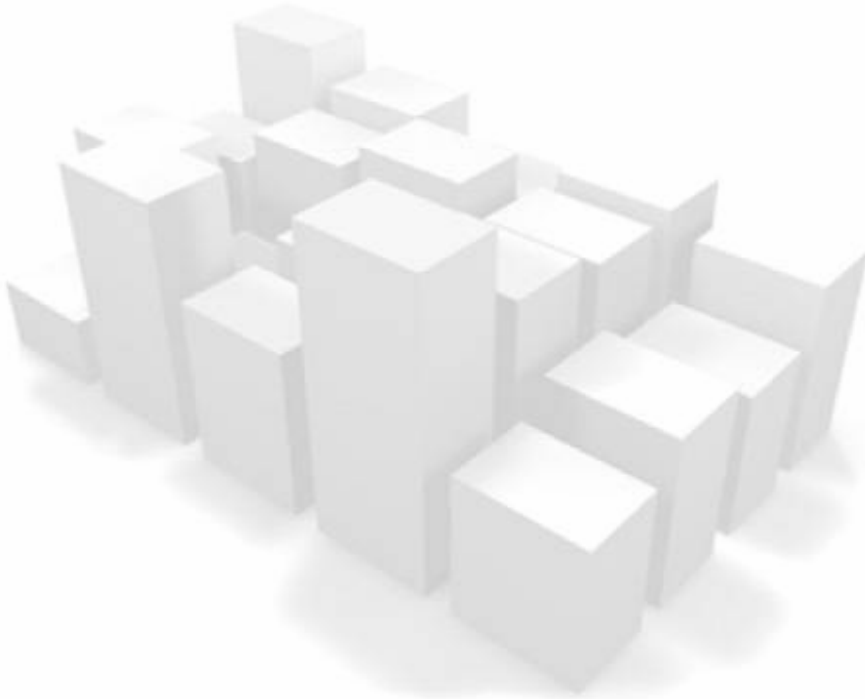


Global Cache Waits
Technical White Paper
José Valerio
Jun 2010



Disclaimer

The following is intended for information purposes only. The author of this White Paper does not make guaranties and accepts no responsibility for the use of this information. Use of any supplied concepts or definitions constitutes acceptance and understanding of these disclaimers.

Content

INTRODUCTION.....	2
About this White Paper.....	2
GLOBAL CACHE SERVICE – (GCS).....	2
GCS - Accessing Block.....	3
GCS - Block Access Cost.....	4
GCS - Block Access Latency	4
GCS - Block modes	4
GCS - Response Time	5
GCS - Monitoring.....	5
Getting the GCS Hit Ratio.....	5
Getting the GCS Hit Ratio.....	6
Getting Blocks involved in busy workloads (Hotblocks)	6
AWR – Global Cache Efficiency Percentages	7
AWR – Messaging Statistics	7
AWR – Messaging Traffic	8
AWR – Network Traffic	8
WAIT EVENTS – I	9
gc current block 2-way – Fig 1	9
gc current block 3-way – Fig 2	9
gc cr block 2-way.....	9
gc cr block 3-way.....	9
WAIT EVENTS - II	11
GC cr/current block congested	11
GC cr/current block busy	11
GC current grant busy	11
GC cr/current block request	11
CONTENTION TYPES.....	11
The block-oriented.....	11
The message-oriented	12
The contention-oriented.....	12
The load-oriented	12
Best Practices	13
General.....	13
Network	13
Hardware	13
Monitoring	14
Storage	14
Conclusions	14
Glossary.....	15
About the author - Summary	16
About Technical Reviewers.....	16
Acknowledgements.....	16

INTRODUCTION

About this White Paper

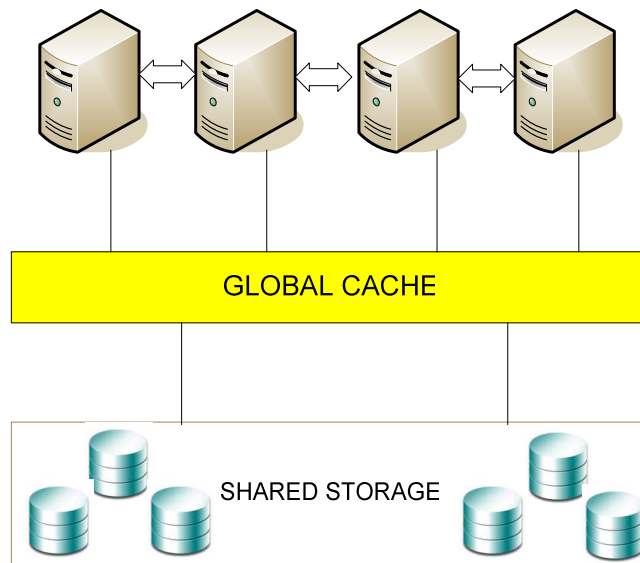
Basically this paper provides good understanding in how nodes communicate between them; this is a must if working in performance. As with single-instance it is very important to avoid disk I/O whenever possible primarily by keeping frequently accessed data in memory. RAC configuration is similar; the data might be in the memory of one of the other instances. Therefore, RAC uses interconnect to request the required data from another instance that has it in memory, rather than by reading it from disk. Each request across interconnect is referred to as a Global Cache request. This document provides all the necessary tools to assist database administrators in understanding the Real Applications waits in the Global Cache.

Who should read this white paper?

This white paper is intended to be accessible to those who are not relatively new to the Real Applications Cluster performance. Familiarity with advanced Oracle concepts and SQL language is assumed.

GLOBAL CACHE SERVICE – (GCS)

Global Cache Service (GCS) is the main component of Oracle Cache Fusion technology. This is represented by background process LMS. The main function of GCS is to track the status and location of data blocks. Status of data block is the mode and role of data block.. GCS is also responsible for block transfer between the grid the instances. GCS is the mechanism that guarantees the data integrity through global access levels use. GCS maintains the block modes for data blocks in the global role. It is also responsible for block transfers between the instances. Upon a request from an Instance, GCS organizes the block shipping and deliver the right lock mode conversions.



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

GCS - Accessing Block

The effect of accessing blocks in the global cache and keep coherency is mapped by "The Global Cache Service" statistics for current and cr blocks, for example, gc current blocks received, gc cr blocks received, and so on.

GCS - Block Access Cost

The process of block retrieval generates:

- Message propagation delay
- Inter process CPU
- Operating System Scheduling
- Block Server Load

To calculate the costs use the formula below.

Block Access Cost = message propagation delay + IPC CPU + Operating System Scheduling + Block Server Load

Note: There is fifth factor called interconnect stability, get in play when there are switch problems. At this time it is not clear how did not provide how to calculate or estimate it exactly.

GCS - Block Access Latency

The following factors impacts the processing time:

- Operating System
- CPU load on other nodes
- Oracle processing time
- Available Interconnect network throughput

GCS - Block modes

A block can exist in multiple buffer caches and can be help by multiples instances in different modes depending on whether the blocks is being read or updated by the instance.

Resource Modes

<i>Null</i>	<i>N</i>	<i>No access rights</i>
<i>Shared</i>	<i>S</i>	<i>Share resources can be read by multiple databases instances but can't be updated by any instance</i>
<i>Exclusive</i>	<i>X</i>	<i>An instance holding a block in exclusive mode can modify a block. Only one instance can hold a block in exclusive mode at a time.</i>

Resource Roles

<i>Local</i>	<i>When a data block is first read into the instance from the disk it has a local role. Meaning that only 1 copy of data block exists in the cache. No other instance cache has a copy of this block.</i>
<i>Global</i>	<i>Global role indicates that multiple copy of data block exists in clustered instance. A user connected to one of the instance request for a data block. This data block is read from disk into an instance. The role granted is local. If another instance request for</i>

same block this block will get copied to the requesting instance and the role becomes global can not be updated by any instance.

This role and mode information is maintained in GRD (Global Resource Directory) by GCS (Global Cache Service).

GCS - Response Time

Response time for cache fusion transfers is determined by the messaging and processing times imposed by the physical interconnect components, the IPC protocol and the GCS protocol. It is not affected by disk I/O factors other than occasional log writes. The cache fusion protocol does not require I/O to data files in order to guarantee cache coherency and Oracle RAC inherently does not cause any more I/O to disk than a non-clustered instance.

GCS - Monitoring

Getting the GCS Hit Ratio

```
--
-- Use SQL*Plus and connect AS SYSDBA
-- Getting the Global Cache Hit Ratio
--
SELECT
  inst_id "Instance #",
  (VALUE+B.VALUE+C.VALUE+D.VALUE)/(E.VALUE+F.VALUE) "GCS HIT RATIO"
FROM
  GV$SYSSTAT A,
  GV$SYSSTAT B,
  GV$SYSSTAT C,
  GV$SYSSTAT D,
  GV$SYSSTAT E,
  GV$SYSSTAT F
WHERE
  NAME='gc gets'
  AND B.NAME='gc converts'
  AND C.NAME='gc cr blocks received'
  AND D.NAME='gc current blocks received'
  AND E.NAME='consistent gets'
  AND F.NAME='db block gets'
  AND B.INST_ID=A.INST_ID
  AND C.INST_ID=A.INST_ID
  AND D.INST_ID=A.INST_ID
  AND E.INST_ID=A.INST_ID
  AND F.INST_ID=A.INST_ID;
```

Instance #	GCS CACHE HIT RATIO
2	.02803656
2	.01279997

Getting the GCS Hit Ratio

```
--
-- Use SQL*Plus and connect AS SYSDBA
-- Getting Block Transfer Ratio
--
SELECT
  INST_ID "Instance #",
  VALUE/B.VALUE "BLOCK TRN RATIO"
FROM
  GV$SYSSTAT A, GV$SYSSTAT B
WHERE
  NAME='gc defers'
  AND B.NAME='gc current blocks served'
  AND B.INST_ID=A.INST_ID;
/
```

Instance #	BLOCK TRN RATIO
2	.052600105
2	.078004479

Note: A value over .3 it's a high value

Getting Blocks involved in busy workloads (Hotblocks)

```
--
-- Use SQL*Plus and connect AS SYSDBA
-- Getting Hot Blocks
--
SELECT INST_ID "Instance #",
NAME,
KIND,
sum(FORCED_READS) "Forced Reads",
sum(FORCED_WRITES) "Forced Writes"
FROM GV$CACHE_TRANSFER
WHERE owner#!=0
GROUP BY INST_ID,NAME,KIND
ORDER BY 1,4 desc,2
/
```

Instance	NAME	KIND	Forced Reads	Forced Writes
1	TT_PROD_IND	INDEX	408	0
1	TTQUEUE	TABLE	44	0
2	TTQUEUE	TABLE	573	0
2	TT_PROD_IND	INDEX	321	0
2	AQ\$_QUEUE_TABLES	TABLE	9	0

Tip: GV\$BH shows the buffer header information for all instances. That is, if you run a multi-instance database, then GV\$BH might be very of great help in order to find the block numbers of blocks experiencing a lot of FORCE_READS and FORCED_WRITES. Then you can find the rows in those blocks.

V\$BH Status values

Free	Resouce Mode	Details
Freee		Buffer is not currently in use
CR	NULL	Consistent read (read only)
SCUR	S	Shared Current Block (read only)
XCUR	X	Exclusive current block (able to modify)
PI	NULL	Past Image (read only)

AWR – Global Cache Efficiency Percentages

```
Global Cache Efficiency Percentages
  • Data blocks retrieved from local cache or remote instance

      Global Cache Efficiency Percentages (Target local+remote 100%)
      ~~~~~
      Buffer access - local cache %: 99.12 <- OK
      Buffer access - remote cache %: 0.75
      Buffer access - disk %: 0.13
```

AWR – Messaging Statistics

The standard average for messages sent should be less than 1 millisecond.

```
Global Cache and Enqueue Services - Messaging Statistics
      ~~~~~
      Avg message sent queue time (ms): 0.4
      Avg message sent queue time on ksxp (ms): 0.2
      Avg message received queue time (ms): 0.0
      Avg GCS message process time (ms): 0.0
      Avg GES message process time (ms): 0.0

      % of direct sent messages: 48.44
      % of indirect sent messages: 23.34
      % of flow controlled messages: 22.61
```

AWR – Messaging Traffic

Global Cache Load Profile

~~~~~

|                               | Per Second | Per Transaction |
|-------------------------------|------------|-----------------|
|                               | -----      | -----           |
| Global Cache blocks received: | 4.30       | 3.65            |
| Global Cache blocks served:   | 23.44      | 19.90           |
| GCS/GES messages received:    | 133.03     | 112.96          |
| GCS/GES messages sent:        | 78.61      | 66.75           |
| DBWR Fusion writes:           | 0.11       | 0.10            |
| Est Interconnect traffic (KB) | 263.20     |                 |

## AWR – Network Traffic

The next formula calculates network traffic use.

### Example

*Network traffic received = Global Cache blocks received \* DB block size =  
4.3 \* 8192 = .01 Mb/sec*

*Network traffic generated = Global Cache blocks served \* DB block size =  
23.44 \* 8192 = .20 Mb/sec*

## WAIT EVENTS – I

RAC wait events are grouped in a category called “Cluster Wait Class” characterized as Current or CR.

- Current - blocks read into memory for the first time
- Consistent Read (CR) - denotes block for read access

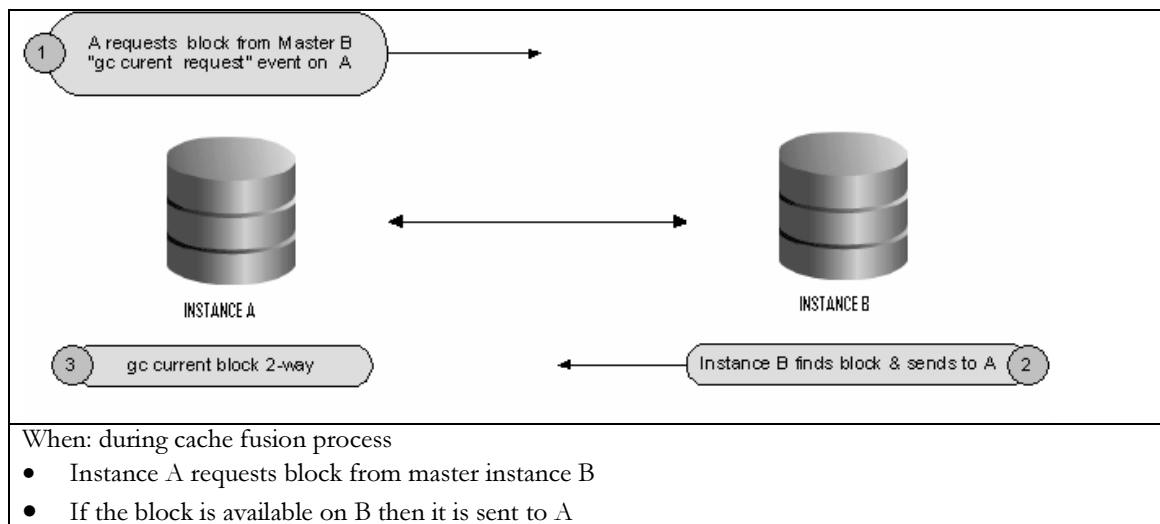
The following wait events shows that the remotely cached blocks were shipped to the local instance without having been busy, pinned or requiring a log flush:

***gc current block 2-way – Fig 1***

***gc current block 3-way – Fig 2***

***gc cr block 2-way***

***gc cr block 3-way***



*Fig. 1*

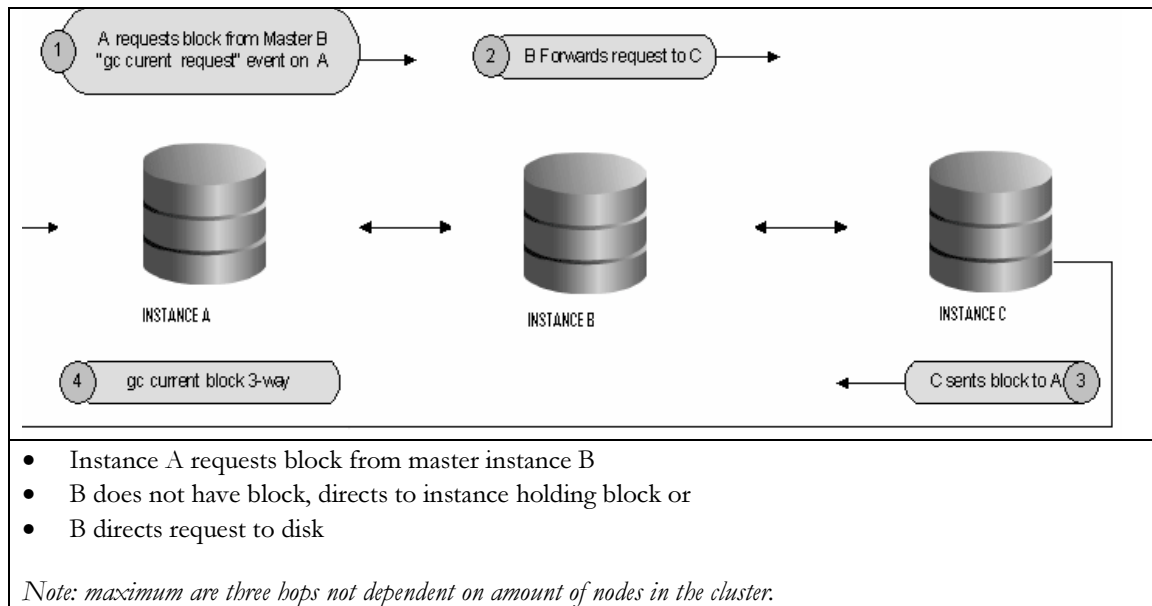


Fig. 2

The object statistics for gc current blocks received and gc cr blocks received enable quick identification of the indexes and tables which are shared by the active instances. You can create an ADDM analysis, in most cases, the analysis result will point you to the SQL statements and database objects that could be affected by inter-instance contention.

Any increases in the average wait times for the events mentioned earlier could be caused by the following issues:

- High load: CPU shortages, long run queues, scheduling delays, hung processes
- Misconfiguration: using public instead of private interconnect for message and block traffic
- Bad SQL execution

If the average wait times are acceptable and no interconnect or load issues can be diagnosed, then the accumulated wait time can be caused by SQL statements which need to be tuned to minimize the number of blocks accessed.

## WAIT EVENTS – II

### **GC cr/current block congested**

- Repeated requests by foreground processes, not serviced by LMS
- Indicates LMS not able to keep up
- Queue lengths & scheduling delays in OS, can cause LMS delays

### **GC cr/current block busy**

- Delay for some reason, before block sent to requestor

### **GC current grant busy**

- Permission to access the block granted, but blocked by other requests ahead of it

### **GC cr/current block request**

- Wait time, cr or current block is being retrieved

*Tip: The number of GCS resource structures is determined by the `_gcs_resources` parameter. Number of free GCS resource structures are in `X$KJBRFX`.*

## CONTENTION TYPES

- **Block-oriented**
  - gc current block 2-way
  - gc current block 3-way
  - gc cr block 2-way
  - gc cr block 3-way
- **Message-oriented**
  - gc current grant 2-way
  - gc cr grant 2-way
- **Contention-oriented**
  - gc current block busy
  - gc cr block busy
  - gc buffer busy acquire/release
- **Load-oriented**
  - gc current block congested
  - gc cr block congested

**The block-oriented** wait event statistics indicate that a block was received as either the result of a 2-way or a 3-way message, means that, the block was sent from either the resource master requiring 1 message and 1 transfer, or was forwarded to a third node from which it was sent, requiring 2 messages and 1 block transfer.

The gc current block busy and gc cr block busy wait events points that the local instance that is making the request will not immediately receive a current or consistent read block. The

term “busy” in these events reflects that the block was delayed on a remote instance. For example, a block cannot be shipped immediately if Oracle has not yet written the redo for the block’s changes to a log file.

Nevertheless “block busy” wait events, a gc buffer busy event means that Oracle cannot immediately grant access to data that is stored in the local buffer cache. This is because a global operation on the buffer is pending and the operation has not yet completed. In other words, the buffer is busy and all other processes that are attempting to access the local buffer must wait to the process will be completed.

The appearance of gc buffer busy events means that there is block contention that comes from multiple requests for access to the local block. Oracle must queue these requests. The length of time that Oracle needs to process the queue depends on the remaining service time for the block. The service time is affected by the processing time that any network latency adds, both from the remote and local instances.

The average wait time and the total wait time should be considered when being alerted to performance issues. Usually, either interconnect or load issues or SQL execution against a large shared working set can be found to be the root cause.

**The message-oriented** wait event statistics indicate that no block was received because it was not cached in any instance. Instead a global grant was given, enabling the requesting instance to read the block from disk or modify it.

If the time consumed by these events is high, then it may be assumed that the frequently used SQL causes a lot of disk I/O (in the event of the cr grant) or that the workload inserts a huge amount of data and needs to find and format new blocks frequently (in the event of the current grant).

**The contention-oriented** wait event statistics shows up that a block was received which was pinned by a session on another node, was deferred because a change had not flushed to disk yet or due to of high concurrency, and therefore could not be shipped immediately. A buffer may also be busy locally when a session has already initiated a cache fusion operation and is waiting for its completion when another session on the same node is trying to read or modify the same data. High service times for blocks exchanged in the global cache may exacerbate the contention, which can be caused by frequent concurrent read and write accesses to the same data.

**The load-oriented** wait events indicate that a delay in processing has occurred in the GCS, which is usually caused by high load, CPU saturation and would have to be solved by additional CPUs, load-balancing, off loading processing at different times frames or a new cluster node. For the events mentioned, the wait time encompasses the entire round trip from the time a session start waiting after initiating a block request until the block arrives.

*The column CLUSTER\_WAIT\_TIME in V\$SQLAREA represents the wait time incurred by individual SQL statements for global cache events and will identify the SQL which may need to be tuned.*

## **Best Practices**

Sr. DBA is needed to successful advice and tune a RAC system, it is complex by default. Across the system life many default configuration needs changes according to your production workload.

One of the most important aspects of RAC tuning is the monitoring and tuning of the global services directory processes. The processes in the Global Service Daemon (GSD) communicate through the cluster interconnects. If the cluster interconnects do not perform properly, the entire RAC structure will compromised.

### ***General***

- Avoid serialization during the application design
- Ensure adequate resources on surviving nodes
- Benchmark cluster configuration
- Load test on single instance first
- Apply few changes at a time
- Fix bad plans, serialization and schemas
- Fix I/O issues avoiding full scans
- Reduce or eliminate the hard parsing
- Avoid the use of non ASSM segments
- Control High Rate DLM's on small cached segments.

### ***Network***

- Monitor dropped packets, timeouts, buffer overflows, transmit and receive errors

### ***Hardware***

- Redundancy – server, storage, network components
- Add HBA cards, switches, disk array controllers
- Load balance LUNs across HBA ports
- Enable hyperthreading at the OS level
- Use asynchronous I/O
- Avoid dissimilar disks within disk group

Verify Set “aio-max-size” and “aio-max-ns”

## **Monitoring**

Monitoring and tuning requires deep RAC skills and knowledge, DBA will need this needs specialized trainings.

- Use Database Control or Grid Control
- View overall system status, status of cluster, alert logs
- Monitor throughput across Interconnect
- Make decisions to add or redistribute resources
- Tune SQL with full scans plans or heavy physical access.
- V\$BH can definitely be of great interest even on non-OPS systems if you want to know which blocks of which objects are currently in your buffer cache and what's happening.
- Verify that Operating System and RAC Best Practices were applied.

## **Storage**

- RAID 1 it's the better performance choice. RAID 5 it is known to be slower for writes, because of the CPU overhead that is required for each write.
- Measure the I/O frequently, be sure your storage can handle the database requests.

## **Conclusions**

- ✓ The integration between development group and database administration is essential to design robust applications.
- ✓ Applications persistence should be carefully considered.
- ✓ Although most applications will run on RAC without modifications many changes could be applied to get a better performance.
- ✓ RAC Performance depends on the application.
- ✓ The Global Cache Service is highly dependent on the data blocks usage.
- ✓ Applications should be minimize in the use of blocks between the instances.
- ✓ Minimizing the interconnect traffic means maximize platform performance and scalability.
- ✓ In a database cluster environment, badly tuned SQL will not run better.
- ✓ Serializing contention makes applications less scalable

## Glossary

|                        |                                                                                                                                                                                                                                                                                                                                                                |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RAC                    | Real Application Clusters. RAC is a shared disk clustered database. Every instance in the cluster has equal access to the database's data on disk                                                                                                                                                                                                              |
| GCS                    | The Global Cache Service is the controlling process that implements Cache Fusion. It maintains the block mode for blocks in the global role. It is responsible for block transfers between instances. The Global Cache Service employs various background processes such as the Global Cache Service Processes (LMSn) and Global Enqueue Service Daemon (LMD). |
| Global Enqueue Service | The Global Enqueue Service Daemon (LMD) is the resource agent process that manages Global Enqueue Service (GES) resource requests. The LMD process also handles deadlock detection Global Enqueue Service (GES) requests. Remote resource requests are requests originating from another instance.                                                             |
| LMSn                   | Oracle process that provides inter-instance resource management                                                                                                                                                                                                                                                                                                |
| Hotblocks              | Most accessed blocks in the buffer cache                                                                                                                                                                                                                                                                                                                       |
| OPS                    | Old version of the Oracle database system designed for massively parallel processors. (Pre RAC)                                                                                                                                                                                                                                                                |
| OS                     | Operating System                                                                                                                                                                                                                                                                                                                                               |
| LUN                    | Logical Unit                                                                                                                                                                                                                                                                                                                                                   |
| Hyperthreading         | Hyper-Threading technology is a technique which enables a single CPU to act like multiple CPU's.                                                                                                                                                                                                                                                               |
| RAID                   | RAID, an acronym for redundant array of independent disks or redundant array of inexpensive disks, is a technology that provides increased storage reliability through redundancy, combining multiple low-cost, less-reliable disk drive components into a logical unit where all drives in the array are interdependent                                       |
| AWR                    | Automatic Workload Repository (Available from Oracle 10g and above). The AWR is used to collect performance statistics.                                                                                                                                                                                                                                        |
| Oracle waits           | When Oracle executes an SQL statement, it is not constantly executing. Sometimes it has to wait for a specific event to happen before it can proceed.                                                                                                                                                                                                          |

## About the author – Summary



Twelve+ years working in Oracle Consulting. Extensive training and field experience defining and implementing technologies strategies at key companies in Latin America. Advance knowledge within all RAC versions from 9.0.x to 11gR2, leading whole project expects. Technical and managerial leadership specializing in infrastructure and high availability.

**ORACLE**  
Certified Professional  
Oracle Database 11g  
Administrator

**ORACLE**  
Certified Expert  
Oracle Real Application  
Clusters 10g Administrator

Personal Blog Site: <http://jose-valerio.com.ar>  
eMail : [contact@jose-valerio.com.ar](mailto:contact@jose-valerio.com.ar),

## About Technical Reviewers

|                        |                                                                                                                                                                                          |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pablo Albeck – Oracle  | Practice Manager at Oracle Corporation, with several years of experience in performance tuning complex Systems. Paul enjoys exploring the world of databases and the process of turning. |
| Juan Carserta - Oracle | Oracle Technology Specialist – Juan was very involved in RAC performance being one of the most experienced professionals in high availability in LAD.                                    |

## Acknowledgements

This simple word paper is dedicated to all those in pursuit of extraordinary performance. I really appreciate the opinions, ideas and concepts proposed by my friends and colleagues, in fact I used many of them, so thank you very much.

Jorge Teodoro – Sr. Engineer – Technologist  
Reinaldo González – Oracle Database Specialist  
Fernando Sciacaluga – Oracle Technology Specialist  
Rosa Zahora – Oracle Database Specialist  
Marcelo Ochoa – Oracle Developer Specialist – Oracle ACE  
Erik Peterson – Oracle Director - Reviewer

## Bibliography

|                                      |                                                                                                                                   |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Oracle Documentation                 | <a href="http://www.oracle.com/technology/documentation/index.html">http://www.oracle.com/technology/documentation/index.html</a> |
| Pro Oracle Database 10g RAC on Linux | Julian Dyke & Steve Shaw                                                                                                          |
| Pro Oracle Database 11g RAC on Linux | Julian Dyke, Martin Bach & Steve Shaw                                                                                             |
| Julian Dyke                          | <a href="http://www.juliandyke.com">http://www.juliandyke.com</a>                                                                 |
| Tom Kyte                             | <a href="http://asktom.oracle.com">http://asktom.oracle.com</a>                                                                   |
| Oracle Performance Survivor          | Guy Harrison                                                                                                                      |
| Personal experience in field         | <a href="http://jose-valerio.com.ar">http://jose-valerio.com.ar</a>                                                               |
| OOW 2006 RAC Performance Experts     | Michael Zoll and Barb Lundhild                                                                                                    |
| Reveal All . Public                  | Portions of this WP was inspired in this presentation.<br>Thanks                                                                  |