System Lock Recovery [6]

If a Cray Research system in an SFS cluster crashes while it holds data locks on a shared file system, other systems in the cluster may be locked out of that file system. For this reason, failures of individual Cray Research systems in an SFS cluster need to be detected and recovered. The UNICOS SFS feature implements *heartbeat* mechanisms that allow properly functioning machines in an SFS cluster to detect failures of other machines in the cluster and to initiate recovery procedures to clean up any locks left behind by other systems in the cluster that have ceased to function.

The semaphore device provides support for system heartbeat monitoring. Each time the semaphore device services a semaphore operation for a client port, it records that fact as a bit in a row in a heartbeat matrix. Part of the response status for each semaphore operation is the client ports column out of the heartbeat matrix. In this manner, each client port can find the identity of the other client ports that have recently accessed the semaphore device. A special kernel thread monitors the heartbeat status of other client systems in an SFS Cluster, and insures that the system it represents keeps a current heartbeat within the semaphore device. If another client system fails to respond within several heartbeat intervals, that system is declared non-responsive.

The SFS heartbeat procedures and system lock recovery mechanisms are initiated automatically when the `sfs_start` command is executed.

6.1 System recovery commands

The commands that check the heartbeat status of the systems in an SFS cluster and initiate recovery when necessary are called automatically, in the following sequence:

- `sfs_start` initiates `codeblue`.
- `codeblue` executes `sfsprefix` (when necessary).
- `sfsprefix` executes `sfsrecover`.
- `sfsrecover` executes `sfsck`, `sfsclrlck`.

The actions of the system recovery commands are summarized as follows:

`codeblue`

`codeblue` registers with the kernel to receive heartbeat state information when it changes. `codeblue` is also responsible for creating and maintaining a log of system lock recovery related events. Upon detecting a system that transitions to a `DOWN` state, `codeblue` initiates a specified recovery action. The default action is to execute `/etc/sfsprefix`.

`/etc/sfsprefix`

`sfsprefix` first attempts to lock the SFS cluster global recovery semaphore. It is likely that all still-running systems will simultaneously notice when one system fails. The first system that successfully locks the recovery semaphore proceeds with the recovery; other systems wait their turn and typically find no recovery work to do. If the system that was conducting a recovery also fails, then the remaining systems have an opportunity to complete the recovery action for all failed systems. `sfsprefix` is a process that initiates recovery actions, and monitors the exit status of those actions. If it is found that a recovery action failed, `sfsprefix` can reinitiate the recovery. One reason that a recovery action would fail would be detection of the failure of yet another system. The default action is to execute `/etc/sfsrecover`.

`/etc/sfsrecover` *port* `#`

`sfsrecover` uses the port number as defined in `/dev/smp` on the machine that failed to indicate which system to recover. `sfsrecover` is an administrative shell script. The purpose of this script is to determine the list of shared file systems that were mounted on the failed system, and to execute the `/etc/sfsck` and `/etc/sfsclrlck` processes against those shared file systems.

`/etc/sfsck`

`sfsck` is the main lock recovery process. All the inodes in the subject shared file system are scanned, using the appropriate SFS semaphore locking protocols. File locks held by the failed system are released, and the inode updated.

`/etc/sfsclrlck` — `sfsclrlck` scans the semaphore device, clearing any semaphores held by the failed system.

System lock recovery actions perform their function while the shared file systems they are recovering are still mounted and used by the remaining Cray Research systems in the SFS cluster, allowing the work being done by other machines in the cluster to continue unaffected.

6.1.1 `codeblue` *program logic*

The following is a summary of the program logic of the `codeblue` program.

The `codeblue` command performs the following processes.

1. Parses the `codeblue` options:

   `-d` *program* — Indicates *program* to run when a machine in the SFS cluster goes down. The default program is `/etc/sfsprefix`.

   `-n` *script* — Indicates *script* to run when a machine in the SFS cluster becomes active for the first time. The default value of *script* is `NULL`.

   `-u` *script* — Indicates *script* to run when a machine in the SFS cluster comes up again after a halt. The default value of *script* is `NULL`.

   `-o` *pathname* — Indicates the path name of the file into which the output of the programs run by `codeblue` should be directed. The default is `/tmp/recovery.log`.

   `-w` *sec* — The number of seconds to delay before checking the heartbeat table for state changes. The default is 5 seconds.

2. Ensures that this is the only copy of `codeblue` running.
3. Ensures that system is running the SFS feature.
4. Makes the process a daemon and opens a log file.
5. Opens the `/dev/sfs` file.
6. Checks the heartbeat table.

7. Reads the heartbeat table.
8. Executes a system call that awaits state changes in the SFS cluster. The process goes to sleep waiting for a change in the heartbeat table. When that occurs, it executes a system call to check on the nature of the state change. If the state change is from `ACTIVE` to `DEAD`, a mask is prepared to indicate that the port is down. If there are ports needing recovery, `/etc/sfsprefix` is spawned (unless a different script has been indicated with the `-d` option).

6.1.2 `sfsrecover` *program logic*

The following is a summary of the program logic of the `sfsrecover` program. The `sfsprefix` script executes the `sfsrecover` command, which performs the following operations.

1. Determines which ports are down and builds a port mask of those systems.
2. Prints the port mask of down systems.
3. Executes `/etc/sfsclrlck` with the `-p` (list of ports) option. This clears any semaphores held by the failed system.
4. Executes `mount -s` to get a list of mounted shared file systems.
5. For all mounted file systems, performs the following actions:
   - Validates file system parameters.
   - Checks if the file system is mounted on any of the systems that are to be recovered.
   - Executes `sfsck` with `-r`, `-P`, and `-p` options. The `-r` option indicates repair mode. The `-P` option monitors the Ports Needing Recovery (PNR) mask. The `-p` option indicates the ports with locks to be released.
   - Checks to see if `sfsck` exited with a return code of 99, which indicates that the PNR mask changed. If so, exits the program so that `codeblue` can be rescheduled.
   - If `sfsck` executed successfully, adds the file system to the list of file systems to be unmounted.
6. For all file systems to be unmounted, executes `umount` with `-p` (list of ports) option.

6.1.3 Sample recovery log

The following shows a sample log from an SFS recovery session.

```
Tue Oct 11 17:12:27 1994 [hosta (port 1)] /etc/codeblue: Awakened.
Tue Oct 11 17:12:27 1994 [hosta (port 1)] /etc/codeblue:   Port :  0 1 2 3 4 5 6 0
Tue Oct 11 17:12:27 1994 [hosta (port 1)] /etc/codeblue:   State:  . ^ ^ . . . . .
Tue Oct 11 17:12:27 1994 [hosta (port 1)] /etc/codeblue:   Resp :  0 0 0 0 0 0 0 0
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue: PNR after re-checking HB table: 0
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue: port 2 state transition
  (NEW -> ACTIVE)
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue: Awakened.
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue:   Port :  0 1 2 3 4 5 6 0
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue:   State:  . ^ ^ . . . . .
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue:   Resp :  0 0 0 0 0 0 0 0
Tue Oct 11 17:12:28 1994 [hosta (port 1)] /etc/codeblue: PNR after re-checking HB table: 0
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue: Awakened.
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue:   Port :  0 1 2 3 4 5 6 0
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue:   State:  . ^ v . . . . .
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue:   Resp :  0 0 14 0 0 0 0 0
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue: port 2 state transition
   (ACTIVE -> DEAD)
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue: PNR after re-checking HB table: 100
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue: down_path: /etc/sfsprefix
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue: Starting recovery for port(s): 2
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/codeblue: spawning ’/etc/sfsprefix 2’
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: TSET’ing recovery sema 2
  (timeout = 6s)
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: Recovery sema 2 SET.
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: Recovery timeout for port(s) 2 = 0s.
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: Reading PNR information from ESD area
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: read OK - PNR mask = 100
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: Recovering for port(s) 2
  (port_mask: 100   PNR mask: 100)
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: Starting ’/etc/sfsrecover’ for
  port(s) 2.
Tue Oct 11 20:22:05 1994 [hosta (port 1)] /etc/sfsprefix: ’/etc/sfsrecover’ for port(s) 2
   started (pid 1160).
Tue Oct 11 20:22:07 1994 [hosta (port 1)] /etc/sfsrecover: INFO - recovery mask pattern =
  ports=??m?????

Tue Oct 11 20:22:07 1994 [hosta (port 1)] /etc/sfsrecover: INFO - calling /etc/sfsclrlck -p 2
Tue Oct 11 20:22:07 1994 [hosta (port 1)] /etc/sfsclrlck:  --- clearing semas for downed
  systems (ports 2) ---
Tue Oct 11 20:22:07 1994 [hosta (port 1)] /etc/sfsclrlck: FSREC Sema state: 1    (last changed
  by port 1)
```

```
Tue Oct 11 20:22:09 1994 [hosta (port 1)] /etc/codeblue: Recovery watchdog sez, ’woof! Checking
  PNR status...’
Tue Oct 11 20:22:09 1994 [hosta (port 1)] /etc/sfsclrlck: cleaned up 0 clear-on-exit semas
  for port 2
Tue Oct 11 20:22:09 1994 [hosta (port 1)] /etc/sfsclrlck: FSREC Sema state: 1    (last changed
  by port 1)
Tue Oct 11 20:22:10 1994 [hosta (port 1)] /etc/codeblue: Checking PNR status...
Tue Oct 11 20:22:10 1994 [hosta (port 1)] /etc/codeblue: PNR status:  orig: 100  cur: 100


        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%
        %%         RECOVERING /dev/dsk/sfs_scsi for ports 2
        %%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Tue Oct 11 20:22:10 1994 [hosta (port 1)] /etc/sfsrecover: INFO - RECOVERING /dev/dsk/sfs_scsi
  for ports 2
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/sfsrecover: INFO - /dev/dsk/sfs_scsi isn’t
  mounted on ports 2
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/sfsprefix: /etc/sfsrecover exited with
   status 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/sfsprefix: Recovery for port(s) 2 completed
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/sfsprefix: PNR bits cleared for port(s) 2. PNR
  mask now: 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/sfsprefix: Clearing sema 2 (timeout = 6s).
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/sfsprefix: sema 2 CLEARED.
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: Checking PNR status...
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: PNR status:  orig: 100  cur: 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: /etc/sfsprefix exited with status 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: recovery succeeded for port(s) 2
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: Awakened.
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue:   Port :  0 1 2 3 4 5 6 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue:   State:  . ^ v . . . . .
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue:   Resp :  0 0 14 0 0 0 0 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: port 2 state transition (ACTIVE -> DEAD)
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: PNR after re-checking HB table: 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: Awakened.
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue:   Port :  0 1 2 3 4 5 6 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue:   State:  . ^ v . . . . .
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue:   Resp :  0 0 14 0 0 0 0 0
Tue Oct 11 20:22:11 1994 [hosta (port 1)] /etc/codeblue: PNR after re-checking HB table: 0
```

6.2 The sfsmddcp command

You may sometimes need to rebuild a leg of a shared file system that includes mirrored legs. When rebuilding the mirrored legs of a shared file system, the operating system must be aware of SFS locking protocols and must be able to rebuild while the shared file system is still mounted and being used by other systems. The sfsmddcp command is a special version of the mddcp(8) utility that you can use for this.