Node Management in Oracle Clusterware – wie war das noch einmal?

Markus Michalewicz
Oracle Corporation
Oracle HQ, Redwood Shores, CA, USA

Key words: Node Management, Oracle Clusterware, Voting Disks, Heartbeats

Introduction

Oracle Clusterware is portable cluster software that allows clustering of independent servers so that they cooperate as a single system. Oracle Clusterware was first released with Oracle Database 10g Release 1 as the required cluster technology for Oracle Real Application Clusters (RAC). Oracle Clusterware is an independent cluster infrastructure, which is fully integrated with Oracle RAC, capable of protecting any kind of application in a failover cluster. Data protection in such clusters has been one of the main concerns when Oracle Clusterware was designed. Stability, accuracy, and consistency were only some of the aspects taken into account when the node management, part of the Cluster Synchronization Services (CSS) layer within Oracle Clusterware, was implemented. It therefore does not surprise that the basics of the node management in Oracle Clusterware have not changed significantly over the last years. The new features of Oracle Clusterware 11g Release 2 discussed in this paper therefore extend a node management approach that has proven to be successful for Oracle RAC for more than two versions.

Improved Availability – Tuning "Under The Hood"

Oracle Clusterware 11g Release 2 has been improved to provide better availability. For example, a new agent-based monitoring system is used for monitoring all resources. These memory resident agents allow more frequent checks using fewer resources. More frequent checks means faster detection of failures and a faster recovery time. In case of the Oracle listener, the average failure detection time was reduced from 5 minutes to 30 seconds, while the check interval was reduced from every 10 minutes to 1 minute.

Simple and efficient node monitoring as the basis

In order to ensure smooth cluster operation, Oracle Clusterware uses simple, yet efficient mechanisms to monitor the nodes in the cluster. Node monitoring is based on two communication channels: A network heartbeat and a disk based communication using Voting Disk(s), which are an essential part of the Oracle Clusterware node management. In simple terms, each node is "pinged" by each node in the cluster using the network heartbeat, while a similar "ping" is exchanged through the Voting Disk(s). If one of these "heartbeats" indicates a failure, a decision is made to evict (forcibly remove) one or more nodes from the cluster.

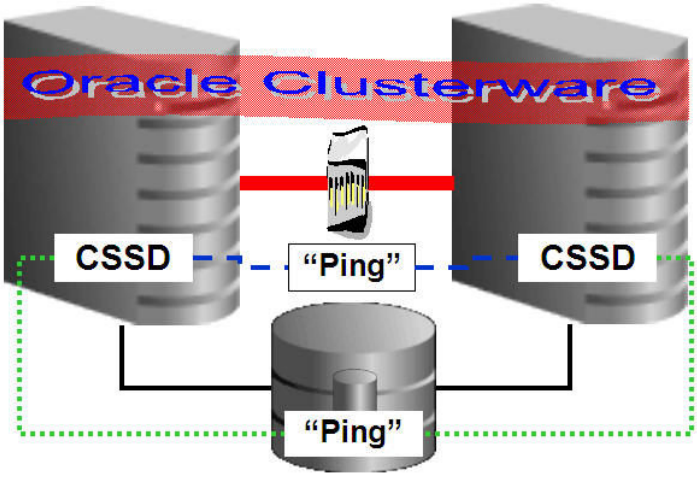

Figure 1: Node monitoring basics

Fencing Flexibility and Third Party Cluster Solution Support

Traditionally, Oracle Clusterware uses a STONITH (Shoot The Other Node In The Head) comparable fencing algorithm to ensure data integrity in cases, in which cluster integrity is endangered and split-brain scenarios need to be prevented. For Oracle Clusterware this means that a local process enforces the removal (eviction) of one or more nodes from the cluster, which, if performed as a preventive measure in the cluster is often referred to as „fencing".

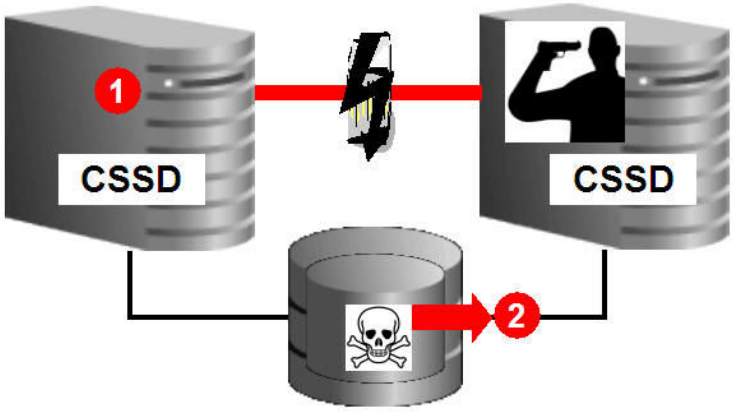

Figure 2: STONITH comparable approach in Oracle Clusterware

In addition to this traditional fencing approach, Oracle Clusterware now supports a new fencing mechanism based on remote node-termination. The concept uses an external mechanism capable of restarting a problem node without cooperation either from Oracle Clusterware or from the operating system running on that node. To provide this capability, Oracle Clusterware 11g Release 2 supports the Intelligent Management Platform Interface specification (IPMI), a standard management protocol.

In order to use IPMI and to be able to remotely fence a server in the cluster, the server must be equipped with a Baseboard Management Controller (BMC), which supports IPMI over a local area network (LAN). Once this hardware is in place in every server of the cluster, IPMI can be activated either during the installation of the Oracle Grid Infrastructure or after the installation in course of a post-installation management task using CRSCTL.

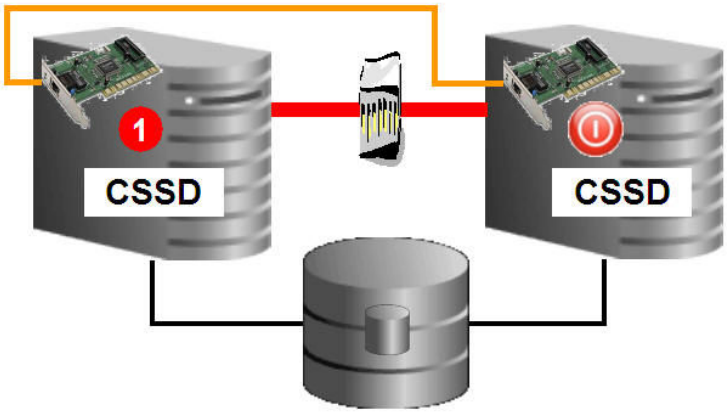

Figure 3: IMPI based node management

Oracle Clusterware also continues to support third party cluster solutions under Oracle Clusterware. For certified solutions (certified solutions can be found in My Oracle Support / Certify) Oracle Clusterware will integrate with the third party cluster solution in a way that node membership decisions are deferred to the third party cluster solution. Only, if a decision is not made within a certain amount of time, Oracle Clusterware will perform corrective actions, using one of the fencing mechanisms described.

Maintaining a third party cluster solution under Oracle Clusterware increases the complexity of the cluster stack and makes the cluster management more difficult. Oracle therefore recommends avoiding having more than one cluster solution on the same system. For Oracle RAC environments it is worth noticing that Oracle Clusterware is mandatory and provides all required functionality. No other third party solution should therefore be required.

Reboot-less node fencing in Oracle Clusterware 11g Release 2

As mentioned, Oracle Clusterware uses a STONITH (Shoot The Other Node In The Head) comparable fencing algorithm to ensure data integrity in cases, in which cluster integrity is endangered and split-brain scenarios need to be prevented. In case of Oracle Clusterware, this means that a local process enforces the removal of one or more nodes from the cluster, which again, if performed as a preventive measure in the cluster is often referred to as „fencing“.

Until Oracle Clusterware 11g Release 2, Patch Set One (11.2.0.2) the fencing of a node was performed by a "fast reboot" of the respective server. A "fast reboot" in this context summarizes a shutdown and restart procedure that does not wait for any IO to finish or for file systems to synchronize on shutdown. With Oracle Clusterware 11g Release 2, Patch Set One (11.2.0.2) this mechanism has been changed in order to prevent such a reboot, if possible.

Already with Oracle Clusterware 11g Release 2 this algorithm was improved so that failures of certain, Oracle RAC-required subcomponents in the cluster do not necessarily cause an immediate fencing (reboot) of a node. Instead, an attempt is made to clean up the failure within the cluster and to restart the failed subcomponent. Only, if a cleanup of the failed component appears to be unsuccessful, a node reboot is performed in order to force a cleanup.

With Oracle Clusterware 11g Release 2, Patch Set One (11.2.0.2) further improvements were made so that Oracle Clusterware will try to prevent a split-brain without rebooting the node. It thereby implements a standing requirement from those customers, who were requesting to preserve the node and to prevent a reboot, since the node runs applications not managed by Oracle Clusterware, which would otherwise be forcibly shut down by the reboot of a node.

With the new algorithm and when a decision is made to evict a node from the cluster, Oracle Clusterware will first attempt to shutdown all resources on the machine that was chosen to be the subject of an eviction. Especially IO generating processes are killed and it is ensured that those processes are completely stopped before continuing. If, for some reason, not all resources can be stopped or IO generating processes cannot be stopped completely, Oracle Clusterware will still perform a reboot or use IPMI to forcibly evict the node from the cluster.

If all resources can be stopped and all IO generating processes can be killed, Oracle Clusterware will shut itself down on the respective node, but will attempt to restart after the stack has been stopped. The restart is initiated by the Oracle High Availability Services Daemon, which has been introduced with Oracle Clusterware 11g Release 2.

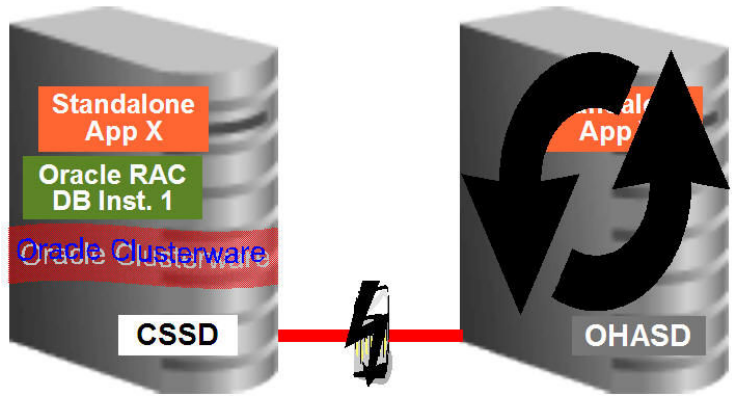

Figure 4: Reboot-less node fencing

Various fallbacks for a thorough protection under all circumstances
Since node management is crucial for a cluster, Oracle Clusterware was designed to protect appliations under all circumstances, even if those are unlikely to occur and hence are often referred to as "corner cases". Corner cases can also refer to situations which are not commonly seen, since dependent on the use case. In either case, a cluster solution needs to cover those to ensure fast recovery and to resume reliable cluster operation. Oracle Clusterware covers these cases, including hangs of processes using various fallbacks. For example: If the Oracle Cluster Sysnchronization Services Daemon (CSSD) is stuck for some reason, its monitoring process (CSSDmonitor, formerly known as OPROCD) will take over.

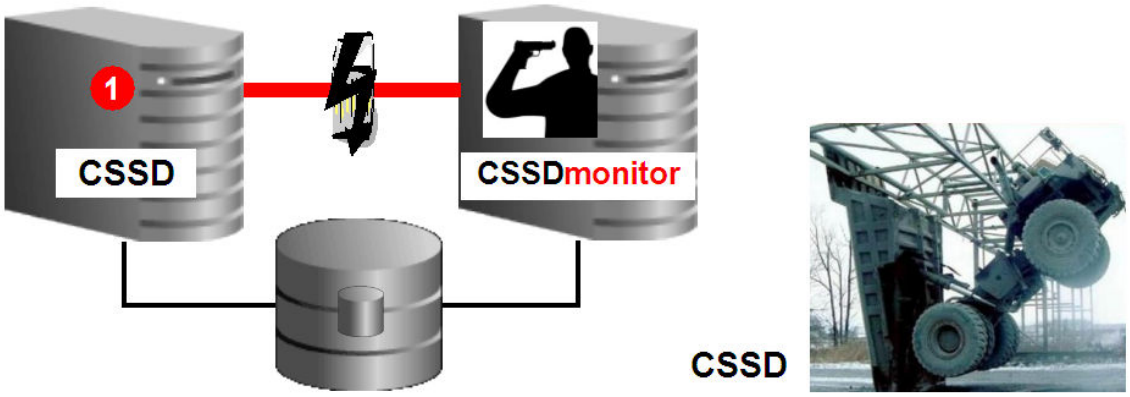

Figure 5: CSSDmonitor covers CSSD

Conclusion
Node Management in Oracle Clusterware is designed to provide a simple and yet efficient way to ensure smooth cluster operation. Nodes are monitored on a constant basis, which allows for instantaneous failure detection. Once a failure is detected, corrective actions are performed, which typically lead to an eviction of one or more nodes in the cluster. Using technologies like IPMI and the new Reboot-less Node Fencing, Oracle Clusterware does not only provide a certain amount of flexibility regarding the fencing mechanism used, but also tries to minimize the impact of a node eviction as much as possible. Corner cases are also considered and various fallbacks are used to ensure that even under those uncommon circumstances data is protected in the cluster and applications are recovered fast and reliably.

Markus Michalewicz
Oracle Corporation
500 Oracle Parkway, MS4OP840
USA – Redwood Shores, CA 94065

Telefon: +1(650)5065444
E-Mail Markus.Michalewicz@oracle.com
Internet: http://www.oracle.com/goto/clusterware