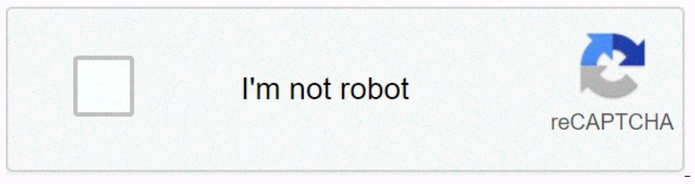

Continue

Oracle 12c rac architecture diagram

Oracle Real Application Group enables multiple instances to access a single database, cases will be run on multiple nods. In a standard Oracle configuration a database can only be mounted by one instance, but in a many RAC environments can be accessed through a single database. Oracle's RAC is heavily dependent on an efficient, high reliable private network speed called the intercontinent, make sure when designing a RAC system that you get the best that you can afford. The following table describes the difference in an oracle standard database (one example) an element RAC Single Element Settings Rac S Example has its own SGA Each instance has its own SGA background process Has its own set of background processing Each instance has its own set of background Data Access processing Not only one Shared Instance by All Instances (Shared Storage) Access Folder Control not only one Shared instance by all instances (Shared Store Page) Online Spindle Logfile Dedicated for writing/reading in one instance only one can write but other cases read during recovery and archiving. If an example is shutdown, changing log log by other instance may force the iled spinal logs to be archived Archived Spindle Logfile Dedicated to the private instance of the example but other instances will need the access to all archive logs needed during recovery media Flash Recovery Log Access not only one instance Shared by all instance (Shared Storage) Log Alert and Trace Files Dedicated to the private instance of each instance, other cases may never read or write these files. ORACLE_HOME multiple instances on the same access to different databases can use the executable files similar to one more instance can be placed on file system sharing that allows a common ORACLE_HOME for all instances of a RAC environment. RAC The major components of an Oracle rac system to shared disk system Oracle Clusterware Clusterware Interconnects Oracle Kernel Components below describe the core architecture of Ora TOC RAC environment here is a list of processes running on a sour architecture freshly installed WITH SAN and NAS disk storage systems today, storage sharing is fairly easy and is required for a RAC environment, you can use the setup below SAN storage (Network Storage) – generally using fire for connecting to the SAN NAS (Secure Storage Network) – generally using a network to connect to the NAS using either NFS, ISCSI JBOD – Storage Direct Attach , the old traditional way and still be used by many companies as a cheap option all of the above solutions can offer multi-way to reduce SPOFs in the RAC environment, there is no reason to not configure multi-path as the price is cheap when adding additional paths to the disk because most of the cost is paid when they come out setup the first path, so an additional controller card and network/cables fiber are all that needed. The last thing to think about is how to setup the disk structure this enclosure is known as a raid level, there are approximately 12 different raid levels that I know cut, here are the most common raid 0 (striping) a number of disks to embrace together to give the appearance of one very large disk. Advantages of improved performance can create very large availability disadvantages (if one disk fails, the failed volume) raid 1 (miroring) A disk is reflected by another disk, if one disk fails the system is affected as it can use its mirror. Advantages of highly available improved performance (if one disk fails to take over) expensive disadvantages (requires double disk amount) raid 5 Raid stands for Redundant Array of Expensive Disks, The disks are tight and aggravated across 3 or more disk, is the parite to use in the event that one of the disks fails, the data on the disk failed to reconstruct using the parity bit. Advantage improved performance (read only) Inexperienced Slow Drawback operation writes (caused by having to create the parite bit) There are many other raid levels that can be used with a particular hardware environment for example EMC storage use the RAID-SID, HP storage uses Auto RAID, so check with the inventory for the best solution that will provide you with the best performance and resistance. Once you have storage attached to the servers, you have three choices on how to configure the Raw Volumes disk – normally used for performance benefits, However they are hard to manage and backup Cluster FileSystem – used to maintain all the Oracle datafiles can be used by windows and linux, its not using widely Automated Storage Management (ASM) – Oracle choice of storage management, it is a portable, dedicated and optimized graphitem file I will only discuss ASM, which I already had a topic about called Automatic Storage Management. Oracle Clusterware Oracle Clusterware software designed to run Oracle in a clusters mode, it can support you to nod 64, it can even be used with a vendor clusters such as Sun Cluster. The Software Clusterware enables the nose to communicate with each other and form the cluster that performs the nose to the work as a single logical server. The software is run by the Ready Cluster Services (CRS) using the Oracle Cluster Registry (OCR) that records and holds the clusters information and node membership information with the voting disc that acts as a tiebreaker during communication failure. Consistent heart rate information travels across the intercourse of the voting disk when the clusters are run. The CRS has four OPROCd components - Monitor Daemon CRSd - CRSS damon, the failure of this daemon result in a have been rebooted to avoid OCSD data corruption – Oracle Cluster Cynchronization Service Daemon (the latest registry) EVMd – Event Volume Manager Daemon damon in OPROCD provides I/O matches for the Oracle click, it uses the hangcheck clock or watchdog clock for integrity of clusters. It is close to memory and run as a real process, failure to result this daemon into the nose being rebooted. Fencing is used to protect the data, if a nose being having food issues presumed the worst and protecting the data thus restarting the nose in question, better it to be saved than sorrow. CRSd processes manage resources such as starting and stopping the services and failover of the application resources, it also spans separate processes to manage application resources. CRS manages the OCR and the current store knows the current state of the clusters, it requires a public, private corner and VIP in order to run. OCSSd provides synchronization services among the nose, it provides access to members of the nose and enables basic clusters services, including customer group service and blocking, failure of this daemon causes the nose to be rebooted to avoid split-in brain situations. The functions below covered by the OCSSd CSS provide basic service group support, it is a distributed group member system that enables applications to coordinate activities to archive a common result. Service Groups uses service blinker cluster vendor when it is available. Closing Service provides the basic clicks-wide function locking serialization, it uses the first, First August (FIFO) mechanism to manage Node service uses OCR to store data and updates the information during reconfiguring, it also manages the OCR data that is otherwise static. The last element is the Event Loger Management, which runs the EVMd process. Daemon's enforcement process is called evmlogger and generates the events when things happen. The Evmlogger passes new child processes on demand and analyzes the directory calls to invoke their calendars. The demise of the EVMd damon will not stand the example and will restart. Quick reminder CRS Process Foncity Failure of the Process running AS OPROCD – Process Monitors provide basic integrity service Node integrity service restart root EVMd – Span events a child processing installation and generate Damone caouts automatically restart, no restart oracle OCSD – Cluster Synchronization Service Basic Bulb Service, Cluster services, basic blocking Node restart oracle CRSD – Monitoring Resource Services Cluster, failover and node recovery daemon restart automatically, node restart the root client-ready service (CRS) is a new component of 10g RAC, is installed in a separate home directory called ORACLE_CRS_HOME. It is a binding component but can be used with a third party cluster (Veritas, Sun Cluster), by default it manages it node functionalities along with managing RAC resources related to sour and RAC services use a member scheme, so any noses want to join the group as they become a member. RAC can avoid any member that it looks like as a problem, its main concern is to protect the data. You can add and remove nodes from the club with members of leverage or decrease, when network issues arrive members become the deciding factor to which party remains as the club and with no avoidance, the use of a voting disk to use which I will talk about later. The Resource Management Foundation manages the resources of the club (disks, volumes), so you can have only one resource foundation per resource. Multiple foundations are not supported as it can lead to undesirable affects. The Oracle Cluster Ready Service (CRS) uses the registry to keep the clusters setup, it should reside on a shared storage and accessible to all nuds in the cluster. This shared repository is known as the Oracle Cluster Registry (OCR) and it's a larger part of the cluster, it is automatically backed up (every 4 hours) the more daemons you can manually back it up. The OCSSd uses the OCR a lot and writes the changes to the OCR registry to keep details of all resources and services, it stores names and pairs of values in information such as resources that are used to manage the resource equivalent by the CRS stack. Resources and stack CRS are components managed by CRS and have information about the good/bad state and named scripts. The OCR is also used to provide botstrap information ports, nodes, etc., it is a binary file. The OCR is loaded as cache on each nerve, each nerve will update the cache then only one nose is allowed to write the cache of the OCR file, the nose is called the owner. The Enterprise Manager also uses the OCR cache, it should be at least 100MB in size. The Damon CRS will update the OCR on the status of the nose at the club during recognition and failure. The voting disk (or quorum disk) is shared by all nations of the group, information about the group is constantly written to the disk, this is known as the pulse. If for any reason a nose is unable to access the voting disk it immediately avoids in the clusters, this protects the group from split-brain (in Instance Member Recovery IMR algorithm used to detect and solve split-brain) as the voting disk decides which party really clusters. Votes the ability to manage members of clients and arbitrary ownership of clients during communication failures between nodes. Votes often confused with the quorum are similar but distinct, below detail what each means vote A is usually a formal expression of opinion or will be in response to a Proposed Decision Quorum defined as the number, usually a majority of Of a body, that when it is rallied legally competent to transact the Business Single Vote count is the quorum vote, the quorum vote defines the client. If a nose or cluster of nodes cannot archive a quorum, they must not initiate any service because they risk conflict with an established quorum. The voting disk has resides on shared storage, it is a small file (20MB) that can be accessed through all noses of the group. In Oracle 10g R1 you may have only one voting disk, but in R2 you may have upto 32 vote disks that allow you to eliminate any SPOF A. The original Virtual IP of Oracle has Transparent Application Failover (TAF), this has bounds, this has now been replaced with group vips. The client VIPs will failover to working nose if a nose should fail, these public IP are configured in DNS so users can access to them. The clusters VIPs are different from the intercontinental IP address and are only used to access the database. Interconsect of clusters is used to synchronize the resources in the RAC clusters, and also is used to transfer some data from one instance to another. This intercontinent should be private, highly available and fast with low latency, preferably they should be on a minimum private 1GB network. What hardware you are using the NIC should use multi-path (Linux - Bonding, Solaris - IPMP). You can use crossover cblesover in a QA/DEV environment but it is not supported in a production environment, whilst crossover cbles your limits to a cluster node. Oracle Kernel Elements The kernel components relate to the background processes, breath cache and shared pools and manage their resources without conflict and corruption requires special handling. In RAC as more than one instance will get access to the resource, cases are required for better coordination at the resource management level. Each nose will have its own sets of defenses, but will be able to request and receive data blocks currently held in caches of another instance. The management of data sharing and exchange is done by the Hidden Global Services (GCS). All the resources in the client group form a central repositatory called the Global Resource Directory (GRD), which is distributed. Each instance snaps some set of resources and simultaneously all can form the GRD. Resources are equally distributed among nations based on their weight. The GRD is managed by two services called Global Caches Services (GCS) and Global Enqueue Services (GES), together to form and manage the GRD. When a nose leaves the group, the GRD portion of that needs to be redistribye to the survival nose, it's an action that seems to be performed when a new nose joins. RAC Background Process each nerve has its own background process and memory structure, there are additional processes than the normal of managing the shared resources, the following additional processes cache coalition through the nose. Cache coherency is the technique of maintaining multiple copies of a consistent defense between different Oracle instances of different noses. Global cache management ensures that accessing a master copy of a data block in one cache detects coordinates with the copy of the block in another defense cache. The sequence of an operation would go like below When e.g. A needs a block of modified data, it reads the bock from disk, before you read it to be informed 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 Now e.g. B wants to modify that same block data, it must be informed GCS, GCS will then require example A to drop close to, thus GCS ensures that example B becomes the latest version of the data block (including example a modification) and then only close it on instance name Bhalf. At any point in time, only one instance has the actual copy of the block, therefore maintaining the integrity of the block. GCS keeps coherent data and coordination by keeping track of all close status in each block that can be read/written to by any nuts in the RAC. GCS is a database memory containing information about current locks on blocks and can waiting to get locks. This is known as Parallel Cache Management (PCM). The Global Resource Manager (GRM) helps coordinate and communicate the close requests to Oracle processes between the instances of the RAC. Each instance has a cache suffered through its SGA, ensuring that each RAC instance finds the block that it needs to meet a query or transaction. RAC uses the two GCS processes and GES that hold records of locked status in each data file and each cache block using a GRD. So what is a resource, it is an identified entity, it basically has a name or a reference, it can be an area of memory, a disk record or an abstract entity. You can own a resource or close to various states (besides or share). Any shared resource is lockable and if it does not share no access conflicts will occur. A global resource is a resource that is visible throughout the nose of the clusters. Data block caches are the most obvious and heavier global resources, anxiety transactions and database structures are other examples. GCS handles block cache data and GES handles all the non-data block resources. All caches of the SGA are either global or local, dictionaries and defense caches are global, large and java pools their local caches. Fusion Cache is used to read the cache of data from another instance instead of getting the block from disk, so Fusion Cache moves the actual copy of data blocks between instances (therefore you need a fast private network), GCS manages the block transfers between instances. Finally we get through the Oracle Processes RAC Daemons and LMSn Process Process Server Manager – This GCS is the cache merger part and the most active process, it touches the consistent copies of blocks that are transferred between instances. It receives requests from LMD to make closing requests. I roll back any unlimited transactions. There may be up to ten LMS processes running and can start dynamically if requested.to manage Service Manager lock requests for GCS resources and send them to a service queue to be handled by the LMSn process. It also handles global detection monitors and monitors for closed conversion timeout. as a winning performance you can increase this processing priority to make sure CPU starvation does not happen you can view the statistics of this damon by looking at the look of X$ KJMSDP LMON Lock Monitoring Process – GES this process manages the GES, it maintains the consistency of GCS memory structure in case of death process. He is also responsible for group reconfirmation and serial recognition (nose join or leave), he checks for demise examples and listens to local messaging. A detailed log file is created that tracks any recognition that occurred. LMD Lock Manager Daemon – This GES manages the service manager requests to enemy for the GCS. It also touches detention detention and requests resources remotely from other cases. you can view the statistics of this daemon by looking at the X$KJMDP LCK0 Lock Process – GES managed instance resource requests and cross-instance call operations for sharing resources. It builds a list of invalid lock elements and closed components during recovery. Diag Daemon's diagnosis is a lightweight process, he uses the DIAGNOSIS foundation to monitor the health of the club. It captures information for later diagnosis of the event of failure. It will make any recovery needed if an operational outcast is detected. detector.