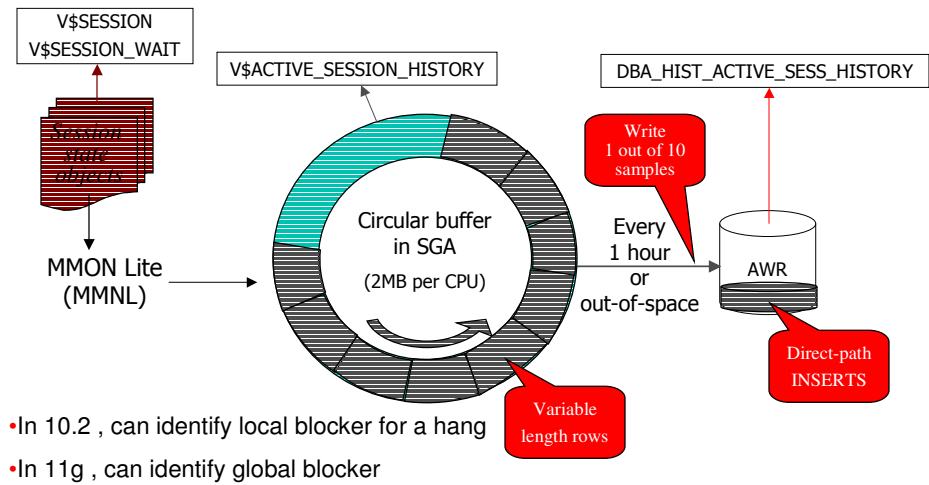
ORACLE

Temporary Slowness and Hang

- Can affect one or more instances in cluster
- Can be related
 - IO issues at log switch time (checkpoint or archiver slow)
 - Process stuck waiting for IO
 - Connection storm
- Hard to establish causality with AWR statistics
- Active Session History

ORACLE

Active Session History



ORACLE

Courtesy of Graham Wood, ORACLE

Accessing ASH data

- Dump to trace file
- V\$ACTIVE_SESSION_HISTORY
- DBA_HIST_ACTIVE_SESS_HISTORY
- ASH report
- EM Diagnostic Pack

ORACLE

Dumping ASH to file

```
>oradebug setmypid
>oradebug dump ashdump 10

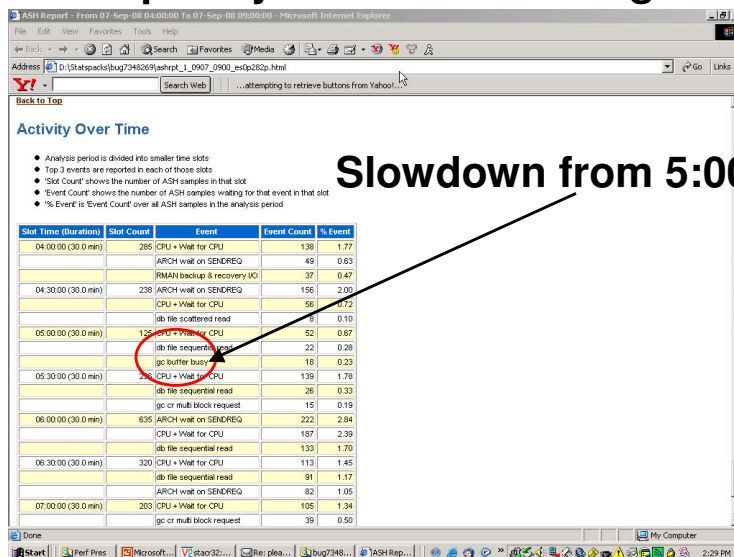
>alter session set events 'immediate trace name
ashdump level 10';
```

- 10 ==> minutes of history you want to dump
- Caveat: no global dump for cluster
- Generated file can be loaded into database using supplied control file `rdbms/demo/ashldr.ctl`

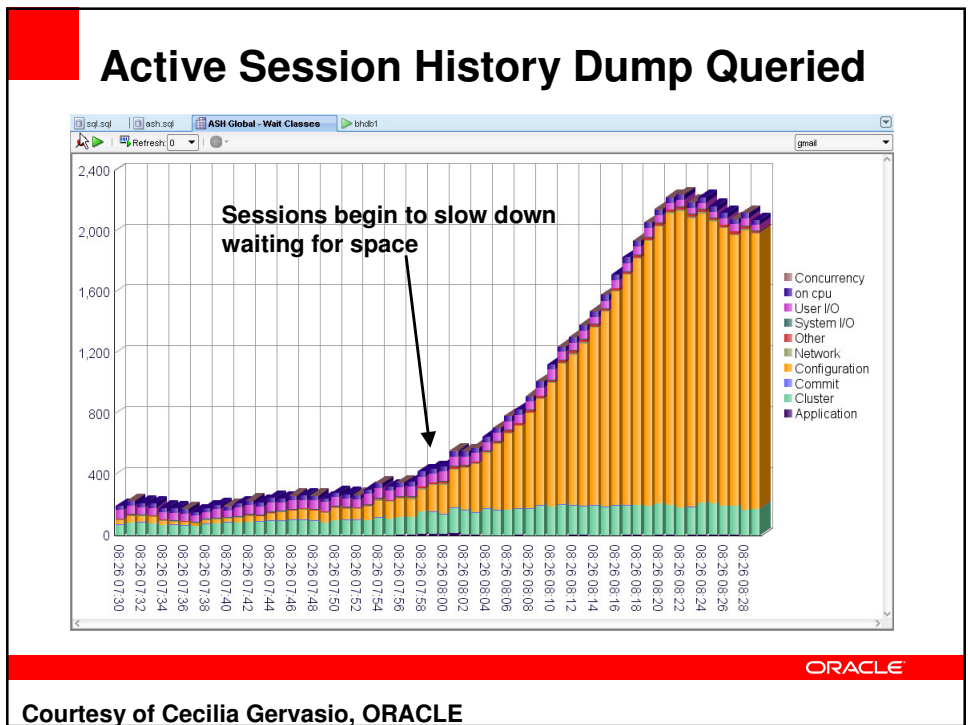
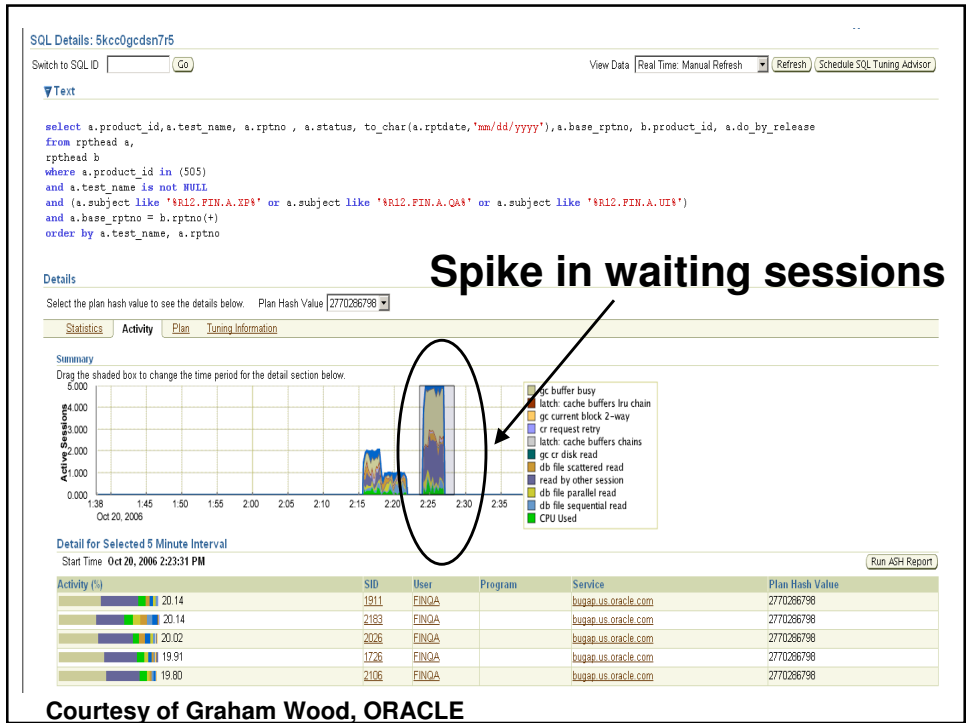
ORACLE

Courtesy of Graham Wood, ORACLE

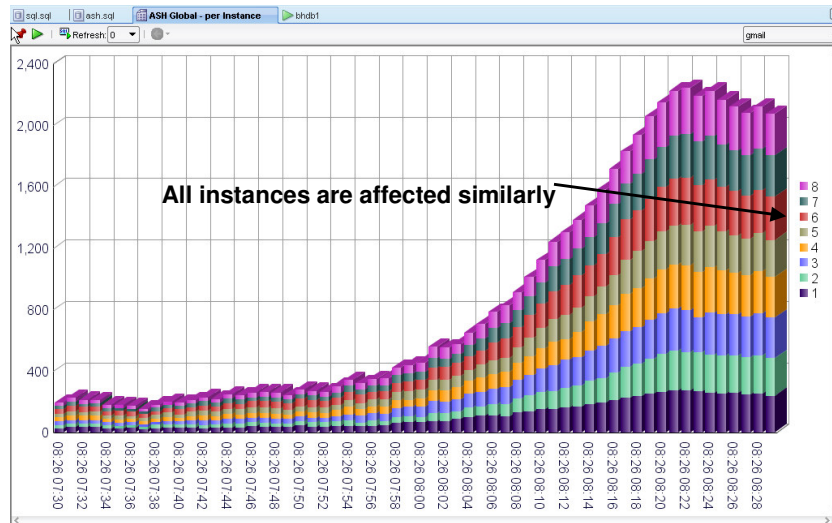
Temporary Slowness or Hang



ORACLE



Active Session History Dump Queried



ORACLE

Courtesy of Cecilia Gervasio, ORACLE

Procedures for transient slowdowns and hangs

- On all instances
 - Produce Active Session History(ASH) report
 - Dump in-memory ASH data and load into database
 - Narrow down to time interval with slowdown

ORACLE

Conclusions



ORACLE

Conclusions

- Anyone can deal with Oracle RAC
- Use Occam's razor :
 - SQL and Schema tuning solves most issues
 - Efforts in testing, configuring and sizing pay off
 - Load control is important
- Any node could be the culprit for a problem, follow the trail the the AWR reports

ORACLE

