Loss in intra-cluster connectivity?

VERITAS

KNOW HOW— VCS handles an intra-cluster connectivity loss differently on UNIX and Windows

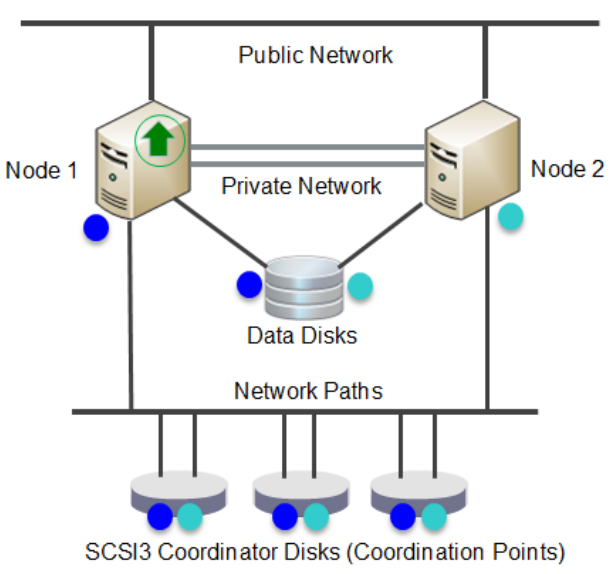
VCS recommends configuring three network links between its cluster nodes, two of which are used for the private network communication between the cluster nodes and the third one is used for the public network communication. All the cluster nodes communicate with each other over the private network link to know if all the nodes are alive and application is online on the active node.

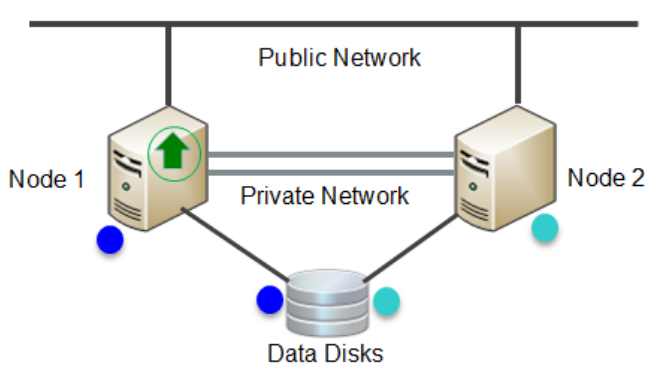
In a scenario where all the private network links are down, the cluster nodes are unable to communicate with each other and fail to know if the active node is still alive and the application is online. Such a condition where the cluster nodes are unable to communicate is called a **split-brain**. As a result of a spilt-brain, a single VCS cluster may form separate sub-clusters and nodes in each sub-cluster consider the other sub-cluster has faulted. Each sub-cluster tries to carry out recovery actions for the departed nodes. As a result, more than one node tries to import the same storage (VxVM LUNs) and may cause data corruption, have an IP address up in two places, or mistakenly run an application in two places at once.

Node membership arbitration guarantees against such spilt-brain conditions. The membership arbitration works differently on UNIX and Windows platforms. This infographic depicts platform-based VCS behavior during membership arbitration.

UNIX vs Windows

Cluster configuration





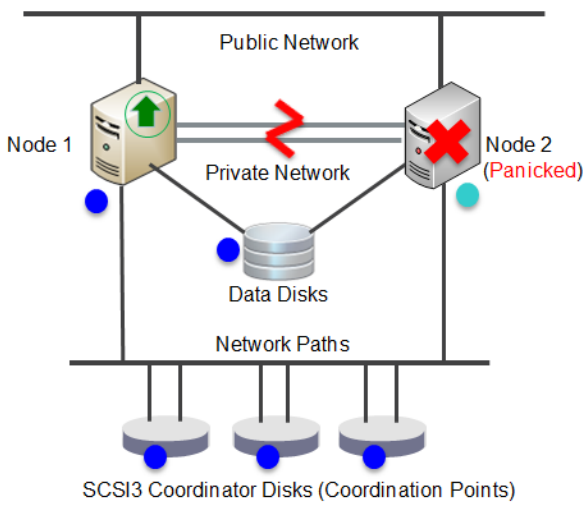
Unique SCSI Keys

Active Node

Note: Coordinator Disks can be replaced by a Coordination Point server.

Spilt-brain condition and VCS behavior

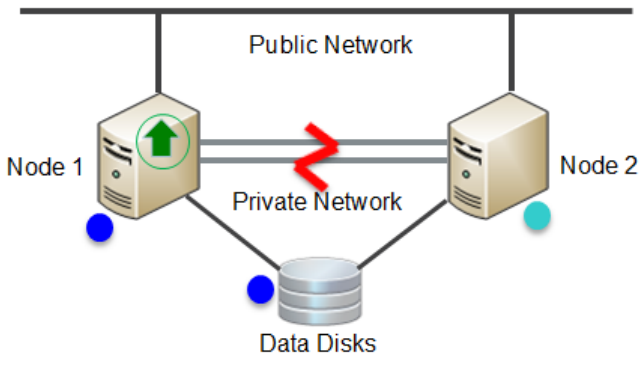
UNIX



The node that retains/acquires reservations on coordination points preempts the keys of other node. The node whose keys are removed then **panics**.

vs

Windows



The node that retains/acquires reservations on coordination points preempts the keys of other node. The node whose keys are removed **does not panic**.

Steps in case of a spilt-brain

UNIX— fencing-based model

1 LLT informs GAB of a heartbeat loss

When LLT on a node no longer receives heartbeat messages for a predefined time, LLT informs GAB of the heartbeat loss.

2 Nodes check for sub-cluster

Nodes synchronize to check which nodes they can still coordinate with. Based on their coordination a single cluster breaks in to sub-clusters.

3 GAB updates node membership

For the nodes with which coordination cannot be achieved, GAB marks the nodes as DOWN, updates the cluster membership, and informs the change to the fencing module.

4 Winning Racer ejects other keys

To connect with the coordination points, the fencing module determines a racer node (node with the lowest node-id) in each sub-cluster.
The Racer nodes race to connect with the coordination points and checks if the reservations of all the nodes from its sub-cluster are still alive.
If the reservations are alive, it has won the race. It then preempts the keys of the nodes that do not belong to its sub-cluster.
Note: By default, the fencing driver favors the sub-cluster with maximum number of nodes during the race for coordination points. With the preferred fencing feature, you can specify how the fencing driver must determine the surviving sub-cluster.

5 Panics losing nodes

The racer node informs the other nodes in its sub-cluster that it has won the race. Similarly, the losing racer node informs the other nodes in its sub-cluster that it has lost the race.
The fencing driver updates VCS about the new cluster membership and panics the losing sub-cluster nodes.

6 Recovers services

The nodes from the winning sub-cluster continue to run and recover services that were running on the nodes in the losing sub-cluster.

7 Nodes attempt to rejoin cluster

After a reboot the nodes that had panicked attempt to rejoin the cluster. If the network links are restored, the nodes succeed to register their keys on the coordination disks and join the cluster. If the network links are not restored, the nodes fail to register their keys on the coordination disks and log an error indicating that the spilt-brain condition prevails. Admin intervention is then required to resolve the issue and add the nodes to the cluster.

vs

Windows— challenge-defense mechanism

1 LLT informs GAB of a heartbeat loss

When LLT on a node no longer receives heartbeat messages for a predefined time, LLT informs GAB of the heartbeat loss.

2 Nodes check for sub-cluster

Nodes synchronize to check which nodes they can still coordinate with. Based on their coordination a single cluster breaks in to sub-clusters.

3 GAB updates node membership

For the nodes with which coordination cannot be achieved, GAB marks these nodes as DOWN and updates the cluster membership.

4 Nodes from each sub-cluster determine a target to host services

The nodes from the sub-cluster determine which node will continue to service or attempt to online the service groups that were running on the active node.

5 Nodes attempt to clear SCSI reservations

The nodes in the sub-cluster attempt to clear the SCSI reservations from the data disks.
This attempt is made by all the nodes in each sub-cluster. Eventually, the node that manages to clear the reservations first wins the challenge-defense race.

6 Winning node reconfirms if the active node is alive

The node that was successful to clear the other reservations waits for two SCSI cycles + 1 second to ensure if the initial active node updates the reservations again.

7 Winning node imports the disk group and starts the services

If the original active node still does not register its reservations, the node that successfully cleared reservations, imports the disk group and starts the services.

Note: The fencing model on UNIX and the challenge-defense mechanism on Windows are applicable in case of single site clusters only. These methods do not apply in case of disaster recovery clusters. For disaster recovery clusters, a Steward Process is followed to handle spilt-brain.

For details about types of spilt-brain conditions, configuring I/O fencing, enabling or disabling preferred fencing policy, and administering I/O fencing, see:

White paper on I/O fencing

Cluster Server Administrator's Guides on SORT