Troubleshooting Oracle Clusterware

Dan Norris, DanNorris.com

Introduction and Terminology

Oracle Clusterware is present in all Oracle RAC configurations, but it is often overlooked as a managed piece of software. Since it isn't a database, many DBAs aren't familiar with its function or techniques for troubleshooting its behaviors.

To better understand and be ready for troubleshooting this relatively new software, we start by examining its components and their purposes. An understanding of how each piece fits together will better enable DBAs to quickly identify the most likely source of problems in the stack. Familiarity with the most useful operating system and Oracle tools enables DBAs to reveal the right information at the right times to debug Oracle Clusterware components and processes.

Finally, adding a troubleshooting process to the sometimes overwhelming stack of software that makes up a RAC cluster can be the key to combining knowledge of Oracle Clusterware architecture and processes with the proper tools to quickly identify the right resolution to most issues. Oracle Clusterware has the ability to reboot servers without warning and sometimes the reasons for its actions are not easily identified. With so much power to affect overall availability, Oracle Clusterware is a critical component and it is important to understand its troubleshooting just as thoroughly as the database layer troubleshooting.

To support the discussion that follows, these definitions may be helpful:

Term	Definition
Clusterware	software that manages cluster membership and monitors the nodes and networks in a cluster
Storage Area Network (SAN)	a storage environment where multiple servers can utilize a single storage array; a storage network is commonly implemented using fiber channel technology
Local Storage	disk space that is available to exactly one node in a cluster; this storage may be part of a SAN or may be direct-attached to the server
Shared Storage	disk space that is available to more than one node in a cluster at the same time; this storage is commonly part of a SAN
Raw Device	the character (unbuffered) special device presented by the operating system
Cluster Filesystem	a special filesystem that can be accessed by multiple cluster nodes at the same time
Fencing	Generically, when a node in the cluster is denied access to a shared resource (like storage), it is known as "fencing." Oracle Clusterware handles fencing by rebooting the node that is to be fenced.
Eviction	When a node is removed from the cluster by one of the current cluster members, the eliminated node is said to be "evicted."

Clusterware Architecture and Processes

Oracle Clusterware provides a mechanism to bind multiple servers together to make databases more highly available. Besides being the cluster manager for Oracle RAC environments in Oracle Database 10g Release 1 and higher, it is also capable of providing database failover capabilities as a standalone product. In order to troubleshoot Oracle Clusterware, you must first understand how it works. While some understanding of Oracle Clusterware is assumed, we'll go over a few of the components and their roles in this section.

Oracle Clusterware has certain hardware and software requirements. These requirements include shared storage, redundant public network interfaces, redundant private network interfaces, and the same operating system platform and version across all nodes in the cluster.

In order to carry out its responsibilities, Oracle Clusterware employs a small repository of metadata called the Oracle Cluster Registry (OCR) and structures called voting disks. In Oracle Clusterware 10g Release 2 and higher, the installer conveniently offers redundant configuration options for each of these components as well (up to 2 OCRs and 3 voting disks). The OCR holds information about cluster members, databases and services they offer, as well as other cluster resources like VIPs, listeners, and ONS processes. The OCR is also where Clusterware tracks the current status of each of these components. The voting disks hold information about current cluster membership. In the event of a partial cluster failure (for example, if the interconnect fails), the voting disks help the cluster determine which of the nodes will survive and remain in the cluster.

CSS, CRS, EVM

To carry out its mission, Oracle Clusterware has a number of background processes. You will observe these processes alive and active on any Oracle Cluster. Each process has specific responsibilities, as noted briefly here.

Process	Description
`crsd`	Performs OCR maintenance and manages application resources; runs as the root user
`evmd`	Event Manager detects cluster disruptions and performs Oracle Clusterware callouts
`ocssd`	Manages cluster membership, runs as the Oracle software owner (usually "oracle"); if this process fails, the node is restarted
`oprocd`	Process Monitor ensures that other processes are running and performs appropriate actions (per the cluster configuration) if they fail; this process does not exist when Oracle Clusterware is integrated with another ("3rd party") clusterware; this process is called OraFenceService on Windows.
Oracle Process Manager Daemon	On Windows, this process is a dependency for all Clusterware processes and provides the necessary delay for other Windows services that may be necessary for Oracle Clusterware (like OCFS) to start and become active. This process does not exist on Linux or UNIX systems.
`racg`	The racg process is used to start, stop, and monitor some of the Oracle Clusterware resources like gsd, ons, vip, and other "built-in" resources

On Linux and UNIX systems, these processes are individual processes with separate process IDs. On Windows systems, the processes are slightly different, but generally follow the pattern Ora<ProcessName>Service except for oprocd. As noted above, oprocd on Windows is known instead as OraFenceService.

Typically, the most interesting logging information comes from the crsd and ocssd processes. On LInux/UNIX systems, the logfiles are placed in `$ORA_CRS_HOME/log/<nodename>/<processname>`. For example, the crsd logs for node1 would be found in `$ORA_CRS_HOME/log/node1/crsd`.

Communication Path(s)

There are several ways that Oracle Clusterware nodes communicate with one another. First, and most obvious, is the direct peer-to-peer communication over the private interconnect network. This network is used by Oracle Clusterware for heartbeat checks of other nodes in the cluster.

The Oracle Cluster Registry (OCR) is also used as a communication device with the administrator since it records information directly from each node regarding status and configuration. All nodes may read and write the OCR directly, so when configuration information changes and is updated in OCR, all other nodes can immediately read that configuration information. OCR is mandatory and critical to Oracle Clusterware operation. It is recommended to configure at least one OCR mirror location.

The Voting Disks are used for cluster membership. In order to be a cluster member, the node must have vote registered on the voting disks. Because voting disks are mandatory and critical, it is recommended that clusters have a minimum of 3 voting disks configured.

Cluster Reconfiguration

The cluster reconfiguration event happens during normal operation in several different scenarios. The first and most common scenario is when a cluster node starts or stops. The start or stop operation could be a normal, graceful startup or shutdown event or it may be an unplanned outage like a server crash or reboot due to a server hang. The reason for the reconfiguration doesn't matter—if there is a change in cluster membership, the cluster will reconfigure itself.

A reconfiguration is the collective name given to a number of steps, mostly handled by the ocssd process, involved in transitioning nodes into and out of the cluster. During a cluster reconfiguration, a number of negotiations occur between the node wishing to join or leave the cluster and its peers. These messages are shown in the logfile `$ORA_CRS_HOME/log/<nodename>/cssd/ocssd.log` with plenty of detail.

The cluster reconfiguration event is a lot of what makes clusterware software proprietary. The act of properly detecting a failure condition and taking the proper action after detection is what makes clusterware software "smart" and also necessarily complex.

With Oracle Clusterware, if a cluster reconfiguration is ordered because of a failed interconnect (which is a split-brain cluster), then one of the nodes in the cluster will reboot itself in order to save the integrity of the cluster and the application(s) it protects. Other clusterware vendors handle the issue differently, but all vendors have to properly detect and handle this condition. This is the source of great debate between some cluster administrators and Oracle's Clusterware team.

Clusterware's role in RAC

Oracle Clusterware has some features that are specific to its role in RAC clusters. For example, it provides a srvctl utility that understands how to handle object types like "database", "service", and "listener." The presence of those object types indicates the presence of agents in the software that know how to start, stop, and check each of those object types.

However, the most important responsibility and role for Oracle Clusterware in RAC environments is to handle cluster membership. RAC delegates that responsibility to Oracle Clusterware and provides no methods for cluster membership management on its own. Managing cluster membership means that Oracle Clusterware is responsible for ensuring that when a failure occurs in the cluster, it is handled safely so that the data and/or database(s) it is protecting are free from corruption. As of Oracle Clusterware 11g Release 1, node reboot is the only fencing mechanism available, so it is possible that one or more nodes are rebooted in some cases.

Oracle Clusterware is required for RAC, but it is also a standalone product that can be used independent of RAC environments. Without RAC, Oracle Clusterware can provide database or other processes a failover capability. Shared storage is still required (at least for Oracle Clusterware) and the same utilities (crs_stat, etc.) can be used to manage the environment. Several whitepapers exist on Oracle TechNet (cited in the references section at the end of this paper) to describe how to configure single-instance databases and some Oracle Fusion Middleware components to perform failover under Oracle Clusterware control.

Tools

Sometimes the hardest part about troubleshooting is trying to figure out how and where to find the information you need to effectively debug the situation. This section provides an introduction to tools from the OS and Oracle Clusterware environments as well.

OS Tools: netstat, pstack, ps, vi, less, strace

There are two main classes of OS tools in my view. The first class includes tools that are "standard" and expected tools that you'll use in every troubleshooting scenario. These include things like ps, vi, more (or less), and netstat, to name a few.

Tool	Platform	Description
`more` (or `less`)	Linux/UNIX	These are tools used to view text files like logfiles and configuration files. They have no editing capability, so they're relatively safe to use on almost all files.
`tail`	Linux/UNIX	This tool allows you to "watch" the end of a file (typically a logfile) as it is being written (with the -f flag).

Tool	Platform	Description
`baretail`	Windows	A (free) Windows-specific utility that provides the same functionality as the tail command on Linux/UNIX.
Process Explorer	Windows	A (free) tool that allows you to get detailed process information like the libraries and files accessed by an individual process.
`ps`	Linux/UNIX	The standard process listing tool to get a list of processes running along with information about who owns the process, etc.
`netstat`	Linux/UNIX	This tool displays information about the network status and configuration, depending on flags. To get interface information use "-ain"; to get socket information use "-an"; to get routing table information use "-rn"
`dd`	Linux/UNIX	useful for reading storage to see if it is readable (not always interested in the contents as much as access or I/O errors that it might reveal)
`ifconfig`	Linux/UNIX	Displays configuration and status information for all network interfaces
`vgdisplay`, `lvdisplay`, `pvdisplay`	Linux/UNIX	Display information about volume manager objects (volume groups, logical volumes, and physical volumes, respectively)

The second class of OS tools are those that aren't used commonly in normal operation. It is relatively common for these tools to provide output that's cryptic and not easy to interpret, unless you have some idea what you're looking for. With these tools, it's important to run them against a normally-running test cluster to see what a normal baseline looks like. You should also use these tools with caution as some of them may interrogate the Oracle Clusterware processes in such a way that it becomes unstable. Again, knowing which tools may cause further instability is something you should attempt to find out on a non-production cluster environment when setting your baseline.

Tool	Platform	Description
`strace`	Linux	stack trace utility – may be likely to cause some instability
`truss`	Solaris	stack trace utility – may be likely to cause some instability
`tusc`	HPUX	stack trace utility – may be likely to cause some instability
`pstack`	Linux, Solaris, HPUX	shows current stack of a process
`procstack`	AIX	shows current stack of a process
`ltrace`	Linux	library call trace utility
`ptree`	Solaris	shows process relationships for all parent and child processes of a given PID
`/proc/<PID>`	Linux	This is a directory (one directory per PID) with many pseudo files and subdirectories in it that provide information about the running process
`wireshark`	Linux, Windows, UNIX	Captures network packets given filter criteria
`tcpdump`	Linux, UNIX	Dumps tcp packets, also provides some analysis capability for files containing captured packets

Tool	Platform	Description
`filemon.exe`	Windows	displays file system activity in real time, it is now part of Process Monitor in Windows XP SP2, Windows Server 2000 SP4, and Windows Server 2003 SP1
`procmon.exe`	Windows	displays real-time filesystem, registry, and process/thread activity; the filemon.exe functionality is now available via this utility on recent Windows versions

Of course, these utilities have various mandatory and optional command-line arguments. The documentation (man pages on Linux/UNIX or help on Windows), should provide enough guidance on how to use them. You'll likely also find information online via Google searches for examples of how others might use these commands in real-world scenarios. It will take some time and practice before you learn which tool provides the necessary information you need to solve the issue you're facing at the moment.

Oracle Clusterware Tools

Oracle Clusterware provides several tools that provide information about the status of resources and processes in the Oracle Clusterware stack. Using these tools is an excellent way to get the status of your cluster. If these tools don't function because the cluster isn't healthy enough to support their use, then you can use that as another diagnostic input to your troubleshooting process.

Tool	Description
`crsctl check crs`	provides an overall status for all Oracle Clusterware processes
`crsctl check {cssd|crsd|evmd}`	provides status information on individual Oracle Clusterware processes
`crsctl check cluster [-node <nodename>]`	gives status on CSS across nodes
`crsctl query css votedisk`	provides the list of voting disks that Oracle Clusterware is using
`crsctl debug statedump {css|crs|evm}`	dumps state information for the given process
`crsctl debug trace {css|crs|evm}`	enables tracing for the given process
`crs_stat [-t] [-v]`	provides status on all cluster-managed resources including databases, services, nodeapps (vip, ons, gsd), and listeners
`ocrdump`	provides a dump of the OCR
`olsnodes`	lists the nodes in the cluster (reveals OCR contents only, no status of the cluster members is provided)
`cluvfy`	Cluster Verification Utility – especially useful for new installations, but can also be used as verification tool during debugging sessions
`oifcfg`	provides a way to access network interface configuration information (private, public designations for NICs)
`diagcollection.pl`	In $ORA_CRS_HOME, this utility can help collect diagnostic information about CRS for sending to support. Sometimes, it is helpful to see what Oracle's utilities collect, since they often include logfiles or configuration information you may not have known about previously.

Like OS tools, these utilities have many and varied command-line arguments. Unfortunately, some of them don't have much documentation, if any. The appendices in the Oracle Clusterware Administration and Deployment Guide 11g Release 1 (11.1) have documentation information for some of these utilities, but not all of them.

Humans

The best tool in the toolbox isn't a utility or a hammer—it's you. While the tools listed here can provide some interesting and useful data, it is humans that must put that data together and draw some conclusions from it. As Tanel Põder preaches in his session "Advanced Oracle Troubleshooting: No magic is needed, a systematic approach will do", use the right tool for the right problem and try to break big problems down into a series or set of smaller problems. This is something that only the human brain can do effectively and quickly.

In my own troubleshooting, I often mention that I try to come up with lots of ideas. Many of them will be wrong, but in almost every case, it's a "bad" idea that eventually triggers someone's brain to come up with the right idea that leads to a solution. Certainly, coming up with ideas that have no basis in fact isn't the right strategy. It only makes sense to focus your efforts on areas that have some possibility of being involved in the problem. That's what Tanel means when he discusses a "systematic" approach. In general, following a systematic approach will always get you to the solution much faster than guessing without any reasoning behind your guesses. Plus, my friend Alex Gorbachev, founder of the Battle Against Any Guess (http://battleagainstanyguess.com/), would be very displeased if you just guessed. If you agree that guessing isn't the way to find a solution, you should consider joining BAAG today.

Logfiles: Finding them is the hard part

As you might have guessed, one of the most important pieces of information in your toolbox is going to be the logging information that the processes create under normal circumstances. The hardest part for beginners that have to debug problems is that they don't know where to find the log output as there is no central point in the documentation that lists all the logfile locations. Here are the most common logfile locations for Oracle Clusterware information.

Location	Platform	Description
`$ORA_CRS_HOME/log/<nodename>/alert<nodename>.log`	All	Clusterware-wide logfile for all processes, contains high-level information, may not contain all detail
`$ORA_CRS_HOME/log/<nodename>/crsd`	All	Logs from the crsd (cluster registry) process
`$ORA_CRS_HOME/log/<nodename>/cssd`	All	Logs from the ocssd (clusterware synchronization) process
`$ORA_CRS_HOME/log/<nodename>/evmd`	All	Logs from the evmd (event manager) process
`$ORA_CRS_HOME/log/<nodename>/racg`	All	Logs from the racg (agent processing) process
`/var/log/messages`	Linux, Solaris	Syslog messages including some Oracle Clusterware output are written to this file
`/var/adm/syslog/syslog.log`	HPUX	Syslog messages including some Oracle Clusterware output are written to this file
dmesg	Linux, most UNIXes	This utility dumps the kernel ring buffer which contains the most recent kernel messages. This utility is in the logfiles section because it just provides logged information.
Event Viewer	Windows	The timeless classic...

This table doesn't cover all platforms, so you might find other locations for interesting logfiles on your platform. Most of the time, when combined with the "`crsctl debug ...`" commands, the logfiles in `$ORA_CRS_HOME/log` will provide sufficient evidence to direct you to the root cause of the problem.

Methods to identify issues

As the Chinese Proverb goes, "Give someone a fish and you feed them for a day. Teach them to fish and you feed them for a lifetime."

Now that we've provided sufficient background and preparation, it is time to put that knowledge to use. So, how do you know what to do when there's a problem that you haven't seen before?

Obviously, experience is helpful even when faced with problems that you haven't experienced before. When you're new to cluster management and administration, this section will provide some guidelines for how you would get started when asked to resolve a problem.

Evaluate symptoms

The first and most important step in any troubleshooting situation is to properly identify the problem. Early in the process, you likely only have a little information, but even that little bit of information can sometimes help direct your efforts. For example, when absolutely no one can connect to the application, you're likely facing a different problem than sporadic connection issues for only a few users.

Stop. Think.

This step is here for two reasons: 1) make you take a breath and settle down in what may be a tense situation and 2) focus some actual effort on making a plan for investigating the problem instead of "shotgunning" and guessing at a lot of things. After all, when a cluster fails, it's usually a pretty big deal as clusters are often expected to be bullet-proof and they commonly run critical applications. If the cluster isn't behaving well, it is likely to bring attention from all levels, including the highest executives.

You don't have to be unpleasant, but if you're the lead investigator in the troubleshooting situation, you are entitled to be direct and ask for help from others that can handle some of the tasks that you may be able to delegate. For example, you can request the storage team to review the storage switch status and error logs, the network team to verify that no errors or dropped packets are observed, and the server admins can help by reviewing the OS logfiles for errors related to Oracle. Sometimes, it's nice to assign tasks to others just because it keeps them busy and that stops them from bothering you while you're investigating. :)

Check status

There are many statuses to check in an Oracle Clusterware environment. Start with the basics like seeing if the nodes are up and online. For each of the items here, I usually see if they're reachable on the network, if they're running, if they're consuming high CPU, and other such attributes. Here's my general list (in order of what I'd check):

1. Nodes running, booted
2. Nodes can reach each other via public, private, and VIP IP addresses
3. Storage accessible, mounted, able to be listed
4. Clusterware processes status

Each of these items have some logs and/or tools that can be used to interrogate it to get more details. Let's connect those dots so that everyone knows which things might be useful to check each of the conditions above.

Checking...	Tools	Logs
Nodes running, booted	ping, ssh (Linux/UNIX), Remote Desktop (Windows), Console inspection	Not really any logs to see if servers are up—possibly console messages
Nodes can reach each other via public, private, and VIP IP addresses	ping, ssh, ifconfig	The syslog or dmesg or Event Viewer might log information about a failed network interface if there's a hardware issue. However, hardware errors may not be present yet the problem still exists.

Checking...	Tools	Logs
Storage accessible, mounted, able to be listed	df, cat /proc/partitions, dir, Windows Explorer (for Windows OCFS)	syslog, dmesg, or Event Viewer
Cluster processes status	crsctl, crs_stat, ps, procexp.exe	$ORA_CRS_HOME/logs/<node>, syslog, Event Viewer

Some of these items can be skipped. For example, checking if the server is online can be done just by attempting to login to it.

At the highest level, all problems fall into one of these four categories. So, as you work down this list, you should encounter something that leads you to the problem. Of course, it is difficult or impossible to create an exhaustive list of tools and logging locations that contain all error conditions, so this starting point should be augmented as your experience grows.

Quite a few of the issues you encounter will have obvious causes. For example, if you see this in a logfile:

```
[    CSSD]2009-03-06 22:03:26.465 [3029851024] >WARNING: clssnmPollingThread: node node1 (1) at 90% heartbeat fatal, eviction in 0.110 seconds
[    CSSD]2009-03-06 22:03:26.577 [3029851024] >TRACE:   clssnmPollingThread: Eviction started for node node1 (1), flags 0x040f, state 3, wt4c 0
[    CSSD]2009-03-06 22:03:26.578 [3008871312] >TRACE:   clssnmEvict: Evicting node 1, node1, birth 130118152, death 130118154, impendingrcfg 1, stateflags 0x40f
```

It should be obvious that node1 is being evicted from the cluster. In a two-node cluster, you may still not be sure which node really has a problem, but it's a good idea to start with the node that was evicted. In the case above, node1 had a failure of its network interconnect interface and that caused the error (as indicated by the first error line pasted above).

Check processes

It will often be the case that one or more failed processes are the first symptom of a problem. Even more often, one of those failed processes will lead to discovery of a misconfiguration or hardware issue that is the real root cause of the problem.

The processes you're most likely to check are the Oracle Clusterware background processes, described earlier in this paper. When you find errors or issues with one of these processes, an understanding of that process' responsibilities should direct you to the potential areas for focus to identify the root cause. In many cases, that cause will be obvious after following the other checks that you'll be conducting.

References

- Oracle Clusterware Administration and Deployment Guide 11g Release 1 (11.1), Chapter 1, Appendices A-H
- Tanel Põder, "Advanced Oracle Troubleshooting: No magic is needed, a systematic approach will do" presentation, Miracle Oracle Open World 2008, Lalandia, Denmark, http://blog.tanelpoder.com/seminar/seminar-files/
- Alex Gorbachev, Battle Against Any Guess, http://battleagainstanyguess.com/
- "Using Oracle Clusterware To Protect A Single Instance Oracle Database 11g" whitepaper, Oracle Technology Network, http://www.oracle.com/technology/products/database/clusterware/pdf/SI_DB_Failover_11g.pdf
- "Using Oracle Clusterware To Protect Oracle Application Server" whitepaper, Oracle Technology Network, http://www.oracle.com/technology/products/database/clusterware/pdf/Using-Clusterware-to-protect-Appserver.pdf
- Metalink 567730.1 – Changes in Oracle Clusterware on Linux with the 10.2.0.4 Patchset
- Metalink 559365.1 – Using Diagwait as a diagnostic to get more information for diagnosing Oracle Clusterware Node evictions
- Metalink 265769.1 – Troubleshooting CRS Reboots
- Metalink 284752.1 – 10g RAC: Steps To Increase CSS Misscount, REboottime and Disktimeout
- Metalink 294430.1 – CSS Timeout Computation in Oracle Clusterware

- Metalink 395878.1 – Heartbeat/Voting/Quorum Related Timeout Configuration for Linux, OCFS2, RAC Stack to avoid unnecessary node fencing, panic, and reboot

From the Lawyers