Clusterware internals

By

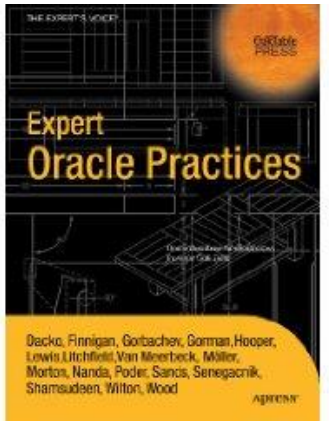
Riyaj Shamsudeen

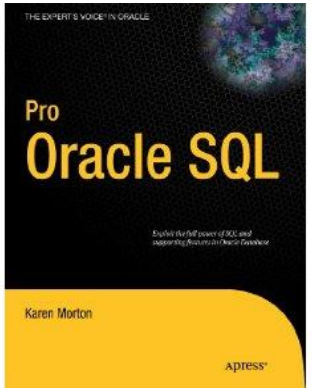
Me

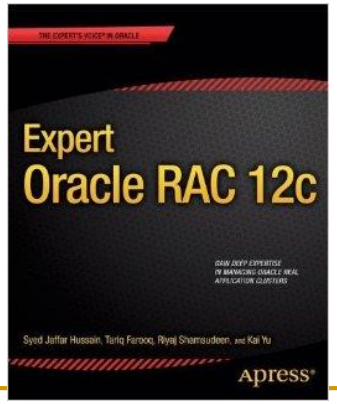
ORACLE ACE Director

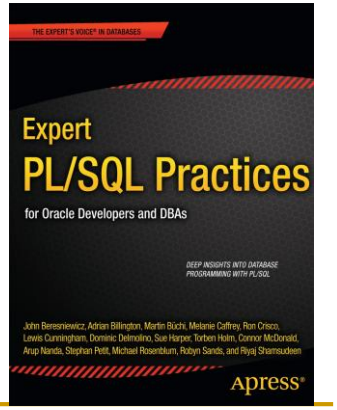
- *23+ years as DBA*
- *OakTable member*
- *Oracle ACE Director*
- *Specializes in RAC, performance tuning and Internals.*
- *Slowly in to BigData*

- *rshamsud@orainternals.com*
- *orainternals.wordpress.com*
- *Web: www.orainternals.com*









WARNING

Most of the topics in this presentations are from my research.

Writing about internals have issues:

a. I completely misunderstood the data and trace files.

b. Future version changed the feature, so, information is outdated.

Tested in version 11g, 12.1.0.2, Linux and Solaris 11 platform.

AGENDA

Clusterware processes

OCR & OLR

CSSD, ocssd.trc, alert

Ctssd, gpnp, oifcfg

Crsctl

Conclusion

Clusterware is a cluster manager

- Clusterware manages the cluster so that cluster is consistent across all nodes.
- Clusterware manages various resources also.
- Prior to 10g, clusterware was usually supplied by a vendor such as VCS , HP etc.
- Almost all of the discussions are in 11gR2/12c, you should be in the latest clusterware anyway ☺

HASD startup

- Ohasd is a daemon started at the system startup through inittab (Linux and Solaris) or RC scripts.

```
h1:3:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1
</dev/null
```

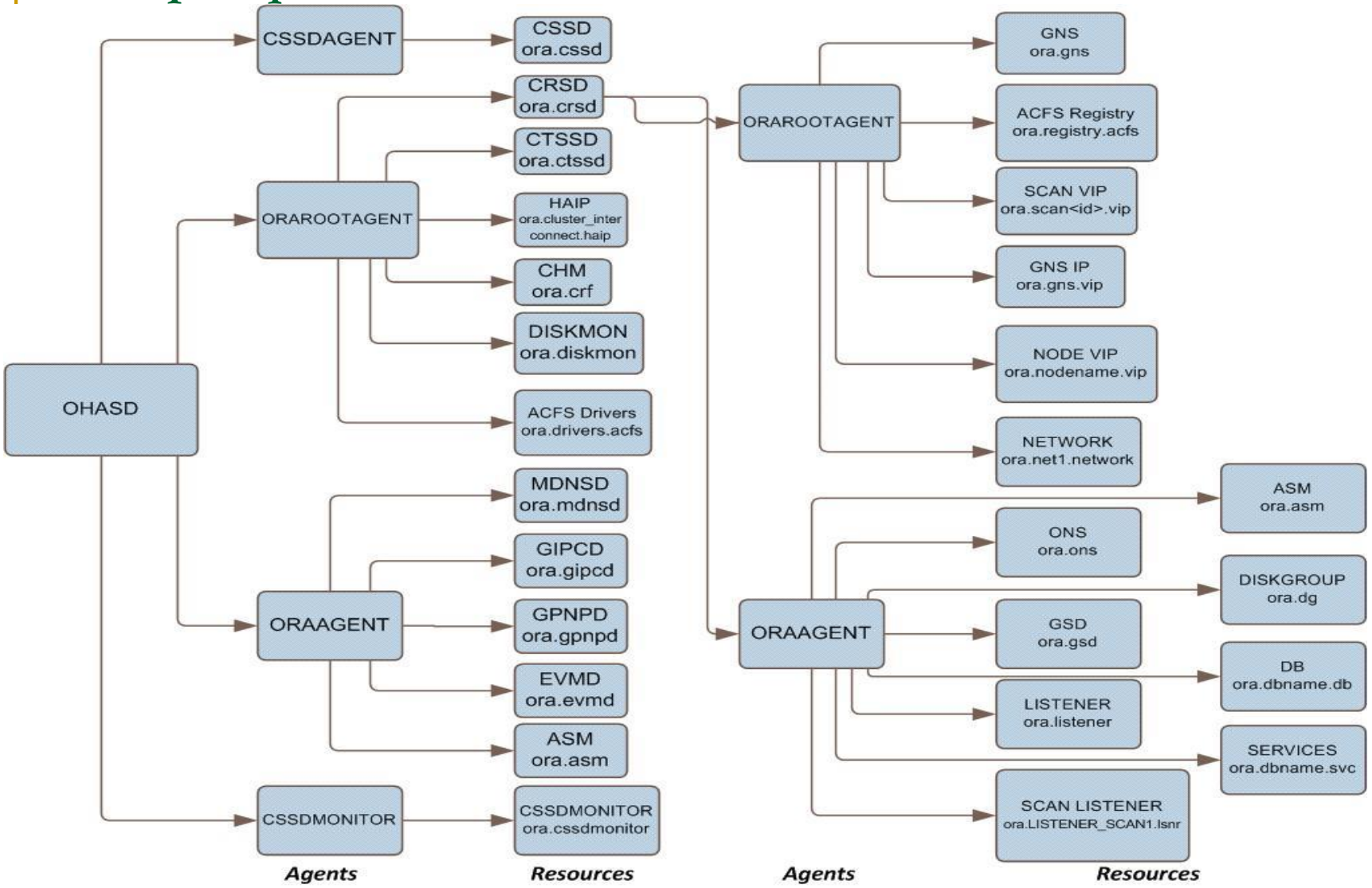
- Init.ohasd is a process that runs always because of the respawn keyword. Even if the process dies, it is restarted immediately by init daemon.

- If the auto start is enabled, then init.ohasd will start ohasd process from GI Home.

```
# cat  /var/opt/oracle/scls_scr/solrac1/root/ohasdrun
restart
```

Demo: cat /etc/inittab and review of file /etc/init.d/init.ohasd

Startup sequence



Note that this picture was copied from a Metalink note.

OCR vs OLR resources (>11.2)

- OLR – Oracle Local Registry

  OCR – Oracle Cluster Registry

- OLR contains all node specific resources.

- HASD bootstrap processes such as CTSS, CRSD, CSSD is also managed by HAS stack.

- OLR is not shared between the nodes and kept in the local file system.

Demo: ocrcheck & ocrcheck –local as root

ocrcheck

```
$ ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       3068
         Available space (kbytes) :     259052
         ID                       : 1527398265
         Device/File Name         :      +DATA
                                    Device/File integrity check succeeded

                                    Device/File not configured

                                    Device/File not configured
...
```

Ocrcheck -local

```
# ocrcheck -local
Status of Oracle Local Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       2628
         Available space (kbytes) :     259492
         ID                       : 1760145455
         Device/File Name         : /u02/app/11.2.0/grid/cdata/solrac1.olr
                                    Device/File integrity check succeeded
...
```

HASD resources

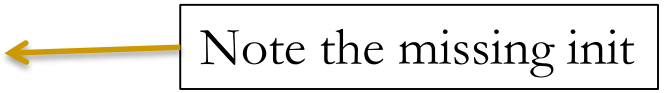
```
$ crsctl stat res -init |  more
NAME=ora.asm
TYPE=ora.asm.type
TARGET=ONLINE
STATE=OFFLINE

NAME=ora.cluster_interconnect.haip
TYPE=ora.haip.type
TARGET=ONLINE
STATE=ONLINE on solrac1

NAME=ora.crf
TYPE=ora.crf.type
TARGET=ONLINE
STATE=ONLINE on solrac1
```

Demo: crsctl

CRS managed resources

- Resources such as DB and listener are managed by CRS.

```
crsctl stat res |more
    NAME=ora.LISTENER.lsnr
    TYPE=ora.listener.type
    TARGET=ONLINE            , ONLINE
    STATE=ONLINE on solrac1, ONLINE on solrac2
    ….
```



- crsd daemon manages CRS resources.

Demo: crsctl stat res

Managing OLR

- OLR is stored in $ORACLE_HOME/cdata/<hostname>.olr.

```
$ ocrdump -local -stdout|more
10/16/2011 21:52:52
/u02/app/11.2.0/grid/bin/ocrdump.bin -local -stdout

[SYSTEM]
UNDEF :
SECURITY : {USER_PERMISSION : PROCR_ALL_ACCESS, GROUP_PERMISSION :
   PROCR_READ, OTHER_PERMISSION : PROCR_READ, USER_NAME : root, GROUP_NAME
   : root}

[SYSTEM.crs]
UNDEF :
SECURITY : {USER_PERMISSION : PROCR_ALL_ACCESS, GROUP_PERMISSION :
   PROCR_READ, OTHER_PERMISSION : PROCR_READ, USER_NAME : root, GROUP_NAME
   : root}
..
```

Demo: ls –lt of olr and ocrdump

OLR backup

- Automatic OLR backup is not supported as of 11.2.

- Backup manually before modifying any resources.

```
# ./ocrconfig -local -manualbackup

solrac1     2011/10/16 21:58:18
   /u02/app/11.2.0/grid/cdata/solrac1/backup_20111016_215818.olr

solrac1     2010/12/17 18:58:25
   /u02/app/11.2.0/grid/cdata/solrac1/backup_20101217_185825.olr

solrac1     2010/10/03 08:48:24
   /u01/app/11.2.0/grid/cdata/solrac1/backup_20101003_084824.olr
```

Demo: ls –lt of olr and ocrdump

OLR export , import

- OLR can be exported to a file. But, use backup and restore for OLR backup.

```
#./ocrconfig -local -export /tmp/backup_2011016.olr
```

- OLR import can be used to import OLR copy.

- But, use restore of backup:

```
#./ocrconfig -local -restore \
   /u02/app/11.2.0/grid/cdata/solrac1/backup_20111016_215818.olr
```

Demo:ocrconfig -local

OCR

- OCR contains cluster resources and must be accessible from all nodes.
- OCR can be in the shared oracle home or ASM.
- OCR is an hyper-critical file. Need to have redundant OCR locations.

```
#./ocrconfig -add +asm_disk_group
Or
# ./ocrconfig -add /file_name_in_certified_file_system
```

- Up to 5 OCRs can be added from 11.2

OCR backup

- OCR is backed up every four hours automatically.

```
# ./ocrconfig -showbackup
solrac1     2011/10/15 19:59:31     /u02/app/11.2.0/grid/cdata/solrac-
   cluster/backup00.ocr
solrac1     2011/10/14 16:50:17     /u02/app/11.2.0/grid/cdata/solrac-
   cluster/backup01.ocr
solrac1     2011/10/14 00:35:07     /u02/app/11.2.0/grid/cdata/solrac-
   cluster/backup02.ocr
solrac1     2011/10/14 00:35:07     /u02/app/11.2.0/grid/cdata/solrac-
   cluster/day.ocr
solrac1     2011/10/14 00:35:07     /u02/app/11.2.0/grid/cdata/solrac-
   cluster/week.ocr
```

- CRSD also keeps end of week OCR, end of day OCR, and three more copies.

Daemons: CSSD

- Clusterware process runs as daemons (UNIX term).
- Few clusterware daemons will run with root permissions since some operations require root permissions.
- CSSD – Cluster Synchronization Services Daemon.
- CSSD runs in each node and heartbeats to CSSD processes running in other nodes.

Demo: pfiles, pmap, truss of cssd daemon

CSSD heartbeats

- Two types of heartbeats:
  - Network based heartbeat

    Timeout of ~30 seconds (css_misscount)
  - Disk based heartbeat (votedisk)

    Time out of 200 seconds

    (600 seconds if vendor clusterware co-exist).
- Failure of heartbeat can lead to node reboots.
- With vendor clusterware such as VCS: Clusterware need to be integrated properly.

Disk heartbeat

- Each node has its own block in the votedisk and maintained by CSSD running in that node.
- CSSD reads votedisk block of other active nodes in the cluster too.
- Missed heartbeat after 200 seconds indicates node trouble.
- Health node's CSSD will ask faulty nodes to reboot itself. (STONITH – Shoot The Other Node in The Head algorithm)

When will the node reboot?

MC: misscount = 30s
DTO: Disk time out =200s

Network ping	Vote disk ping	reboot
Completes< misscount	Completes < MC	N
Completes < misscount	Competes>mc, but <DTO	N
Completes < misscount	Completes > DTO	Y
Takes > misscount	Doesn't matter	Y

$ crsctl get css disktimeout
CRS-4678: Successful get disktimeout 200 for Cluster Synchronization Services.

$ crsctl get css misscount
CRS-4678: Successful get misscount 30 for Cluster Synchronization Services.

Note: This is taken from MOS note 294430.1

Votedisk

- Configuring odd number of votedisk is a best practice.
- If a node can not access more than 50% of the configured votedisk devices, node will reboot.
- With 3 votedisks, as long as the node has access to 2 votedisks, it will survive.
- M=2*N+1

  M = # of voting disk to configure

  N = # of disk failure to sustain.

Votedisk in ASM

- From 11g onwards, votedisk can be kept in ASM.
- But, cssd is started much before ASM.
- So, Votedisks are identified outside of ASM by the clusterware.

```
$ crsctl query css votedisk
##  STATE    File Universal Id                File Name Disk group
--  -----    -----------------                --------- ---------
 1. ONLINE   68c4e3d9dee94fc0bf9dc32b1187e589 (/dev/rdsk/c2t10d0s1) [DATA]
```

Votedisk & ASM redundancy

- Depending upon redundancy of ASM disk group, multiple votedisk(s) *can* be configured.

DG redundancy	Maximum
External	1
Normal	3
High	5

- For normal redundancy, you must have at least three fail groups configured: crsctl replace css votedisk +dgnormal.

```
Errors in file
   /u01/app/oracle/diag/asm/+asm/+ASM2/trace/+ASM2_ora_2069.trc:
ORA-15274: Not enough failgroups (3) to create voting files
```

Quorum disk

- Quorum disks are special type of failure groups.

```
CREATE DISKGROUP ocr NORMAL REDUNDANCY
  FAILGROUP fg1 DISK '/devices/disk1'
  FAILGROUP fg2 DISK '/devices/disk2'
  QUORUM FAILGROUP fg3 DISK '/devices/disk3';
```

- Consider, two failure groups with 10 disks of 1GB.
- If we need to add a third failure group, then we need to have additional 10 disks of 1G(or 10G in total). Not in the case of quorum failure group.

Votedisk backup

- Votedisks are automatically backed up in to OCR in 11.2.
- Until 11.1, dd command can be used to backup the voting disk.
- From 11.2 onwards, dd command to backup votedisk is not supported.
- 11.2 also simplifies the restore as votedisk can be re-initialized.

```
$ crsctl replace votedisk +DATA
```

CSSD performance

- CSSD performance is very important.
- CSSD process runs in elevated CPU priority (RT in Solaris).
- CSSD won't miss heartbeat due to CPU starvation.
- But, swap/paging issues of CSSD can lead to Node reboots.
- GI file system also should have proper mount options.

Demo: ps –efl|grep cssd

Agent

- OCSSD daemon is started by ocssdagent.
- Both ocssdagent and cssdmonitor monitors ocssd process.
- If ocssd process is stuck, then cssdagent or cssdmonitor will restart the node.
- Log files in:

$ORACLE_HOME/log/<node>/agent/ohasd/oracssdagent_root

$ORACLE_HOME/log/<node>/agent/ohasd/cssdmonitor_root

Demo: Log directory structure

Agent log

```
2011-10-19 11:38:45.083: [ USRTHRD][12] clsnproc_needreboot: Impending reboot at 50% of
   limit 28500; disk timeout 27849, network timeout 28500, last heartbeat from CSSD at
   epoch seconds 1319042310.521, 14562 milliseconds ago based on invariant clock
   862571; now polling at 100 ms

...

2011-10-19 11:38:51.905: [ USRTHRD][12] clsnproc_needreboot: Impending reboot at 75% of
   limit 28500; disk timeout 27849, network timeout 28500, last heartbeat from CSSD at
   epoch seconds 1319042310.521, 21384 milliseconds ago based on invariant clock
   862571; now polling at 100 ms

…

2011-10-19 11:38:56.218: [ USRTHRD][12] clsnproc_needreboot: Impending reboot at 90% of
   limit 28500; disk timeout 27849, network timeout 28500, last heartbeat from CSSD at
   epoch seconds 1319042310.521, 25697 milliseconds ago based on invariant clock
   862571; now polling at 100 ms

2011-10-19 11:38:56.218: [ USRTHRD][12] clsnwork_queue: posting worker thread

2011-10-19 11:38:56.218: [ USRTHRD][14] clsnwork_process_work: calling sync

2011-10-19 11:38:56.244: [ USRTHRD][14] clsnwork_process_work: sync completed

2011-10-19 11:38:56.244: [ USRTHRD][14] clsnSyncComplete: posting omon
```

Demo: kill –STOP <cssdpid as root>, monitor cssdagent and cassdmonitor, review alert log.

CSSD in 12c

- Split brain in equal sub-cluster is an issue.

- 12.1.0.2 introduces the concept of node weights using a new algorithm. Sub-cluster with more weight will survive.

- 2017-02-14 12:22:05.361480 : CSSD:50308864: clssnmCheckDskInfo: Checking disk info...
  2017-02-14 12:22:05.361485 : CSSD:50308864: clssnmCheckSplit: nodenum 1 curts_ms 1053414138 readts_ms 1053413998
  2017-02-14 12:22:05.361493 : CSSD:50308864: clssnmCheckSplit: Node 1, node2, is alive, DHB (1487103723, 1054849738) more than disk timeout of 27000 after the last NHB (1487103695, 1054821958)
  2017-02-14 12:22:05.361514 : CSSD:50308864: clssnmrCheckNodeWeight**: node(1) has weight stamp(0), pebble(0)**
  2017-02-14 12:22:05.361515 : CSSD:50308864: clssnmrCheckNodeWeight: **node(2) has weight stamp(377182490), pebble(0)**
  2017-02-14 12:22:05.361517 : CSSD:50308864: clssnmrCheckNodeWeight: stamp(377182490), completed(1/2)
  2017-02-14 12:22:05.361518 : CSSD:50308864: clssnmrCheckSplit: Waiting for node weights, stamp(377182490)

Lastgasp files

- Clusterware also writes lastgasp files.

```
$ cd /var/opt/oracle/lastgasp
$ strings  cssmonit_solrac1.lgl
   OLG01,0309,0000,solrac-
cluster,5122f3e8b3745f69bf67820791872 3c2,solrac1,cssmoni
t,L-2011-10-19-11:38:59.022,Rebooting after limit 28500
exceeded; disk timeout 27849, network timeout 28500,
last heartbeat from CSSD at epoch seconds
1319042310.521, 28501 milliseconds ago based on
invariant clock value of 862571
```

Demo: lastgasp

messages

- Messages at OS will show that user requested for a panic.

```
Oct 19 11:38:59 solrac1 ^Mpanic[cpu0]/thread=ffffffff967a4000:
Oct 19 11:38:59 solrac1 genunix: [ID 156897 kern.notice] forced crash dump
   initiated at user request
Oct 19 11:38:59 solrac1 unix: [ID 100000 kern.notice]
Oct 19 11:38:59 solrac1 genunix: [ID 655072 kern.notice] fffffe80014f6eb0
   genunix:kadmin+517 ()
Oct 19 11:38:59 solrac1 genunix: [ID 655072 kern.notice] fffffe80014f6f00
   genunix:uadmin+c7 ()
Oct 19 11:38:59 solrac1 genunix: [ID 655072 kern.notice] fffffe80014f6f10
   unix:brand_sys_syscall+21d ()
Oct 19 11:38:59 solrac1 unix: [ID 100000 kern.notice]
Oct 19 11:38:59 solrac1 genunix: [ID 672855 kern.notice] syncing file
   systems…
...
```

Demo: messages

HA stack

- HA stack is managed by hasd and CRS stack is by crsd
- crsd is an HA resource and started by ora.crsd resource.

```
$ crsctl stat res -init -t|more
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.asm
      1        ONLINE  ONLINE       solrac1                  Started
ora.cluster_interconnect.haip
      1        ONLINE  ONLINE       solrac1
ora.crf
      1        ONLINE  ONLINE       solrac1
ora.crsd
      1        ONLINE  ONLINE       solrac1
```

Demo: crsctl stat res –init –t in each node.

HA resources are local

- OLR controls a few HA resources too.
- To see OLR resources, crsctl need to be executed in each node, with –init modifier.
- Other resources such as ASM, HAIP, GPNPD, mDNSd are HA resources and local.
- Key difference between HA stack and CRS stack resource is that HA stack resources are not failed over. (But, few can be relocated)

Static configuration

- Static configuration of the resource can be queried with –p flag.

```
$ crsctl stat res  ora.cluster_interconnect.haip -init  -p |more
NAME=ora.cluster_interconnect.haip
TYPE=ora.haip.type
ACL=owner:root:rw-,pgrp:oinstall:rw-,other::r--,user:oracle:r-x
ACTION_FAILURE_TEMPLATE=
ACTION_SCRIPT=
ACTIVE_PLACEMENT=0
AGENT_FILENAME=%CRS_HOME%/bin/orarootagent%CRS_EXE_SUFFIX%
…
```

- Log file for HAIP is in:

```
/u02/app/11.2.0/grid/log/solrac1/agent/ohasd/orarootagent_root
```

Runtime configuration

- Flag –v can be used to query run time configuration.

```
$ crsctl stat res  ora.cluster_interconnect.haip -init  -v |more
NAME=ora.cluster_interconnect.haip
TYPE=ora.haip.type
LAST_SERVER=solrac1
STATE=ONLINE on solrac1
TARGET=ONLINE
...
FAILURE_COUNT=0
FAILURE_HISTORY=
ID=ora.cluster_interconnect.haip 1 1
INCARNATION=1
LAST_RESTART=10/19/2011 11:44:27
LAST_STATE_CHANGE=10/19/2011 11:44:27
STATE_DETAILS=
INTERNAL_STATE=STABLE
```

Demo: commands in this slide

Daemons: ctssd

- This daemon keeps consistent time in the cluster nodes.
- If another time daemon (such as NTP) is configured, this runs in an observer mode.
- If not, adjusts the time in cluster nodes keeping one node as a reference node.
- CSSD runs in each node and heartbeats to CSSD processes running in other nodes.

Ctssd: log file

```
2011-10-19 15:00:57.919: [    CTSS][12]ctssslave_swm: Received from master
   (mode [0xcc] nodenum [2] hostname [solrac1] )
2011-10-19 15:00:57.919: [    CTSS][12]ctsselect_msm: Sync interval
   returned in [1]
2011-10-19 15:00:57.919: [    CTSS][10]ctssslave_msg_handler4_3:
   slave_sync_with_master finished sync process. Exiting
   clsctssslave_msg_handler
2011-10-19 15:01:05.938: [    CTSS][12]sclsctss_gvss9: NTPD is not active.
2011-10-19 15:01:05.938: [    CTSS][12]sclsctss_gvss8: Return [0] and NTP
   status [1].
2011-10-19 15:01:05.938: [    CTSS][12]ctss_check_vendor_sw: Vendor time
   sync software is not detected. status [1].
2011-10-19 15:01:05.938: [    CTSS][12]ctsselect_msm: CTSS mode is [0xc4]
```

Ctssd: time sync in action

```
22011-10-19 15:01:05.938: [    CTSS][12]ctssslave_swm1_2: Ready to initiate
   new time sync process.
2011-10-19 15:01:05.940: [    CTSS][10]ctsscomm_recv_cb2: Receive incoming
   message event. Msgtype [2].
2011-10-19 15:01:05.949: [    CTSS][12]ctssslave_swm2_1: Waiting for time
   sync message from master. sync_state[2].
2011-10-19 15:01:05.949: [    CTSS][10]ctssslave_msg_handler4_1: Waiting
   for slave_sync_with_master to finish sync process. sync_state[3].
2011-10-19 15:01:05.949: [    CTSS][12]ctssslave_swm2_3: Received time sync
   message from master.
2011-10-19 15:01:05.949: [    CTSS][12]ctssslave_swm15: The CTSS master is
   behind this node. The local time offset [-16454 usec] is being adjusted.
   Sync method [2]
2011-10-19 15:01:05.949: [    CTSS][12]ctssslave_swm17: LT [1319054465sec
   940739usec], MT [1319054465sec 924285usec], Delta [2380usec]
2011-10-19 15:01:05.950: [    CTSS][12]ctssslave_swm19: The offset is
   [16454 usec] and sync interval set to [1]
```

Date adjustment

- If you adjust the time in the master node to an higher value, that can cause the nodes to have different time, until ctssd catches up.

```
2011-10-19 15:14:52.111: [    CTSS][12]ctssslave_swm: The magnitude of the
   systime diff is larger than max adjtime limit. Offset [-995418] usec
   will be changed to max adjtime limit [+/- 131071].
```

```
2011-10-19 15:14:52.111: [    CTSS][12]ctssslave_swm15: The CTSS master is
   ahead this node. The local time offset [131071 usec] is being adjusted.
   Sync method [2]
```

```
2011-10-19 15:14:52.111: [    CTSS][12]ctssslave_swm17: LT [1319055292sec
   103097usec], MT [1319055293sec 98515usec], Delta [2588usec]
```

- An adjustment exceeding a day can even lead to node reboots.

Demo: as root date 1234.56 to adjust time to 12:34:56

NTP and ctssd

- If vendor daemon (such as NTP) is configured and running, ctssd will start in observer mode.

```
Crsctl stat res -init -t|more
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.ctssd

      1        ONLINE  ONLINE       solrac1                  OBSERVER
```

- NTP daemon should run 'slew always' mode since higher increase can cause clusterware problem (bug 5015469)

GPnP

- gpnpd daemon maintains the global plug and play profile.
- It's job to ensure that profile xml files are in sync.
- This profile provides essential information to join the cluster.
- Profile is in :

```
$ORACLE_HOME/gpnp/${NODE}/profiles/peer
```

- For example, cluster_guid is in this xml file:

```
ClusterUId="5122f3e8b3745f69bf678207918723c2
  " ClusterName="solrac-cluster"
```

Demo: cd profiles directory, show the xml file.

More about GPnP

- In RAC, GPnP profile also stores the ASM spfile location:

```
orcl:ASM-Profile id="asm" DiscoveryString="" SPFile="+DATA/s
olrac-cluster/asmparameterfile/registry.253.731407017"/>
```

- When you add a new node to a cluster, this profile is pushed to the new node and provides all essential information.

- Of course, corruption in profile can cause the node not able to join.

gpnptool

- Profile is protected and so, modification is tightly controlled.

- A specific key can be obtained too:

```
$ gpnptool getpval  -asm_spf
Warning: some command line parameters were defaulted. Resulting
   command line:

/u02/app/11.2.0/grid/bin/gpnptool.bin getpval -asm_spf -
   p=profile.xml -o-

+DATA/solrac-cluster/asmparameterfile/registry.253.731407017
```

- Gpnptool getpval -? For all options.

Demo: gpnptool get

Sign/unsign

- If you must modify the profile.xml, you need a specific procedure.
- Loss of complete ASM can mandate this type of change:

```
# Stop CRS in al nodes
cd /u02/app/11.2.0/grid/gpnp/solrac1/profiles/peer
#Remove signature
gpnptool unsign -p=profile.xml -ovr -o=profile.xml
#Change attributes in profile.xml
#Sign again
gpnptool sign -p=profile.xml -ovr -o=profile.xml -
w=file:../../wallers/peer -rmws
```

CRSD

- CRS stack is managed by CRS daemon.
- Ora.crsd is an HA stack resource and started by HA daemon.

```
$ crsctl stat res -init -t |more
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
...
ora.crsd
      1        ONLINE  ONLINE       solrac1
...
```

Demo:

CRS resources

- There are two types of resources CRS manages: Grid owner owned and root owned.
- There are two types of agents starting these resources:

  orarootagent and oraagent

```
$ crsctl stat res -t |more
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA.dg
               ONLINE  ONLINE       solrac1
               ONLINE  ONLINE       solrac2
ora.DGNORMAL.dg
               ONLINE  ONLINE       solrac1
               ONLINE  ONLINE       solrac2
...
```

Demo:

Root or GI owned?

- You can use –p flag to identify the agent that will start the resource, as below:

```
$ crsctl stat res ora.scan1.vip  -p |grep AGENT_FILE
AGENT_FILENAME=%CRS_HOME%/bin/orarootagent%CRS_EXE_SUFFIX%


$ crsctl stat res ora.asm  -p |grep AGENT_FILE
   AGENT_FILENAME=%CRS_HOME%/bin/oraagent%CRS_EXE_SUFFIX%
```

- Log files:

```
$ ls -lt $ORACLE_HOME/log/`uname -n`/agent/*/*
/u02/app/11.2.0/grid/log/solrac1/agent/crsd/oraagent_oracle:
total 426002
-rw-r--r--   1 oracle   oinstall 3405577 Oct 23 17:32
   oraagent_oracle.log
```

Demo:

DB Instance name

- DB instance name is derived from node number of that node (and static).

```
$ olsnodes -n
db001     1
db002     2
db003     3
…
db013     12
db012     13
...
```

```
$ crsctl stat res ora.tstdb.db -p |more
USR_ORA_INST_NAME@SERVERNAME(db001)=TSTDB1
USR_ORA_INST_NAME@SERVERNAME(db002)=TSTDB2
USR_ORA_INST_NAME@SERVERNAME(db003)=TSTDB3
..
USR_ORA_INST_NAME@SERVERNAME(db012)=TSTDB12
USR_ORA_INST_NAME@SERVERNAME(db013)=TSTDB13
..
```

Demo: output of commands

ASM Instance name

- ASM instance name is derived from node number of that node (and dynamic).

```
$ olsnodes -n
db001     1
db002     2
db003     3
…
db013     12
db012     13
...
```

```
$ crsctl stat res ora.asm -p |more
..
USR_ORA_INST_NAME=+ASM%CRS_CSS_NODENUMBER%
..
```

Demo: output of commands

Node number

- Each node acquires a lease on node number when the node joins the cluster initially.
- That lease expires only if the node is unpinned and the node is down for more than a week.
- Running root.sh in parallel can cause this type of node number mismatch. To keep the mapping, run root.sh in serial.
- To avoid lease expiry, nodes can be pinned.

```
$ olsnodes -t -n
$ crsctl pin css -n <nodename>
```

Demo: output of commands

crsctl

- crsctl is an option-rich tool. Older tools such as crs_start, crs_stop are deprecated.
- CRS is enabled by default to auto start.
- It can be disabled. scls_scr directory entry is modified.

```
$cat /var/opt/oracle/scls_scr/`uname -n`/root/ohasdstr
Enable

# /u02/app/11.2.0/grid/bin/crsctl disable crs
CRS-4621: Oracle High Availability Services autostart is
   disabled.

$cat /var/opt/oracle/scls_scr/`uname -n`/root/ohasdstr
disable
```

Sockets

- Many Clusterware daemons uses sockets to communicate to other processes.
- These network sockets are externalized as temp files.
- In Solaris, these files are stored in /var/tmp/.oracle

```
$ ls -lt /var/tmp/.oracle/ |more
total 2
srwxrwxrwx   1 oracle   oinstall       0 Oct 24 11:39 sLISTENER_SCAN1
srwxrwxrwx   1 oracle   oinstall       0 Oct 24 11:39 sLISTENER
srwxrwxrwx   1 root     root           0 Oct 24 11:39 sora_crsqs
srwxrwxrwx   1 root     root           0 Oct 24 11:39 sCRSD_UI_SOCKET
...
```

- Other processes connect using these socket files. For example, connection to ohasd from 'crsctl check crs'

```
access("/var/tmp/.oracle/sOHASD_UI_SOCKET", F_OK) = 0
connect(5, 0xFFFFFD7FFFDFA310, 110, SOV_DEFAULT) = 0
```

Crsctl status

- Status can be used to check the status of HA or CRS resources.

```
$  crsctl stat res -t|more
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA.dg
               ONLINE  ONLINE       solrac1
               ONLINE  ONLINE       solrac2
...
```

- Specific resource status also can be checked:

```
$ crsctl stat res ora.DGNORMAL.dg
NAME=ora.DGNORMAL.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE           , ONLINE
STATE=ONLINE on solrac1, ONLINE on solrac2
```

Demo:

Crsctl query

- Query can be used to check active version, votedisk etc.

```
$ crsctl query crs activeversion
Oracle Clusterware active version on the cluster is [11.2.0.2.0]

$ crsctl query css votedisk
##  STATE    File Universal Id                File Name Disk group
--  -----    -----------------                --------- ---------
 1. ONLINE   68c4e3d9dee94fc0bf9dc32b1187e589 (/dev/rdsk/c2t10d0s1) [DATA]
```

- Relocate is also useful if you have to manually relocate a resource.

```
$ crsctl relocate resource  ora.solrac1.vip -s source -n target
```

Demo: various crsctl commands

oifcfg

- oifcfg command can be used to configure the network interfaces.

```
$ oifcfg iflist -p -n
e1000g0  172.16.0.0  PUBLIC  255.255.0.0
e1000g1  1.3.1.0  PUBLIC  255.255.255.0
e1000g1  169.254.0.0  PUBLIC  255.255.128.0
e1000g1  169.254.128.0  PUBLIC  255.255.128.0
e1000g2  1.99.1.0  PUBLIC  255.255.255.0
```

- oifcfg iflist prints the *available* interfaces to configure.
- For example, /sbin/ifconfig –a in Solaris will show that iflist is showing the subnets configured.

```
$ /sbin/ifconfig -a
...
e1000g0:3: flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu 1500 index
   2
        inet 172.16.140.151 netmask ffff0000 broadcast 172.16.255.255
e1000g1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 1.3.1.170 netmask ffffff00 broadcast 1.3.1.255
```

Demo: oifcfg commands

Interface names

- An interface can be configured as global or local (node specific).
- Global interface: All nodes have same interface name, subnet mapping.
- Best practice is to have all interfaces to have same name connect to a subnet.
- For example, 1.3.1.0 subnet is associated with e1000g1 interface.

```
solrac1:
e1000g1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 1.3.1.180 netmask ffffff00 broadcast 1.3.1.255

Solrac2:
e1000g1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 1.3.1.170 netmask ffffff00 broadcast 1.3.1.255
```

Demo: oifcfg commands

Getif & setif

- Oifcfg getif is to query the current configuration:

```
$ oifcfg getif
e1000g0  172.16.0.0  global  public
e1000g1  1.3.1.0  global  cluster_interconnect
```

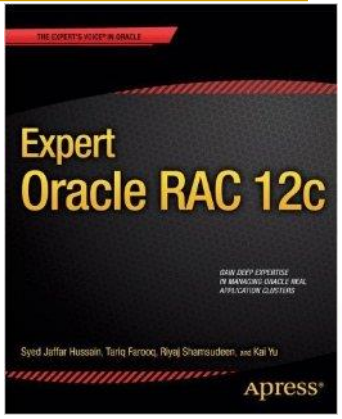
- To modify above cluster_interconnect to another interface e1000g3 and subnet:

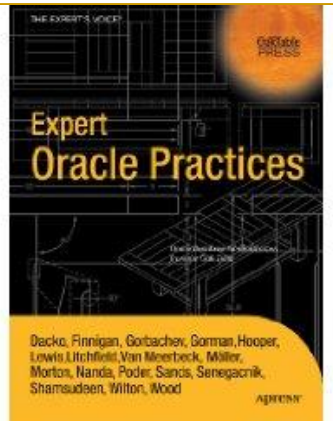
```
# set the interface to e1000g3 and 1.4.1.0 subnet.
Oifcfg setif -global /e1000g3/1.4.1.0/:cluster_interconnect

# delete the old interface
Oifcfg delif -global /e1000g1/1.3.1.0/
```

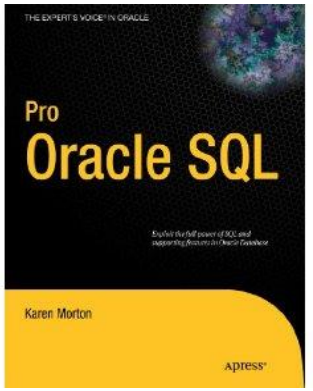
- This modifies OCR. So, CRS need to be restarted for the change to take effect.

Demo: oifcfg commands

THANK YOU







- *Email: rshamsud@orainternals.com*
- *Blog : orainternals.wordpress.com*
- *Web: www.orainternals.com*