Oracle RAC Private Network

Paresh Patel, Member of Technical Staff, PayPal Core Data Platform

Agenda

1. Who am I and Introduction to PayPal
2. Usage of Private network in RAC
3. RAC related background processes
4. SGA in RAC Database
5. Parameters Influencing RAC behavior
6. Wait events in RAC Database
7. Hardware and OS support
8. Infiniband
9. UDP vs. RDS
10. Monitoring
11. Troubleshooting
12. Putting it all together
13. Questions?

Disclaimer: Some of the observations here may not be applicable to your environment so test them out or contact Oracle before implementing.

Who am I

- MTS 2 - Database Engineer, Oracle Database Engineering
- Oracle RAC Certified Professional with more than a decade's experience starting with Oracle 9i
- Oracle RAC, ADG, performance tuning and GoldenGate expert
- Conversant with MongoDB, Cassandra and Couchbase

Introduction to PayPal

Two decades ago, our founders invented payment technology to make buying and selling faster, secure, and easier—and put economic power where it belongs: **In the hands of people.**

Our customers can accept payments in >**100** currencies, withdraw funds to their bank accounts in **56** currencies, and hold balances in their PayPal accounts in **25** currencies.

Almost **8,000** PayPal team members provide support to our customers in over **20** languages.

We are a trusted part of people's financial lives and a partner to merchants in 200+ markets around the world.

Usage of Private Network

✓ Clusterware

- Inter-node communication to maintain cluster integrity
  - Cluster Synchronization service
  - octssd to avoid time drift
  - crs resources and crsd agent processes

✓ Database

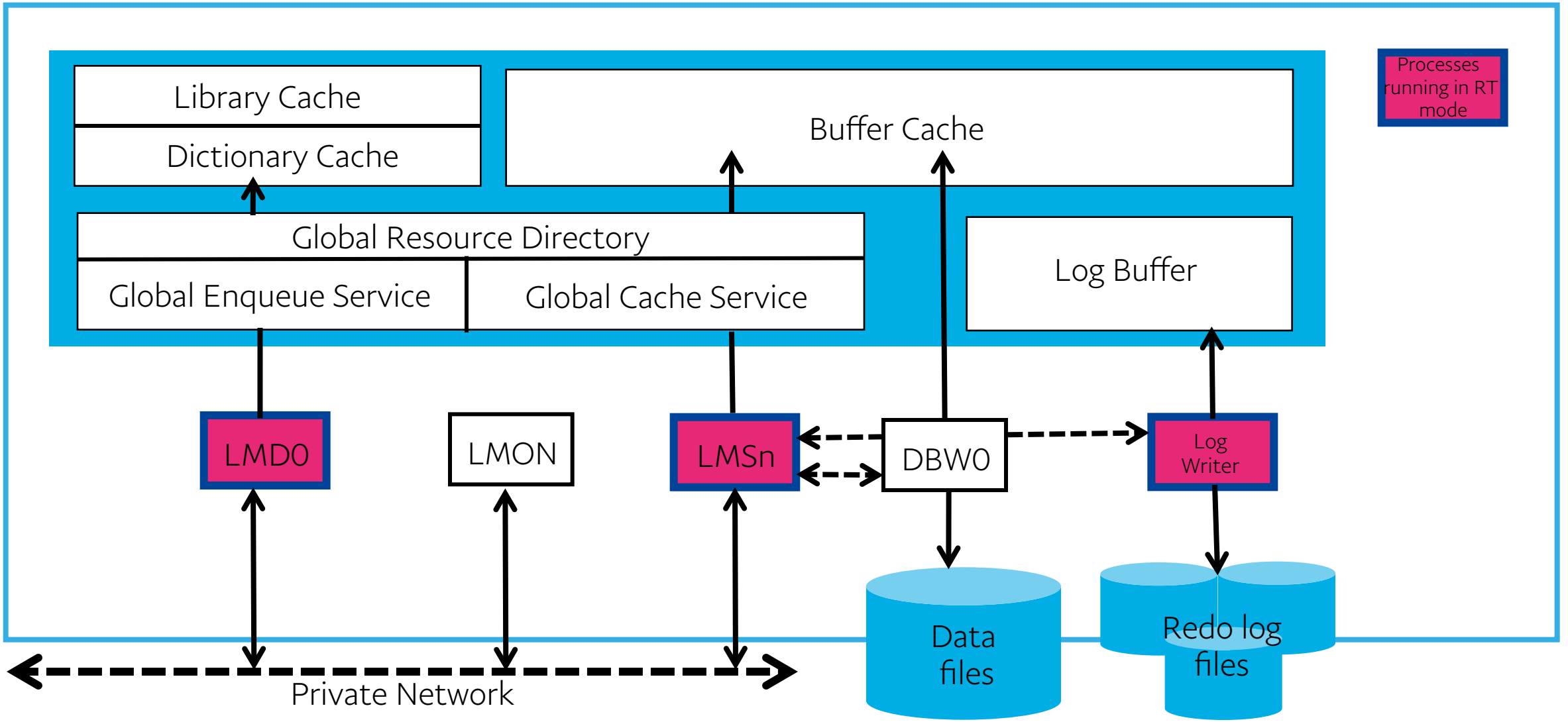
- Global Cache Services(GCS)/Parallel Cache Management(PCM)
  - Allocating, de-allocating and locking Data buffer cache resources
- Global Enqueue Services(GES)/non-Parallel Cache Management(non-PCM)
  - Dictionary cache and Library cache resources

RAC related background processes

✓ Critical Oracle Background Processes using private interconnect

- LMSn (Global Cache Service Process)
  - Handles cache fusion(GCS) (Block transfers, messages)
  - Maintains records of each data file and block open in a cache in GRD (Global Resource Directory)
  - Controls flow of messages of remote instance(s)
  - Tracks global data block access
- LCK0 (Instance Enqueue Background Process)
  - Handles requests related to non-fusion resources such as library and row cache requests
- LMD0 (Global Enqueue Service Daemon)
  - Manages global enqueues
  - Detection of deadlocks
- LMON (Global Enqueue Service Monitor)
  - Maintenance instance membership
  - Reconfiguration of GCS/GES during recovery from instance crash or startup
- LMHB (Global Cache/Enqueue Service Heartbeat Monitor)
  - Monitors LMON, LMD and LMSn Processes

SGA in RAC Database



Parameters Influencing RAC behavior

1. Always run critical background processes in RT priority
   - _lm_lms_priority_dynamic = false
   - _high_priority_processes = 'LMS*|VKTM|LMD*|LGWR'
2. Disable Undo DRM during instance start after crash
   - _gc_undo_affinity = false
3. Disable DRM to avoid unacceptable and unpredictable freezes
   - _gc_policy_time = 0
4. Minimize reconfiguration time for bigger SGA
   - _lm_tickets
   - gcs_server_processes
5. Monitoring related critical background parameters
   - Heartbeat ping to local processes
     - _lm_rcvr_hang_check_frequency
     - _lm_rcvr_hang_allow_time
     - _lm_rcvr_hang_kill
   - Heartbeat ping to peer process on remote instances
     - _lm_idle_connection_check
     - _lm_idle_connection_check_interval
     - _lm_idle_connection_kill
6. Fairness and light work rule (_fairness_threshold)

Wait events in RAC Database

1. gc cr/current grant 2-way (message wait)
   - Block does not exist in cache and LMS grants to FG process to read from disk
2. gc cr/current block 2-way/3-way (block wait)
   - Requested for read and write operations
   - Every execution triggers this since SCN advances
3. gc cr/current block congested (block congestion waits)
   - LMS didn't process within 1 ms due to CPU shortage or scheduling delays
4. gc cr/current block busy (block concurrency/contention waits)
   - Indicates high concurrency
   - LMS needs to perform additional work to prepare block in requested mode
5. gcs log flush sync
   - Before sending reconstructed CR/CURR block, LMS request LGWR to flush redo vectors
6. gc cr failure/gc cr retry
   - Inefficiencies with interconnect or invalid block request or checksum error
7. gc cr/current block corrupt/lost
   - Dropped packets due buffer overflow
   - Misconfigured interconnect

Hardware and OS support

✓ Supported networks

- Ethernet
- Ethernet with Jumbo frames
- Infiniband

✓ Network protocols

- UDP
- RDS

✓ Redundancy

- Oracle Clusterware HAIP
- In-built on HCA (in the case of Infiniband)
- Bonding on Linux
- Bonding such as IPMP on Solaris

InfiniBand

- High speed communication link
- Built in availability and load balance features
- Port failover on dual-port HCA(Host Channel Adapter)
- Onboard processor
- Supports both UDP and RDS protocols
- Integrated with Zero-copy mechanism and RDMA (Remote Direct Memory Access)
- Provides higher bandwidth(40Gb/s) and throughput(Network PPS)
- Ultra low latency(less than 80μs) and high-efficiency
- Fabric consolidation for cluster and storage
- Oracle Exadata uses for both interconnect and storage
- Failure of IB Card makes database dysfunctional

```
18: ib0.f004: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 7000 qdisc pfifo_fast state UP qlen 1024
    link/infiniband 80:00:00:4a:fe:80:00:00:00:00:00:00:00:10:e0:00:01:91:62:d9 brd 00:ff:ff:ff:ff:12:40:1b:f0:04:00:00:00:00:00:00:ff:ff:ff:ff
    inet 192.168.1.3/24 brd 192.168.1.255 scope global ib0.f004
    inet 169.254.125.213/17 brd 169.254.127.255 scope global ib0.f004:1
    inet6 fe80::210:e000:191:62d9/64 scope link
       valid_lft forever preferred_lft forever
19: ib1.f004: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 7000 qdisc pfifo_fast state UP qlen 1024
    link/infiniband 80:00:00:4b:fe:80:00:00:00:00:00:00:00:10:e0:00:01:91:62:da brd 00:ff:ff:ff:ff:12:40:1b:f0:04:00:00:00:00:00:00:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global ib1.f004
    inet 169.254.193.14/17 brd 169.254.255.255 scope global ib1.f004:1
    inet6 fe80::210:e000:191:62da/64 scope link
       valid_lft forever preferred_lft forever
```

UDP vs RDS

✓ UDP

- Implemented reliability in USER mode(acking/windowing/fragmenting/re-ordering)
- Kernel consumes higher CPU cycles
- Results in retransmits and lost datagrams under heavy CPU utilization
- Requires all memory to be pre-registered

✓ RDS

- Ultra low latency, highly reliable and high bandwidth IPC protocol
- Driver controls with reliable delivery rather than OS kernel
- Compatible to existing IPC models using in Oracle RAC
- Runs on Infiniband
- Unlike UDP, consumes very low system CPU cycles
- Supports up to 1 MB Datagram Payload

Monitoring

✓ Capacity/Performance measures and metrics to track in Database

1. Estimate Global Cache traffic flowing in/out a given node from AWR

- Messages are typically 200 bytes in size or less while CR/Curr blocks are in 8k(Same as DB_BLOCK_SIZE) in size
- Goal is to keep network Packets/sec under 70% of estimate throughput of interconnect device
- DBA_HIST_SYSSTAT provides data related to all GC wait events, block/message transfers
- DBA_HIST_IC_DEVICE_STATS provides stats like packets received/transmitted/dropped for each interface
- DBA_HIST_IC_CLIENT_STATS provides usage of interconnect by area(IPQ, DLM and Cache)

Begin SnapID	Date and Time	CR Blocks - Served	CR Blocks - Recd	CURR Blocks - Served	CURR Blocks - Recd	GCS Messages - Sent	GCS Messages - Recd	GES Messages - Sent	GES Messages - Recd	Estd. Traffic
005899	25-OCT-16_01:45	497,271	3,403,272	1,099,257	3,457,773	15,304,842	0	170,851	0	78,537
005900	25-OCT-16_02:00	523,275	3,629,648	1,140,903	3,587,882	16,162,008	0	163,385	0	82,036
005901	25-OCT-16_02:15	579,853	4,246,162	1,189,130	4,361,968	18,046,899	0	169,351	0	96,087
005902	25-OCT-16_02:30	543,690	3,793,673	1,201,856	3,799,555	16,890,334	0	163,888	0	86,424
005903	25-OCT-16_02:45	542,762	3,795,403	1,200,702	3,809,292	16,685,774	0	182,148	0	86,659
005904	25-OCT-16_03:00	566,251	3,785,220	1,224,125	3,881,049	16,922,233	0	179,242	0	87,382
005905	25-OCT-16_03:15	618,169	4,487,084	1,339,603	4,946,569	19,587,570	0	171,978	0	105,428
005906	25-OCT-16_03:30	595,135	4,001,571	1,311,175	4,172,392	18,080,367	0	205,834	0	93,467
005907	25-OCT-16_03:45	599,172	4,012,597	1,296,741	4,028,561	18,228,523	0	170,961	0	92,016
005908	25-OCT-16_04:00	575,364	3,954,142	1,314,352	4,133,448	17,829,586	0	175,135	0	94,377

Monitoring

✓ Capacity/Performance measures and metrics to track in Database

2. Estimate Global Cache traffic flowing in/out a given node from AWR

- Goal is to keep avg wait time for GC * grant wait events below 0.5 ms.
- Goal is to keep avg wait time for GC * block transfer wait events below 1 ms

	Wait		Event		Wait Time			Summary Avg Wait Time (ms)				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt
*		DB CPU	N/A	N/A	1,599,417.54	N/A	65.82					4
	Applicatio	enq: TX - row lock contention	1,466,368	1.5	311,134.98	212.2	20.31	212.33	210.70	213.97	2.31	2
	User I/O	db file sequential read	630,685,513	0.0	275,374.03	0.4	11.33	0.44	0.43	0.44	0.00	4
	Cluster	gc current block 2-way	1.064523E+09	0.0	157,796.05	0.1	6.49	0.15	0.15	0.15	0.00	4
	Cluster	gc current block 3-way	526,151,094	0.0	118,935.87	0.2	4.89	0.23	0.22	0.23	0.00	4
	Cluster	gc cr grant 2-way	426,768,530	0.0	52,460.22	0.1	2.16	0.12	0.12	0.12	0.00	4
	System I/O	log file parallel write	218,957,624	0.0	37,825.98	0.2	1.56	0.18	0.15	0.22	0.03	4
	Commit	log file sync	96,775,172	0.0	36,217.24	0.4	1.49	0.35	0.29	0.39	0.04	4
	Cluster	gc cr block busy	47,301,703	0.0	31,469.79	0.7	1.30	0.67	0.65	0.67	0.01	4
	System I/O	db file parallel write	181,716,855	0.0	27,260.16	0.2	1.12	0.15	0.15	0.16	0.00	4

		% of Waits						
Event	Total Waits	<1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<=1s	>1s
gc buffer busy acquire	4789	97.0	.9	.5	.5	.3	.2	.7	
gc buffer busy release	286	77.3	9.1	2.4	1.4	2.1	6.3	1.4	
gc cr block 2-way	191.9	99.8	.1	.1	.0	.0	.0	.0	
gc cr block 3-way	215.8	99.7	.1	.1	.0	.0	.0	.0	
gc cr block busy	62.6K	98.8	1.0	.1	.1	.0	.0	.0	
gc cr block congested	5254	99.6	.0	.2	.1	.0		.1	
gc cr failure	490	99.8	.2						
gc cr grant 2-way	888.7	99.9	.1	.0	.0	.0	.0	.0	
gc cr grant congested	7112	99.5	.1	.1	.1	.1	.1	.1	
gc cr multi block request	4	100.0							
gc current block 2-way	2185.	99.9	.1	.0	.0	.0	.0	.0	
g current block 3-way	1164.	99.7	.1	.1	.1	.0	.0	.0	
gc current block busy	44.1K	96.9	2.7	.2	.1	.0	.0	.0	
gc current block congested	36.8K	99.4	.1	.1	.1	.1	.0	.1	
gc current grant 2-way	400.4	99.9	.0	.0	.0	.0	.0	.0	
gc current grant busy	124.9	99.5	.2	.1	.1	.0	.0	.1	
gc current grant congested	2995	99.6	.0	.1	.1	.0	.0	.1	
gc current multi block req	1015	99.5	.2	.1	.2				
gc current retry	69	97.1	1.4	1.4					
gc current split	104	77.9	4.8	7.7	4.8	4.8			

Monitoring

✓ Performance monitoring from OS

1. OSWatcher

   Add “node:STORAGE” in /opt/oracle.cellos/image.id to collect IB data when using RDS

2. OS Commands
   - nmon utility (AIX)
   - netstat -i -I ibd1 -P udp 1 (Solaris, AIX)

```
    input   ibd1      output        input   (Total)    output
packets errs  packets errs  colls  packets errs  packets errs  colls
10621   0     8981    0     0      48977   0     38061   0     0
10678   0     8979    0     0      46569   0     34689   0     0
10531   0     8892    0     0      46015   0     34066   0     0
8592    0     7104    0     0      39050   0     28561   0     0
9430    0     7609    0     0      41647   0     29762   0     0
8556    0     7274    0     0      38055   0     28249   0     0
```

   - collectl –s x

```
#<----------InfiniBand---------->
#    KBIn   PktIn    KBOut  PktOut Errs
     1732   16025     4432   16027    0
     2547   21107     6479   20903    0
      794    8039     2791    8271    0
     2200   21627     6744   21643    0
     1341   13379     4136   13389    0
     1804   17755     5361   17754    0
     2309   22880     7373   22892    0
     1883   17861     5149   17860    0
```

Troubleshooting

✓ GIPCD log file

1. css uses UDP to check network heartbeat
2. Rank below 99 indicates some trouble with private network

```
2016-11-05 21:12:19.081: [GIPCDMON][1677719296] gipcdMonitorSaveInfMetrics: inf[ 0]  ib0.8004          - rank   99, avgms 0.000001 [ 31 / 29 / 29 ]
2016-11-05 21:12:19.081: [GIPCDMON][1677719296] gipcdMonitorSaveInfMetrics: inf[ 1]  ib2.8004          - rank   99, avgms 1.000000 [ 32 / 30 / 30 ]
2016-11-05 21:12:19.081: [GIPCDMON][1677719296] gipcdMonitorSaveInfMetrics: inf[ 2]  ib1.8004          - rank   99, avgms 0.645161 [ 31 / 31 / 31 ]
2016-11-05 21:12:19.081: [GIPCDMON][1677719296] gipcdMonitorSaveInfMetrics: inf[ 3]  ib3.8004          - rank   99, avgms 1.034483 [ 31 / 29 / 29 ]
```

✓ CSSD log files

1. cssd/gipcd establishes communication between nodes when node joins
2. Starting 11.2.0.2, multicast communication is MUST

```
oracle@testppdb1 > nm $GRID_HOME/lib/libskgxp11.so | grep skgxp_rds_enabled
00000000000e5754 r skgxp_rds_enabled
2016-10-12 18:12:11.940: [GIPCHTHR][8] gipchaWorkerCreateInterface: created remote bootstrap multicast interface for node 'testppdb1', haName 'CSS_PRM-test-cls1', inf 'mcast://224.0.0.251:42424/192.168.2
.2:45979'
2016-10-12 18:12:11.940: [GIPCHTHR][8] gipchaWorkerCreateInterface: created remote bootstrap multicast interface for node 'testppdb1', haName 'CSS_PRM-test-cls1', inf 'mcast://230.0.1.0:42424/192.168.2.2
:45979'
2016-10-12 18:12:11.941: [GIPCHTHR][8] gipchaWorkerCreateInterface: created remote bootstrap broadcast interface for node 'testppdb1', haName 'CSS_PRM-test-cls1', inf 'udp://192.168.2.255:42424'
2016-10-12 18:12:41.970: [    CSSD][26]clssnmconnect: connecting to addr gipcha://testppdb1:nm2_PRM-test-cls1
2016-10-12 18:12:41.972: [GIPCHGEN][9] gipchaNodeDelete: performing final delete of node 1029f6990 { host 'testppdb1', haName 'CSS_PRM-test-cls1', srcLuid abde5596-6a482a44, dstLuid 00000000-00000000 num
Inf 0, contigSeq 0, lastAck 0, lastValidAck 0, sendSeq [1 : 1], createTime 69505907, sentRegister 1, localMonitor 1, flags 0x1e0 }
2016-10-12 18:12:41.973: [    CSSD][26]clssscConnect: endp a4da - cookie 10118f590 - addr gipcha://testppdb1:nm2_PRM-test-cls1
2016-10-12 18:12:41.973: [    CSSD][26]clssnmconnect: connecting to node(1), endp(a4da), flags 0x10002
2016-10-12 18:12:41.973: [GIPCHGEN][8] gipchaNodeCreate: adding new node 101923690 { host 'testppdb1', haName 'CSS_PRM-test-cls1', srcLuid abde5596-6c0047d3, dstLuid 00000000-00000000 numInf 0, contigSeq
 0, lastAck 0, lastValidAck 0, sendSeq [0 : 0], createTime 69536938, sentRegister 0, localMonitor 0, flags 0x0 }
2016-10-12 18:12:41.973: [GIPCHALO][8] gipchaLowerSend: deffering startup of hdr 1029e9758 { len 232, seq 0, type gipchaHdrTypeSend (1), lastSeq 0, lastAck 0, minAck 0, flags 0x0, srcLuid 00000000-000000
00, dstLuid 00000000-00000000, msgId 0 }, node 101923690 { host 'testppdb1', haName 'CSS_PRM-test-cls1', srcLuid abde5596-6c0047d3, dstLuid 00000000-00000000 numInf 0, contigSeq 0, lastAck 0, lastValidAc
k 0, sendSeq [0 : 0], createTime 69536938, sentRegister 0, localMonitor 0, flags 0x0 }
2016-10-12 18:12:41.973: [GIPCHALO][8] gipchaLowerProcessNode: DEBUG node 101923690 { host 'testppdb1', haName 'CSS_PRM-test-cls1', srcLuid abde5596-6c0047d3, dstLuid 00000000-00000000 numInf 0, contigSe
q 0, lastAck 0, lastValidAck 0, sendSeq [1 : 1], createTime 69536938, sentRegister 0, localMonitor 1, flags 0x0 } now 69536938 diff 0 connectTime 30000
2016-10-12 18:12:41.973: [GIPCHALO][8] gipchaLowerSendEstablish: sending establish message for node '101923690 { host 'testppdb1', haName 'CSS_PRM-test-cls1', srcLuid abde5596-6c0047d3, dstLuid 00000000-
00000000 numInf 0, contigSeq 0, lastAck 0, lastValidAck 0, sendSeq [1 : 1], createTime 69536938, sentRegister 0, localMonitor 1, flags 0x4 }'
2016-10-12 18:12:41.973: [GIPCHALO][8] gipchaLowerProcessNode: no valid interfaces found to node for 69536938 ms, node 101923690 { host 'testppdb1', haName 'CSS_PRM-test-cls1', srcLuid abde5596-6c0047d3,
 dstLuid 00000000-00000000 numInf 0, contigSeq 0, lastAck 0, lastValidAck 0, sendSeq [1 : 1], createTime 69536938, sentRegister 0, localMonitor 1, flags 0x4 }
2016-10-12 18:12:42.128: [    CSSD][22]clssgmWaitOnEventValue: after CmInfo State  val 3, eval 1 waited 0
2016-10-12 18:12:42.645: [    CSSD][24]clssnmSendingThread: sending join msg to all nodes
2016-10-12 18:12:42.645: [    CSSD][24]clssnmSendingThread: sent 5 join msgs to all nodes
2016-10-12 18:12:42.971: [    CSSD][18]clssnmvDHBValidateNcopy: node 1, testppdb1, has a disk HB, but no network HB, DHB has rcfg 331472367, wrtcnt, 43431291, LATS 69537936, lastSeqNo 43431290, uniquenes
s 1476310616, timestamp 1476321161/788020146
```

Troubleshooting

✓ OSWatcher

1. oswnetstat
   - Received packets
   - Transmitted packets
   - Dropped packets
2. osw_ib_diagnostics
   - Interface port status
   - processor utilization
3. osw_rds_diagnostics
   - IB connections
   - RDS connections
   - RDS sockets
   - Checks remote node reachable over RDS
   - Various RDS counters
     - send_queue_full
     - cong_send_error
     - send_delayed_retry
     - ib_tx_stalled
     - ib_rx_total_frags

Putting it all together

- GCS/GES drives private network workload
- Slow private network impacts all activities in RAC cluster
- Redundancy is MUST for cluster to function without any disruptions
- Capacity analysis and Monitoring is essential to stay ahead of problem
- RDS on Infiniband to achieve ultra low latency and high throughput
- Always run critical background processes in RT priority
- Stable interconnect is the key for stable cluster performance
- If application doesn't scale well in single instance won't scale well in RAC

Questions?