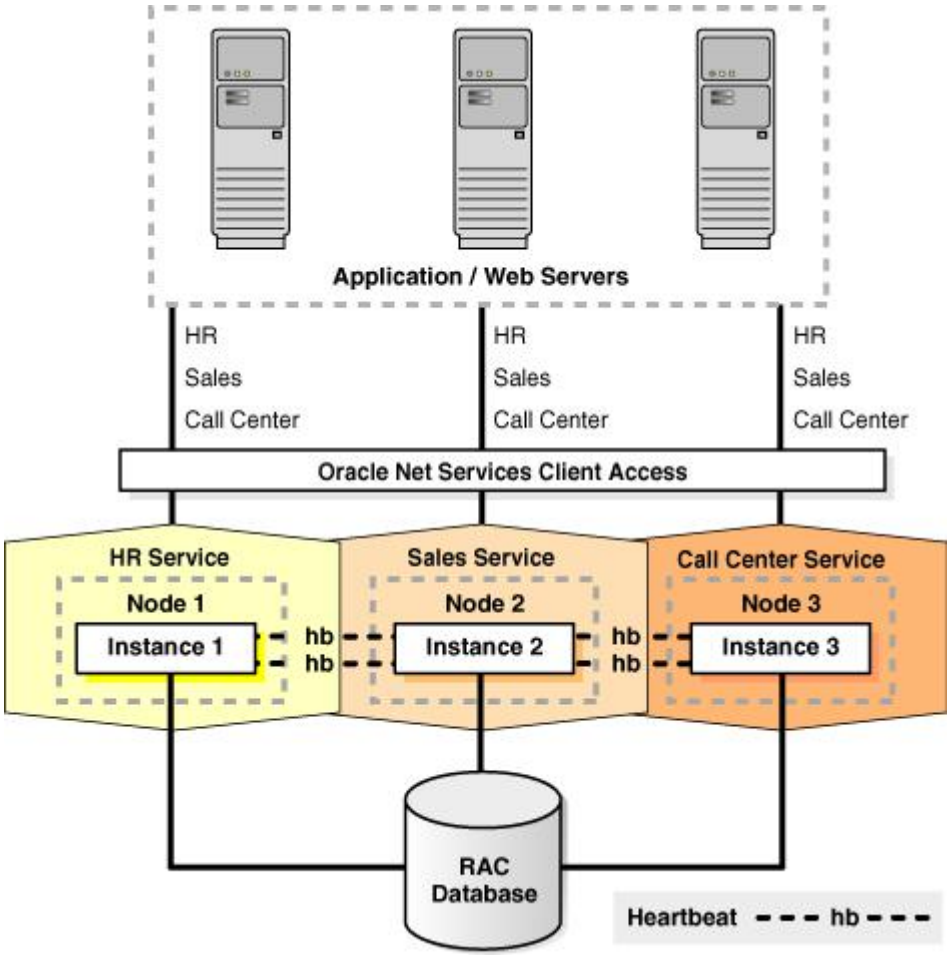
数据库 RAC 安装、操作手册

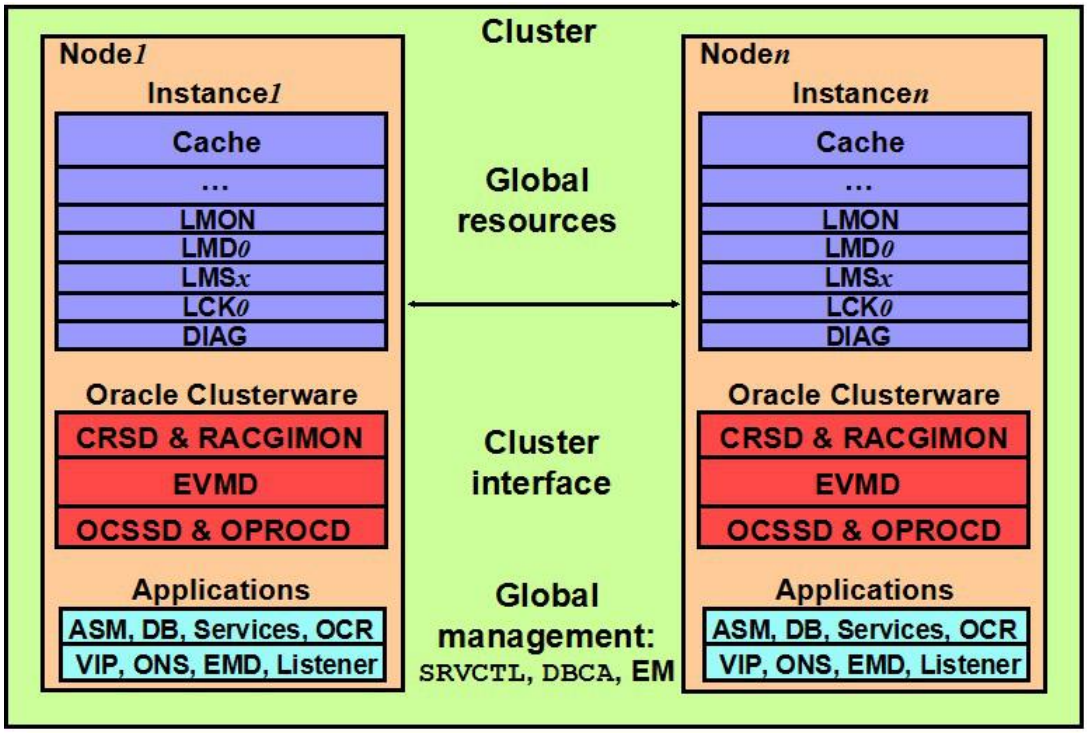
目录

1. RAC 系统架构



Oracle RAC databases differ architecturally from noncluster Oracle databases in that each Oracle RAC database instance also has:

- At least one additional thread of redo for each instance
- An instance-specific undo tablespace



The Oracle Clusterware requires two clusterware components: a voting disk to record node membership information and the Oracle Cluster Registry (OCR) to record cluster configuration information.

The voting disk and the OCR must reside on shared storage. The Oracle Clusterware requires that each node be connected to a private network by way of a private interconnect.

Oracle recommends that you configure a redundant interconnect to prevent the interconnect from being a single point of failure. Oracle also recommends that you use User Datagram Protocol (UDP) on a Gigabit Ethernet for your cluster interconnect.

The Oracle Clusterware manages node membership and prevents split brain syndrome in which two or more instances attempt to control the database. This can occur in cases where there is a break in communication between nodes through the interconnect.

2. 11G 新特性

2.1. RAC One Node

RAC One Node 是 11.2 的新特性，是 RAC 数据库中的一个实例运行在 GRID 集群中，并且可以实现 failover。

这个功能有些类似于我们以前俗称的"HA 数据库" <HA - high availability>，"HA 数据库"是利用其他厂商的集群软件来管理 Oracle 单机数据库，实现数据库的高可用。而 RAC One Node 是通过 ORACLE 集群软件(GRID)来管理数据库，实现数据库可以在集群中节点上切换（failover/relocate），达到数据库高可用的特点。RAC One Node 是完全由 Oracle 提供的一整套高可用的解决方案。

在 11.2 之前，为了实现数据库的高可用<俗称"HA 数据库">，通常的做法是将单机数据库部署在其他厂商集群环境中（比如 HP MC/SG, IBM HACMP 等）管理，来实现数据库的高可用。即单机数据库运行在主节点上，当主节点需要维护或者异常中断的情况下，通过厂商集群软件将服务 IP 资源组和数据文件资源组切换（failover）到备节点，将数据库在备用节点重新启动。这个过程我们一般称为 cold failover，因为数据库在切换的过程中是先 shutdown 再 open。

RAC One Node 的原理与以前的 HA 数据库不同，RAC One Node 是基于 RAC 数据库，并且通过 Oracle 集群软件(GRID)管理实现只启动 RAC 数据库的一个实例，当运行实例的节点需要维护停机的情况下，可以通过 online database relocation 的方式将数据库实例切换到集群中的其他节点上运行。

参考文档：Oracle RAC One Node -- Changes in 11.2.0.2 (Doc ID 1232802.1)

2.2. Job Role Separation (集群件任务角色分离)

在 11gR2，操作系统用户 grid 成为了集群件（GI）的 owner,并且 ASM 成为了集群件的一部分，所以 grid 用户也成为了 ASM 磁盘的 owner。

ASM 磁盘的 group 是 asmadmin，这意味着组 asmadmin 中的成员可以对 asm 磁盘进行读写操作，当然 grid 用户也可以。而其他用户，例如 oracle，则需要通过 oracle_home/bin 下的 oracle 可执行文件访问 asm 磁盘。

这意味着 oracle 可执行文件不仅需要黏着位（stick bit），还需要是设置 group 为 asmadmin。当使用 srvctl（srvctl start database/instance）启动数据库时 oracle 会自动调用<rdbms_home>/bin/setasmgid 设置 oracle 可执行文件的 group 为 asmadmin。

所以，如果问题出现在 oracle 不能访问 asm 磁盘，需要检查以下的内容。当然由于 oracle 可以直接访问 asm 磁盘，而不需要通过 asm 实例，所以问题的症状可能很多，甚至 ora-600 错误都可能是这个原因。

1. asmlib 标识过的磁盘的权限和 group 设置

```
brw-rw---- 1 grid asmadmin 8, 49 Dec 31 12:14 DATA
```

2. 裸设备或者块设备的权限和 group 设置

```
crw-rw---- 1 grid asmadmin 162, 1 Jul 18 21:40 /dev/raw/raw1
```

3. RDBMS 和 GI 主目录下的 oracle 可执行文件的权限和 group 设置

```
RDBMS_HOME: -rwsr-s--x 1 oracle asmadmin 188832561 Oct 30 21:22 oracle
GI_HOME: -rwsr-s--x 1 grid oinstall 166530359 Nov 16 14:31 oracle
```

注意黏着位(stick bit)的设置：

最后我们对 11gR2 中安装 oracle 集群件和数据库软件中的一些 group 进行简单的介绍。

- oinstall：这个 group 是 GI 和 RDBMS 软件的拥有者。
- dba：这个 group 是数据库的 dba group，对数据库具有最高权限。
- asmdba：这个 group 是 asm 实例的 dba group，可以启动/关闭实例，挂载/卸载 asm 磁盘组。
- asmadmin：这个 group 是 asm 的管理员 group,它包含 asmdba 的全部权限，同时还可以增加/删除 asm 磁盘，磁盘组等。

2.3. SCAN(Single Client Access Name)

从 11gR2 Grid Infrastructure (CRS/clusterware)开始，引入了一个新功能叫 SCAN (Single Client Access Name)，SCAN 是一个域名，可以解析至少 1 个 IP，最多解析 3 个 SCAN IP，客户端可以通过这个 SCAN 名字来访问数据库，SCAN 的好处就是当集群中新增加了节点或者删除了节点，不需要额外维护客户端。在 11gR2 上，客户端仍然可以继续使用原有的 VIP，但是 oracle 推荐使用 SCAN。

SCAN ip 必须与 public ip 和 VIP 在一个子网，同时 oracle 推荐使用 DNS 或者 GNS(11gR2 新功能)来解析 SCAN，如果没有使用 DNS 或者 GNS 的话，可以使用 hosts 文件，但是这个办法不是 oracle 推荐的，因为这个方法只能定义一个 SCAN IP。

GRID 集群中有 2 类资源是与 SCAN 有关的，一类是 SCAN IP，另一类是 SCAN Listener，SCAN IP 和 SCAN Listener 是成对出现的，也就是说如果有 3 个 SCAN IP，就会同时有 3 个 SCAN Listene。SCAN IP 就是 DNS 解析的 IP 地址，SCAN Listener 的作用是接受客户端的连接请求。查看 SCAN IP 信息和 SCAN Listener 信息的方法在下文介绍。

数据库的初始化参数 remote_listener 默认被设置为 SCAN Listener，目的是为了让 SCAN Listener 可以监听所有的实例，记录所有实例的压力，以便于按照负载均衡的方式来转发客户端的请求。

```
$ srvctl config scan
SCAN name: rac-cluster-scan, Network: 1/192.168.1.0/255.255.255.0/
SCAN VIP name: scan1, IP: /rac-cluster-scan/192.168.1.12
SCAN VIP name: scan2, IP: /rac-cluster-scan/192.168.1.13
SCAN VIP name: scan3, IP: /rac-cluster-scan/192.168.1.14
```

ORACLE SUPPORT 网站上查看下面 2 篇文章，都是关于 SCAN 的介绍，希望能够帮助您解决关于 SCAN 的问题。

11gR2 Grid Infrastructure Single Client Access Name (SCAN) Explained (Doc ID 887522.1)

How to Setup SCAN Listener and Client for TAF and Load Balancing [Video] (Doc ID 1188736.1)

2.4. HAIP (Redundant Interconnect with Highly Available IP)

从 11.2.0.2 开始，Oracle 的集群软件 Grid Infrastructure(GI)中新增了 Redundant Interconnect with Highly Available IP(HAIP)，以实现集群私网的高可用性和负载均衡。

在 11.2.0.2 之前，私网的冗余一般是通过在 OS 上做网卡绑（如 bonding，EtherChannel 等）实现的，有了 HAIP 之后，无需使用网卡绑定就可以实现私网网卡的冗余。

安装后，HAIP 地址自动设置为 169.254.*.*,这个地址不可以手动设置。HAIP 最少为 1 个,最多为 4 个(1 块网卡,1 个 HAIP;2 块网卡,2 个 HAIP; 3 块及以上,4 个 HAIP),均匀的分布在私网的网卡上。

多个私网网卡可以在安装阶段定义，也可以在后来使用 oifcfg 更改。ora.cluster_interconnect.haip 资源将从“link-local” IP 范围(169.254.*.*)中为每个私网网卡选取一个高可用的虚 IP(HAIP)。默认地，私网流量会在所有活动的内联网卡上进行负载均衡，如果一个私网网卡损坏或者无法通信，Oracle GI 软件会透明地将相应的 HAIP 地址移到其中一个剩余的在工作的网卡上面。相比于第三方网卡绑定技术，在提供高可用性的同时又有效利用了带宽。

```
[oracle@rac11g1 ~]$ ifconfig -a
eth2      Link encap:Ethernet  HWaddr 08:00:27:D3:6A:D6
          inet addr:10.10.1.10  Bcast:10.10.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed3:6ad6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:762187 errors:0 dropped:0 overruns:0 frame:0
          TX packets:713836 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:522044523 (497.8 MiB)  TX bytes:500807612 (477.6 MiB)

eth2:1    Link encap:Ethernet  HWaddr 08:00:27:D3:6A:D6
          inet addr:169.254.2.227  Bcast:169.254.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

注意：HAIP 地址失败不会对 ocssd 产生影响，也就是说 HAIP 失败，不会导致节点重启。

注意：HAIP 是不允许被手动停止或禁用的，除非是由于某些版本或者平台不支持。

Grid Infrastructure Redundant Interconnect and ora.cluster_interconnect.haip (Doc ID 1210883.1)

2.5. 11gR2 Agent

在 10gR2 当中，crsd 负责对集群中的资源进行管理。具体说来，crsd 调用相关的 racg 脚本，产生 racg 进程对资源进行管理，例如 racgvip 脚本用来管理 vip 资源。这种管理办法，由于是 racg 进程进行对资源的操作，有时候会存在一些问题。从 11gR2 GI 开始，agent 作为一个全新的架构对 GI 中所有的资源进行管理，这种全新的 agent 架构使资源管理更加强壮，性能更好。

agent 拥有一些 EP(Entry Point)，类似于可以对资源执行的动作。

Start：启动资源

Stop：停止资源

Check：检查资源的状态，如果发现了资源状态改变，则 agent 会通知 GI，资源状态发生了改变。

Clean：清理资源，一般来说清理资源会在资源存在问题，需要重新启动或 failover 之前发生。

Abort：中止资源。

任意 EP 结束之后，会返回以下返回值中的一个，而这些返回值也对应着资源的状态。

ONLINE：在线。对应资源的 online 状态

OFFLINE：离线。对应资源的 offline 状态。对于 offline 状态，可以细分为 planed offline 和 unplaned offline。Planed offline 是指 GI 倾向于这个资源处在 offline 状态，例如我们使用 GI 相关的工具（srvctl，crsctl）停止了一个资源，这种情况，GI 就认为资源应该处于 offline 状态，因为停止资源的操作是通过 GI 来实现的。同时，对于 planed offline 的资源，它的 target 状态也会被修改为 offline 状态，这意味着，如果在资源的 target 状态为 offline 时重启 GI stacks，除非资源的 auto_start 属性设置为 always，否则，该资源不会被自动启动。对于 unplaned offline，是指资源被 GI 以外的工具停止，例如使用 sqlplus 手动关闭数据库，在这种情况下，GI 并不认为该资源应该处于 offline 状态，资源的 target 状态仍然为 online，所以，资源在重新启动 GI 时仍然会被启动，当然除非资源的 auto_start 属性设置为 never。

UNKNOWN：未知，对应资源的 unknown 状态。在这种状态下，agent 会继续对该资源进行 check。

PARTIAL：资源部分在线，对应资源的 intermediate 状态。 在这种情况下 agent 会继续对该资源进行 check，并及时更新资源状态。

FAILED：失败。该返回值说明资源存在问题，不能正常工作，agent 会首先执行 clean EP，之后根据资源的相关属性进行 failover 或 restart 操作。

如何避免资源不自启动

```
[oracle@rac11g2 ~]$ crsctl stat res ora.crmdb.db -p | grep AUTO_START
AUTO_START=restore

[oracle@rac11g2 ~]$ crsctl modify resource ora.crmdb.db -attr
"AUTO_START=always"

[oracle@rac11g2 ~]$ crsctl stat res ora.crmdb.db -p | grep AUTO_START
AUTO_START=always
```

2.6. Rebootless Restart

从版本 11.2.0.2 开始，oracle 新特性 rebootless restart 被介绍。当出现以下情况的时候，集群件（GI）会重新启动集群管理软件，而不是将节点重启。

1.当某个节点连续丢失网络心跳超过 misscount 时。

2.当某个节点不能访问大多数表决盘（VF）时。

3.当 member kill 被升级成为 node kill 的时候。

在之前的版本，以上情况，集群管理软件（CRS）会直接重启节点。

GI 在重启集群之前，首先要对集群进行 graceful shutdown，基本的步骤如下。

1. 停止本地节点的所有心跳（网络心跳，磁盘心跳和本地心跳）。

2. 通知 cssd agent，ocssd.bin 即将停止

3. 停止所有注册到 css 的具有 i/o 能力的进程，例如 lmon。

4. cssd 通知 crsd 停止所有资源，如果 crsd 不能成功的停止所有的资源，节点重启仍然会发生。

5. Cssd 等待所有的具有 i/o 能力的进程退出，如果这些进程在 short i/o timeout 时间内不能不能全部推迟，节点重启仍然会发生。

6. 通知 cssd agent 所有的有 i/o 能力的进程全部退出。

7. Ohasd 重新启动集群。

8. 本地节点通知其他节点进行集群重配置。

2.7. CHM(Oracle Cluster Health Monitor)

Cluster Health Monitor（以下简称 CHM）是一个 Oracle 提供的工具，用来自动收集操作系统的资源（CPU、内存、SWAP、进程、I/O 以及网络等）的使用情况。相对于 OSWatcher，CHM 直接调用 OS 的 API 来降低开销，而 OSWatcher 则是直接调用 UNIX 命令。另外，CHM 的实时性更强，每秒收集一次数据(在 11.2.0.3，改为了 5 秒一次)。

OSWatcher 的优点是可以用 traceroute 命令检测私网间的连通性，而且生成的数据的保留时间可以设置得很长。如果可以的话，最好是两个工具都使用。

这些系统资源数据对于诊断集群系统的节点重启、Hang、实例驱逐(Eviction)、性能问题等是非常有帮助的。另外，用户可以使用 CHM 来及早发现一些系统负载高、内存异常等问题，从而避免产生更严重的问题。

在集群中，可以通过下面的命令查看 CHM 对应的资源(ora.crf)的状态：

```
[grid@rac11g1 ~]$ crsctl stat res -t -init
ora.crf
      1        ONLINE  ONLINE       rac11g1
```

CHM 主要包括两个服务：

1). System Monitor Service(osysmond)：这个服务在所有节点都会运行，osysmond 会将每个节点的资源使用情况发送给 cluster logger service，后者将会把所有节点的信息都接收并保存到 CHM 的资料库。

```
[grid@rac11g1 ~]$ ps -ef | grep osysmond
root     2560     1  0 Jul29 ?       00:10:28 /oracle/app/11.2.0/grid/bin/osysmond.bin
```

2)Cluster Logger Service(ologgerd)：在一个集群中的，ologgerd 会有一个主机点(master)，还有一个备节点(standby)。当 ologgerd 在当前的节点遇到问题无法启动后，它会在备用节点启用。

主节点：

```
[grid@rac11g1 ~]$ ps -ef|grep ologgerd
root     2625     1  0 Jul28 ?       00:08:20 /oracle/app/11.2.0/grid/bin/ologgerd -M -d
/oracle/app/11.2.0/grid/crf/db/rac11g1
```

备节点：

```
[oracle@rac11g2 ~]$ ps -ef|grep ologgerd
root     2639     1  0 Jul29 ?       00:01:26 /oracle/app/11.2.0/grid/bin/ologgerd -m rac11g1 -r
-d /oracle/app/11.2.0/grid/crf/db/rac11g2
```

备注：该进程在 11.2.0.3 的 Linux 平台上有 bug，会造成 CPU 持续 100%。

CHM 诊断日志：如果 CHM 的运行异常，可以查看下面的日志：

$GRID_HOME/log/<nodename>/crflogd/crflogd.log

$GRID_HOME/log/<nodename>/crfmond/crfmond.log

GI 中的服务 ora.crf 是 CHM 对应的资源，可以使用下面的命令来启停 CHM（不推荐停止该服务）：

用 root 用户：

$GRID_HOME/bin/crsctl stop res ora.crf -init

$GRID_HOME/bin/crsctl start res ora.crf -init

CHM Repository：用于存放收集到数据，默认情况下，会存在于 Grid Infrastructure home 下，需要 1 GB 的磁盘空间，每个节点大约每天会占用 0.5GB 的空间，您可以使用 OCLUMON 来调整它的存放路径以及允许的空间大小(最多只能保存 3 天的数据)。

下面的命令用来查看它当前设置：

$ oclumon manage -get reppath

```
CHM Repository Path = /u01/app/11.2.0/grid/crf/db/rac2
Done
```

$ oclumon manage -get repsize

```
CHM Repository Size = 68082 <====单位为秒
Done
```

修改路径：

$ oclumon manage -repos reploc /shared/oracle/chm

修改大小：

$ oclumon manage -repos resize 68083 <==在 3600(小时) 到 259200(3 天)之间

```
rac11g1 --> retention check successful
rac11g2 --> retention check successful
New retention is 68083 and will use 1188674713 bytes of disk space

CRS-9115-Cluster Health Monitor repository size change completed on all nodes.

 Done
```

获得 CHM 生成的数据的方法有两种：

1. 一种是使用 Grid_home/bin/diagcollection.pl：

1). 首先，确定 cluster logger service 的主节点：

```
$ oclumon manage -get master
   Master = rac2
```

2).用 root 身份在主节点 rac2 执行下面的命令：

```
# <Grid_home>/bin/diagcollection.pl -collect -chmos
-incidenttime inc_time -incidentduration duration
```

inc_time 是指从什么时间开始获得数据，格式为 MM/DD/YYYY24HH:MM:SS，duration 指的是获得开始时间后多长时间的数据。

比如：# diagcollection.pl -collect -crshome /u01/app/11.2.0/grid -chmoshome /u01/app/11.2.0/grid -chmos -incidenttime 06/15/201215:30:00 -incidentduration 00:05

3).运行这个命令之后，CHM 的数据会生成在文件 chmosData_rac2_20120615_1537.tar.gz。

2. 另外一种获得 CHM 生成的数据的方法为 oclumon:

```
$ oclumon dumpnodeview [[-allnodes] | [-n node1 node2] [-last
"duration"] | [-s "time_stamp" -e "time_stamp"] [-v] [-warning]] [-h]
```

-s 表示开始时间，-e 表示结束时间

```
$ oclumon dumpnodeview -allnodes -v -s "2012-06-15 07:40:00" -e
"2012-06-15 07:57:00" > /tmp/chm1.txt
$ oclumon dumpnodeview -n node1 node2 node3 -last
"12:00:00" >/tmp/chm1.txt
$ oclumon dumpnodeview -allnodes -last "00:15:00" >/tmp/chm1.txt
```

Cluster Health Monitor (CHM) FAQ (Doc ID 1328466.1)

2.8. HM (Hang Manager)

在我们诊断数据库问题的时候，经常会遇到一些数据库/进程 hang 住的问题。对于 hang 的问题。

一般来说，常见的原因有以下两种：

死锁（cycle）。对于这种 hang，除非循环被打破，问题会永远存在。

某个堵塞者（blocker） 进程在持有了某些资源后堵住了其他进程。当然，根据堵塞的情况，我们可以把 blocker 分为直接堵塞进程（immediate blocker）和根堵塞进程（root blocker）。而 root blocker 在通常情况下会处于两种状态。

- 根堵塞进程处于空闲状态，对于这种情况，终止这个进程能够解决问题。
- 根堵塞进程正在等待某些和数据库无关的资源（例如：等待 I/O），对于这种情况，终止这个进程也许能解决问题。但是，从数据库的角度来讲，这已经超出了数据库的范畴。

接下来，我们对每个步骤进行具体的介绍。

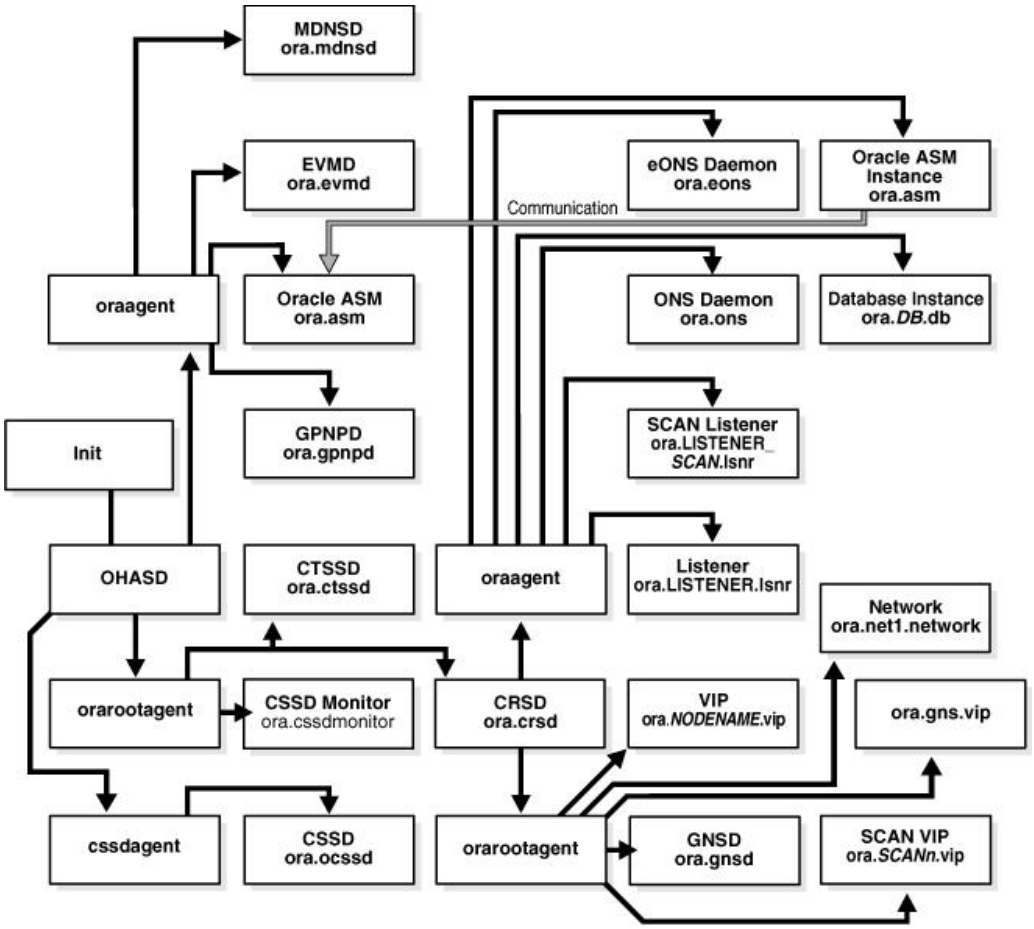
步骤 1: ORACLE 会分配一部分内存空间，我们称之为 hang analysis cache，用来存放搜集的 hang analyze dump i 信息。这部分内存空间在每个节点的数据库实例上都存在。

步骤 2：oracle 会定期搜集 hang analyze 信息，由于，HM 特性是针对 RAC 数据库的特性，hang analyze 的级别会包括本地和全局。另外，负责搜集这些 dump 信息的后台进程是 DIA0(这个进程从 11g 才被介绍)。默认情况下每 3 秒钟搜集本地级别 hang analyze dump，每 10 秒搜集全局级别 hang analyze dump。

步骤 3：因为，每个节点都会搜集 hang analyze dump 信息，那么，意味着每个实例都会拥有自己的 DIA0 进程，负责完成本地的 hang 分析。但是，对于 RAC 数据库，很多 hang 的情况会包含多个实例的进程。所以，我们需要一个实例上的 DIA0 进程作为 master，来对多个实例搜集到的信息进行分析。对于 11g 版本，节点号最小的实例的 DIA0 进程会成为 HM 的 master 进程。当然，在实例级别发生了重新配置后，主（master）DIA0 进程会重新在存在的实例中重新被选举出来。

3. Oracle Clusterware 组件

3.1. 集群启动流程



Short summary of the startup sequence: INIT spawns init.ohasd (with respawn) which in turn starts the OHASD process (Oracle High Availability Services Daemon). This daemon spawns 4 processes.

1. Level 1

OHASD Spawns:

- cssdagent - Agent responsible for spawning CSSD.
- orarootagent - Agent responsible for managing all root owned ohasd resources.
- oraagent - Agent responsible for managing all oracle owned ohasd resources.

- cssdmonitor - Monitors CSSD and node health (along wth the cssdagent).

2. Level 2

OHASD rootagent spawns:

- CRSD - Primary daemon responsible for managing cluster resources
- CTSSD - Cluster Time Synchronization Services Daemon
- Diskmon
- ACFS (ASM Cluster File System) Drivers

3. Level 2

OHASD oraagent spawns:

- MDNSD - Used for DNS lookup
- GIPCD - Used for inter-process and inter-node communication
- GPNPD - Grid Plug & Play Profile Daemon
- EVMD - Event Monitor Daemon
- ASM - Resource for monitoring ASM instances

4. Level 3

CRSD spawns:

- orarootagent - Agent responsible for managing all root owned crsd resources.
- oraagent - Agent responsible for managing all oracle owned crsd resources.

5. Level 4

CRSD rootagent spawns:

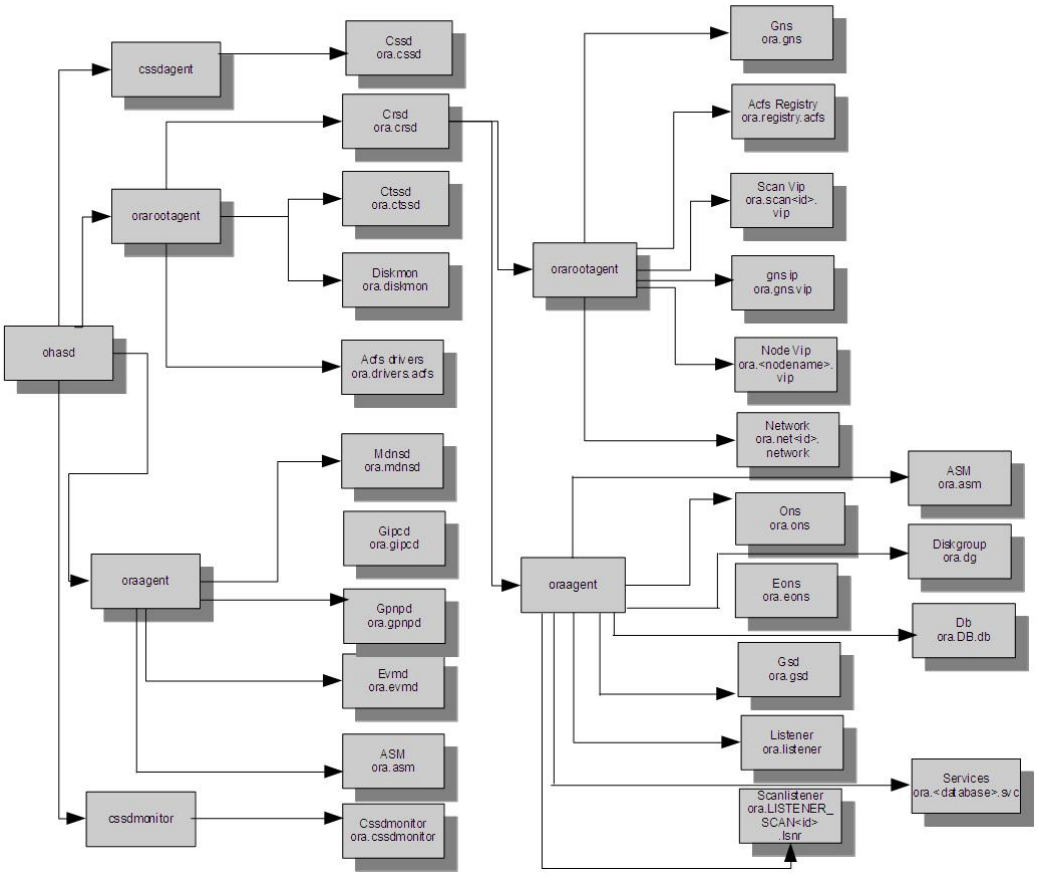
- Network resource - To monitor the public network
- SCAN VIP(s) - Single Client Access Name Virtual IPs
- Node VIPs - One per node
- ACFS Registery - For mounting ASM Cluster File System
- GNS VIP (optional) - VIP for GNS

6. Level 4

CRSD oraagent spawns:

- ASM Resouce - ASM Instance(s) resource

- Diskgroup - Used for managing/monitoring ASM diskgroups.
- DB Resource - Used for monitoring and managing the DB and instances
- SCAN Listener - Listener for single client access name, listening on SCAN VIP
- Listener - Node listener listening on the Node VIP
- Services - Used for monitoring and managing services
- ONS - Oracle Notification Service
- eONS - Enhanced Oracle Notification Service
- GSD - For 9i backward compatibility
- GNS (optional) - Grid Naming Service - Performs name resolution



3.2. 集群核心进程

3.2.1. ocssd.bin

这是一个很核心的进程，如果它异常终止会导致这个节点的集群或者主机重启。这个进程主要用于检查表决盘能否正常访问，节点私网间的通信是否正常。数据库实例上的 LMON 进程也会注册到 CSSD 上，这样 CSSD 会通过 LMON 来了解数据库实例的健康情况。

如果节点发生了主机自动重启，需要查看 ocssd 的日志，位于：<CRS_HOME>/log/<host>/cssd。

3.2.2. crsd.bin

这个进程主要用于管理集群中的资源。用来启动、停止检查一些资源，比如数据库实例、ASM、监听、磁盘组、VIP 等。在 11.2，这些具体的操作由对应的 agent 执行。另外，OCR 的维护也是由 CRSD 完成的。

当发现某些资源异常终止后，首先需要查看 crsd 的日志：<CRS_HOME>/log/<host>/crsd。

3.2.3. evmd.bin

事件监控(event monitor)进程，由它来发布集群事件，比如实例启动、停止等事件。

3.2.4. ons 进程

Oracle Notification Service daemon，它用于接收 evmd 发来的集群事件，然后将这些事件发送给应用预订者或者本地的监听，这样就可以实现 FAN(Fast Application Notification)，应用能够接收到这些事件并进行处理。

3.2.5. gsd

只有当 CRS 或者 GI 上需要管理 9i 的数据库时才需要。在 11.2，gsd 默认就是 offline 的。

可以参考 My Oracle Support 文档： GSD Is Used Only if 9i RAC Database is Present (Doc ID 429966.1)

3.2.6. oprocd.bin

该进程是 10g 和 11.1 特有的。

Oracle Clusterware Process Monitor Daemon，用来监控主机 hang，如果发现主机 hang 后会发起主机重启。只有未使用第三方的集群软件时才运行，Linux 上从 10.2.0.4 开始使用。

oprocd 的日志会在：/etc/oracle/oprocd/*.log.* 或者 /var/opt/oracle/oprocd/*.log.*。

3.2.7. oclsomon.bin

该进程是 10g 和 11.1 特有的。用来监控 ocssd 进程是否 hang，如果发现 hang，会发起 reboot。

以下进程为 11.2 特有：

3.2.8. ohasd.bin

在 GI 启动时，最先启动的就是 ohasd,然后由它启动 agent(orarootagent,oraagent, cssdagnet 和 cssdmonitor)，各个 agent 再启动对应的进程。如果 GI 启动过程有问题，可以查看一下 ohasd 的日志：<GRID_HOME>/log/<host>/agent/ohasd

3.2.9. cssdagent(11.2)

这个进程由 ohasd 启动，然后由它来启动、停止、检查 ocssd 进程，以 root 身份运行。它的日志位于：<GRID_HOME>/log/<host>/agent/ohasd/oracssdagent_root

3.2.10. cssdmonitor(11.2)

监控 cssdagent，并且检查节点 hang（类似于 oprocd），监控 ocssd 进程是否 hang（类似于 oclsomon）,监控 vendor clusterware（类似于 vmon），以 root 身份运行。它的日志位于：<GRID_HOME>/log/<host>/agent/ohasd/oracssdmonitor_root

3.2.11. mdnsd.bin

这个进程通过多播（Multicast）发现集群中的节点和所有的网卡信息。一定要确定集群中的网卡支持多播，而且节点间的通信正常。它的日志位于：
<GRID_HOME>/log/<host>/mdnsd

3.2.12. gpnpd.bin (Grid Plug and Play Daemon)

发布构建集群所需要的 bootstrap 信息，并且在集群的所有节点之间同步 gpnp profile。它的日志位于：<GRID_HOME>/log/<host>/gpnpd

首先，gpnp profile(这是个 xml 文件)用于存放构建集群的 bootstrap 信息，或者可以称为构建集群的最基本的信息，其中包括，集群名称，集群 GUID，ASM discovery string，公网和私网信息等等。所以，当我们在启动集群的某一个节点时，需要读取这个文件(默认文件名为$GRID_HOME/gpnp/$hostname/profiles/peer/profile.xml)，从而获得构建集群的基本信息。另外，由于这个文件中保存的是整个集群的基本信息，所以这个文件在所有节点之间都应该是相同的。同时，我们还需要一个守护进程，也就是 gpnpd.bin(资源名为 ora.gpnpd) 来对 gpnp profile 进行维护。

```
<gpnp:HostNetwork id="gen" HostName="*">
<gpnp:Network id="net1" IP="192.168.56.0" Adapter="eth0" Use="public"/>
<gpnp:Network id="net2" IP="10.10.1.0" Adapter="eth2" Use="cluster_interconnect"/>
</gpnp:HostNetwork>
<orcl:CSS-Profile id="css" DiscoveryString="+asm" LeaseDuration="400"/>
<orcl:ASM-Profile id="asm" DiscoveryString="/dev/asm*"
SPFile="+GRID_ACCT/acct/spfileacct.ora"/>
```

下面，我们对 gpnpd 守护进程的功能进行一些介绍。

1. 这个进程是由 ohasd 的 oraagent 负责管理的。

2. 通过 gpnp 的 wallet 文件进行验证并负责读取 gpnp prfofile。当然，如果 gpnpd 发现本地的 gpnp profile 无法读取，会尝试从 OLR 的信息中重建 gpnp profile。

```
[root@rac11g2 rac11g2]# ocrcheck -local
[root@rac11g2 rac11g2]# ocrdump -local -stdout -xml
```

3. 对 gpnp profile 的客户（例如 ocssd.bin）发布信息。

4. 发现集群中其他节点的 gpnpd 守护进程，如果有需要，通过 mdns 同步在节点间同步 gpnp profile。

5. 如果集群的配置发生改变，有必要的话，修改 gpnp profile.

```
[root@rac11g1 ~]# gpnptool get -o-
<?xml version="1.0" encoding="UTF-8"?>
<gpnp:GPnP-Profile Version="1.0" xmlns="http://www.grid-pnp.org/2005/11/gpnp-profile"
xmlns:gpnp="http://www.grid-pnp.org/2005/11/gpnp-profile"
xmlns:orcl="http://www.oracle.com/gpnp/2005/11/gpnp-profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.grid-pnp.org/2005/11/gpnp-profile gpnp-profile.xsd"
ProfileSequence="6" ClusterUId="1423a3327c1c5fd1ff69f6dec4e6add4"
ClusterName="rac11g-cluster" PALocation="">
<gpnp:Network-Profile>
<gpnp:HostNetwork id="gen" HostName="*">
<gpnp:Network id="net1" IP="192.168.56.0" Adapter="eth0" Use="public"/>
<gpnp:Network id="net2" IP="10.10.1.0" Adapter="eth2" Use="cluster_interconnect"/>
</gpnp:HostNetwork>
</gpnp:Network-Profile>
<orcl:CSS-Profile id="css" DiscoveryString="+asm" LeaseDuration="400"/>
<orcl:ASM-Profile id="asm" DiscoveryString="/dev/asm*"
SPFile="+GRID_ACCT/acct/spfileacct.ora"/>
......
</gpnp:GPnP-Profile>
```

3.2.13. gipcd.bin

这个进程负责管理集群中所有的私网（cluster interconnect）网卡。私网信息是通过 gpnpd 获得的。它的日志位于：<GRID_HOME>/log/<host>/gipcd

集群私网主要负责两类数据的通信。第一种：集群层面的数据通信，例如：ocssd.bin 网络心跳，crsd.bin 之间的通信等；第二种：oracle RAC 通信，例如：ASM 实例间的通信，数据库实例间的通信等。而且，第二种数据通信的工作负载要远远大于第一种。

3.2.14. gnsd(可选)

Grid Naming Service. 相当于子 DNS，功能和 DNS 类似，会取代使用/etc/hosts 进行主机的解析。它的日志位于：<GRID_HOME>/log/<host>/gnsd

3.2.15. octssd.bin:

The Cluster Time Sync Service(CTSS) 用于各个节点间的时钟同步，集群中的一个节点的时钟会作为参照节点，其它节点和这个节点进行时钟同步。注意：当第三方时间同步软件（例如：NTP） 存在时，CTSS 会以‘观察者’的方式运行，并不修改节点时间，但是，如果 CTSS 没有发现第三方时间同步软件，它会开始修改节点时间以便和参考节点同步。它的日志位于：<GRID_HOME>/log/<host>/ctssd。

3.2.16. osysmond.bin

这是 Oracle Cluster Health Monitor(CHM)的主要进程，这个进程在所有节点都会运行，sysmond 会将每个节点的资源使用情况发送给 cluster logger service，后者将会把所有节点的信息都接收并保存到 CHM 的资料库。它的日志位于：
<GRID_HOME>/log/<host>/crfmond/crfmond.log

3.2.17. ologgerd

这是 Oracle Cluster Health Monitor(CHM)的另一个主要进程，在一个集群中的，ologgerd 会有一个主机点(master)，还有一个备节点(standby)。当 ologgerd 在当前的节点遇到问题无法启动后，它会在备用节点启用。它的日志位于：
<GRID_HOME>/log/<host>/crflogd/crflogd.log

3.2.18. <XXX>agent.bin

在 11.2，各个资源的启动、停止和检查都是由 agent 来执行的。ohasd 会把这些 agent 启动。

Agent 包括两种

oraagent.bin
orarootagent.bin
oraagent.bin
orarootagent.bin
scriptagent.bin
oraagent.bin

- 一种是 ohasd 的：

oraagent_grid： 启动/停止/检查/清除 ora.asm, ora.evmd, ora.gipcd, ora.gpnpd, ora.mdnsd 等资源。

orarootagent_root：启动/停止/检查/清除 ora.crsd, ora.ctssd, ora.diskmon, ora.drivers.acfs, ora.crf (11.2.0.2)等资源。

oracssdagent_root: 启动/停止/检查 ocssd 进程。

oracssdmonitor_root：监控 cssdagent 进程。

它们的日志位于：<GRID_HOME>/log/<host>/agent/ohasd

- 另一种是 crsd 的：

oraagent_grid: 启动/停止/检查/清除 asm, ora.eons, ora.LISTENER.lsnr, SCAN listeners, ora.ons, diskgroup 等资源

oraagent_oracle: 启动/停止/检查/清除 service, database 等资源

orarootagent_root: 启动/停止/检查/清除 GNS, VIP, SCAN VIP and network 等资源.

scriptagent_grid: 应用服务定制的服务。

它们的日志位于：<GRID_HOME>/log/<host>/agent/crsd

3.2.19. 核心进程启动依赖

```
[grid@rac11g1 agent]$ crsctl stat res ora.cssd -init -dependency
```

```
==============================================================================
Resource Start Dependencies
==============================================================================
-----------------------------------ora.cssd-----------------------------------
ora.cssd(ora.cssd.type)->
| ora.diskmon(ora.diskmon.type)[weak:concurrent]
| | ora.cssd(ora.cssd.type)[weak:concurrent,pullup:always]
| ora.cssdmonitor(ora.cssdmonitor.type)[hard]
| | ora.gpnpd(ora.gpnp.type)[weak]
| | | ora.mdnsd(ora.mdns.type)[weak]
| | ora.cssd(ora.cssd.type)[pullup:always]
| ora.gpnpd(ora.gpnp.type)[hard,pullup]
| | ora.mdnsd(ora.mdns.type)[weak]
| ora.gipcd(ora.gipc.type)[hard,pullup]
| | ora.gpnpd(ora.gpnp.type)[hard]
| | | ora.mdnsd(ora.mdns.type)[weak]
------------------------------------------------------------------------------
```

3.3. 集群数据库核心进程

New Background Processes In 11g (Doc ID 444149.1)
Database Administration -> Background Processes

```
[oracle@nmacct1 ~]$ ps -ef | grep -E 'lmon|lmd|lms' | grep -v grep
oracle   28459     1  0 19:39 ?       00:00:03 ora_lmon_acct1
```

```
oracle   28461     1  0 19:39 ?       00:00:05 ora_lmd0_acct1
oracle   28463     1  1 19:39 ?       00:00:10 ora_lms0_acct1
```

- LMON：全局派队服务监视器
- LMD0：全局派队服务守护程序
- LMSx：全局高速缓存服务进程，其中 x 的值介于 0 和 j 之间
- LCK0：锁进程
- DIAG：可诊断性进程

3.3.1. LMD: Global Enqueue Service Daemon

LMD 进程主要处理从远程节点发出的资源请求。大概过程如下：

+ 一个连接发出了 global enqueue 请求
+ 这个请求会被发给本节点的 LMD0 进程
+ 这个前台进程会处于等待状态
+ LMD0 会找到这个资源的 master 节点是谁
+ LMD0 会把这个请求发送给 master 节点
+ 如果需要的话，master 节点会增加一个新的 master 资源
+ 这时从 master 节点可以获知谁是 owner, waiter
+ 当这个资源被 grant 给 requestor 后，master 节点的 LMD0 进程会告知 requestor 节点的 LMD0
+ 然后 requestor 节点的 LMD0 会通知申请资源的前台进程

也就是说 LMD 主要处理 global enqueue 的请求，而 LCK0 主要处理本实例的 lock。另外，RAC 上的 global deadlock 也是由 LMD 来发现的。

3.3.2. LCK0: Instance Enqueue Process

LCK0 进程主要处理非 cache fustion 的资源请求，比如 library 和 row cache 请求。

LCK0 处理在实例一级的锁：

- Row cache entries
- Library cache entries
- Result cache entries

这些实例级的锁的 owner，waiter 是 LCK0 进程。

只要这个实例的锁的 owner 是 LCK0，那么这个实例的任何一个连接都可以使用这种 cached 的 metedata.

如果本地的实例没有拥有这个 lock，那么需要申请这个 lock，前台进程会等待 DFS Lock Handle。

另外，当 shared pool 出现压力需要释放一些内存来存放新的 cursor 时，LCK 进程会将 dictionary cache 的一些内存进行释放。

3.3.3. LMON:Global Enqueue Service Monitor

LMON 用于监控整个集群的 global enqueues 和 resources，而且会执行 global enqueue recovery。实例异常终止后，会由 LMON 来进行 GCS 内存方面的处理。当一个实例加入或者离开集群后，LMON 会对 lock 和 resource 进行 reconfiguration。另外 LMON 会在不同的实例间进行通讯检查，如果发现对方通讯超时，就会发出节点 eviction，所以很多时候节点发生 eviction 后(ORA-481，ORA-29740 等)，我们需要查看 LMON 的 trace 来了解 eviction 的原因。

还有，在 DRM(Dynamic Resource management)中，LMD 会监控需要进行 remaster 的 queue，然后把任务发送给 LMON 进程，LMON 进程来实施 remaster。

3.3.4. LMS:Global Cache Service Process

LMS 进程会维护在 Global Resource Directory (GRD)中的数据文件以及每个 cached block 的状态。LMS 用于在 RAC 的实例间进行 message 以及数据块的传输。LMS 是 Cache Fusion 的一个重要部分。LMS 进程可以说是 RAC 上最活跃的后台进程，会消耗较多的 CPU.一般每个实例会有多个 LMS 进程，每个 Oracle 版本的默认的 LMS 进程数目会有所不同，大部分版本的默认值是：MIN(CPU_COUNT/2, 2))

3.3.5. DIAG:Diagnostic Capture Process

用来打印诊断信息。diag 进程会响应别的进程发出的 dump 请求，将相关的诊断信息写到 diag trace 文件中。在 RAC 上，当发出 global oradebug 请求时，会由每个实例的 diag 进程来打印诊断信息到 diag trace 中。

比如：下面的命令用了“-g”,那么生成的 dump 信息会分别写到每个实例的 diag trace 文件中：

```
SQL>oradebug -g all hanganalyze 3
SQL>oradebug -g all dump systemstate 266
```

3.3.6. ASMB:ASM Background Process

用于和 ASM 实例进行通讯，用来管理 storage 和提供 statistics。当使用 ASMCMD 的 cp 命令时，需要用到 ASM 实例上的 ASMB 进程，数据库实例的 spfile 如果位于存于 ASM 上，那么也会用到 ASMB 进程。如果 OCR 存放在 ASM 中，也会用到 ASMB。

3.3.7. RBAL:ASM Rebalance Master Process

作为 ASM 磁盘组进行 rebalance 时的协调者(Coordinator)。在数据库实例上，由它来管理 ASM 磁盘组。

3.3.8. Onnn:ASM Connection Pool Process

是从数据库实例连接到 ASM 实例上的一些连接池，通过这些连接池，数据库可以发送消息给 ASM 实例。比如，由它将打开文件的请求发送给 ASM 实例，这些连接池只处理一些较短的请求，不处理创建文件这种较长的请求。

3.3.9. PZ:PQ slaves

PZnn 进程（从 99 开始）用于查询 GV$视图，这种查询需要在每个实例上并行执行。如果需要更多的 PZ 进程，会自动生成 PZ98, PZ97,...（降序）。

3.3.10. PING:Interconnect Latency Measurement Process

用来检查集群中各个实例间的私网通讯状况。每个实例每隔几秒会发送给其它实例一些消息，这些消息会由其它实例的 PING 进程收到。发送和接收信息花费的时间会被记录下来并判断是否正常。

3.3.11. LMHB:Global Cache/Enqueue Service Heartbeat Monitor

监控本地的 LMON，LMD，LCK0,RMS0 and LMSn 等进程是否运行正常，是否被阻塞或者已经 hang 了。

3.3.12. RMSn:Oracle RAC Management Process

完成对 RAC 的一些管理任务，比如当一个新的实例加入到集群后，给这个实例创建相关的资源。

3.3.13. RSMN:Remote Slave Monitor Process

管理后台的 slave 进程的创建，作为远程实例的协调者来完成一些任务。

3.3.14. GTXn:Global Transaction Process

在 RAC 环境中对于 XA 事务提供透明支持，维护在 RAC 中的 XA 事务的 global 信息，完成 global 事务的两阶段提交。

3.3.15. RCBG:Result Cache Background Process

这个进程用来处理 RAC 上 Result Cache 相关的消息。

3.3.16. ACMS:Atomic Control File to Memory Service Process

作为每个实例上的 agent 来保证 SGA 的更新在 RAC 的所有实例上都是同步的，或者是全局成功提交，或者由于一些问题而导致全局回滚。

3.4. 操作系统进程

- CRSD 和 RACGIMON：提供高可用性操作的引擎
- OCSSD：提供对节点成员资格和组服务的访问权
- EVMD：扫描调出目录，并调用调出，作为对检测到事件做出的反应
- OPROCD：是集群的进程监视器（不用在 Linux 和 Windows 上）

表决文件实质上由 Cluster Synchronization Services 守护程序使用，用于获取集群中的节点监视信息。

Oracle 集群注册表(OCR) 文件也是 Oracle Clusterware 的一个关键组件。

集群同步服务守护程序(OCSSD)：此进程既可以在供应商集群件环境中运行，也可以在非供应商集群件环境中运行。它能够与现有供应商集群件（如果存在）相集成。OCSSD 的主要工作是监视节点间运行状况（主要使用网络互联和表决磁盘来执行此工作），以及通过组服务发现数据库/ASM 实例端点。OCSSD 以 oracle 用户身份运行，在因故障退出时可以使计算机重新启动，以防止在出现裂脑(split brain) 时数据遭到损坏。

进程监视器守护程序(OPROCD)：此进程是在除 Linux 和 Windows 之外的任意非供应商集群件环境中衍生的，在 Linux 和 Windows 环境中，Oracle Clusterware 将使用内核驱动程序（如 hangcheck-timer）执行相同的操作。如果 OPROCD 检测到问题，它就会终止相应节点。 它以 root 身份运行。此守护程序用于检测计算机硬件问题及驱动程序冻结问题。如果某台计算机被冻结了很长时间，以致其他节点将其从集群中逐出，则该计算机需要自行终止，以防止在集群的其余节点重新配备锁定后向磁盘重新发出任何 I/O。

集群就绪服务守护程序(CRSD)：此进程是高可用性操作的引擎。它管理由 Oracle Clusterware 注册的应用程序，它通过特殊操作脚本启动、停止和检查这些应用程序，并对其进行故障转移。CRSD 可衍生名为 RACGIMON 的专用进程，后者用于监视数据库和 ASM 实例的运行状况，并存储各种功能线程，如快速应用通知(FAN)。将为每个实例衍生一个 RACGIMON 进程。CRSD 将在 OCR（Oracle 集群注册表）中维护配置概要文件以及资源状态。它以 root 身份运行，在出现故障时可以自动重新启动。

此外，CRSD 可以衍生临时子项，用于执行某些特殊操作，例如：

- racgeut（按计时器执行），用于终止在某个时间段后未完成的操作
- racgmdb（管理数据库），用于启动/停止/检查实例
- racgchsn（更改服务名称），用于添加/删除/检查实例的服务名称
- racgons，用于向 OCR 添加/删除 ONS 配置
- racgvip，用于启动/停止/检查实例虚拟 IP

事件管理守护程序(EVMD)：此进程可在发生事件时转发集群事件。它可以衍生一个永久性的子 evmlogger，而后者可根据需要衍生用于调用调出的子项（如 racgevtf）。它以 oracle 身份运行，在出现故障时可以自动重新启动。

3.5. 核心进程

3.5.1. Cluster Synchronization Services (CSS)

Manages the cluster configuration by controlling which nodes are members of the cluster and by notifying members when a node joins or leaves the cluster. If you are using third-party clusterware, then the css process interfaces with your clusterware to manage node membership information.

```
[root@nmacct1 ~]# pstree `ps -ef | grep ocssd.bin | grep -v grep | awk '{print $3}'`
init.cssd─────ocssd.bin─────19*[{ocssd.bin}]

[root@nmacct1 ~]# lsof -p `ps -ef | grep ocssd.bin | grep -v grep | awk '{print $2}'`

[root@nmacct1 ~]# lsof -p `ps -ef | grep ocssd.bin | grep -v grep | awk '{print $2}'` | grep raw | awk '{print $1, $2, $3, $9}' | sort | uniq
ocssd.bin 4512 oracle /dev/raw/raw3
ocssd.bin 4512 oracle /dev/raw/raw4
ocssd.bin 4512 oracle /dev/raw/raw5
```

3.5.2. Cluster Ready Services (CRS)

The primary program for managing high availability operations within a cluster. Anything that the crs process manages is known as a cluster resource which could be a database, an instance, a service, a Listener, a virtual IP (VIP) address, an application process, and so on.

The crs process manages cluster resources based on the resource's configuration information that is stored in the OCR.

When you have installed Oracle RAC, crs monitors the Oracle instance, Listener, and so on, and automatically restarts these components when a failure occurs. By default, the crs process makes **five** attempts to restart a resource and then does not make further restart attempts if the resource does not restart.

```
[root@nmacct1 ~]# pstree `ps -ef | grep init.crsd | grep -v grep |
awk '{print $2}'`
init.crsd---crsd.bin---43*[{crsd.bin}]


init.crsd─────crsd.bin─┬─racgmain─┬─racgeut─────racgvip
                       |          └──2*[{racgmain}]
                       └──44*[{crsd.bin}]


[root@nmacct1 ~]# lsof -p `ps -ef | grep crsd.bin | grep -v grep |
awk '{print $2}'`


[root@nmacct1 ~]# lsof -p `ps -ef | grep crsd.bin | grep -v grep |
awk '{print $2}'` | grep oradata | awk '{print $1, $2, $3, $9}' | sort
| uniq
crsd.bin 3994 root /oradata/ocr/ocr1
crsd.bin 3994 root /oradata/ocr/ocr2
```

3.5.3. Event Management (EVM)

A background process that publishes events that crs creates.

```
[root@nmacct1 ~]# lsof -p `ps -ef | grep evmd.bin | grep -v grep |
awk '{print $2}'` | grep -E 'oradata|raw' | awk '{print $1, $2, $3,
$9}' | sort | uniq
evmd.bin 3967 oracle /oradata/ocr/ocr1
evmd.bin 3967 oracle /oradata/ocr/ocr2
```

3.5.4. Oracle Notification Service (ONS)

A publish and subscribe service for communicating Fast Application Notification (FAN) events.

3.5.5. RACG

Extends clusterware to support Oracle-specific requirements and complex resources. Runs server callout scripts when FAN events occur.

3.5.6. Process Monitor Daemon (OPROCD)

This process is locked in memory to monitor the cluster and provide I/O fencing. OPROCD performs its check, stops running, and if the wake up is beyond the expected time, then OPROCD resets the processor and reboots the node.

3.5.7. 核心服务对应进程

Oracle Clusterware Component	Linux/Unix Process	Windows Services	Windows Processes
Process Monitor Daemon	oprocd (r)	OraFenceService	
RACG	racgmain, racgimon		racgmain.exe racgimon.exe
Oracle Notification Service (ONS)	ons		ons.exe
Event Manager	evmd (r), evmd.bin, evmlogger	OracleEVMService	evmlogger.exe, evmd.exe
Cluster Ready	crsd.bin (r)	OracleCRSService	crsd.exe
Cluster Synchronization Services	init.cssd (r), ocssd (r), ocssd.bin	OracleCSService	ocssd.exe

List of Processes and Windows Services associated with Oracle Clusterware

3.6. 集群软件组件

The SGA size requirements for Oracle RAC are greater than the SGA requirements for single-instance Oracle databases due to Cache Fusion.

To ensure that each Oracle RAC database instance obtains the block that it needs to satisfy a query or transaction, Oracle RAC instances use two processes, the Global Cache Service (GCS) and the Global Enqueue Service (GES).

The GCS and GES maintain records of the statuses of each data file and each cached block using a Global Resource Directory (GRD). The GRD contents are distributed across all of the active instances, which effectively increases the size of the System Global Area for an Oracle RAC instance.

These Oracle RAC processes and the GRD collaborate to enable Cache Fusion. The Oracle RAC processes and their identifiers are as follows:

LMS: Global Cache Service Process

LMD: Global Enqueue Service Daemon

LMON: Global Enqueue Service Monitor

LCK0: Instance Enqueue Process

3.7. 集群高可用性

3.7.1. Voting Disk and Oracle Cluster Registry

- **Voting Disk:** Manages cluster membership by way of a health check and arbitrates cluster ownership among the instances in case of network failures.The voting disk must reside on shared disk. The Oracle Clusterware enables multiple voting disks but you must have an odd number of voting disks, such as three, five, and so on. If you define a single voting disk, then you should use external mirroring to provide redundancy.

- **Oracle Cluster Registry (OCR):** Maintains cluster configuration information as well as configuration information about any cluster database within the cluster.The OCR also manages information about processes that Oracle Clusterware controls. The OCR stores configuration information in a series of key-value pairs within a directory tree structure. The OCR must reside on shared disk that is accessible by all of the nodes in your cluster. The Oracle Clusterware can multiplex the OCR and Oracle recommends that you use this feature to ensure cluster high availability. You can replace a failed OCR online, and you can update the OCR through supported APIs such as Enterprise Manager, the Server Control Utility (SRVCTL), or the Database Configuration Assistant (DBCA).

3.7.2. 管理 Multiple Cluster Interconnects

The CLUSTER_INTERCONNECTS initialization parameter requires the IP address of the interconnect instead of the device name.

The CLUSTER_INTERCONNECTS initialization parameter is useful only in a UNIX-based environments where UDP IPC is enabled.

The CLUSTER_INTERCONNECTS parameter enables you to specify an interconnect for all IPC traffic to include Oracle Global Cache Service (GCS), Global Enqueue Service (GES), and Interprocessor Parallel Query (IPQ).

Specify the CLUSTER_INTERCONNECTS initialization parameter in the parameter file, setting a different value for each database instance.

3.8. OCR (Oracle Cluster Registry)

The OCR contains information about the cluster node list, instance-to-node mapping information, and information about Oracle Clusterware resource profiles for applications that you have customized.

3.8.1. 自动备份 OCR

- OCR 内容对于 Oracle Clusterware 至关重要。
- 实际上，OCR 会在以下时间自动进行备份：
  - 每 4 小时：CRS 会保留最后 3 个副本。
  - 每天结束时：CRS 会保留最后 2 个副本。
  - 每周结束时：CRS 会保留最后 2 个副本。

```
[root@nmacct2 ~]# ocrconfig -showbackup

nmacct1     2013/11/26 10:00:34     /oracle/product/10.2.0/crs/cdata/crs-acct

nmacct2     2013/11/26 05:08:22     /oracle/product/10.2.0/crs/cdata/crs-acct

nmacct2     2013/11/26 01:08:22     /oracle/product/10.2.0/crs/cdata/crs-acct

nmacct2     2013/11/25 13:08:22     /oracle/product/10.2.0/crs/cdata/crs-acct

nmacct2     2013/11/26 01:08:22     /oracle/product/10.2.0/crs/cdata/crs-acct
```

- 更改自动备份的默认位置：

```
# ocrconfig –backuploc /shared/bak
```

3.8.2. 手动备份 OCR

- 每天都应将 OCR 自动备份备份到其他存储设备：
  - 使用首选备份工具。
- 在进行重大更改之前和之后应对 OCR 进行逻辑备份：

```
# ocrconfig –export file name
```

3.8.3. 使用物理备份恢复 OCR

1.找到物理备份：

```
$ ocrconfig –showbackup
```

2.检查其内容：

```
# ocrdump –backupfilefile_name
```

3.停止所有节点上的 Oracle Clusterware：

```
# crsctl stop crs
```

4.还原 OCR 物理备份：

```
# ocrconfig –restore <CRS HOME>/cdata/jfv_clus/day.ocr
```

5.重新启动所有节点上的 Oracle Clusterware：

```
# crsctl start crs
```

6.检查 OCR 完整性：

```
$ cluvfy comp ocr -n all
```

3.8.4. 使用逻辑备份恢复 OCR

1. 找到使用 OCR 导出文件创建的逻辑备份。

2. 停止所有节点上的 Oracle Clusterware：

```
# crsctl stop crs
```

3. 还原逻辑 OCR 备份：

```
# ocrconfig –import /shared/export/ocrback.dmp
```

4. 重新启动所有节点上的 Oracle Clusterware：

```
# crsctl start crs
```

5. 检查 OCR 完整性：

```
$ cluvfy comp ocr -n all
```

3.8.5. OCR 配置信息

- Linux 和 AIX 上的/etc/oracle/ocr.loc 中
- Solaris 和 HP-UX 上的/var/opt/oracle/ocr.loc 中
- Windows 上的注册表项 HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\ocr 中

修复 OCR 配置

3.8.6. 添加，替换， 修复，删除 OCR

The Oracle installation process for Oracle RAC gives you the option of automatically mirroring the OCR. This creates a second OCR to duplicate the original OCR. You can put the mirrored OCR on an Oracle cluster file system disk, on a shared raw device, or on a shared raw logical volume.

Note:

The operations in this section affect the OCR cluster-wide: they change the OCR configuration information in the ocr.loc file on UNIX-based systems and the Registry keys on Windows-based systems. However, the ocrconfig command cannot modify OCR configuration information for nodes that are shut down or for nodes on which Oracle Clusterware is not running.

```
[root@nmacct1 oracle]# ocrconfig
```

```
Name:
      ocrconfig - Configuration tool for Oracle Cluster Registry.
```

```
Synopsis:
       ocrconfig [option]
       option:
              -export <filename> [-s online]
                                          - Export cluster register contents to a file
              -import <filename>             - Import cluster registry contents from a file
              -upgrade [<user> [<group>]]
                                          - Upgrade cluster registry from previous version
              -downgrade [-version <version string>]
                                          - Downgrade cluster registry to the specified
version
              -backuploc <dirname>            - Configure periodic backup location
              -showbackup                    - Show backup information
              -restore <filename>             - Restore from physical backup
              -replace ocr|ocrmirror [<filename>] - Add/replace/remove a OCR device/file
              -overwrite                     - Overwrite OCR configuration on disk
              -repair ocr|ocrmirror <filename>    - Repair local OCR configuration
              -help                          - Print out this help information

Note:
       A log file will be created in
       $ORACLE_HOME/log/<hostname>/client/ocrconfig_<pid>.log. Please ensure
       you have file creation privileges in the above directory before
       running this tool.
```

1. 添加 Oracle Cluster Registry

Oracle RAC environments do not support more than two OCRs, a primary OCR and a second OCR.

Run the following command to add an OCR location using either destination_file or disk to designate the target location of the additional OCR:

```
ocrconfig -replace ocr destination_file or disk
```

Run the following command to add an OCR mirror location using either destination_file or disk to designate the target location of the additional OCR:

```
ocrconfig -replace ocrmirror destination_file or disk
```

添加 mirror 之前，镜像文件必须存在，否则会报错。

```
[root@nmacct1 oracle]# ocrconfig -replace ocrmirror
/oradata/ocr/ocr4
```

PROT-21: Invalid parameter

```
[root@nmacct1 ocr]# ls
ocr1  ocr2  ocr3  votedisk1  votedisk2  votedisk3
[root@nmacct1 ocr]# cp ocr3 ocr4
[root@nmacct1 ocr]# ocrconfig -replace ocrmirror /oradata/ocr/ocr4
```

Note:

You must be root user to run ocrconfig commands.

2. 替换 Oracle Cluster Registry

注意

当 OCR 存放在集群文件系统或 NFS 上时，操作前需要先创建 OCR 文件。

Run the following command to replace the OCR using either destination_file or disk to indicate the target OCR:

```
ocrconfig -replace ocr destination_file or disk
```

Run the following command to replace an OCR mirror location using either destination_file or disk to indicate the target OCR:

```
ocrconfig -replace ocrmirror destination_file or disk
```

3. 修复本地节点 Oracle Cluster Registry 配置

You may need to repair an OCR configuration on a particular node if your OCR configuration changes while that node is stopped.

Note:

You cannot perform this operation on a node on which the Oracle Clusterware daemon is running.

Use the following procedure to repair an OCR configuration:

a. Run the following command to stop Oracle Clusterware on all nodes:

```
crsctl stop crs
```

b. Run the following command on one node to take a backup of the OCR configuration:

```
ocrconfig -export export_filename
```

In the preceding command, export_filename is the name of the of the file to which you backed up OCR. You import this file after you repair the OCR configuration.

c. Run the following command on all nodes to repair the OCR configuration:

```
ocrconfig -repair
```

d. Run the following command to import the backup to the repaired OCR configuration:

```
ocrconfig -import export_filename
```

e. Run the following command on one node to overwrite the OCR configuration on disk:

```
ocrconfig -overwrite
```

f. Run the following command on one node to verify the OCR configuration:

```
ocrcheck
```

4. Removing an Oracle Cluster Registry

To remove an OCR location, at least one other OCR must be online.

a. Ensure that at least one OCR other than the OCR that you are removing is online.

Caution:

Do not perform this OCR removal procedure unless there is at least one other active OCR online.

b. Run the following command on any node in the cluster to remove the OCR:

```
ocrconfig -replace ocr
```

Run the following command on any node in the cluster to remove the mirrored OCR:

```
ocrconfig -replace ocrmirror
```

Note:

When removing an OCR location, the remaining OCR must be online. If you remove a primary OCR, then the mirrored OCR becomes the primary OCR.

3.8.7. 使用 OCR 备份文件进行恢复

The first method uses automatically generated OCR file copies and the second method uses manually created OCR export files.

The Oracle Clusterware automatically creates OCR backups every four hours.

每 4 小时自动备份一次。

The CRSD process that creates the backups also creates and retains an OCR backup for each full day and at theend of each week.

The default location for generating backups on UNIX-based systems is CRS_home/cdata/cluster_name where cluster_name is the name of your cluster.

```
[oracle@nmacct1 ocr]$ ocrconfig -showbackup
nmacct1     2013/12/17 03:38:11
/oracle/product/10.2.0/crs/cdata/crs-acct
nmacct1     2013/12/16 23:38:11
/oracle/product/10.2.0/crs/cdata/crs-acct
```

Note:

You must be root user to run ocrconfig commands. 必须使用 root 用户操作。

1. 使用自动备份恢复

a. Identify the OCR backups using the ocrconfig -showbackup command.

```
[oracle@nmacct2 cssd]$ ocrconfig -showbackup
nmacct1     2013/12/17 03:38:11
/oracle/product/10.2.0/crs/cdata/crs-acct
```

b. Stop Oracle Clusterware on all the nodes in your Oracle RAC cluster by running the following command as root:

```
# crsctl stop crs
```

run the following command to verify that all processes except init.cssd fatal are inactive:

```
# ps -ef|grep cssd
```

c. Perform the restore by applying an OCR backup file that you identified in Step 1 using the following command where file_name is the name of the OCR that you want to restore.

```
[oracle@nmacct2 cssd]$ ocrconfig -restore file_name
```

d. Start Oracle Clusterware on all the nodes in your Oracle RAC cluster by running the following command as root:

```
# crsctl start crs
```

e. Run the following command to verify the OCR integrity where the -n all argument retrieves a listing of all of the cluster nodes that are configured as part of your cluster:

```
[oracle@nmacct2 cssd]$ cluvfy comp ocr -n all [-verbose]
```

3.8.8. 使用 OCRDUMP 和 OCRCHECK 工具诊断 OCR 问题

1. Using the OCRDUMP Utility

Use the OCRDUMP utility to write the OCR contents to a file so that you can examine the OCR content.

2. Using the OCRCHECK Utility

Use the OCRCHECK utility to verify the OCR integrity.

3.9. Votedisk

3.9.1. 备份 Voting Disk

```
dd if=voting_disk_name of=backup_file_name
```

3.9.2. 恢复 Voting Disk

```
dd if=backup_file_name of=voting_disk_name
```

Note:

If you have multiple voting disks, then you can remove the voting disks and add them back into your environment using the crsctl delete css votedisk path and crsctl add css votedisk path commands respectively, where path is the complete path of the location on which the voting disk resides.

- 建议使用符号链接。
- 请使用 dd 命令备份一个表决磁盘。
  - 在安装 Oracle Clusterware 之后
  - 在添加或删除节点之后
  - 可以联机执行

```
$ crsctl query css votedisk
$ dd if=<voting disk path> of=<backup path> bs=4k
```

- 可以使用以下方法恢复表决磁盘：使用 dd 命令恢复第一个表决磁盘，然后根据需要对该磁盘进行多路复用。
- 如果没有可用的表决磁盘备份，则应重新安装 Oracle Clusterware。

3.9.3. 更改 Voting Disk 路径

Run the following command as the root user to add a voting disk:

```
crsctl add css votedisk path
```

Run the following command as the root user to remove a voting disk:

```
crsctl delete css votedisk path
```

- 可以动态更改表决磁盘配置。
- 要添加新的表决磁盘，请使用以下命令：

```
# crsctl add css votedisk<new voting disk path>
```

- 要删除表决磁盘，请使用以下命令：

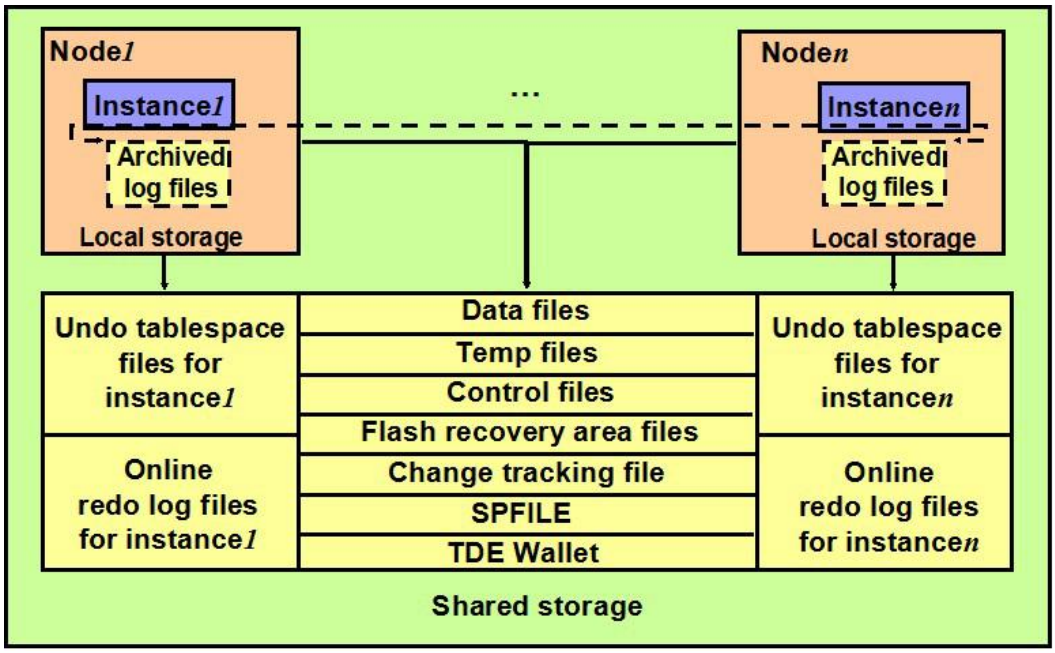
```
# crsctl delete css votedisk <old voting disk path>
```

- 如果所有节点上的 Oracle Clusterware 都已关闭，请使用-force 选项：

```
# crsctl add css votedisk <new voting disk path> -force
# crsctl delete css votedisk <old voting disk path> -force
```

其中 path 是全限定路径。

4. 共享存储



Storage for Oracle Real Application Clusters (Oracle RAC) databases must be shared.

In other words, datafiles must reside in an Automatic Storage Management (ASM) disk group, on a cluster file system, or on shared raw devices.

4.1. 文件系统

4.1.1. Datafile Access in Oracle Real Application Clusters

All Oracle RAC instances must be able to access all datafiles.

Then run the ALTER SYSTEM CHECK DATAFILES statement on each instance to verify datafile access. If you add a datafile to a disk that other instances cannot access, then verification fails.

```
SQL> ALTER SYSTEM CHECK DATAFILES;
System altered.
```

校验信息在 alert.log 日志文件中。

```
Wed Dec 18 17:09:50 CST 2013
Errors in file
/oracle/product/10.2.0/db/admin/acct/bdump/acct1_dbw0_5948.trc:
ORA-01186: file 6 failed verification tests
ORA-01157: cannot identify/lock data file 6 - see DBWR trace file
ORA-01110: data file 6: '/home/oracle/test.dbf'
Wed Dec 18 17:09:50 CST 2013
File 6 not verified due to error ORA-01157
```

4.1.2. Redo Log File Storage in Oracle Real Application Clusters

Each instance has its own online redo log groups which are referred to as an instance's thread of online redo.

Each instance must have at least two groups of online redo log files in its thread. When the current group fills, an instance begins writing to the next log file group.

4.1.3. Automatic Undo Management in Oracle Real Application Clusters

Oracle automatically manages undo segments within a specific undo tablespace that is assigned to an instance. Only the instance assigned to the undo tablespace can modify the contents of that tablespace.

4.2. Automatic Storage Management

ASM automatically optimizes storage to maximize performance by managing the storage configuration across the disks that ASM manages.

4.2.1. The syntax for the DBCA silent mode command is:

```
dbca -silent -nodeList nodelist -configureASM -asmPassword asm_pwd
[-diskList disk_list] [-redundancy redundancy_option]
[-diskGroupName dgname]  [-diskString disk_discovery_string]
[-recoveryGroupName recovery_dgname] [-recoveryRedundancy
redundancy_option]
```

4.2.2. Administering ASM Instances with SRVCTL in Oracle RAC

- add configuration information about an existing ASM instance

```
srvctl add asm -n node_name -i asm_instance_name -o oracle_home
```

- remove an ASM instance

```
srvctl remove asm -n node_name [-i asm_instance_name]
```

- enable an ASM instance

```
srvctl enable asm -n node_name [-i ] asm_instance_name
```

- disable an ASM instance

```
srvctl disable asm -n node_name [-i asm_instance_name]
```

- start an ASM instance

```
srvctl start asm -n node_name [-i asm_instance_name] [-o start_options]
[-c <connect_str> | -q]
```

- stop an ASM instance

```
srvctl stop asm -n node_name [-i asm_instance_name] [-o stop_options]
[-c <connect_str> | -q]
```

- configuration of an ASM instance

```
srvctl config asm -n node_name
```

- status of an ASM instance

```
srvctl status asm -n node_name
```

5. 集群高可用

5.1. 高可用工作负载介绍

- Services

  Oracle Database 10g introduces a powerful automatic workload management facility, called services, to enable the enterprise grid vision.

- Connection Load Balancing

  A feature of Oracle Net Services that balances incoming connections across all of the instances that provide the requested database service.

- High Availability Framework

  An Oracle RAC component that enables the Oracle Database to maintain components in a running state at all times.

- Fast Application Notification(FAN)

  The notification mechanism that Oracle RAC uses to quickly alert applications about configuration and workload service level changes.

- Load Balancing Advisory

  Provides information to applications about the current service levels that the database and its instances are providing.

- Fast Connection Failover

  This is the ability of Oracle Clients to provide rapid failover of connections by subscribing to FAN events.

- Runtime Connection Load Balancing

  This is the ability of Oracle Clients to provide intelligent allocations of connections in the connection pool based on the current service level provided by the database instances when applications request a connection to complete some work.

5.2. Service Deployment Options

5.2.1. Using Oracle Services

When you define a service, you define which instances normally support that service. These are known as the PREFERRED (首选) instances. You can also define other instances to support a service if the service's preferred instance fails. These are known as AVAILABLE (可用) instances.

When a service moves to an available instance, Oracle does not move the service back to the PREFERRED instance when the PREFERRED instance restarts because:

- The service is already running on the desired number of instances
- Maintaining the service on the current instance provides a higher level of service availability
- Not moving the service back to the initial PREFERRED instance prevents a second outage

You can, however, easily automate fail back by using FAN callouts.

5.2.2. Default Service Connections

The database also supports the following two internal services:

- SYS$BACKGROUND is used by the background processes only
- SYS$USERS is the default service for user sessions that are not associated with any application service

5.2.3. Connection Load Balancing

There are two types of load balancing that you can implement: client-side and server-side load balancing. 负载均衡分为服务器端和客户端两种。

Client-side load balancing balances the connection requests across the Listeners.

With server-side load balancing, the Listener directs a connection request to the best instance currently providing the service by using the load balancing advisory.

- **Long**: Use the LONG connection load balancing method for applications that have long-lived connections.

```
EXECUTE DBMS_SERVICE.MODIFY_SERVICE (service_name => 'POSTMAN'
       , clb_goal => DBMS_SERVICE.CLB_GOAL_LONG);
```

- **Short**: Use the SHORT connection load balancing method for applications that have short-lived connections.

```
EXECUTE DBMS_SERVICE.MODIFY_SERVICE (service_name => 'ORDER'
       , CLB_GOAL => DBMS_SERVICE.CLB_GOAL_SHORT);
```

5.3. Fast Application Notification

FAN is a notification mechanism that Oracle RAC uses to notify other processes about configuration and service level information such as includes service status changes, such as UP or DOWN events.

For cluster configuration changes, the Oracle RAC high availability framework publishes a FAN event immediately when a state change occurs in the cluster.

5.4. Administering Services with SRVCTL

5.4.1. Creating Services with SRVCTL

```
srvctl add service -d database_unique_name -s service_name -r
preferred_list
[-a available_list] [-P TAF_policy]
```

5.4.2. Starting and Stopping Services with SRVCTL

```
srvctl start service -d database_unique_name [-s service_name_list]
[-i inst_name]
[-o start_options] [-c connect_str | -q]
```

```
srvctl stop service -d database_unique_name -s service_name_list [-i
inst_name]
[-o start_options] [-c connect_str | -q]
```

5.4.3. Enabling and Disabling Services with SRVCTL

```
srvctl enable service -d database_unique_name -s service_name_list
[-i inst_name]
```

```
srvctl disable service -d database_unique_name -s service_name_list
[-i inst_name]
```

5.4.4. Relocating Services with SRVCTL

```
srvctl relocate service -d apps -s crm -i apps1 -t apps3
```

5.4.5. Obtaining the Statuses of Services with SRVCTL

```
srvctl status service -d apps -s crm
```

5.4.6. Obtaining the Configuration of Services with SRVCTL

```
srvctl config service -d apps -s crm -a
```

6. 内核参数

6.1. HP

6.1.1. udp_smallest_anon_port

使用默认 udp_smallest_anon_port 设置，当系统压力较大时，会出现进程无法连接数据库的问题。这次 alert 日志中，会有错误日志，信息与 OS IPC 有关，该错误的错误代码为：227。

```
Errors in file
/oracle/OracleHomes/admin/bsjfdb/udump/bsjfdb2_ora_18719.trc:
ORA-00603: ORACLE server session terminated by fatal error
ORA-27504: IPC error creating OSD context
ORA-27300: OS system dependent operation:bind failed with status: 227
ORA-27301: OS failure message: Can't assign requested address
ORA-27302: failure occurred at: sskgxpcre3
```

```
# define ENOTSOCK 216 /* Socket operation on non-socket */
# define EADDRNOTAVAIL 227 /* Can't assign requested address */
```

判断该问题的方法：

1) 系统参数检查：

```
$ ndd -get /dev/udp udp_largest_anon_port
$ ndd -get /dev/udp udp_smallest_anon_port
```

2) 检查当时系统使用的 udp 数量

```
$ netstat -an | grep -i udp | wc -l
```

3) 步骤 1 与步骤 2 进行对比

当步骤 2 返回的值大于步骤 1 中 (udp_largest_anon_port - udp_smallest_anon_port) 的值，系统会出现该错误。Metalink 中指出，该错误是系统参数设置问题，而非 oracle 的 bug。

因此解决该问题的办法，是调整操作系统参数。

- **HP-UX 操作系统修改办法如下：**

For TCP we use:

```
# /usr/bin/ndd -set /dev/tcp tcp_smallest_anon_port 15000
# /usr/bin/ndd -set /dev/tcp tcp_largest_anon_port 61000
```

For UDP connections we use:

```
# /usr/bin/ndd -set /dev/udp udp_smallest_anon_port 15000
# /usr/bin/ndd -set /dev/udp udp_largest_anon_port 61000
```

To make the change persistent after reboot we can append the following entries in /etc/rc.config.d/nddconf

```
TRANSPORT_NAME[0]=tcp
NDD_NAME[0]=tcp_largest_anon_port
NDD_VALUE[0]=65500

TRANSPORT_NAME[1]=tcp
NDD_NAME[1]=tcp_smallest_anon_port
NDD_VALUE[1]=9000

TRANSPORT_NAME[0]=udp
NDD_NAME[0]=udp_largest_anon_port
NDD_VALUE[0]=65500

TRANSPORT_NAME[1]=udp
NDD_NAME[1]=udp_smallest_anon_port
NDD_VALUE[1]=9000
```

- **Linux 操作系统修改办法如下：**

```
echo "15000 61000" > /proc/sys/net/ipv4/ip_local_port_range
```

To make this change persistent after reboots, we can use sysctl. Edit /etc/sysctl.conf and append the entry:

```
net.ipv4.ip_local_port_range = 15000 61000

# sysctl -p
```

7. Cluster Verification Utility

7.1. 11g (使用 grid 用户运行)

```
cluvfy comp crs [ -n node_list] [-verbose]
cluvfy comp ocr [ -n node_list] [-verbose]
cluvfy comp clu [-verbose]
cluvfy comp nodereach -n node_list [ -srcnode node ] [-verbose]
cluvfy comp nodecon -n node_list -i interface_list [-verbose]
```

7.2. Enabling Tracing

```
set SRVM_TRACE=true; export SRVM_TRACE
```

日志文件存放在 CRS_home/cv/log/cvutrace.log.0 目录下。

8. 集群日志代码命名规则

PRKA—Cluster Node Applications Messages

PRKC—Cluster Command Messages

PRKD—Global Services Daemon Messages

PRKE—Global Services Daemon Controller Utility Messages

PRKH—Server Manager (SRVM) Messages

PRKI—Cluster Pre-Install Messages

PRKN—Server Manager (SRVM) System Library Messages

PRKO—Server Control (SRVCTL) Utility Messages

PRKP—Cluster Database Management Messages

PRKR—Cluster Registry Messages

PRKS—Automatic Storage Management Messages

PRKU—Command-Line Parser Utility Messages

PRKV—Virtual IP Configuration Assistant Messages

9. 常用视图

10. 应急操作

10.1. 更改 VIP 地址

1. 确定用于支持 VIP 的接口：

```
$ ifconfig -a
```

2. 停止依赖 VIP 的所有资源：

```
$ srvctl stop instance -d DB -i DB1
$ srvctl stop asm -n node1
# srvctl stop nodeapps -n node1
```

3. 验证 VIP 不再处于运行状态：

```
$ ifconfig -a
$ crs_stat -t
```

4. 更改/etc/hosts 中的 IP 和 DNS。

5.使用 srvctl 修改 VIP 地址：

```
# srvctl modify nodeapps -n node1 -a
192.168.2.125/255.255.255.0/eth0
```

6.启动 nodeapps 及依赖它的所有资源：

```
# srvctl start nodeapps -n node1
```

7.对下一节点重复执行从步骤 1 开始的步骤。

10.2. 更改公用/互联 IP 子网配置

可使用 oifcfg 添加或删除 OCR 中的网络接口信息：

```
$ <CRS HOME>/bin/oifcfg getif
eth0 139.2.156.0 global public
eth1 192.168.0.0 global cluster_interconnect
```

```
$ oifcfg delif -global eth0
$ oifcfg setif –global eth0/139.2.166.0:public

$ oifcfg delif –global eth1
$ oifcfg setif –global eth1/192.168.1.0:cluster_interconnect

$ oifcfg getif
eth0 139.2.166.0 global public
eth1 192.168.1.0 global cluster_interconnect
```

10.3. 所有 OCR 文件损坏

```
# crsctl start crs –excl (on one node only)
# crsctl delete css votedisk FUID
# crsctl add css votedisk path_to_voting_disk
```

11. RMAN 配置

11.1. Configuring Channels to Use a Specific Channel

```
CONFIGURE CHANNEL DEVICE TYPE sbt CONNECT
'SYS/change_on_install@node1'
CONFIGURE CHANNEL DEVICE TYPE sbt CONNECT
'SYS/change_on_install@node2'
```

Therefore, the node that is performing the recovery must be able to access all of the archived logs that are needed for the recovery operation.

You can use the PARALLEL clause in the RECOVER command to change this number.

11.2. Guidelines and Considerations for Archived Redo Logs

For any archived redo log configuration, uniquely identify the archived redo logs with the LOG_ARCHIVE_FORMAT parameter.

Parameter	Description	Example
%r	Resetlogs identifier	log_1_62_23452345
%R	Padded resetlogs identifier	log_1_62_0023452345
%s	Log sequence number, not padded	log_251
%S	Log sequence number, left-zero-padded	log_0000000251
%t	Thread number, not padded	log_1
%T	Thread number, left-zero-padded	log_0001

11.2.1. Initialization Parameter Settings for the Cluster File System Archiving Scheme

In the cluster file system scheme, each node archives to a directory that is identified with the same name on all instances within the cluster database.

```
sid1.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
sid2.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
sid3.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
```

11.2.2. Initialization Parameter Settings for Non-Cluster File System Local Archiving

In the non-cluster file system local archiving scheme, each node archives to a uniquely named local directory. If recovery is required, then you can configure the recovery node so that it can access directories on the other nodes remotely.

```
sid1.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_1"
sid2.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_2"
sid3.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_3"
```

11.3. 备份恢复

In a non-cluster file system environment, each node can back up only its own local archived redo logs.

You do not need to perform a full backup after a resetlogs operation.

RMAN backs up earlier incarnation logs on running BACKUP ARCHIVELOG ALL or BACKUP ARCHIVELOG FROM TIME or BACKUP ARCHIVELOG FROM SCN command.

```
RUN
{
   ALLOCATE CHANNEL CH1 CONNECT 'user1/pwd1@node1';
   ALLOCATE CHANNEL CH2 CONNECT 'user2/pwd2@node2';
   ALLOCATE CHANNEL CH3 CONNECT 'user3/pwd3@node3';
   BACKUP DATABASE PLUS ARCHIVED LOG;
}
```

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT
'SYS/oracle@node1';
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT
'SYS/oracle@node2';
```

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT
'SYS/oracle@node3';
DELETE ARCHIVELOG ALL BACKED UP n TIMES TO DEVICE TYPE device_type;
```

12. 诊断 RAC 集群

12.1. 集群 debug

12.1.1. Dynamic Debugging

```
crsctl debug log crs "CRSRTI:1,CRSCOMM:2"
crsctl debug log evm "EVMCOMM:1"
crsctl debug log res "resname:1"
```

Debugging information remains in the Oracle Cluster Registry (OCR) for use during the next startup.

12.1.2. Component Level Debugging

Run the following command to obtain component names where module_name is the name of the module, crs, evm, or css:

```
crsctl lsmodules module_name
```

Running crsctl commands as follows where module_name is the name of the module, crs, evm, or css and debugging_level is a level from 1 to 5:

```
crsctl debug log module_name component:debugging_level
```

12.1.3. Resource Debugging

You can use crsctl commands to enable resource debugging using the following syntax:

```
[root@nmacct1 nmacct1]# crs_stat -p ora.nmacct1.vip
crsctl debug log res "ora.nmacct1.vip:1"
```

This has the effect of setting the environment variable USER_ORA_DEBUG, to 1, before running the start, stop, or check action scripts for the ora.node1.vip resource.

12.2. 集群起停管理 (必须使用 root 用户)

You can use a crsctl command as follows to stop Oracle Clusterware and its related resources on a specific node:

```
crsctl stop crs
```

You can use a crsctl command as follows to start Oracle Clusterware and its related resources on a specific node:

```
crsctl start crs
```

12.3. 集群守护进程自启动管理

Run the following command to enable startup for all of the Oracle Clusterware daemons:

```
crsctl enable crs
```

Run the following command to disable the startup of all of the Oracle Clusterware daemons:

```
crsctl disable crs
```

12.4. 集群时间同步

12.4.1. 11G CTSS

```
# crsctl check ctss
CRS-4701: The Cluster Time Synchronization Service is in Active mode.
CRS-4702: Offset (in msec): 0
```

12.4.2. 使用何种时间同步

```
$ cluvfy comp clocksync -verbose
```

12.5. 诊断信息收集脚本 (必须使用 root 用户)

Use the diagcollection.pl script to collect diagnostic information from an Oracle Clusterware installation.

$CRS_HOME/bin/diagcollection.pl

12.6. 集群健康状态检测

Use the crsctl check command to determine the health of your clusterware as in the following example:

```
Usage: crsctl check  crs        - checks the viability of the CRS
stack
      crsctl check  cssd       - checks the viability of CSS
      crsctl check  crsd       - checks the viability of CRS
      crsctl check  evmd      - checks the viability of EVM
```

12.7. 集群日志文件

Oracle retains one current log file and five older log files that are 20 MB in size (120 MB of storage) for the cssd process, and one current log file and 10 older log files that are 10 MB in size (110 MB of storage) for the crsd process.

12.7.1. The Cluster Ready Services Daemon (crsd) 日志文件

Log files for the CRSD process (crsd) can be found in the following directories:

$CRS_HOME/log/`hostname`/crsd

12.7.2. Oracle Cluster Registry (OCR)日志文件

The Oracle Cluster Registry (OCR) records log information in the following location:

$CRS_HOME/log/`hostname`/client

12.7.3. Cluster Synchronization Services (CSS) 日志文件

You can find CSS information that the OCSSD generates in log files in the following locations:

$CRS_HOME/log//`hostname`/cssd

12.7.4. Event Manager (EVM) 日志

Event Manager (EVM) information generated by evmd is recorded in log files in the following locations:

$CRS_HOME/log//`hostname`/evmd

12.7.5. RACG Log Files

The Oracle RAC high availability trace files are located in the following two locations:

CRS home/log/hostname/racg

$ORACLE_HOME/log/hostname/racg

12.8. 诊断 Oracle Cluster Registry (OCR)

12.8.1. 使用 OCRDUMP 查看 Oracle Cluster Registry 内容

The OCRDUMP utility enables you to view the OCR contents by writing OCR content to a file or stdout in a readable format.

OCRDUMP also creates a log file in CRS_Home/log/hostname/client. To change the amount of logging, edit the file CRS_Home/srvm/admin/ocrlog.ini.

Note:

Make sure that you have file creation privileges in the CRS Home/log/hostname/client directory before using the OCRDUMP utility.

12.8.2. OCRDUMP Utility Syntax and Options

```
ocrdump [<filename>|-stdout] [-backupfile <backupfilename>]
[-keyname <keyname>] [-xml] [-noheader]
```

```
ocrdump
ocrdump MYFILE
ocrdump -stdout -keyname SYSTEM
ocrdump -stdout -xml
```

12.8.3. Using the OCRCHECK Utility

The OCRCHECK utility displays the version of the OCR's block format, total space available and used space, OCRID, and the OCR locations that you have configured.

```
[root@nmacct1 ~]# ocrcheck
Status of Oracle Cluster Registry is as follows :
        Version                  :        2
        Total space (kbytes)     :     204576
        Used space (kbytes)      :       4884
        Available space (kbytes) :      199692
        ID                       : 1090417786
        Device/File Name         : /oradata/ocr/ocr2
                                   Device/File integrity check succeeded
        Device/File Name         : /oradata/ocr/ocr1
                                   Device/File integrity check succeeded

        Cluster registry integrity check succeeded
```

OCRCHECK performs a block-by-block checksum operation for all of the blocks in all of the OCRs that you have configured.

OCRCHECK creates a log file in the directory CRS_Home/log/hostname/client. To change amount of logging, edit the file CRS_Home/srvm/admin/ocrlog.ini.

12.9. 诊断 RAC 集群组件

Information about ORA-600 errors appear in the alert_SID.log file for each instance where SID is the instance identifier. For troubleshooting, you may need to also provide files from the following bdump locations:

$ORACLE_HOME/admin/db_name/bdump on UNIX-based systems

%ORACLE_HOME%\admin\db_name\bdump on Windows-based systems

12.10. Cluster Verification Utility

You do not have to be the root user to use the CVU and the CVU assumes that the current user is the oracle user.

You can enable tracing by setting the environment variable SRVM_TRACE to true.

The CVU trace files are created in the CV_HOME/cv/log directory.

Oracle automatically rotates the log files and the most recently created log file has the name cvutrace.log.0.

You can use CVU's configuration file to define specific inputs for the execution of the CVU. The path for the configuration file is CV_HOME/cv/admin/cvu_config.

12.10.1. Cluster Verification Utility 系统检测

```
cluvfy comp sys [ -n node_list ] -p { crs | database } }  [-r { 10gR1 | 10gR2 } ] [ -osdba osdba_group ] [ -orainv orainventory_group ] [-verbose]
```

12.10.2. Cluster Verification Utility 存储校验

```
cluvfy comp ssa [ -n node_list ] [ -s storageID_list ] [-verbose]
cluvfy comp ssa  -n all -verbose
cluvfy comp ssa  -n all -s /dev/sda

cluvfy comp space  [ -n node_list ] -l storage_location -z disk_space {B|K|M|G}  [-verbose]
cluvfy comp space  -n all -l  / home/dbadmin/products  –z 2G -verbose
cluvfy comp cfs  [ -n node_list ] -f file_system [-verbose]
cluvfy comp cfs -f /oradbshare –n all -verbose
```

12.10.3. Cluster Verification Utility 连接测试

```
cluvfy comp nodereach -n node_list [ -srcnode node ]  [-verbose]
cluvfy comp nodecon -n node_list [ -i interface_list ]  [-verbose]
cluvfy comp nodecon -n all [-verbose]
cluvfy comp nodecon -n node1,node2,node3 –i eth0 -verbose
```

12.10.4. Cluster Verification Utility 用户权限验证

```
cluvfy comp admprv  [ -n node_list ]  [-verbose]
              | -o user_equiv [-sshonly]
              | -o crs_inst  [-orainv orainventory_group ]
              | -o db_inst  [-orainv orainventory_group ] [-osdba
osdba_group ]
              | -o db_config  -d oracle_home
```

12.10.5. Cluster Verification Utility 节点校验

```
cluvfy comp peer [ -refnode node ] -n node_list [-r { 10gR1 | 10gR2 } ]
[ -orainv orainventory_group ] [ -osdba osdba_group ]  [-verbose]
cluvfy comp peer -n node_list [-r 10gR2] [-verbose]
```

12.10.6. Cluster Verification Utility 安装校验

```
cluvfy stage -pre crsinst -n node_list
 [ -c ocr_location ] [-r { 10gR1 | 10gR2 } ][ -q voting_disk ]
 [ -osdba osdba_group ]
 [ -orainv orainventory_group ]  [-verbose]
```

12.10.7. 解决 CVU 运行异常的办法

1. 开启 CVU 日志监控

```
[oracle@nmacct1 ~]$ export SRVM_TRACE=TRUE
```

2. 查看 CVU 调用日志

```
[oracle@nmacct1 ~]$ cd $CRS_HOME/cv/log
[oracle@nmacct1 log]$ vi cvutrace.log.0
```

日志中存在脚本没有权限的问题

```
[Worker 1] [10:25:55:260] [RuntimeExec.runCommand:158]
/tmp/CVU_10.2.0.5.0.1_oinstall/exectask.sh: line 12: exec:
/tmp/CVU_10.2.0.5.0.1_oinstall/exectask: cannot execute: Permission
denied; Mon Dec 30 10:25:55 CST 2013
```

3. 修改脚本执行权限

修改的脚本为$CRS_HOME/cv/remenv 目录下的脚本,这些脚本在运行时会被拷贝到/tmp 目录下。

```
[oracle@nmacct1 crs]$ find ./ -name exectask.sh
./cv/remenv/exectask.sh

[oracle@nmacct1 cv]$ cd remenv/
[oracle@nmacct1 remenv]$ ls
exectask  exectask.sh
[oracle@nmacct1 remenv]$ chmod 755 *
```

4. 重新执行 cvu 脚本

```
[oracle@nmacct1 ~]$ cluvfy stage -pre crsinst -n all

Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "nmacct1".


Checking user equivalence...
User equivalence check passed for user "oracle".

Checking administrative privileges...
User existence check passed for "oracle".
Group existence check passed for "oinstall".
Membership check for user "oracle" in group "oinstall" [as Primary]
passed.

Administrative privileges check passed.
```

12.11. 11G ADRCI 命令行诊断工具

```
adrci> help

adrci> show home;
```

```
adrci> set home diag/rdbms/acct/acct1

adrci> help show tracefile;

adrci> show tracefile %lmon% -t

adrci> show alert -tail 20
```

13. 日常维护命令

13.1. 检查点

ALTER SYSTEM SET CHECKPOINT LOCAL 语句只会影响当前连接到的实例，而不会影响默认实例或所有实例。

ALTER SYSTEM CHECKPOINT LOCAL 会影响当前实例。

ALTER SYSTEM CHECKPOINT 或 ALTER SYSTEM CHECKPOINT GLOBAL 会影响集群数据库中的所有实例。

13.1.1. 当前实例

```
ALTER SYSTEM CHECKPOINT LOCAL;
```

13.1.2. 所有实例

```
ALTER SYSTEM CHECKPOINT GLOBAL;
```

13.2. 日志切换

ALTER SYSTEM SWITCH LOGFILE 只会影响当前实例。

要强制执行全局日志切换，请使用 ALTER SYSTEM ARCHIVE LOG CURRENT 语句。

ALTER SYSTEM ARCHIVE LOG 的 INSTANCE 选项使您可以对特定实例的每个联机重做日志文件进行归档。

13.2.1. 当前实例

```
ALTER SYSTEM SWITCH LOGFILE;
```

13.2.2. 所有实例

```
ALTER SYSTEM ARCHIVE LOG CURRENT;
```

13.3. 主机信息检查

13.3.1. 通过指定 IP 地址进行 PING 操作

```
[oracle@nmacct1 ~]$ ping -I 10.10.10.11 -c 2 10.10.10.12
```

13.4. 节点层

13.4.1. olsnodes

```
[oracle@nmacct1 ~]$ olsnodes -n -p -i
nmacct1 1      nmacct1-priv    nmacct1-vip
nmacct2 2      nmacct2-priv    nmacct2-vip
```

看不到所有节点，可以通过 olsnodes -v 检查。

13.5. 集群层

```
[oracle@nmacct1 ~]$ clscfg -concepts
```

13.5.1. 集群运行状态

- 10g

```
[oracle@rac10g1 ~]$ crsctl check crs
```

- 11g

```
[grid@rac11g1 ~]$ crsctl check cluster -all
```

- 资源信息检查

```
[grid@rac11g1 ~]$ crsctl status resource -w "TYPE co 'ora.asm'" -t
```

13.5.2. 超时设置

- MISSCOUNT
  - 表示网络脉动超时
  - 重新配置期间磁盘 I/O 超时的测定
  - 默认值为 30 秒（对于 Linux 为 60 秒）
  - 使用供应商（非 Oracle）集群件时默认值为 600 秒
  - 不能进行更改

```
[oracle@nmacct1 bin]$ crsctl get css misscount
60
```

- DISKTIMEOUT
  - 表示重新配置以外的磁盘 I/O 超时
  - 默认值为 200 秒
  - 当表决磁盘遇到很长时间的 I/O 延迟时可临时进行更改：
    1. 关闭除一个节点之外的所有节点上的 Oracle Clusterware。
    2. 在可用节点上以 root 身份使用以下命令：crsctl set css disktimeout M+1
    3. 重新启动可用节点。
    4. 重新启动所有其他节点。

```
[oracle@nmacct1 bin]$ crsctl get css disktimeout
200
```

13.5.3. 查看集群名称

```
[oracle@nmacct1 ~]$ cemutlo -n
crs-acct
```

13.5.4. 集群通信地址

```
select * from v$cluster_interconnects;
select INST_ID, NAME_KSXPIA, IP_KSXPIA from X$KSXPIA where PUB_KSXPIA
= 'N';
```

13.6. 网络层

13.6.1. OIFCFG

```
[oracle@oeldb1 ~]$ oifcfg

Name:
       oifcfg - Oracle Interface Configuration Tool.

Usage:  oifcfg iflist [-p [-n]]
       oifcfg setif {-node <nodename> | -global}
{<if_name>/<subnet>:<if_type>}...
       oifcfg getif [-node <nodename> | -global] [ -if
<if_name>[/<subnet>] [-type <if_type>] ]
       oifcfg delif {{-node <nodename> | -global}
[<if_name>[/<subnet>]] [-force] | -force}
       oifcfg [-help]

       <nodename> - name of the host, as known to a communications
network
       <if_name>  - name by which the interface is configured in the
system
       <subnet>   - subnet address of the interface
       <if_type>  - type of the interface { cluster_interconnect |
public }
```

- **Public**: An interface that can be used for communication with components external to Oracle RAC instances, such as Oracle Net and Virtual Internet Protocol (VIP) addresses
- **Cluster interconnect**: A private interface used for the cluster interconnect to provide inter-instance or Cache Fusion communication.

```
   If an interface is configured as both a global and a node-specific
interface, the node-specific definition takes precedence over the
global definition. A network interface specification is in the form
of:

A network interface specification is in the form of:
interface_name/subnet:interface_type
```

```
qfe0/204.152.65.32:cluster_interconnect

oifcfg setif -global hme0/139.185.141.0:cluster_interconnect
oifcfg setif -global cms0/139.185.142.0:cluster_interconnect
oifcfg delif -global qfe0/204.152.65.0
oifcfg delif -global
```

13.7. 服务层 SRVCTL

13.7.1. SRVCTL 命令

Command	Description
srvctl add	Adds the node applications, database, database instance, ASM instance, or service.
srvctl remove	Removes the node applications, database, database instance, ASM instance, or service.
srvctl config	Lists the configuration for the node applications, database, ASM instance, or service.
srvctl enable	Enables the database, database instance, ASM instance, or service.
srvctl disable	Disables the database, database instance, ASM instance, or service.
srvctl start	Starts the node applications, database, database instance, ASM instance, or service.
srvctl stop	Stops the node applications, database, database instance, ASM instance, or service.
srvctl modify	Modifies the node applications, database, database instance, or service configuration.
srvctl relocate	Relocates the service from one instance to another.
srvctl status	Obtains the status of the node applications, database, database instance, ASM instance, or service.
srvctl getenv	Displays the environment variable in the configuration for the node applications, database, database

	instance, or service.
srvctl setenv and unsetenv	Sets and unsets the environment variable in the configuration for the node applications, database, database instance, or service.

13.7.2. SRVCTL 对象

Object Noun Name	Abbreviation	Purpose
asm	asm	To add, configure, enable, start, obtain the status of, stop, disable, and remove ASM instances.
database	db	To add, configure, modify, manage environment variables for, enable, start, obtain the status of, stop, disable, and remove databases.
instance	inst	To add, configure, modify, manage environment variables for, enable, start, obtain the status of, stop, and remove database instances.
nodeapps	no abbreviation	To add, configure, modify, manage environment variables for, start, obtain the status of, stop, and remove node applications.
service	serv	To add, configure, modify, manage environment variables for, enable, start, obtain the status of, relocate, disable, stop, and remove services from your cluster database.

13.8. 诊断信息收集

13.8.1. diagcollection.pl

```
[root@nmacct2 ~]# cd $ORA_CRS_HOME
[root@nmacct2 bin]# ./diagcollection.pl -collect
```

开始执行脚本之前，必须以 root 用户身份登录，并且必须设置以下环境变量：
ORACLE_BASE、ORACLE_HOME、ORA_CRS_HOME、HOSTNAME。

basData.tar.gz 主要包含 $ORACLE/BASE/admin 目录中的日志文件。
crsData.tar.gz 文件包含 $ORA_CRS_HOME/log/<hostname> 目录中的日志文
ocrData.tar.gz 文件包含 ocrdump 和 ocrcheck 以及 ocr 备份列表的结果。
oraData.tar.gz 文件包含 $ORACLE_HOME/log/<hostname> 目录中的日志文件。

13.9. Cluvfy

```
[oracle@nmacct2 ~]$ cd $ORA_CRS_HOME
[oracle@nmacct2 bin]$ ./cluvfy stage -post hwos -n "all"
```

- -post hwos：对硬件和操作系统进行后检查
- -pre cfs：对 CFS 设置进行预检查
- -post cfs：对 CFS 设置进行后检查
- -pre crsinst：对 CRS 安装进行预检查
- -post crsinst：对 CRS 安装进行后检查
- -pre dbinst：对数据库安装进行预检查
- -pre dbcfg：对数据库配置进行预检查

13.9.1. -fixup

生成修复脚本

```
cluvfy stage -pre crsinst -n node1 -fixup -fixupdir /db11202/fixit.sh
```

13.10. 主机频繁 root

禁止 crs 自启动

```
[root@nmacct1 ~]# crsctl disable crs
```

解决问题后，在启用自启动

```
[root@nmacct1 ~]# crsctl enable crs
```

13.11. 集群自启动配置

```
[oracle@rac10g1 ~]$ cd /etc/oracle/scls_scr/`hostname`/root
[oracle@rac10g1 root]$ ls
clsomonpid  crsdboot  crsstart  cssfboot  cssrun  daemonpid
noclsmon  noclsvmon  oprocdpid

[oracle@rac10g1 root]$ more crsstart
enable
```

13.12. 集群管理命令

Command	Description
crs_getperm	Inspects the permissions associated with a resource.
crs_profile	Creates, validates, deletes, and updates an Oracle Clusterware application profile
crs_register	Registers configuration information for an application with the OCR.
crs_relocate	Relocates an application profile to another node.
crs_setperm	Sets permissions associated with a resource.
crs_stat	Lists the status of an application profile.
crs_start	Starts applications that have been registered.
crs_stop	Stops an Oracle Clusterware application.
crs_unregister	Removes the configuration information for an application profile from the OCR.

13.13. 集群启动

13.13.1. OHASD 进程启动的情况下，启动所有节点

```
[root@rac11g1 ~]# crsctl start cluster -all
```

13.13.2. OHASD 进程启动的情况下，启动指定节点

```
[root@rac11g1 ~]# crsctl start cluster -n racnode1 racnode4
```

13.13.3. 启动整个集群

```
[root@rac11g1 ~]# crsctl start crs
```

13.14. 集群停止

13.14.1. 停集群资源

```
[root@rac11g1 ~]# crsctl stop cluster -all
```

13.14.2. 停指定节点集群

```
[root@rac11g1 ~]# crsctl stop cluster -n racnode1 racnode3
```

13.14.3. 停集群所有资源

```
[root@rac11g1 ~]# crsctl stop crs -all -f
```

13.15. 集群调试

```
[root@rac11g1 ~]# crsctl lsmodules css
List CSSD Debug Module: CLSF
List CSSD Debug Module: CSSD
List CSSD Debug Module: GIPCCM
```

```
List CSSD Debug Module: GIPCGM
List CSSD Debug Module: GIPCNM
List CSSD Debug Module: GPNP
List CSSD Debug Module: OLR
List CSSD Debug Module: SKGFD

[root@rac11g1 ~]# crsctl debug log css CSSD:1
CRS-4151: DEPRECATED: use crsctl set log {css|crs|evm}
Set CSSD Module: CSSD  Log Level: 1

# crsctl debug log component module:debugging_level
# crsctl debug log res resource_name:debugging_level
```

13.16. 禁用集群

```
[root@rac11g1 ~]# crsctl disable crs
CRS-4621: Oracle High Availability Services autostart is disabled.

[root@rac11g2 root]# pwd
/etc/oracle/scls_scr/rac11g2/root
[root@rac11g2 root]# more ohasdstr
Disable

/var/log/message
Jul 31 07:49:20 rac11g2 logger: Oracle HA daemon is disabled for
autostart.
```

13.17. 启用集群

```
[root@rac11g1 ~]# crsctl enable crs
CRS-4622: Oracle High Availability Services autostart is enabled.

[root@rac11g1 root]# pwd
/etc/oracle/scls_scr/rac11g1/root
[root@rac11g1 root]# more crsstart
enable
```

13.18. 10g 与 11g 集群命令对比

Deprecated Command	Replacement Commands
crs_stat	crsctl check cluster -all crsctl stat res -t
crs_register	crsctl add resource crsctl add type crsctl modify resource crsctl modify type
crs_unregister	crsctl stop resource crsctl modify resource resource_name -attr "AUTO_START=never"
crs_start	crsctl start resource crsctl start crs crsctl start cluster
crs_stop	crsctl stop resource crsctl stop crs crsctl stop cluster
crs_getperm	crsctl getperm resource crsctl getperm type
crs_profile	crsctl add resource crsctl add type crsctl status resource crsctl status type crsctl modify resource crsctl modify type
crs_relocate	crsctl relocate resource
crs_setperm	crsctl setperm resource crsctl setperm type
crsctl check crsd	crsctl check crs
crsctl check cssd	crsctl check css
crsctl check evmd	crsctl check evm
crsctl debug res log resource_name:level	crsctl set log
crsctl set css votedisk	crsctl add css votedisk crsctl delete css votedisk crsctl query css votedisk crsctl replace votedisk

Deprecated Command	Replacement Commands
crsctl start resources	crsctl start resource -all
crsctl stop resources	crsctl stop resource -all

14. 技巧

14.1. 修改连接标识

```
[oracle@nmacct1 ~]$ cd $ORACLE_HOME/sqlplus/admin
[oracle@nmacct1 admin]$ vi glogin.sql
```

```
-- Default for XQUERY
COLUMN result_plus_xquery HEADING 'Result Sequence'
SET SQLPROMPT '_CONNECT_IDENTIFIER> '   --新增
```

```
[oracle@nmacct1 ~]$ sqlplus / as sysdba
```

```
SQL*Plus: Release 10.2.0.5.0 - Production on Thu Dec 19 10:55:47 2013


acct1>
```

15. 官方支持信息

MetaLink Note: 278132.1