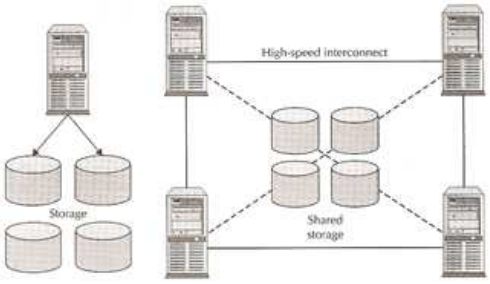
RAC Architecture

Oracle Real Application clusters allows multiple instances to access a single database, the instances will be running on multiple nodes. In an standard Oracle configuration a database can only be mounted by one instance but in a RAC environment many instances can access a single database.



Oracle's RAC is heavy dependent on a efficient, high reliable high speed private network called the interconnect, make sure when designing a RAC system that you get the best that you can afford.

The table below describes the difference of a standard oracle database (single instance) an a RAC environment

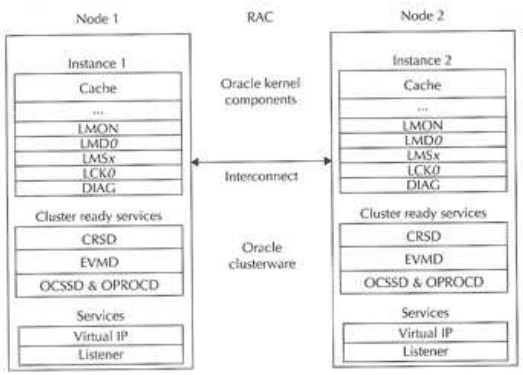
Component	Single Instance Environment	RAC Environment
SGA	Instance has its own SGA	Each instance has its own SGA
Background processes	Instance has its own set of background processes	Each instance has its own set of background processes
Datafiles	Accessed by **only** one instance	Shared by all instances (shared storage)
Control Files	Accessed by **only** one instance	Shared by all instances (shared storage)
Online Redo Logfile	Dedicated for write/read to **only** one instance	**Only** one instance can write but other instances can read during recovery and archiving. If an instance is shutdown, log switches by other instances can force the idle instance redo logs to be archived
Archived Redo Logfile	Dedicated to the instance	Private to the instance but other instances will need access to all required archive logs during media recovery
Flash Recovery Log	Accessed by **only** one instance	Shared by all instances (shared storage)
Alert Log and Trace Files	Dedicated to the instance	Private to each instance, other instances never read or write to those files.
ORACLE_HOME	Multiple instances on the same server accessing different databases ca use the same executable files	Same as single instance plus can be placed on shared file system allowing a common ORACLE_HOME for all instances in a RAC environment.

RAC Components

The major components of a Oracle RAC system are

- Shared disk system
- Oracle Clusterware
- Cluster Interconnects
- Oracle Kernel Components

The below diagram describes the basic architecture of the Oracle RAC environment



Here are a list of processes running on a freshly installed RAC

```
[root@hpevo1 ~]# ps -ef|egrep -i 'oracle|u01|crs|css'
root      3322     1  0 10:41 ?        00:00:00 /bin/su -l oracle -c sh -c 'ulimit -c unlimited; cd /u01/app/crs/log/hpevo1/evmd; exec /u01/app/crs/b
h
root      3323     1  0 10:41 ?        00:00:00 /bin/sh /etc/init.d/init.cssd fatal
root      3324     1  1 10:41 ?        00:00:02 /u01/app/crs/bin/crsd.bin reboot
root      3703  3323  0 10:42 ?        00:00:00 /bin/sh /etc/init.d/init.cssd daemon
oracle    3718  3322  0 10:42 ?        00:00:00 /u01/app/crs/bin/evmd.bin
root      3835  3703  0 10:42 ?        00:00:00 /bin/su -l oracle -c /bin/sh -c 'ulimit -c unlimited; cd /u01/app/crs/log/hpevo1/cssd;  /u01/app/crs/b
d  || exit $?'
oracle    3836  3835  0 10:42 ?        00:00:00 /bin/sh -c ulimit -c unlimited; cd /u01/app/crs/log/hpevo1/cssd;  /u01/app/crs/bin/ocssd  || exit $?
oracle    3858  3836  0 10:42 ?        00:00:00 /u01/app/crs/bin/ocssd.bin
oracle    4085  3718  0 10:42 ?        00:00:00 /u01/app/crs/bin/evmlogger.bin -o /u01/app/crs/evm/log/evmlogger.info -l /u01/app/crs/evm/log/evmlogge
oracle    4892     1  0 10:42 ?        00:00:00 asm_pmon_+ASM1
oracle    4894     1  0 10:42 ?        00:00:00 asm_diag_+ASM1
oracle    4896     1  0 10:42 ?        00:00:00 asm_psp0_+ASM1
oracle    4898     1  0 10:42 ?        00:00:00 asm_lmon_+ASM1
oracle    4900     1  0 10:42 ?        00:00:00 asm_lmd0_+ASM1
oracle    4902     1  0 10:42 ?        00:00:00 asm_lms0_+ASM1
oracle    4930     1  0 10:42 ?        00:00:00 asm_mman_+ASM1
oracle    4932     1  0 10:42 ?        00:00:00 asm_dbw0_+ASM1
oracle    4941     1  0 10:42 ?        00:00:00 asm_lgwr_+ASM1
oracle    4943     1  0 10:42 ?        00:00:00 asm_ckpt_+ASM1
oracle    4945     1  0 10:42 ?        00:00:00 asm_smon_+ASM1
oracle    4947     1  0 10:42 ?        00:00:00 asm_rbal_+ASM1
oracle    4949     1  0 10:42 ?        00:00:00 asm_gmon_+ASM1
oracle    4951     1  0 10:42 ?        00:00:00 asm_lck0_+ASM1
oracle    5201     1  0 10:42 ?        00:00:00 /u01/app/oracle/product/10.2.0/db_1/bin/racgimon daemon ora.hpevo1.ASM1.asm
root      7568  3323  0 10:44 ?        00:00:00 /bin/sh /etc/init.d/init.cssd runcheck
root      7577  7568  0 10:44 ?        00:00:00 /bin/sh /etc/init.d/init.cssd runcheck
root      7579  4596  0 10:44 pts/1    00:00:00 egrep -i oracle|u01|crs|css
```

Disk architecture

With today's SAN and NAS disk storage systems, sharing storage is fairly easy and is required for a RAC environment, you can use the below storage setups

- SAN (Storage Area Networks) - generally using fibre to connect to the SAN
- NAS (Network Attached Storage) - generally using a network to connect to the NAS using either NFS, ISCSI
- JBOD - direct attached storage, the old traditional way and still used by many companies as a cheap option

All of the above solutions can offer multi-pathing to reduce SPOFs within the RAC environment, there is no reason not to configure multi-pathing as the cost is cheap when adding additional paths to the disk because most of the expense is paid when out when configuring the first path, so an additional controller card and network/fibre cables is all that is need.

The last thing to think about is how to setup the underlining disk structure this is known as a raid level, there are about 12 different raid levels that I know off, here are the most common ones

raid 0 (Striping)	A number of disks are concatenated together to give the appearance of one very large disk. Advantages Improved performance Can Create very large Volumes Disadvantages Not highly available (if one disk fails, the volume fails)
raid 1 (Mirroring)	A single disk is mirrored by another disk, if one disk fails the system is unaffected as it can use its mirror. Advantages Improved performance Highly Available (if one disk fails the mirror takes over) Disadvantages Expensive (requires double the number of disks)
raid 5	Raid stands for Redundant Array of Inexpensive Disks, the disks are striped with parity across 3 or more disks, the parity is used in the event that one of the disks fails, the data on the failed disk is reconstructed by using the parity bit. Advantages Improved performance (read only) Not expensive Disadvantages Slow write operations (caused by having to create the parity bit)

There are many other raid levels that can be used with a particular hardware environment for example EMC storage uses the RAID-S, HP storage uses Auto RAID, so check with the manufacture for the best solution that will provide you with the best performance and resilience.

Once you have you storage attached to the servers, you have three choices on how to setup the disks

- Raw Volumes - normally used for performance benefits, however they are hard to manage and backup
- Cluster FileSystem - used to hold all the Oracle datafiles can be used by windows and linux, its not used widely
- Automatic Storage Management (ASM) - Oracle choice of storage management, its a portable, dedicated and optimized cluster filesystem

I will only be discussing ASM, which i have already have a topic on called Automatic Storage Management.

Oracle Clusterware

Oracle Clusterware software is designed to run Oracle in a cluster mode, it can support you to 64 nodes, it can even be used with a vendor cluster like Sun Cluster.

The Clusterware software allows nodes to communicate with each other and forms the cluster that makes the nodes work as a single logical server. The software is run by the Cluster Ready Services (CRS) using the Oracle Cluster Registry (OCR) that records and maintains the cluster and node membership information and the voting disk which acts as a tiebreaker during communication failures. Consistent heartbeat information travels across the interconnect to the voting disk when the cluster is running.

The CRS has four components

- OPROCd - Process Monitor Daemon
- CRSd - CRS daemon, the failure of this daemon results in a node being reboot to avoid data corruption
- OCSSd - Oracle Cluster Synchronization Service Daemon (updates the registry)
- EVMd - Event Volume Manager Daemon

The OPROCd daemon provides the I/O fencing for the Oracle cluster, it uses the hangcheck timer or watchdog timer for the cluster integrity. It is locked into memory and runs as a realtime processes, failure of this daemon results in the node being rebooted. Fencing is used to protect the data, if a node were to have problems fencing presumes the worst and protects the data thus restarts the node in question, its better to be save than sorry.

The CRSd process manages resources such as starting and stopping the services and failover of the application resources, it also spawns separate processes to manage application resources. CRS manages the OCR and stores the current know state of the cluster, it requires a public, private and VIP interface in order to run. OCSSd provides synchronization services among nodes, it provides access to the node membership and enables basic cluster services, including cluster group services and locking, failure of this daemon causes the node to be rebooted to avoid *split-brain* situations.

The below functions are covered by the OCSSd

- CSS provides basic Group Services Support, it is a distributed group membership system that allows applications to coordinate activities to archive a common result.
- Group services use vendor clusterware group services when it is available.
- Lock services provide the basic cluster-wide serialization locking functions, it uses the First In, First Out (FIFO) mechanism to manage locking
- Node services uses OCR to store data and updates the information during reconfiguration, it also manages the OCR data which is static otherwise.

The last component is the Event Management Logger, which runs the EVMd process. The daemon spawns a processes called *evmlogger* and generates the events when things happen. The *evmlogger* spawns new children processes on demand and scans the callout directory to invoke callouts. Death of the EVMd daemon will not halt the instance and will be restarted.

Quick recap

CRS Process	Functionality	Failure of the Process	Run AS
OPROCd - Process Monitor	provides basic cluster integrity services	Node Restart	root
EVMd - Event Management	spawns a child process event logger and generates callouts	Daemon automatically restarted, no node restart	oracle
OCSSd - Cluster Synchronization Services	basic node membership, group services, basic locking	Node Restart	oracle
CRSd - Cluster Ready Services	resource monitoring, failover and node recovery	Daemon restarted automatically, no node restart	root

The cluster-ready services (CRS) is a new component in 10g RAC, its is installed in a separate home directory called ORACLE_CRS_HOME. It is a mandatory component but can be used with a third party cluster (Veritas, Sun Cluster), by default it manages the node membership functionality along with managing regular RAC-related resources and services

RAC uses a membership scheme, thus any node wanting to join the cluster as to become a member. RAC can evict any member that it seems as a problem, its primary concern is protecting the data. You can add and remove nodes from the cluster and the membership increases or decrease, when network problems occur membership becomes the deciding factor on which part stays as the cluster and what nodes get evicted, the use of a voting disk is used which I will talk about later.

The resource management framework manage the resources to the cluster (disks, volumes), thus you can have only have one resource management framework per resource. Multiple frameworks are not supported as it can lead to undesirable affects.

The Oracle Cluster Ready Services (CRS) uses the registry to keep the cluster configuration, it should reside on a shared storage and accessible to all nodes within the cluster. This shared storage is known as the Oracle Cluster Registry (OCR) and its a major part of the cluster, it is automatically backed up (every 4 hours) the daemons plus you can manually back it up. The OCSSd uses the OCR extensively and writes the changes to the registry

The OCR keeps details of all resources and services, it stores name and value pairs of information such as resources that are used to manage the resource equivalents by the CRS stack. Resources with the CRS stack are components that are managed by CRS and have the

information on the good/bad state and the callout scripts. The OCR is also used to supply bootstrap information ports, nodes, etc, it is a binary file.

The OCR is loaded as cache on each node, each node will update the cache then only one node is allowed to write the cache to the OCR file, the node is called the master. The Enterprise manager also uses the OCR cache, it should be at least 100MB in size. The CRS daemon will update the OCR about status of the nodes in the cluster during reconfigurations and failures.

The voting disk (or quorum disk) is shared by all nodes within the cluster, information about the cluster is constantly being written to the disk, this is know as the heartbeat. If for any reason a node cannot access the voting disk it is immediately evicted from the cluster, this protects the cluster from split-brains (the Instance Membership Recovery algorithm IMR is used to detect and resolve split-brains) as the voting disk decides what part is the really cluster. The voting disk manages the cluster membership and arbitrates the cluster ownership during communication failures between nodes. Voting is often confused with quorum the are similar but distinct, below details what each means

Voting	A vote is usually a formal expression of opinion or will in response to a proposed decision
Quorum	is defined as the number, usually a majority of members of a body, that, when assembled is legally competent to transact business

The only vote that counts is the quorum member vote, the quorum member vote defines the cluster. If a node or group of nodes cannot archive a quorum, they should not start any services because they risk conflicting with an established quorum.

The voting disk has to reside on shared storage, it is a a small file (20MB) that can be accessed by all nodes in the cluster. In Oracle 10g R1 you can have only one voting disk, but in R2 you can have upto 32 voting disks allowing you to eliminate any SPOF's.

The original Virtual IP in Oracle was Transparent Application Failover (TAF), this had limitations, this has now been replaced with *cluster VIPs*. The *cluster VIPs* will failover to working nodes if a node should fail, these public IPs are configured in DNS so that users can access them. The *cluster VIPs* are different from the cluster interconnect IP address and are only used to access the database.

The cluster interconnect is used to synchronize the resources of the RAC cluster, and also used to transfer some data from one instance to another. This interconnect should be private, highly available and fast with low latency, ideally they should be on a minimum private 1GB network. What ever hardware you are using the NIC should use multi-pathing (Linux - bonding, Solaris - IPMP). You can use crossover cables in a QA/DEV environment but it is not supported in a production environment, also crossover cables limit you to a two node cluster.

Oracle Kernel Components

The kernel components relate to the background processes, buffer cache and shared pool and managing the resources without conflicts and corruptions requires special handling.

In RAC as more than one instance is accessing the resource, the instances require better coordination at the resource management level. Each node will have its own set of buffers but will be able to request and receive data blocks currently held in another instance's cache. The management of data sharing and exchange is done by the Global Cache Services (GCS).

All the resources in the cluster group form a central repository called the Global Resource Directory (GRD), which is distributed. Each instance masters some set of resources and together all instances form the GRD. The resources are equally distributed among the nodes based on their weight. The GRD is managed by two services called Global Caches Services (GCS) and Global Enqueue Services (GES), together they form and manage the GRD. When a node leaves the cluster, the GRD portion of that instance needs to be redistributed to the surviving nodes, a similar action is performed when a new node joins.

RAC Background Processes

Each node has its own background processes and memory structures, there are additional processes than the norm to manage the shared resources, theses additional processes maintain cache coherency across the nodes.

Cache coherency is the technique of keeping multiple copies of a buffer consistent between different Oracle instances on different nodes. Global cache management ensures that access to a master copy of a data block in one buffer cache is coordinated with the copy of the block in another buffer cache.

The sequence of a operation would go as below

1. When instance A needs a block of data to modify, it reads the bock from disk, before reading it must inform the GCS (DLM). GCS keeps track of the lock status of the data block by keeping an exclusive lock on it on behalf of instance A
2. Now instance B wants to modify that same data block, it to must inform GCS, GCS will then request instance A to release the lock, thus GCS ensures that instance B gets the latest version of the data block (including instance A modifications) and then exclusively locks it on instance B behalf.
3. At any one point in time, **only one** instance has the current copy of the block, thus keeping the integrity of the block.

GCS maintains data coherency and coordination by keeping track of all lock status of each block that can be read/written to by any nodes in the RAC. GCS is an in memory database that contains information about current locks on blocks and instances waiting to acquire locks. This is known as *Parallel Cache Management* (PCM). The Global Resource Manager (GRM) helps to coordinate and communicate the lock requests from Oracle processes between instances in the RAC. Each instance has a buffer cache in its SGA, to ensure that each RAC instance obtains the block that it needs to satisfy a query or transaction. RAC uses two processes the GCS and GES which maintain records of lock status of each data file and each cached block using a GRD.

So what is a resource, it is an identifiable entity, it basically has a name or a reference, it can be a area in memory, a disk file or an abstract entity. A resource can be owned or locked in various states (exclusive or shared). Any shared resource is lockable and if it is not shared no

access conflict will occur.

A global resource is a resource that is visible to all the nodes within the cluster. Data buffer cache blocks are the most obvious and most heavily global resource, transaction enqueue's and database data structures are other examples. GCS handle data buffer cache blocks and GES handle all the non-data block resources.

All caches in the SGA are either global or local, dictionary and buffer caches are global, large and java pool buffer caches are local. Cache fusion is used to read the data buffer cache from another instance instead of getting the block from disk, thus cache fusion moves current copies of data blocks between instances (hence why you need a fast private network), GCS manages the block transfers between the instances.

Finally we get to the processes

Oracle RAC Daemons and Processes		
LMSn	Lock Manager Server process - GCS	this is the cache fusion part and the most active process, it handles the consistent copies of blocks that are transferred between instances. It receives requests from LMD to perform lock requests. I rolls back any uncommitted transactions. There can be up to ten LMS processes running and can be started dynamically if demand requires it. they manage lock manager service requests for GCS resources and send them to a service queue to be handled by the LMSn process. It also handles global deadlock detection and monitors for lock conversion timeouts. as a performance gain you can increase this process priority to make sure CPU starvation does not occur you can see the statistics of this daemon by looking at the view X$KJMSDP
LMON	Lock Monitor Process - GES	this process manages the GES, it maintains consistency of GCS memory structure in case of process death. It is also responsible for cluster reconfiguration and locks reconfiguration (node joining or leaving), it checks for instance deaths and listens for local messaging. A detailed log file is created that tracks any reconfigurations that have happened.
LMD	Lock Manager Daemon - GES	this manages the enqueue manager service requests for the GCS. It also handles deadlock detention and remote resource requests from other instances. you can see the statistics of this daemon by looking at the view X$KJMDDP
LCK0	Lock Process - GES	manages instance resource requests and cross-instance call operations for shared resources. It builds a list of invalid lock elements and validates lock elements during recovery.
DIAG	Diagnostic Daemon	This is a lightweight process, it uses the DIAG framework to monitor the health of the cluster. It captures information for later diagnosis in the event of failures. It will perform any necessary recovery if an operational hang is detected.

Previous Menu Next