Configuring the SFS Environment Manually [4]

This chapter describes how to configure a Cray Research computer system to use the Shared File System (SFS) feature and how to create SFS file systems manually, as an alternative to using the UNICOS installation and configuration menu system. If you configured the SFS environment by using the menu system, you can ignore this chapter.

The examples in this chapter initialize an SFS environment using HIPPI or GigaRing I/O-basedIdisks.

4.1 Configuring SFS device nodes

Configuring a Cray Research system to use the SFS feature involves the creation of several special-purpose device nodes for each file system that you define.

The following is a list of device nodes you will create when configuring the SFS environment, along with a brief description of the purpose of each device.

Device	Description
/dev/smp	The low-level raw interface to the semaphore device.
/dev/hdd/smp, /dev/xdd/smp	The IPI-3 and GigaRing driver nodes, respectively. They contain the information for physically accessing the semaphore device and are used by /dev/smp. This is named smp by convention.
/dev/sfs	The interface to the SFS software administration programs. This device driver is responsible for the management of the Shared Lock Region. The Shared Lock Region is a small piece of the shared media pool used by the SFS feature to record common information such as semaphore assignment, system state, etc.
/dev/dsk/slr	The path to the Shared Lock Region slice on the shared media, used in defining /dev/sfs.

`/dev/dsk/smnt`	The path to the Shared Mount Table, which is contained within the Shared Lock Region of the shared media pool.

4.1.1 Configuring the semaphore device

Configuring the semaphore device for shared file systems is a two-step process, which you perform for each system in the SFS cluster. You must use the same major and minor numbers for each system in the SFS cluster.

The first step is to create the device node which describes the I/O path in order to reach the semaphore device from the Cray Research system you are configuring.

The following is an example of using the `mknod` command to create this node for a HIPPI-based semaphore device:

```
/etc/mknod /dev/hdd/smp c 60 1 1 20 0 0 34 0 255 6769
```

Parameter	Description
`/dev/hdd/smp`	This is the node name being created. For GigaRing devices, this would more properly be named `/dev/xdd/smp`.
60	Major number for IPI-3 (`hdd`) disk driver. The major number for GigaRing I/O disk driver (`xdd`) is 33.
1	Minor number. Do not use 0 as the minor number for a `/dev/hdd` node, as this causes a conflict with the `hddmon` command.
1	Type. An HD-16 device is type 1, which indicates a 16 kbyte sector device. An HD-32 device is type 2, which indicates a 32 kbyte sector device. An HD-64 device is type 1, which indicates a 64 kbyte sector device. For GigaRing devices, use 0.
20	I/O path
0	Start
0	Length
34	Flags
0	Reserved
255	Unit

6769	I-field (smp 's address on the HIPPI network when accessed from this system) or 0 for a GigaRing device

Define the physical device driver, /dev/smp, as in the following example:

```
/etc/mknod /dev/smp c 73 0 4 0 /dev/hdd/smp 0 0
```

Parameter	Description
73	Major number for smp device.
0	Minor number
4	smp device type (4 indicates HIPPI semaphore, 6 is used for xdd).
0	Port number. On HIPPI-based semaphore systems, you must put a unique number in this field for every machine in the cluster. This number is used to uniquely identify each machine in the cluster.
/dev/hdd/smp	Path name to the hdd node created in the previous step (/dev/xdd/smp if using GigaRing channel).
0	Reserved
0	Reserved

Caution: When defining the physical device driver, take care to ensure that the port number is a unique number for every machine in the cluster.

You can execute the /etc/sema command to determine whether the semaphore box you have defined is available to the system.

Caution: Do not run the sema command after you have started the system daemons by running sfs_start.

The following shows an example of the sema command. This command executes a TEST command against semaphore 1 in the /dev/smp device and prints the results.

```
/etc/sema TEST 1 /dev/smp
```

4.1.2 Defining the sfs *device*

The sfs device must be configured on each system of the SFS environment.

To define the sfs device, first define the area of the shared media that you will use for the Shared Lock Region. To do this, define a physical disk device and then define

a logical disk device which contains a single slice consisting of the physical disk device you have just created.

The following example defines a physical disk device:

```
mknod /dev/hdd/slr c 60 3 1 20 0 1024 040 0 16 6769
```

Parameter	Description
/dev/hdd/slr	This is the device being created. For GigaRing devices, this would more appropriately be /dev/xdd/slr.
60	Major number for IPI-3 disk driver. For GigaRin devices, this would be 33 for the GigaRing (xdd) disk driver.
3	Minor number for device
1	Type (1 indicates 16 kbyte sector device, 2 indicates 32 kbyte sector device, 3 indicates 64 kbyte sector device)
20	I/O path
0	Starting sector
1024	Number of sectors
040	flags
0	Reserved (must be 0)
16	Unit (facility ID)
6769	I-field (disk's address on the HIPPI network when accessed from this system). Set to 0 for GigaRing devices.

The following example defines the logical disk device containing the physical disk device defined in the previous example (for HIPPI).

```
mknod /dev/dsk/slr b 34 200 0 0 /dev/hdd/slr
```

The following command shows an example of sfs node configuration:

```
/etc/mknod /dev/sfs c 48 0 0 0 /dev/dsk/slr 0 0
```

Parameter	Description
48	Major number for sfs device
0	Minor number
0	Type
0	Reserved
/dev/dsk/slr	Path name to Shared Lock Region slice, defined in the previous step
0	Reserved

0	Reserved

4.1.3 Defining the Shared Mount Table

The Shared Mount Table takes its space from the area you have defined as the Shared Lock Region. Use the `mknod` command to define the Shared Mount Table configuration. As the following example shows, 75 is the major device number for the Shared Mount Table and is the only significant parameter for this device type. The remainder of the arguments can be set to 0.

```
/etc/mknod /dev/smntent c 75 0 0 0 0 0 0 0 0 0
```

4.2 Creating SFS file systems

You create SFS file systems the same way as you create standard NC1FS file systems. An SFS file system is laid out in the same way as an ordinary NC1FS file system. It may be as simple as a single slice, or it may use the logical device constructs that incorporate mirroring and/or striping.

Note: Mirroring is not supported on CRAY T3D or CRAY T3E systems.

4.2.1 Describing slices on a HIPPI disk

You describe the slices on a HIPPI disk the same way as you define a slice for a standard NC1FS file system. The following example defines an hdd slice, which has a major device number of 60 and a minor device number of 101.

```
/etc/mknod /dev/hdd/h01 c 60 101 1 01230 0 10000 0 0 2 7
/etc/mknod /dev/dsk/hfs01 b 34 120 0 0 /dev/hdd/h01
```

Caution: Device names and major/minor numbers must be the same across all of the systems in an SFS cluster. Further, minor numbers must be less then 256 and multiple partitions from the same device should not be SFS exported (See "SFS Restrictions and Limitations," Chapter 9, for details).

Only the I/O path and the I-field information may be different across systems. The rest of the disk definition must be identical for a given SFS file system across all systems in an SFS cluster.

4.2.2 Describing slices on a GigaRing-based disk

You describe the slices on a GigaRing FCN, MPN or HPN disk the same way as you define a slice for a standard NC1FS file system. The following example defines an xdd slice, which has a major device number of 33 and a minor device number of 101.

```
/etc/mknod /dev/xdd/x01 c 33 101 1 01230 0 10000 0 0 2 7
/etc/mknod /dev/dsk/xfs01 b 34 120 0 0 /dev/xdd/h01
```

Caution: Device names and major/minor numbers must be the same across all of the systems in an SFS cluster. Further, minor numbers must be less then 256 and multiple partitions from the same device should not be SFS exported (See "SFS Restrictions and Limitations," Chapter 9, for details).

Only the I/O path may be different across systems. The rest of the disk definition must be identical for a given SFS file system across all systems in an SFS cluster.

4.2.3 Creating a shared file system

You create an SFS file system by using the mkfs(8) command. All standard NC1FS options of mkfs can be used when defining an SFS file system.

When creating an SFS file system, you must use the -s option of the mkfs command. The -s option requires an argument, which defines the number of semaphores to be assigned to the system at mount time. This number is recorded in the superblock of the SFS file system.

An SFS file system requires multiple semaphores to operate. The first assigned semaphore is used to protect global file system related metadata, specifically the superblock, dynamic block, allocation map, and Inode allocation maps. Remaining semaphores are then used, in a hashed manner, to protect individual inode sectors during inode updates.

The more semaphores you assign to a file system, the better the performance. This is because more semaphores result in less contention for inode semaphores, allowing more simultaneous inode sector updates.

You cannot assign more than 2048 semaphores in total to shared file systems. If you create more than one shared file system, you must allocate your semaphores according to your anticipated needs for each file system. When apportioning semaphores among shared file systems, you should assign a greater percentage of available semaphores to the more heavily used file systems.

If you attempt to mount a file system that will bring the number of assigned semaphores over the limit of available semaphores, the file system will not mount and an error message will result. If this should occur, you can use the -s option of the setfs command to change the number of semaphores assigned to a file system that

is not mounted. You can also use this option to increase the number of semaphores assigned to a file system if you are removing existing shared file systems.

4.2.4 Changing file system types

You can change a file system created as a shared file system (file system type SFS) to an NC1FS file system and, conversely, you can change a file system created as an NC1FS file system to an SFS file system by using the -s option of the setfs(8) command. This allows more flexibility and speed in system administration, supporting such actions as making and restoring a file system in non-shared (fully cached) mode, and then marking the file system as shareable.

For more information about using the -s option of the setfs command, see the setfs(8) man page.

4.2.5 Adding SFS entries to the /etc/fstab file

After defining an SFS file system, you may choose to add the file system name to the /etc/fstab file of every system in the SFS cluster. SFS entries in /etc/fstab are in the same format as NC1FS entries, with SFS specified as the file system type. The following is an example of an fstab file that includes both NC1FS and SFS entries.

```
/dev/dsk/root                    NC1FS   rw,CRI_RC="YES" 1       2
/dev/dsk/usr          /usr       NC1FS   rw,CRI_RC="YES" 1       2
/dev/dsk/src          /usr/src   NC1FS   rw,CRI_RC="YES" 1       2
/dev/dsk/sfs_usr1     /sfs/usr1  SFS     rw,CRI_RC="YES" 1       2
```

If you list an SFS file system in /etc/fstab, the file system will be checked and mounted when sfs_start is run, as described in Chapter 5, page 33. This is the recommended way of mounting shared file systems because sfs_start performs valuable cross-system integrity checks on your shared file systems before mounting them.

4.3 The /etc/config/sfs file

A UNICOS SFS configuration supports multiple SFS arbitration devices. Each Cray Research system in the SFS cluster that will be sharing file systems must include an /etc/config/sfs file that contains the names of each Cray Research system in the cluster and which SFS arbiters are associated with each system. The

`/etc/config/sfs` file also describes the identity of the SFS arbiters and defines the character special devices that support each arbiter. For a description of the format of this file, see the `sfs`(4) man page.

You must maintain identical `/etc/config/sfs` files on all systems in an SFS cluster.