VOTING DISK INTERNALS

Introduction

In RAC, CSSD processes (Cluster Services Synchronization Daemon) monitor the health of RAC nodes employing two distinct heart beats: Network heart beat and Disk heart beat. Healthy nodes will have continuous network and disk heartbeats exchanged between the nodes. Break in heart beat indicates a possible error scenario. There are few different scenarios possible with missing heart beats:

1. Network heart beat is successful, but disk heart beat is missed.
2. Disk heart beat is successful, but network heart beat is missed.
3. Both heart beats failed.

In addition, with numerous nodes, there are other possible scenarios too. Few possible scenarios:

1. Nodes have split in to N sets of nodes, communicating within the set, but not with members in other set.
2. Just one node is unhealthy.

Nodes with quorum will maintain active membership of the cluster and other node(s) will be fenced/rebooted. I can't discuss all possible scenarios in a blog entry, so we will discuss a simplistic 2-node single voting disk alone here.

Voting disks are used to monitor the disk heart beats. It is preferable to have at least 3 voting disks or odd number of voting disks greater than or equal to 3.

CSSD is a multi-threaded process

Voting disks are shared between the nodes and should be visible from all nodes, stating the obvious. CSSD process is a multi-threaded process and a thread of the CSSD process monitors the disk heart beat. The disk HB (Heart Beat) thread is scheduled approximately every second and that thread verifies the disk heart beat from all active nodes in the cluster. Also, another thread of CSSD monitors the network heart beat. Pstack (Solaris) of CSSD process would show the threads of CSSD process.

Details: write calls

CSSD process in each RAC node maintains it heart beat in a block of size 1 OS block, in the voting disk. In Solaris VM that I was testing, OS block size is 512 bytes (We will discuss just Solaris alone in this post). In addition to maintaining its own disk block, CSSD processes also monitors the disk blocks maintained by the CSSD processes running in other cluster nodes.

CSSD process writes a 512 block to the voting disk in a specific offset. The written block has a header area with the node name. For example, in the pwrite call below, node name of solrac1 is in the first few bytes of the block. Third line printed below is keeping track of heart beat counter. This heart beat counter looks similar to the SCN mechanism, "0F 9D02" is the sequence number for the first write.

Also, Notice that the offset for the pwrite call 0×04002400. Node solrac1 writes a 512 byte block starting at the offset 0×04002400.

```
"/14:  pwrite(256, 0x019F4A00, 512, 0x04002400)      = 512"
"/14:     e t o V02\0\0\00104\v02\0\0\0\0 s o l r a c 1\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 c JD2\n0F9D02\003\0\0\0"
"/14:    \00303\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    F0 xBE L e01EC\0 e ;\0\0\0\0\0003\0\0\0 { =BE L1C87A8 L\0\001\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
```

Next write call from the local node CSSD process shows an increase in the counter. Value of that counter in line 3 went up from "0F 9D02" -> "10 9D02". BTW, I am removing few lines printed with to improve readability. So, the counter is incremented for every heart beat.

```
"/14:  pwrite(256, 0x019F4A00, 512, 0x04002400)      = 512"
"/14:     e t o V02\0\0\00104\v02\0\0\0\0 s o l r a c 1\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 c JD2\n109D02\003\0\0\0"
"/14:    \00303\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
...
"/14:    F1 xBE L N05EC\0 f ;\0\0\0\0\0003\0\0\0 { =BE L1C87A8 L\0\001\0"
...
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
```

Details: read calls

After the successful write, CSSD process also reads the blocks maintained by CSSD processes from other nodes. For example, in the pread output below, CSSD process in the solrac1 node is reading the block maintained by CSSD@solrac2 node using a pread call. Third line in the listing below has a sequence value of "FB 9702" for the node solrac2. A different sequence number is used by each node.

Also, Notice the offset for the pread call is 0×04002200 and that offset is different between pread and pwrite calls. Essentially, Node solrac2 is writing its heart beat starting at offset 0×04002200 and solrac1 is writing its heart beat at an offset of 0×04002400. The difference between these two offset values are exactly 0×200, which is 512 bytes.

In a nutshell, node Solrac2 maintains the heart beat disk block at an offset of 0×04002200 and the node solrac1 maintains the heart beat in the next 512 byte disk block.

```
"/14:  pread(256, 0x019F5000, 512, 0x04002200)       = 512"
"/14:     e t o V01\0\0\00104\v02\0\0\0\0 s o l r a c 2\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 c JD2\nFB9702\003\0\0\0"
"/14:    \00303\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    F0 xBE L KD6E9\0 /\t\0\0\0\0\0\003\0\0\0 m oBE L1C87A8 L\0\001\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
```

Next read from the offset 0×04002200 shows that solrac2 also increased the counter from “FB 9702" to “FC 9702".

```
"/14:  pread(256, 0x019F5000, 512, 0x04002200)       = 512"
"/14:     e t o V01\0\0\00104\v02\0\0\0\0 s o l r a c 2\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0 c JD2\nFC9702\003\0\0\0"
"/14:    \00303\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
...
"/14:    F1 xBE L 5DAE9\0 0\t\0\0\0\0\0\003\0\0\0 m oBE L1C87A8 L\0\001\0"
...
"/14:    \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"
```

Summary

In essence, disk heart beat is maintained in the voting disk by the CSSD processes. If the disk block is not updated in a short timeout period, that node is considered unhealthy and may be rebooted depending upon quorum of that node(or Shot in the head) to avoid split brain situation.

As evidenced in this blog, there isn’t really any useful data kept in the voting disk. So, if you lose voting disks, you can simply add them back without losing any data. But, of course, losing voting disks can lead to node reboots. If you lose all voting disks, then you will have to keep the CRS daemons down, then only you can add the voting disks.

This blog also begs the question about performance. How many I/O calls are performed against these voting disks? As the number of nodes increases, I/O also increases. For 2 node RAC, there are 2 reads (CSSD also reads another block, not sure why though) and 2 writes per second. With 6 nodes in the cluster, it will be 35 reads and 6 writes per second. From 11g onwards, you could keep voting disks in ASM.