Netra ft™ 1800
400 MHz CPUset Upgrade

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part No. 806-4639-10
April 2000, Revision A

Send comments about this document to: docfeedback@sun.com

Contents

Preface

This document describes two approaches for upgrading a Netra ft 1800 system from 300 MHz CPUsets to 400 MHz CPUsets:

- Using Split Mode
- Using system shutdown.

It is recognized that the relative advantages and disadvantages of both approaches are dependant not only on the application downtime, but also on the complexity of the approach. The split mode procedure, whilst more complex, is likely to result in the least amount of downtime.

Typographic Conventions

TABLE P-1 Typographic Conventions

Typeface	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the *User's Guide.* These are called *class* options. You *must* be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type `rm` *filename.*

Shell Prompts

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	*machine_name*%
C shell superuser	*machine_name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#
OpenBoot PROM	ok
Kernel debugger	kdb

Ordering Sun Documentation

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at:

`http://www1.fatbrain.com/documentation/sun`

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@sun.com`

Please include the part number (806-4639-01) of your document in the subject line of your email.

CHAPTER 1

Before You Start

The system must be set up according to the instructions in the *Netra ft 1800 User's Guide* and *Netra ft 1800 Hardware Reference Manual*. In particular, the system must be running Solaris 2.6 Netra ft 1800 Update 01 software, with Sun Enterprise Volume Manager (SEVM) 2.5.

Prerequisites

The configuration of the Netra ft 1800 *must* consist of at least the following components:

- Two 300 Mhz CPUsets, both at Update 01 firmware levels
- Two HDDs, one in A-DSK0 and one in B-DSK0, creating one root mirror
- Two CAFs
- At least two PSUs per side, depending on configuration

Optional components include:

- PCI cards
- Data (non-boot) HDDs

▼ To Check the Software, Firmware and Hardware Levels

Use the following procedure to check that the system is at Update 01 level.

Check software level

1. **At the system prompt, type:**

```
# uname -a
SunOS foo 5.6 107548-10 sun4u sparc SUNW,Ultra-4FT
```

to confirm the software level. The response should be as shown, with `foo` replaced by the system hostname.

Check firmware level

2. **Type:**

```
# /usr/platform/SUNW,Ultra-4FT/SUNWcms/sbin/cmsfruinfo -l
A-MBD EE_EEPROM | grep FWARE
EE_MBD_RCP_FWARE_PARTNO=2587132
EE_MBD_RCP_FWARE_DASH=02

EE_MBD_BRIDGE_FWARE_PARTNO=2587771
EE_MBD_BRIDGE_FWARE_DASH=10
```

Then type:

```
# /usr/platform/SUNW,Ultra-4FT/SUNWcms/sbin/cmsfruinfo -l
B-MBD EE_EEPROM | grep FWARE
EE_MBD_RCP_FWARE_PARTNO=2587132
EE_MBD_RCP_FWARE_DASH=02

EE_MBD_BRIDGE_FWARE_PARTNO=2587771
EE_MBD_BRIDGE_FWARE_DASH=10
```

This confirms the firmware levels.

Check OBP level

3. **Type:**

```
# prtconf -V
OBP 3.7.28.0 1999/10/26 10:32
```

This confirms the OpenBoot PROM firmware level, which should be 28.

CHAPTER 2

Using Split Mode to Upgrade a CPUset

This procedure describes how to upgrade from 300 MHz CPUsets to 400 MHz CPUsets on a Netra ft 1800, using split mode.

The steps are as follows:

1. The system is split, with the split winner (side A) continuing to provide service.
2. The split loser's (side B) CPUset is upgraded.
3. The application is migrated from side A to side B.
4. Side A's CPUset is upgraded.
5. The system is merged.

Note – Remember that the nodename and hostid of the split loser will be different at the end of this procedure. You should ensure that the applications, connections, and disk layout are able to handle this.

Note – You will need the two SEVM license keys that were originally supplied by the License Centre.

▼ To Upgrade the CPUset

Split the system

1. **Split the system and wait for the split loser to reboot.**

   Refer to “Splitting the System” on page 13.

Clear part numbers

2. **Clear the part numbers of both CPUsets on the split losing side (side B) using** `cmsconfig`**. This will prevent the CPUsets from going** `enable_failed` **when they are enabled.**

   Clear the part numbers for A-CPU and B-CPU on the B side as follows:

   a. Start `cmsconfig`**.**

   b. Type the item number for A-CPU and press Return.

   c. Type the item number for the part number and press Return.

   d. Press Return to enter a null part number.

   e. Press `q` **and Return to return to the main menu.**

   f. Type the item number for B-CPU and press Return.

   g. Type the item number for the part number and press Return.

   h. Press Return to enter a null part number.

   i. Press `q` **and Return twice to exit to the prompt.**

Shut down split loser to the PROM

3. **Shut down the split loser (side B) to the PROM:**

```
# init 0
```

Replace 300 MHz CPUset with 400 MHz CPUset

4. **Replace the existing 300 MHz CPUset with the new 400 MHz CPUset on the split loser's side (side B, that is, the left-hand CPUset labeled B-CPU).**

Note – The *Power* and *Split* LEDs will be illuminated; this is normal at this stage.

Refer to “To Remove a CPUset” on page 42 and “To Insert a CPUset” on page 44.

After inserting the new CPUset, and providing it has been prepared as described in “To Prepare the New CPUset” on page 44, the new CPUset should boot automatically.

New CPUset boots

5. Wait for the split loser to boot.

Migrate application to new CPUset

6. Migrate the application from the split winner (300 MHz CPUset) to the split loser (the new 400 MHz CPUset).

Refer to Appendix B ”Application Migration”.

a. Checkpoint the application state (for stateful applications).

b. Shutdown the application on the split winner.

c. Restore the application state on the split loser (for stateful applications).

d. Start up the application on the split loser (the new 400 MHz CPUset).

Clear split winner’s part number

7. Clear the running CPUset's part number on the split winner (side A) using `cmsconfig` as follows:

a. Start `cmsconfig`.

b. Type the item number for A-CPU and press Return.

c. Type the item number for the part number and press Return.

d. Press Return to enter a null part number.

e. Press `q` and Return to return to the main menu.

Shut down winner to the PROM

8. Shut down the split winner to the PROM:

```
# init 0
```

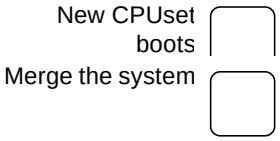
Replace 300 MHz CPUset with 400 MHz CPUset

9. Replace the existing 300 MHz CPUset with the new 400 MHz CPUset on the split winner’s side (side A).

Note – The *Power* and *Split* LEDs will be illuminated; this is normal at this stage.

Refer to “To Remove a CPUset” on page 42 and “To Insert a CPUset” on page 44.

After inserting the new CPUset, and providing it has been prepared as described in “To Prepare the New CPUset” on page 44, the new CPUset should boot automatically.



10. Wait for the split winner to boot.

11. Merge the system and wait for the new 400 MHz CPUsets to come into sync.

Refer to “Merging the System” on page 28.

Step 6 constitutes the only downtime for the application because the application can continue running during the other steps in the procedure. The downtime resulting from the migration of an application from the split winner to the split loser is dependant on the four parts of this step, which are themselves determined by the application itself. Hence, the downtime for an application using a Netra ft 1800 depends on the application:

- For stateless applications, the downtime is immediate because the application can be started on the split loser and switched over from the split winner immediately;
- For stateful applications, the downtime is dependant on all four steps, including the time to checkpoint the state of the application and then restore this state.

CHAPTER 3

Using System Shutdown to Upgrade a CPUset

This procedure describes how to replace 300 MHz CPUsets with 400 MHz CPUsets on a Netra ft 1800 system by shutting the system down.

Refer also to Appendix B ”Application Migration” on page 35.

▼ To Upgrade the CPUset

Checkpoint application state ☐

1. **Checkpoint the application state.**

Shut down application ☐

2. **Shut down the application.**

Clear CPUset part numbers ☐

3. **Clear the part numbers of both CPUsets using** `cmsconfig`**.**

   a. **Start** `cmsconfig`**.**

   b. **Type the item number for A-CPU and press Return.**

   c. **Type the item number for the part number and press Return.**

   d. **Press Return to enter a null part number.**

   e. **Press** `q` **and Return to return to the main menu.**

   f. **Type the item number for B-CPU and press Return.**

   g. **Type the item number for the part number and press Return.**

   h. **Press Return to enter a null part number.**

   i. **Press** `q` **and Return twice to exit to the prompt.**

Check SEVM state ☐

4. **Check the state of SEVM:**

```
# vxprint
```

   Ensure all volumes are fully mirrored, that is, all volumes and plexes are active.

Shut down system ☐

5. **Shut down the system:**

```
# init 5
```

Replace 300 MHz CPUsets ☐

6. **Replace the 300 MHz CPUsets with the new 400 MHz CPUsets.**

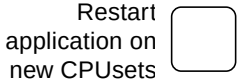
   Refer to Appendix C ”CPUset Replacement”.

System booted and in sync ☐

7. **Power on the system using the On buttons on the CAFs and wait for the system to boot and the CPUsets to come into sync.**

   Check that the CPUsets are configured correctly – refer to “To Prepare the New CPUset” on page 44.

Note – The system will only be fault tolerant if the disks are fully mirrored.



8. Restore the application state and start up the application.

Step 2 through Step 8 constitute the downtime for the application.

The shutdown and startup time must take into account the dependency of the application on the resources of a Netra ft 1800 system, such as `ft_networks`, disks etc. For example, a more complex disk setup will require more shutdown and startup time.

APPENDIX **A**

Splitting and Merging a System for CPUset Upgrade

Note – The procedures described here are specific to CPUset upgrade and should not be used for any other purpose.

Assume that the Netra ft 1800 system that is going through the upgrade cycle is:

- Called `foo`
- Running Solaris 2.6 Netra ft 1800 Update 01 software
- Running in combined mode

In addition, assume that the system is called `foo-2` after the merge.

To minimize service unavailability due to upgrades, at least two fault-tolerant networks are on the original fault-tolerant system, one of which is on a movable resource. The primary connection (this is the base name or the node name of the original fault-tolerant system) is called `foo`. An optional secondary connection is called `foo-1` and the mandatory secondary interface is called `foo-3`. Note that both the primary connection, `foo`, and the secondary connection `foo-1` are on the CAF, which is a fixed module in the split-mode sense.

Therefore, `foo-3` is used for the service that the application is providing, as this software resource is on a movable PCI module. The naming convention used here is reflected in the relevant ICN and split daemon configuration files where the name `foo-2` is reserved for the loser of the split operation.

When the split operation has completed, you can boot the loser of the split, side B, with its new identity (`foo-2`).

Meanwhile, it is possible that the server running on side A has updated the data that will be used by the upgraded server now running on side B.

Note that while in split mode the split daemons running on each side will communicate with each other using channel 0 of the ICN.

To minimize service unavailability, you must:

1. Disable the `ft_network` service that uses the backup PCI card, using the `cmsconfig` utility running on that side.
2. Transfer ownership of the backup PCI card (residing on side A in this example) to side B:

```
foo# splitconf -l <slot> -o B
```

   where <*slot*> is the location of the PCI card. Note that **`-l`** is 'minus ell', not 'minus 1'.

3. Enable it using `cmsconfig` on side B.
4. Define it as a component of the ft network interface, `foo-3`, on side B.
5. Enable the `ft_network` service using host name `foo-3` on side B, with the `cmsconfig` utility running on side B.

At this point, if the service is stateless, the service can be restored and clients of the service will see the entire upgrade procedure as the loss of their connection to the server. These clients will only need to reconnect to the server (running at the same host, `foo-3`) to take advantage of the upgraded service.

You can now merge the system and re-establish the fault tolerant pairs. The new system will have a different identity (host id and node name) from the original fault tolerant system, but that is not an issue if the applications, connections, and disk layout are properly thought out when you initially set up system.

While this procedure is sufficient for stateless servers or new services running on the system, the procedure for upgrading stateful applications is more complex and, therefore, more time consuming. This is because checkpointing and recovery of state must happen before the service can be correctly restored. The current state must be checkpointed before you merge the system and restart the upgraded service.

Note – This should happen after the service on the primary PCI card has been disabled. Once the system is merged, you can recover the state of the server application and restore the service.

Splitting the System

▼ To Prepare the System

Open xterms

1. **Open two terminals, one for each of the following:**
   - A-side console, to monitor the progress of the split on the A-side
   - B-side console, to monitor the progress of the split on the B-side

Set OBP variables

2. **In order to ensure that the split loser boots, set the OBP variables to the following values:**
   - `use-nvramrc?=true`
   - `auto-boot?=true`
   - `boot-device=a-dsk0 b-dsk0`
   - `diag-device=a-dsk0 b-dsk0`
   - `diag-switch?=true`

```
foo# eeprom use-nvramrc?=true
foo# eeprom auto-boot?=true
foo# eeprom boot-device="a-dsk0 b-dsk0"
foo# eeprom diag-device="a-dsk0 b-dsk0"
foo# eeprom diag-switch?=true
```

Quiesce losing side devices

3. **Quiesce all modules that will be used by the split loser after the split.**

   You need to ensure that no applications are running which use the hard disk drives or PCI cards.

   Assigning modules that contain open devices to the new system causes the split operation to fail, as such modules will not be disabled and the split operation will time out waiting for them to be disabled. The split daemon attempts to disable all modules that will be owned by the loser of the split operation before splitting the system. Each disable operation is expected to terminate within a timeout value. The default value is 60 seconds, and this can be modified by either a command line (using the `-t`) option to the `splitadm` command or by modifying the split daemon's configuration file where the timeout value is expressed in seconds. If one or more of the modules cannot be disabled within the specified time an error message identifying the list of modules that failed to disable is displayed. The split process then stops. At this point all the modules that have been successfully disabled will

remain disabled and the system will remain in the combined (or fault tolerant) mode. The operation can be retried with a longer timeout so that the device can be quiesced. If increasing the timeout value fails, it is likely that there is one or more busy devices that cannot be disabled. At this point it may be necessary to stop the use of these devices by all applications so that the split operation can succeed. As it may not be possible to quiesce some devices, the split mode software provides a `force` option. However, the use of this option is not recommended and is only acceptable as a last resort.

▼ Initial Configuration

Check system in combined mode

4. **Check that the system (`foo`) is in combined mode using the command:**

```
foo# splitinfo -a
attributes=combined
```

Ensure system not router

5. **Ensure that the Netra ft 1800 combined system is not acting as a router.**

   Create the file `/etc/notrouter` by issuing the command:

```
foo# /usr/sbin/in.routed -q
```

   The `/etc/notrouter` file will contain this command at the next reboot.

Disable data disks

6. Manually disable the data disks by removing all split loser data disks from SEVM control.

The exact procedure depends on the data disk configuration. The following is an example based on four data disks in two sets of mirrors as follows:

Location	SEVM name	CMS name
Disk group Appdg		
A-DSK1	AppMain	c2t1d0s2
B-DSK1	AppMirror	c3t1d0s2
Disk group Usersdg		
A-DSK2	UsersMain	c2t2d0s2
B-DSK2	UsersMirror	c3t2d0s2

The procedure is:

a. Remove the data disks AppMirror and UsersMirror from their respective disk groups:

```
foo# vxdg -g Appdg -k rmdisk AppMirror
foo# vxdg -g Usersdg -k rmdisk UsersMirror
```

The console should display messages similar to the following, indicating that the volumes are no longer fault tolerant but consist of just the disks on the split winner:

```
vxvm:vxconfigd: NOTICE: Detached plex AppVol-02 in volume AppVol
vxvm:vxconfigd: NOTICE: Detached plex UsersVol-02 in volume UsersVol
```

b. Remove the mirror data disk devices from SEVM control:

```
foo# vxdisk rm c3t1d0s2
foo# vxdisk rm c3t2d0s2
```

Disable losing side data disks

7. Use `cmsconfig` **to disable all losing side data disks (in this case,** `B-DSK1` **and** `B-DSK2`**).**

▼ To Configure for Split Mode

Add IP addresses

8. Add IP addresses to `/etc/hosts` for all networks, including `icn`:

```
#
# Internet host table
#
127.0.0.1       localhost       loghost
129.156.203.152 foo
129.156.203.18  foo-2
192.168.1.1     foo-i0
192.168.1.2     foo-2-i0
129.156.203.15  foo-1
129.156.203.16  foo-3
```

In addition, `localhost` has been set as the `loghost` in order to ensure that logging is performed on the local machine after a split. Otherwise, logging would be directed across the network to the other side of the Netra ft 1800.

Edit `splitd.conf` file

9. Edit the `/etc/splitd.conf` file to give the two hostnames required after the split:

```
.
.
.
.
.
host_prim foo-i0 foo-2-i0
host_alt foo foo-2
.
.
.
hostnames foo foo-2
```

The first `host-prim` name (`foo-i0`) is the split winner, and the second (`foo-2-i0`) is the split loser.

Edit nsswitch.conf file

10. If the system is an NIS system, edit the `/etc/nsswitch.conf` **file.**

Edit the `hosts:` line to ensure that the keyword `files` occurs before the keyword `xfn`, as follows:

```
.
.
hosts:     files xfn nis [NOTFOUND=return]
.
.
```

Add an icn network

11. Use `cmsconfig` **to add an ICN network.**

At the prompt, type **`i icn`** to add the network.

Set winner's ICN hostname

12. Use `cmsconfig` **to set the** *hostname* **entry in** `icn 0` **to the IP address of the split winner's ICN network.**

a. Enter the item number for `icn 0`**.**

b. Enter the item number for `hostname`**.**

```
Modify: icn0 hostname (foo-i0)

--------------------------------------------------

New value:
```

c. Enter the IP address of the split winner's ICN network and press Return.

d. Press `q` **and Return twice to exit.**

Set split master

13. Set the split master to side A:

```
foo# splitconf -m A
master = A
```

Confirm system is running combined

14. Use `cmsconfig` **to check** `ft_network 0`**.**

Check that `ft_network 0` is operating in fault-tolerant mode, that is, both A and B controllers are online.

```
Select: ft_network 0

Item  Name                     Value                Page 1 of 1
-------------------------------------------------------------
0     state                    online_up
1     description              network multiplexor
2     user_label
3     hostname                 foo
4     preferred_controller     none
5     controllerA_FRU          A-CAF
6     controllerA_Funct        Net_0
7     controllerB_FRU          B-CAF
8     controllerB_Funct        Net_0
9     info
10    busylock                 no
11    devpath                  pnet0
12    link                     100 Mbps half-duplex link up
13    transceiver              internal transceiver selected
14    usable_controllers       A (online) & B (online)
15    controller_in_use        A
```

If any other `ft_networks` are configured, however, make sure they are only online on the winning side, or the loser may try to duplicate IP addresses already in use by the winning side.

Check SEVM state

15. Check the state of SEVM by issuing the following commands:

a. Ensure the SEVM daemon is running:

```
foo# ps -ef -oargs | grep vx
vxconfigd -m boot
```

b. Ensure the SEVM daemon is enabled:

```
foo# vxdctl mode
mode: enabled
```

c. Ensure that redundant root volumes exist:

```
foo# vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c2t0d0s2     sliced    rootdisk     rootdg       online
c3t0d0s2     sliced    disk01       rootdg       online
```

In addition, issue the `vxprint` command to determine the redundant state of SEVM. The output will resemble the following:

```
foo# vxprint
 Disk group: rootdg

TY  NAME         ASSOC          KSTATE       LENGTH      PLOFFS   STATE    TUTIL0  PUTIL0
dg rootdg        rootdg         -            -           -        -        -       -

dm disk01        c3t0d0s2       -            17538444    -        -        -       -
dm rootdisk      c2t0d0s2       -            17538444    -        -        -       -

v   opt          fsgen          ENABLED      4097331     -        ACTIVE   -       -
pl  opt-01       opt            ENABLED      4097331     -        ACTIVE   -       -
sd  rootdisk-04  opt-01         ENABLED      4097331     0        -        -       -
pl  opt-02       opt            ENABLED      4097331     -        ACTIVE   -       -
sd  disk01-01    opt-02         ENABLED      4097331     0        -        -       -

v   rootvol      root           ENABLED      1106028     -        ACTIVE   -       -
pl  rootvol-01   rootvol        ENABLED      1106028     -        ACTIVE   -       -
sd  rootdisk-B0  rootvol-01     ENABLED      1           0        -        -       Block0
sd  rootdisk-02  rootvol-01     ENABLED      1106027     1        -        -       -
pl  rootvol-02   rootvol        ENABLED      1106028     -        ACTIVE   -       -
sd  disk01-02    rootvol-02     ENABLED      1106028     0        -        -       -

v   swapvol      swap           ENABLED      1052163     -        ACTIVE   -       -
pl  swapvol-01   swapvol        ENABLED      1052163     -        ACTIVE   -       -
sd  rootdisk-01  swapvol-01     ENABLED      1052163     0        -        -       -
pl  swapvol-02   swapvol        ENABLED      1052163     -        ACTIVE   -       -
sd  disk01-03    swapvol-02     ENABLED      1052163     0        -        -       -

v   usr          fsgen          ENABLED      7185591     -        ACTIVE   -       -
pl  usr-01       usr            ENABLED      7185591     -        ACTIVE   -       -
sd  rootdisk-05  usr-01         ENABLED      7185591     0        -        -       -
pl  usr-02       usr            ENABLED      7185591     -        ACTIVE   -       -
sd  disk01-04    usr-02         ENABLED      7185591     0        -        -       -

v   var          fsgen          ENABLED      4097331     -        ACTIVE   -       -
pl  var-01       var            ENABLED      4097331     -        ACTIVE   -       -
sd  rootdisk-03  var-01         ENABLED      4097331     0        -        -       -
pl  var-02       var            ENABLED      4097331     -        ACTIVE   -       -
sd  disk01-05    var-02         ENABLED      4097331     0        -        -       -
```

All plex KSTATE entries should be ENABLED, and all plex STATE entries should be ACTIVE. These states are required in order for split mode to succeed. Recover any plexes that are not in these states.

CPUsets in sync

16. Use `cmsconfig` **to confirm that the CPUsets are in sync.**

```
Item Name          Fault Loc    State                    Page 1 of 2
 ------------------------------------------------------------------------
 0    A-MBD 0             A-MBD  enabled
 1    B-MBD 0             B-MBD  enabled
 2    CAF 0               A-CAF  enabled
 3    CAF 1               B-CAF  enabled
 4    CPU 0               A-CPU  enabled
 5    CPU 1               B-CPU  enabled
 6    DSK 0               A-DSK  enabled
 7    DSK 1               B-DSK  enabled
 8    HDD 0               A-DSK0 enabled
 9    HDD 6               B-DSK0 enabled
 10   PSU 0               A-PSU0 enabled
 11   PSU 1               A-PSU1 enabled
 12   PSU 3               B-PSU0 enabled
 13   PSU 4               B-PSU1 enabled
 14   RMM 0               A-RMM  enabled
 15   RMM 1               B-RMM  enabled
 16   ft_alarm 0                 usable
 17   ft_core 0                  enabled      Fault tolerant
 18   ft_network 0               online_up    A (online) & B (online)
 19   ft_serial 0                online
(H)elp, (I)nclude, (E)xclude, (S)elect, (P)age, (V)iew, (Q)uit or <Number> ?
```

The `ft_core` object should read `Fault tolerant`, indicating that the CPUsets are in sync.

Split daemon running

17. Ensure that the split daemon, `u4ftsplitd`**, is running by issuing the command:**

```
foo# ps -elf -oargs | grep u4ftsplitd
/usr/platform/SUNW,Ultra-4FT/SUNWcms/lib/u4ftsplitd
```

and noting that the output contains `u4ftsplitd`, as shown above. If it is not running, restart it with the command:

```
foo# /etc/init.d/u4ftsplit start
```

Confirm config

18. Confirm the following before proceeding:

- `/etc/hosts` contains all IP addresses (for `foo`, `foo-1`, `foo-2` and `foo-3` and the ICN networks configured in `/etc/splitd.conf`) (Step 8).
- `/etc/splitd.conf` names, `foo` and `foo-2` (Step 9) and at least one of the ICN networks, in this case `foo-i0` and `foo-2-i0` (Step 12).
- For each ICN network instance configured in `/etc/splitd.conf`, the `hostname` field of the appropriate object in `cmsconfig` is set (Step 12). For example, if ICN network instance 0 is configured, the host name for `icn 0` in `cmsconfig` should be set.
- The `ft_network 0` is configured in fault-tolerant mode (Step 14).
- The boot disks are mirrored and the mirroring process has completed (Step 15). A fully-mirrored volume has an `ACTIVE` state for the volume and plex fields in the output from the `vxprint` command.
- The CPUs are running in sync (Step 16).

Caution – Do not power cycle, or perform an operation that results in a power cycle of, either side when in split mode unless all modules owned by that side (including the motherboard) are disabled in the CMS on the other side. Failure to do this will result in undesirable failures occurring in certain modules on the other side, including the motherboards.

Issue split command

19. Issue the `split` command from either side, making side A the winner:

```
splitadm -w a split
```

Note – This will cause many warning messages from SEVM as it loses sight of its mirror. This is normal.

Start `cmsconfig`

20. Start `cmsconfig` on the split winner (side A), which should display information similar to the following:

```
 Item Name          Fault Loc    State                    Page 1 of 2
 ------------------------------------------------------------------------
 0    A-MBD 0             A-MBD  enabled
 1    B-MBD 0             B-MBD  enabled
 2    CAF 0               A-CAF  enabled
 3    CAF 1               B-CAF  disabled
 4    CPU 0               A-CPU  enabled
 5    CPU 1               B-CPU  disabled
 6    DSK 0               A-DSK  enabled
 7    DSK 1               B-DSK  disabled
 8    HDD 0               A-DSK0 enabled
 9    HDD 6               B-DSK0 disabled
 10   PSU 0               A-PSU0 enabled
 11   PSU 1               A-PSU1 enabled
 12   PSU 3               B-PSU0 enabled
 13   PSU 4               B-PSU1 enabled
 14   RMM 0               A-RMM  enabled
 15   RMM 1               B-RMM  disabled
 16   ft_alarm 0                 usable
 17   ft_core 0                  enabled      Running on A-CPU
 18   ft_network 0               online_up    A (online)
 19   ft_serial 0                online       2 (unusable)
(H)elp, (I)nclude,(E)xclude, (S)elect, (P)age, (V)iew, (Q)uit or <Number>? p

Item Name          Fault Loc    State                    Page 2 of 2
 ------------------------------------------------------------------------
 20   icn 0                      enabled
 21   icn 1                      disabled
 22   icn 2                      disabled
 23   icn 3                      disabled
 24   icn_system 0               enabled
 (H)elp, (I)nclude, (E)xclude, (S)elect, (P)age, (V)iew, (Q)uit or <Number> ?
```

Notice that all the B-side modules except B-MBD are disabled, and the split winner is running on the A-CPU. Also notice that `icn_system` is `enabled`, indicating that the ICN network has been initialized.

Bring up losing side

21. Bring up the losing side

The losing (B) side will attempt a reboot from B-DSK0, finally resulting in a prompt to enter SEVM license keys.

After successfully entering the license keys a second reboot of side B will occur.

When side B has successfully rebooted for the second time, you can log in to both sides. The winning side (A) retains the identity of `foo`, and the losing side (B) has the new identity `foo-2`.

Check system in split mode

22. Check that the system is actually in split mode by issuing the command `splitinfo` from both sides:

Side A:

```
foo# splitinfo -mad
domain = A
attributes = split,master
master = A
```

Side B:

```
foo-2# splitinfo -mad
domain = B
attributes = split
master = A
```

Start cmsconfig

23. Start `cmsconfig` **on side B (**`foo-2`**).**

The output should be similar to the following:

```
 Item Name          Fault Loc    State                    Page 1 of 2
 ------------------------------------------------------------------------
 0    A-MBD 0            A-MBD  enabled
 1    B-MBD 0            B-MBD  enabled
 2    CAF 0              A-CAF  enable_faile FRU owned by other side
 3    CAF 1              B-CAF  enabled
 4    CPU 0              A-CPU  enable_faile FRU owned by other side
 5    CPU 1              B-CPU  enabled
 6    DSK 0              A-DSK  enable_faile FRU owned by other side
 7    DSK 1              B-DSK  enabled
 8   HDD 0            A-DSK0 enable_faile Can not be enabled until DSK 0 i
 9    HDD 6              B-DSK0 enabled
 10   PSU 0              A-PSU0 enable_faile FRU owned by other side
 11   PSU 1              A-PSU1 enable_faile FRU owned by other side
 12   PSU 3              B-PSU0 enabled
 13   PSU 4              B-PSU1 enabled
 14   RMM 0              A-RMM  enable_faile FRU owned by other side
 15   RMM 1              B-RMM  enabled
 16   ft_alarm 0                usable
 17   ft_core 0                 enabled      Running on B-CPU
 18   ft_network 0              online_up    B (online)
 19   ft_serial 0               online       0 (unusable)
(H)elp, (I)nclude,(E)xclude, (S)elect, (P)age, (V)iew, (Q)uit or <Number>? p

 Item Name          Fault Loc    State                    Page 2 of 2
 ------------------------------------------------------------------------
 20   icn 0                     enabled
 21   icn 1                     disabled
 22   icn 2                     disabled
 23   icn 3                     disabled
 24   icn_system 0              enabled
(H)elp, (I)nclude, (E)xclude, (S)elect, (P)age, (V)iew, (Q)uit or <Number> ?
```

The losing side (B) modules are in an `enable_failed` state. This is correct because the split loser inherits its state from the split winner in which these modules were enabled. Hence, the split loser attempts to enable them and fails because it no longer has access to any modules on the A-side.

Ensure communication

24. To ensure that the sides are correctly communicating, issue the following commands on both sides:

```
foo# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
     inet 127.0.0.1 netmask ff000000
pnet0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
     inet 129.156.203.152 netmask ffffff00 broadcast 129.156.203.255
     ether 8:0:20:91:8f:54
icn0: flags=8843<UP,BROADCAST,RUNNING,MULTICAST,PRIVATE> mtu 40945
     inet 192.168.1.1 netmask ffffff00 broadcast 192.168.1.255
     ether 8:0:20:91:0:52
```

```
foo# u4ftctl -d /dev/icn status
(icn#0): Online + Exporting + Importing
(icn#1): Not initialised
(icn#2): Not initialised
(icn#3): Not initialised
```

Note – If either side is not exporting, use `cmsconfig` to disable and re-enable `icn0` on that side.

Check split daemon

25. Check that the split daemon is running on each side:

Side A:

```
foo# ps -efo args | grep u4ftsplitd
/usr/platform/SUNW,Ultra-4FT/SUNWcms/lib/u4ftsplitd
```

Side B:

```
foo-2# ps -efo args | grep u4ftsplitd
/usr/platform/SUNW,Ultra-4FT/SUNWcms/lib/u4ftsplitd
```

The output should contain `u4ftsplitd`, as shown above.

If the daemon is not running, start it using the `u4ftsplit` command:

Side A:

```
foo# /etc/init.d/u4ftsplit start
```

Side B:

```
foo-2# /etc/init.d/u4ftsplit start
```

The system is now split.

Merging the System

Confirm split daemons

1. Confirm the split daemons are communicating.

Before a merge can occur, the split daemons on the two sides of the machine must be communicating with each other over the chosen network (set up in `/etc/splitd.conf`), which is typically `icn0`. Confirm this by attempting to change split mastership on both sides:

```
foo# splitconf -m <side>
master = A
```

<*side*> is dependent on the existing master: if the existing master is A, <*side*> should be B; if the existing master is B, <*side*> should be A.

- If the change of mastership proceeds without producing an error and the `splitinfo` command shows <*side*> as the master on both sides, the split daemons are communicating.
- If this is not the case, stop the split daemon on each side using the command

```
foo# /etc/init.d/u4ftsplit stop
```

which should result in the split daemons stopping and then restarting. If necessary, recheck split daemon communication using the above procedure.

Issue merge command

2. Issue the `merge` command.

You can issue the `merge` command from either side. In this case, it is issued from the merge winner (side B):

```
foo-2# splitadm -w B merge
icn network shutdown (via CMS): icn.keepalive done.
domain = C
attributes = combined
master = A
```

This will force the losing side (A) to reboot (this may take some time), and the side A console displays the following:

```
icn network shutdown (via CMS): icn.keepalive done.

INIT: New run level: 6
The system is coming down.  Please wait.
System services are now being stopped.
Print services stopped.
Stopping the syslog service.
syslogd: going down on signal 15
Killed download daemon.
Oct 11 11:26:33 snmpdx: received signal 15
The system is down.

INIT: failed write of utmpx entry:"s6"

INIT: failed write of utmpx entry:"rb"
syncing file systems... done
rebooting...
Resetting ...
```

At this point the side A console may go dead because the newly-combined system has the A-CAF disabled.

Check fault tolerance

3. Wait for the CPUsets to become fault tolerant.

Check by running `cmsconfig` and noting that *ft_core* has the value `Fault tolerant`.

Bring FRUs online

4. Bring the losing side modules back online.

Issue the `cmsconfig` command on the merge winner (`foo-2`, side B):

```
Item  Name         Fault   Loc     State                Page 1 of 1
-------------------------------------------------------------------------
0     A-MBD 0              A-MBD   enabled
1     B-MBD 0              B-MBD   enabled
2     CAF 0                A-CAF   enable_faile FRU owned by other side
3     CAF 1                B-CAF   enabled
4     CPU 0                A-CPU   busy
5     CPU 1                B-CPU   enabled
6     DSK 0                A-DSK   enable_faile FRU owned by other side
7     DSK 1                B-DSK   enabled
8     HDD 0                A-DSK0  enable_faile Can not be enabled until DSK 0 i
9     HDD 7                B-DSK0  enabled
10    ft_alarm 0                   usable
11    ft_core 0                    enabled      Waiting for A-CPU to be ready
12    ft_network 0                 online_up    B (online)
13    ft_serial 0                  online       0 (unusable)
14    icn_system 0                 disabled

(H)elp, (I)nclude, (E)xclude, (S)elect, (V)iew, (Q)uit or <Number> ?
```

Disable and then re-enable each A-side module in turn.

Remirror new boot disk

5. Remirror the root disk mirror.

After the merge, you can check the state of the SEVM using the `vxdisk list` command, as follows:

```
foo-2# vxdisk list
DEVICE      TYPE      DISK       GROUP      STATUS
c2t0d0s2    sliced    -          -          online
c3t0d0s2    sliced    disk01     rootdg     online
-           -         rootdisk   rootdg     failed was:c2t0d0s2
```

The boot disk used by `foo-2` is `c3t0d0`, known to SEVM as `disk01`. The original disk used by `foo` before the upgrade is `c2t0d0`, known to SEVM as `rootdisk`, which has been failed by SEVM.

The merge process results in SEVM losing sight of `rootdisk` because the CMS tells it that it is disabled (which occurs because the merged system inherits its CMS state from the merge winner in which `rootdisk` was disabled).

Inform SEVM that `rootdisk` is still present in the Netra ft 1800. To do this, use the `vxdiskadm` command, main menu option 5 'Replace a failed or removed disk'.

The sequence will be similar to the following series of screens.

```
foo-2# vxdiskadm

Volume Manager Support Operations
Menu: VolumeManager/Disk

 1       Add or initialize one or more disks
 2       Encapsulate one or more disks
 3       Remove a disk
 4       Remove a disk for replacement
 5       Replace a failed or removed disk
 6       Mirror volumes on a disk
 7       Move volumes from a disk
 8       Enable access to (import) a disk group
 9       Remove access to (deport) a disk group
 10      Enable (online) a disk device
 11      Disable (offline) a disk device
 12      Mark a disk as a spare for a disk group
 13      Turn off the spare flag on a disk
 list    List disk information

 ?       Display help about menu
 ??      Display help about the menuing system
 q       Exit from menus

Select an operation to perform: 5
```

```
Replace a failed or removed disk
Menu: VolumeManager/Disk/ReplaceDisk

 Use this menu operation to specify a replacement disk for a disk
 that you removed with the "Remove a disk for replacement" menu
 operation, or that failed during use.  You will be prompted for
 a disk name to replace and a disk device to use as a replacement.
 You can choose an uninitialized disk, in which case the disk will
 be initialized, or you can choose a disk that you have already
 initialized using the Add or initialize a disk menu operation.

Select a removed or failed disk [<disk>,list,q,?] list
```

```
Disk group: rootdg

DM NAME       DEVICE       TYPE     PRIVLEN  PUBLEN   STATE

dm rootdisk   -            -        -        -        NODEVICE

Select a removed or failed disk [<disk>,list,q,?] rootdisk

Select disk device to initialize [<address>,list,q,?] list
```

```
DEVICE       DISK         GROUP        STATUS
c2t0d0       -            -            online
c3t0d0       disk01       rootdg       online

Select disk device to initialize [<address>,list,q,?] c2t0d0
```

```
This disk device is currently listed as in use by another host.
If you are certain that the other host is not using the disk, you
can choose to clear the use status. To use the disk the use status
must be cleared.
Output format: [Device_Name,Disk_Access_Name,Hostid]

[c2t0d0,c2t0d0s2,foo]

Clear use status? [y,n,q,?] (default: n) y
```

```
The following disk you selected for use appears to already have
been initialized for the Volume Manager.  If you are certain the
disk has already been initialized for the Volume Manager, then you
do not need to reinitialize the disk device.
Output format: [Device_Name]

c2t0d0

Reinitialize this device? [y,n,q,?] (default: y) y
Continue y/n y
```

```
Replacement of disk rootdisk in group rootdg with disk device
c2t0d0 completed successfully.

Replace another disk? [y,n,q,?] (default: n) q
```

Check Volume Manager state

6. Check the state of SEVM using the `vxdisk` list command:

```
foo-2# vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c2t0d0s2     sliced    rootdisk     rootdg       online
c3t0d0s2     sliced    disk01       rootdg       online
```

Note that there are two devices, both of which are online and mapped to two SEVM disks. At this point, SEVM knows about `c2t0d0` and has assigned `rootdisk` to it, as before. The mirroring from `disk01` to `rootdisk` can now proceed.

Check `vxrecover`

7. Check for the existence of a process `vxrecover`, which resynchronizes the root disk mirror:

```
foo-2# ps -ef | grep recover
root  9764  4250  0 17:13:10 console  0:00  grep recover
root  9173  1     0 17:12:00 ?        0:00  vxrecover -sb -g rootdg rootdisk
```

If it does not exist, issue the `vxrecover` command. The progress of this remirroring can be followed using either `vxprint`, or the X-windows graphical interface `vxva`.

If SEVM is unable to find the disk device `c2t0d0` for `rootdisk`, add the disk and initialize it by selecting option 1 ’Add or initialize one or more disks’ before you select option 5 ’Replace a failed or removed disk’.

APPENDIX **B**

Application Migration

This section describes the procedure for migrating software.

Considerations

The first issue that needs to be considered is the 'type' of the application. An application can be either 'stateless' or 'stateful'. If an application is stateless then that application does not need to recover its previous state before it can provide a service. Many applications, however, are stateful and require a recovery of their previous state before they can operate correctly. Such applications require to checkpoint and recover their state if they need to be restarted. Many of these applications already have mechanisms for doing this, for example database management systems, and such mechanisms can be used by the system administrators to correctly set up the services when such an application is upgraded. The split mode software does not provide any such facilities and it is the responsibility of the system administrator to ensure that stateful applications are started correctly following an upgrade or migration.

A second area that can impact the usefulness of split mode operation is that of disk layout and volume management. If an application is likely to be going through an migration procedure, then that application should have its own disk group, in the Volume Manager sense, so that it can be mirrored independently of other data and applications. FIGURE B-1 depicts the data layout of the initial fault tolerant system from the volume management perspective.

Note that the root disk group is placed in exactly two (mirrored) disks and that no other data is kept on these two disks. Similarly, the application software that will be migrated is kept on its own disk group, `Appdg`, and it is mirrored on disks A-DSK1 and B-DSK1. The data, which is assumed to be read-write, is also mirrored and is in the `Usersdg` disk group.

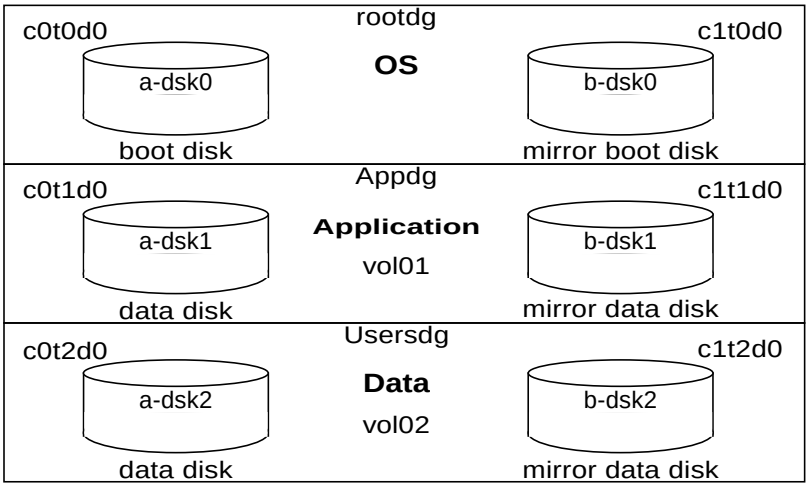


FIGURE B-1 The Volume Management Layout of the Initial Fault Tolerant System

To minimize service unavailability due to migration there will be at least two fault tolerant networks on the original fault tolerant system, where one will be on a movable resource.

The primary connection (this is the base name or the node name of the original fault tolerant system) is called `foo`. An optional secondary connection is called `foo-1` and the mandatory secondary interface is called `foo-3`.

Note that both the primary connection, `foo`, and the secondary connection `foo-1` are on the CAF, which is a fixed module in the split mode sense. Therefore, `foo-3` is used for the service that the application is providing, as this software resource is on a movable PCI module. The naming convention used here is reflected in the relevant ICN and split daemon configuration files where the name `foo-2` is reserved for the loser of a split operation.

The system at this initial stage is shown in FIGURE B-2.

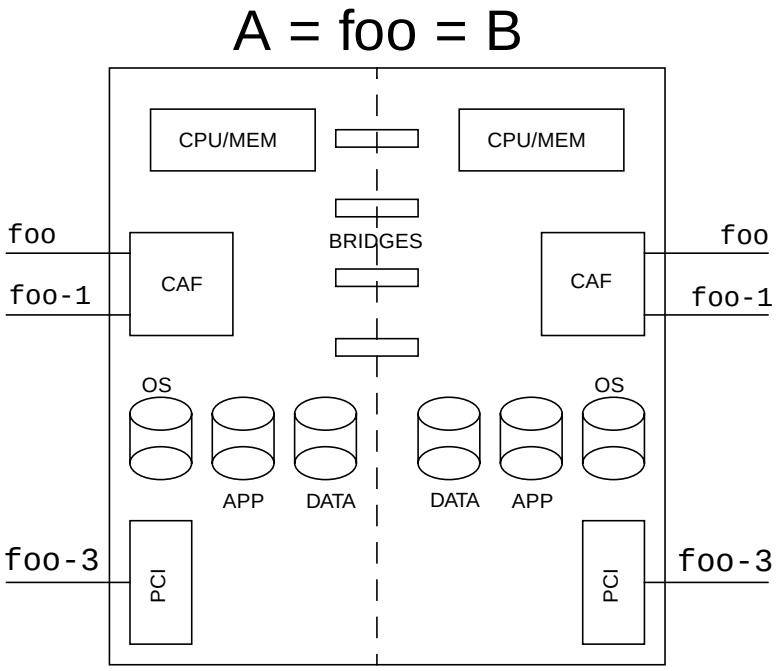


FIGURE B-2 The Initial Set Up of the Fault Tolerant System

Migrating a Stateless Application

The system can be split into two sides, where side A is chosen as the winner of the split operation. Assume that all the required steps for the split operations have been taken and that side A continues to run the service provided by the fault tolerant Netra ft 1800 system on its secondary interface `foo-3`. In order to minimize the unavailability of service, the network `foo-3` (the connection of the server to its external clients) is kept as a fault tolerant network.

After the split operation is completed, the loser of the split, side B, can be booted with its new identity. This system, called `foo-2`, can have its copy of the application software stopped by the system administrator. Note that the split daemons running on each side are communicating with each other using channel 0 of the ICN. The system at this stage of the procedure is shown in FIGURE B-3.

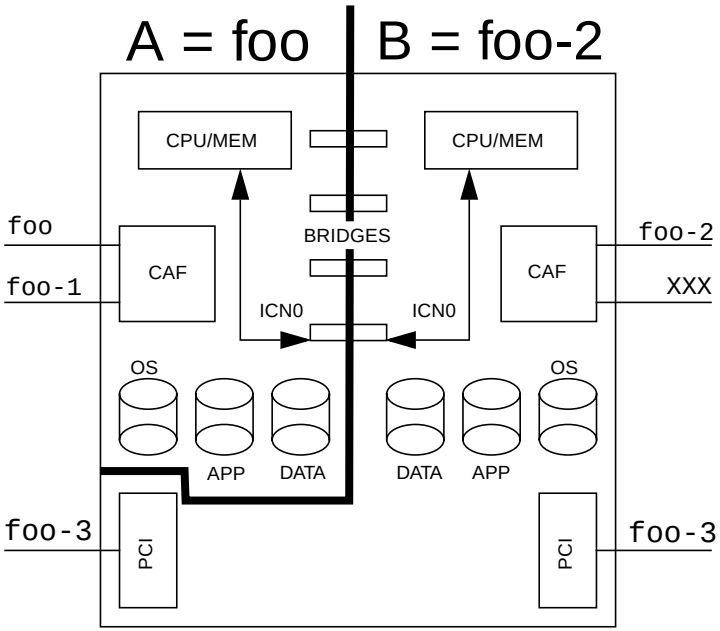


FIGURE B-3 The System in Split Mode

In order to minimize service unavailability:

1. Ownership of the back-up PCI card (in this example residing on side B) must be transferred to side A.
2. It must then be enabled using `cmsconfig`, and defined as a component of the ft network interface, `foo-3`, without enabling the `ft_network`.
3. The primary PCI card (residing on side A) and the `ft_network` service that uses it are then disabled using the `cmsconfig` utility running on that side.
4. The `ft_network` service is immediately enabled using hostname `foo-3` on side B, using the `cmsconfig` utility running on side B.

Note that in order to disable the primary PCI card, that card must have been quiesced.

At this point, the service can be restored and clients of the service will see the entire procedure as the loss of their connection to the server. Such clients will simply need to reconnect to the server, running on the same host, `foo-3`, to take advantage of the service.

The system can be now merged and fault tolerant pairs re-established. The new system will have a different identity (hostid and node name) from the original fault tolerant system, but that is not an issue if the applications, connections, and disk layout are properly thought about during the initial set up of the system.

Migrating a Stateful Application

While the above procedure is sufficient for stateless servers, or new services running on the system, the procedure for migrating stateful applications is more complex and therefore more time consuming. This is due to the need for the check pointing and recovery of state that must happen before the service can be correctly restored.

The current state must be checkpointed before the system is merged and the service is restarted. Note that this should happen after the service on the primary PCI card has been disabled. Once the system is merged the state of the server application can be recovered and the service restored. FIGURE B-4 shows the new fault tolerant system after a successful merge operation where the direction of data mirroring is shown to indicate that the data was updated on side A while the application software was stopped on side B. Further note that in this figure the fault tolerant pairs for the network connections are restored to the original configuration with the exception of the primary interface which is now called `foo-2`.

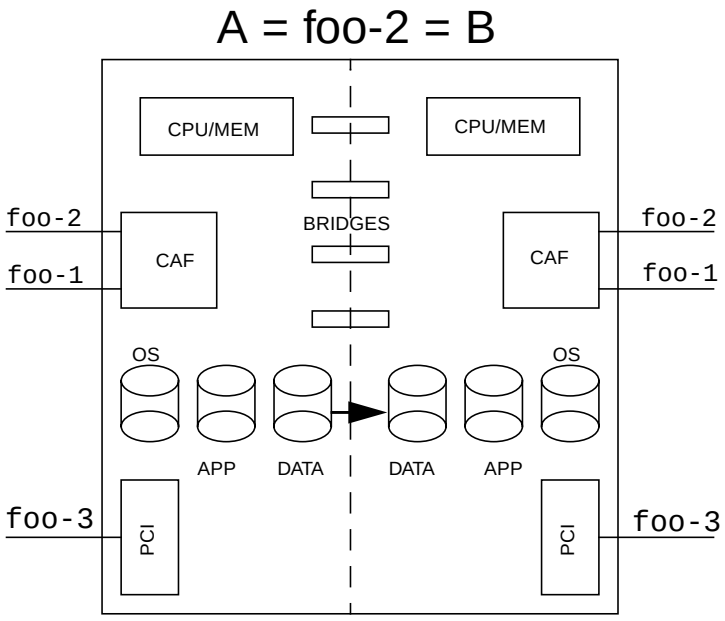


FIGURE B-4 The Final Fault Tolerant System

APPENDIX **C**

CPUset Replacement

Module Injector/Ejector Mechanisms

CPUset modules have two injector/ejector levers. The main feature is a slide which engages and disengages the CPUset's electrical connection to the motherboard, and a lever which physically engages and disengages the CPUset. When the latch is disengaged, a red dot is exposed. This facilitates the identification of unlatched injectors.

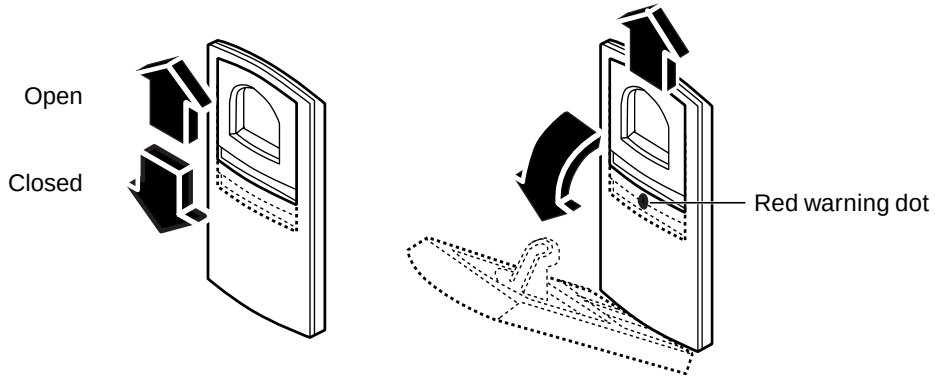


FIGURE C-1 Module Injector/ejector Lever

The CPUset is disengaged from its electrical connection when the slide is moved towards the rounded end of the lever, exposing the red warning dot.

▼ To Remove a CPUset

Disengage the injector levers

Raise/lower the injector levers

Remove the CPUset

1. **Move the slides in the levers on the CPUset to the disengaged position.**

   This will expose the red warning dots.

2. **Lower the bottom lever and raise the top lever simultaneously.**

   The CPUset will slide out a small amount when the levers are fully raised/lowered.

3. **Slide the CPUset out of its slot, using the handle.**

   As you pull out the CPUset module, the handle in the top panel pops up and must be depressed again manually in order to withdraw the module fully from the chassis (see FIGURE C-2). Once the handle is clear of the crossbar and has popped up again, it can be used to take the weight of the module.

Caution – CPUset modules are very heavy. The weight warning label on the CPUset is for guidance only. The actual weight of a CPUset depends on its configuration. Both the front and top handles must be used simultaneously once the module has been withdrawn as illustrated in FIGURE C-2.

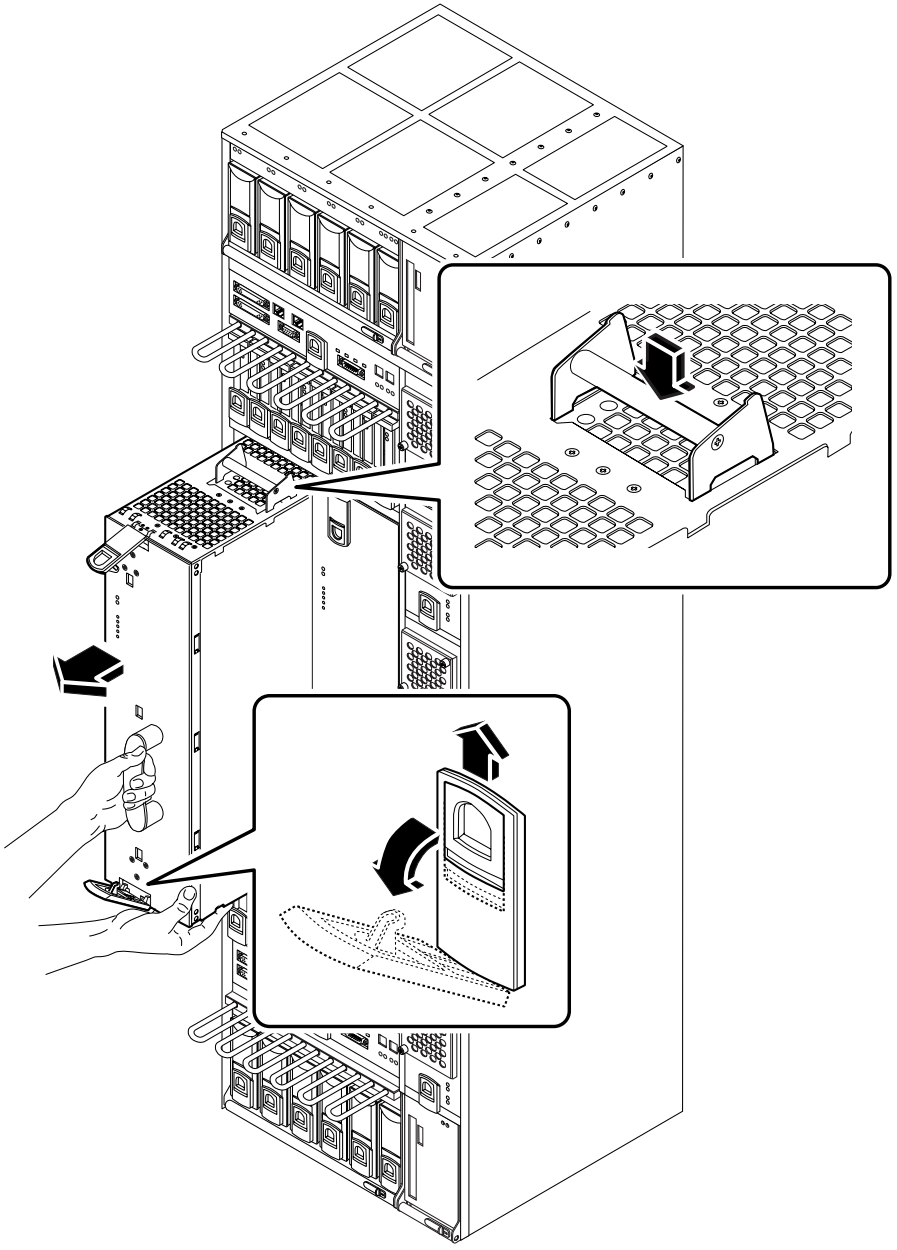


FIGURE C-2 Removing a CPUset Module

▼ To Insert a CPUset

Insert the CPUset

1. **Slide the CPUset into its slot but not fully home.**

   On inserting the CPUset module the top handle must be depressed in order to push the module fully into the chassis.

Push the CPUset fully home

2. **When the injector levers engage with the chassis, raise the bottom injector and lower the top injector simultaneously to push the CPUset fully home.**

Engage the slides

3. **Move the slides in the injector levers into the engaged position.**

▼ To Prepare the New CPUset

After inserting the new 400MHz CPUsets, it is necessary to ensure that they are set up correctly before they are integrated into the system.

Ensure that:

- There are no hardware faults
- The NVRAM variables are the same as those on the existing 300MHz CPUsets
- The OBP variables are set correctly:
  - `use-nvramrc?=true`
  - `auto-boot?=true`
  - `boot-device=a-dsk0 b-dsk0`
  - `diag-device=a-dsk0 b-dsk0`
  - `diag-switch?=true`

Go to OBP prompt

1. **Ensure the new CPUset is at the OBP prompt.**

   a. **If the `auto-boot` variable is set to `false`, this will happen automatically and you can proceed to Step 2.**

b. If the `auto-boot` variable is set to `true`, the CPUset will attempt to boot; in this case, send a break command from the console.

If the `ok` prompt appears, you can proceed to Step 2.

If the kernel debugger (kdb) prompt appears, type:

```
kdb $q
```

to go to the OBP prompt.

Set OBP variables

2. Enter the following commands:

```
ok set-defaults
ok setenv use-nvramrc?=true
ok setenv auto-boot?true
ok setenv boot-device=a-dsk0 b-dsk0
ok setenv diag-device=a-dsk0 b-dsk0
ok setenv diag-switch?=true
```

Note – Local PROM options must be re-entered before issuing the final boot command.

Boot the CPUset

3. Type:

```
ok reset-all
ok boot
```