

1 Software Architecture

Introduction

This chapter discusses the software architecture of the Auspex M2000. The M2000 utilizes Sun's Solaris 2.6 operating system on the Sparc host processor, and Auspex's proprietary M16 Functional Multiprocessing Kernel (FMK) on the Network Processor(s) and File and Storage Processor(s). This chapter focuses on the differences that Solaris introduces to the M2000 product, as compared to the Auspex NetServer product line. Auspex-specific changes to Solaris will also be highlighted.

Objectives

By the end of this lesson, you will be able to:

- ▲ Explain the role that Solaris plays in the operation of the Auspex M2000
- ▲ Understand the separation of Solaris configuration and Auspex configuration
- ▲ Refer to devices using the proper nomenclature
- ▲ Identify the present command set and map the command set from the Auspex NetServer product to the M2000
- ▲ Map configuration files used in the Auspex NetServer product to the M2000

Software Architecture Overview

The Auspex software is built on top of Solaris 2.6

Most Auspex software is located in /usr/AXbase/

Auspex code is now dynamically loaded at boot-time, and no longer must be compiled into the kernel

The system can be booted without Auspex functionality, to function as a simple Solaris 2.6 system

The Auspex separates NFS services from standard UNIX services to optimize NFS and isolate NFS service from UNIX problems

DataGuard exploits this separation of NFS and allows the host processor to be rebooted without affecting NFS service

Software Architecture Overview

Software Architecture Overview

The M2000 software is built on top of Sun Solaris 2.6. Note that the relationship between Solaris and Auspex on the M2000 differs significantly from that between SunOS and Auspex on the NetServer product line; with the M2000, most Auspex-specific files have been moved to their own directory hierarchy in the filesystem, /usr/AXbase. More importantly, Auspex code is no longer compiled directly into the kernel, but rather dynamically loaded at boot time. This avoids the need to recompile the kernel when upgrading or applying patches.

As was mentioned in the previous chapter, the Auspex hardware components are much more loosely coupled with the host processor in the M2000. This is reflected in the software architecture of the system. Solaris and Auspex-specific disks, filesystems, and network interfaces are now configured separately. It is no longer necessary for Auspex software to be loaded for the system to boot to single- or multi-user level (though you will not be able to access FSP attached storage or NP attached networks). These boot levels will be detailed below in the section on the boot process. This separation between Solaris on the host and the Auspex software on the FMP components is exploited to provide the capability to reboot the host processor without affecting NFS service. This capability is available when the optional product DataGuard is installed.

Auspex vs. Solaris Configuration

Network Configuration

The host ethernet interface (hme0) is intended for diagnostic uses only, not to serve data

hme0 is configured through standard Solaris means: hostname.hme0 specifies a hostname, which must match an IP address in /etc/hosts

NP-attached network interfaces are configured by the /usr/AXbase/etc/iftab file

Disk Configuration

The M2000 root disk is attached to the host scsi adapter, allowing the system to boot without Auspex functionality

An extra slot is available on the host processor to attach a backup root disk, which is highly recommended

The root disk is partitioned as a standard Solaris root disk, and should not contain any user data

Disks attached to the FSP must be set up as RAID arrays on the Mylex controller

Virtual partitions are set up in /usr/AXbase/etc/vpartab

Network and Disk Configuration

Auspex vs. Solaris Configuration

Network configuration

The M2000 includes a host-attached fast ethernet interface, in addition to interfaces present on the Network Processor(s). The host ethernet is intended for diagnostic tasks rather than server use, as it is not part of the FMP data path. Using this interface to serve data will result in degraded performance; in addition, if DataGuard is installed, clients who mount this interface will hang during host processor reboots, while those mounted through the Network Processor interface will not.

The host ethernet is configured through standard Solaris means — the `hostname.hme0` file specifies a hostname which matches an IP address in `/etc/hosts`, and further configuration options can be specified either as `ndd` commands in an `rc` file or in `/etc/system`.

The Auspex network interfaces on the Network Processor are configured in `/usr/AXbase/etc/iftab`, but must also have an `/etc/hosts` entry.

Disk Configuration

The M2000 utilizes a host-attached root disk, in contrast to the NetServer's use of the first disk on the first Storage Processor. This not only allows the system to boot to a usable state to diagnose problems with the storage subsystem, but also allows booting from a backup root disk without having to move disks between slots. An extra slot is provided on the host for this purpose of a backup root disk, and users cannot be encouraged strongly enough to ensure that a current backup root disk is always available.

The root (and backup root) disk is partitioned as a standard Solaris disk. It may not utilize RAID or virtual partitions, and should not be used to house any user data.

Disks attached to the FSP are set up as RAID arrays, which can also be part of virtual partitions. Raid configuration is performed through the `ax_storage` utility, and virtual partitions are defined by

Auspex vs. Solaris Configuration Filesystem Configuration

Host filesystems are set up in `/etc/vfstab`

Host filesystems are UFS filesystems, and do not support LFS features like filesystem degradation

Auspex filesystems are LFS, which is an implementation of the High Throughput Filesystem (HTFS)

LFS filesystems are specified in `/usr/AXbase/etc/lfstab`

NFS Configuration

Only LFS filesystems may be exported as NFS filesystems through the FSP

NFS exported filesystems are configured with standard Solaris mechanisms

The file `/etc/dfs/dfstab` contains share commands to export NFS filesystems at boot-time

Filesystem and NFS Configuration

/usr/AXbase/etc/vpartab and instantiated by ax_loadvpar. Both procedures will be detailed in later sections.

Filesystem Configuration

Host filesystems (/ (root), /usr, /var, et al) are specified in the standard /etc/vfstab. These are UFS filesystems, and do not support the advanced features included in LFS (such as filesystem degradation).

Auspex implements the High Throughput FileSystem (HTFS) on the FSP. These filesystems are specified in /usr/AXbase/etc/lfstab. This will also be discussed in detail in the Filesystems Module.

NFS Configuration

Only LFS filesystems may be exported as NFS filesystems through the FSP. While host filesystems may be exported via NFS, doing so defeats the purpose of the Auspex hardware and software design. Also note that at the time of this writing, there is a problem exporting host filesystems through NP-attached network interfaces (a bug is open on the problem).

NFS exported filesystems are specified through standard Solaris mechanisms, the share command and the file /etc/dfs/dfstab. The dfstab file contains share commands which are executed at boot-time, as will be shown in the Filesystems Module.

Boot Process

The M2000 uses the standard SysV boot procedure, with configuration scripts in a /sbin/rcN directory for each run level

To boot the system without Auspex functionality, specify “-auspex” as a boot argument

Boot Sequence

General Solaris boot, with detection of attached devices

The host SCI card is detected

Other nodes are detected

Auspex IPC is started to manage processors on other boards and boards load their images

Boot Process

Boot Process

The M2000 uses the standard SysV run command (rc) script framework as Solaris. Each init state has a corresponding /sbin/rcN directory with individual scripts to start (S-files) or stop (K-files) individual components of the operating system, depending on whether the system is moving to a higher or lower runlevel.

The system can be booted without Auspex functionality by specifying the -auspex flag in the boot command. This will cause the system to ignore all Auspex components and run as a simple Solaris 2.6 machine. This may be useful for troubleshooting, and is also required when upgrading the Auspex software.

Init States (Run Levels)

- ▲ **Init State 0** is the shutdown level, and moving to this state will bring the system to the ok prompt (**Note:** the ok prompt is the console level prompt; the HP> prompt is no longer used).
- ▲ **Init State 1** is the single-user state. Only console access is available, and only root and /usr filesystems are mounted.
- ▲ **Init State 2** is a multiuser state where basic network access is available, but NFS services are not enabled. It is in this state that Auspex specific commands are run to bring up Auspex hardware.
- ▲ **Init State 3** is the normal operations state, and NFS services are available.
- ▲ **Init State 6** is used to shutdown to run level 0 and reboot.

Auspex Boot Sequence

While the details of the boot procedure are beyond the scope of this module, it is important to understand the general boot sequence, especially the timing of the Auspex board initialization. This will enable you to better recognize the cause of a boot problem.

```

cpu0: SUNW,UltraSPARC-IIi (upaid 0 impl 0x12 ver 0x13 clock 270 MHz)
SunOS Release 5.6 Version Generic_105181-05 [UNIX(R) System V Release 4.0]
Copyright (c) 1983-1997, Sun Microsystems, Inc.
mem = 131072K (0x8000000)
avail mem = 125353984
Ethernet address = 8:0:20:9e:bb:2
root nexus = SPARCengine(tm)Ultra(tm) AXi (UltraSPARC-IIi 270MHz)
pci0 at root: UPA 0x1f 0x0
pci0 is /pci@1f,0
PCI-device: pci@1,1, simba #0
PCI-device: pci@1, simba #1
glm0: Rev. 4 Symbios 53c875 found.
glm0 is /pci@1f,0/pci@1/scsi@1
glm1: Rev. 4 Symbios 53c875 found.
glm1 is /pci@1f,0/pci@1/scsi@1,1
sd1 at glm0: target 1 lun 0
sd1 is /pci@1f,0/pci@1/scsi@1/sd@1,0
    <Auspex 9GB cyl 8668 alt 1 hd 16 sec 128>
sd3 at glm0: target 3 lun 0
sd3 is /pci@1f,0/pci@1/scsi@1/sd@3,0
    <Auspex 9GB cyl 8668 alt 1 hd 16 sec 128>
sd6 at glm0: target 6 lun 0
sd6 is /pci@1f,0/pci@1/scsi@1/sd@6,0
root on /pci@1f,0/pci@1/scsi@1/disk@3,0:a fstype ufs
80420 at ebus0: offset 14,300060
kb_ps20 at 80420: reg=0, name=offset="0"
kb_ps20 is /pci@1f,0/pci@1,1/ebus@1/8042@14,300060/kb_ps2@0
keyboard is </pci@1f,0/pci@1,1/ebus@1/8042@14,300060/kb_ps2@0> major <87> minor <0>
kdmouse0 at 80420: reg=1, name=offset="1"
kdmouse0 is /pci@1f,0/pci@1,1/ebus@1/8042@14,300060/kdmouse@1
mouse is </pci@1f,0/pci@1,1/ebus@1/8042@14,300060/kdmouse@1> major <88> minor <0>
stdin is </pci@1f,0/pci@1,1/ebus@1/8042@14,300060/kb_ps2@0> major <87> minor <0>
SUNW,m64B0 is /pci@1f,0/pci@1,1/ATY,3DCHARGER@4
m64#0: 1024x768, 4M mappable, rev 4755.9a
stdout is </pci@1f,0/pci@1,1/ATY,3DCHARGER@4> major <110> minor <0>
se0 at ebus0: offset 14,400000
se0 is /pci@1f,0/pci@1,1/ebus@1/se@14,400000
su_pnp0 at ebus0: offset 14,3803f8
su_pnp0 is /pci@1f,0/pci@1,1/ebus@1/su_pnp@14,3803f8
su_pnp1 at ebus0: offset 14,3602f8su_pnp1 is /pci@1f,0/pci@1,1/ebus@1/su_pnp@14,3602f8
SUNW,hme0: CheerIO 2.0 (Rev Id = c1) Found
SUNW,hme0 is /pci@1f,0/pci@1,1/network@1,1
dump on /dev/dsk/c0t3d0s4 size 204776K
SUNW,hme0: Using Internal Transceiver
SUNW,hme0: 10 Mbps half-duplex Link Up
NOTICE: Link Controller LC2 found
NOTICE: Rev C PSB found

WARNING: sci address 60660000 60668000 61668000
NOTICE: Host is the SCRUBBER node

WARNING: sci: high level handler required.

```

Host SCI card detection

Sample Boot Messages

The boot sequence proceeds as follows:

- ▲ General Solaris boot, with detection of attached devices
- ▲ The host SCI card is detected
- ▲ Other nodes are detected
- ▲ Auspex IPC is started to manage processors on other boards and boards load their images

```

WARNING: sci: high level handler required.
pc11c8,6581 is /pci@1f,0/pci@1/pc11c8,658@2
NOTICE: Number of nodes is 1
NOTICE: Performing SCI configuration
ASSIGNED NEW NODE ID - OLD = ff0a, NEW = ffea
ASSIGNED NEW NODE ID - OLD = ffea, NEW = ff0a
Peer Node Fifo Address is e2000000
pseudo-device: axipc0axipc0 is /pseudo/axipc@0
pseudo-device: axipc1
axipc1 is /pseudo/axipc@1
[repeats for pseudo-devices axipc2-30]
pseudo-device: axipc31
axipc31 is /pseudo/axipc@31
WARNING: axipc: Panic state, dumping Auspex board images

```

```

    dump req done for 2
    dump req done for 3
NOTICE: Waiting 90 seconds for 1 node(s) to reset
.....
.....
set node 2 state to AUS_HW_READY
set node 3 state to AUS_HW_READY
sci_soft_reset: finished scanning nodes after reset.
    have instance 1 of DE 21140.
    have instance 2 of DE 21140.
    have instance 3 of DE 21140.
    have instance 4 of DE 21140.

```

```

pseudo-device: anp0
anp0 is /pseudo/anp@0
    set AXIPC_ONLINE
pseudo-device: afe0
afe0 is /pseudo/afe@0
pseudo-device: afel
afel is /pseudo/afe@1
pseudo-device: afe2
afe2 is /pseudo/afe@2
pseudo-device: afe3
afe3 is /pseudo/afe@3
pseudo-device: pm0
pm0 is /pseudo/pm@0
pseudo-device: tod0
tod0 is /pseudo/tod@0
pseudo-device: vol0
vol0 is /pseudo/vol@0FSP_0[473]:
High Throughput File System, Release 3.1.0m
Copyright (c) 1992-1998 Programmed Logic Corporation
Built
Mon Nov 16 02:42:15 98
on      build-nuptse
from    /export/scm/build/2.0A12+inprogress/solaris/src/ml6/fsp/fp/plc/fs/htfs
Static File System
Copyright (c) 1997 Programmed Logic Corporation
Available memory = 205127680

```

} Auspex board dump
from prior reboot

}
} Sync up with other boards

Sample Boot Messages (continued)

Empty rectangular box for student notes.

Student Notes

```

Auspex SP disk
Auspex SP VP
Auspex SP RAID
Auspex SP Tape
Auspex SP SnapShot
Auspex SP Auto
Auspex SP MDAC RAID
Auspex SP MDAC Tape
Auspex SP MDAC Auto
Auspex NVRAM
NVRAM: No AFX8000 found or failed to init.
NVRAM: logging disabled.
fsp0m0c0t0: QUANTUM QM318200 [ Disk ] [ Wide_16 ] [ Tags ]
fsp0m0c0t1: QUANTUM QM318200 [ Disk ] [ Wide_16 ] [ Tags ]
fsp0m0c0t2: QUANTUM QM318200 [ Disk ] [ Wide_16 ] [ Tags ]
fsp0m0rd0: [ 17366 MB ] [ RAID1 ] [ Online ] [ dev: 0x4180000 ]
fsp0m0rd1: [ 17366 MB ] [ RAID7 ] [ Online ] [ dev: 0x4180010 ]

```

Detected FSP-attached disks
 Configured RAID arrays

Sample Boot Messages (continued)

Student Notes

DataGuard

DataGuard allows the host processor to be rebooted without affecting standard NFS traffic

DataGuard host reboots may be user-initiated or automatic (vmunix panics and other boards resetting a unresponsive host processor)

DataGuard commands are hpreboot, hphalt, and hpshutdown, which are functional equivalents to their namesakes, except they only affect the host processor

No new mounts are possible while the host processor is rebooting

Locking services are unavailabe during host reboots

DataGuard

DataGuard

DataGuard is an optional product which takes advantage of the FMP structure to allow host processor reboots without affecting normal NFS traffic. These reboots can be automatic or user-initiated. User-initiated host reboots may be required when installing host-attached storage or when some problem has arisen in UNIX which can only be resolved by rebooting. Automatic reboots occur when a UNIX kernel panic is encountered, or when another board in the system becomes aware that the host is no longer responding to probe messages.

Installing and Enabling DataGuard

DataGuard is installed with the standard Sun pkgadd command, and is called AXdguard.

User-initiated Host Reboots

Once DataGuard is installed, three additional commands will be available in /usr/sbin: **hpreboot**, **hphalt**, and **hpshutdown**. These commands are functionally equivalent to the standard Solaris reboot, halt, and shutdown commands, but only effect the host processor. The other boards in the Auspex will continue to function in order to serve NFS data. Services which require the host processor will no longer be accessible, however.

Caveats

While standard NFS traffic will continue to be served while the host processor is down or rebooting, some functionality will be unavailable. Most importantly, no new mounts are possible while the host processor is down. The **mountd** process which answers mount requests is a host processor utility. Locking service is also unavailable during a host processor reboot.

Device Nomenclature

Solaris creates a more hierarchical /dev directory tree

Virtual partitions are located in /dev/{r}axvp

Raid devices are located in /dev/{r}axmrd

It is no longer possible to access single disks (adN) directly

Disk slices are referred to as fspFmMcCtTsS (where F is the FSP, M is the Mylex card, C is the controller on the Mylex card, T is the target (disk), and S is the slice number)

Raid arrays are referred to as fspFmMrdR (where R is the RAID device number, specified when the RAID array is created); Raid arrays also have slices (fspFmMrdRsS)

Virtual partitions are referred to as fspFvpV (where V is the virtual partition number)

Device Nomenclature

Device Nomenclature

Solaris brings a new structure to the /dev directory, which is more hierarchical in nature than that of SunOS. /dev now contains numerous subdirectories which reduce the clutter created by having an entry for each slice of each disk on each controller, multiple names to specify access methods for each tape device, etc. We can no longer access disks as adN, but rather must specify by FSP, Mylex card, controller, and target. In addition, we now deal with slices 0-16 rather than partitions a-h.

New /dev structure

Virtual partitions and RAID devices each have two subdirectories under /dev, one for character devices and one for block devices. The virtual partition special devices are found in /dev/axvp/ (block devices) and /dev/raxvp/ (character devices), while the RAID devices are found in /dev/axmrd and /dev/raxmrd. All access to FSP-attached disks must go through these interfaces — direct access to disks (adN) is no longer supported.

Storage nomenclature for Auspex commands

In the Auspex storage management commands, which will be detailed later, disks are referred to as: fspFmMcCtTsS, where F is the FSP number, M is the Mylex card, C is the controller, T is the target, and S is the slice (formerly known as partition). Note that slices are now numbers rather than letters, and 16 slices are supported by the M2000, 0-15. Slice 2 refers to the entire device. Since RAID arrays exist on the Mylex level, they are referred to as fspFmMrdR, where R is the RAID array number which will be assigned when the array is created. RAID arrays also contain slices. Virtual partitions exist on the FSP level, and are thus referred to as fspFvpV, where V is the virtual partition instance number, assigned in the virtual partition table, vpartab.

Auspex File and Command Equivalencies Overview

Network configuration is no longer restricted to /etc/rc.boot, but is instead handled in disparate files such as /etc/netmasks and /usr/AXbase/etc/iftab.

/etc/vpartab has moved to /usr/AXbase/etc/vpartab with nomenclature changes, but virtual partition commands (ax_loadvpar and ax_vpstat) remain the same.

/etc/raidtab no longer exists and ax_raid and ax_diskconf functions have been merged into ax_storage

Disk labeling utilities (ax_label, ax_lslabel) remain the same, with new syntax

/etc/exports functionality is handled by /etc/dfs/dfstab and Solaris share replaces SunOS exportfs

Auspex File and Command Equivalencies Overview

Auspex File and Command Equivalencies Overview

Below are highlighted some of the major equivalencies (and inequalities) between the existing NetServer product and the M2000. For further information, please refer to the “Comparative List of Sunos Commands...” and “Comparative List of Commands...” which have been provided.

- ▲ Network configuration is no longer restricted to `/etc/rc.boot`, but is instead handled in disparate files such as `/etc/netmasks` and `/usr/AXbase/etc/iftab`.
- ▲ `/etc/vpartab` has moved to `/usr/AXbase/etc/vpartab` with nomenclature changes, but virtual partition commands (`ax_loadvpar` and `ax_vpstat`) remain the same.
- ▲ `/etc/raidtab` no longer exists and `ax_raid` and `ax_diskconf` functions have been merged into `ax_storage`
- ▲ Disk labeling utilities (`ax_label`, `ax_lslabel`) remain the same, with new syntax
- ▲ `/etc/exports` functionality is handled by `/etc/dfs/dfstab` and Solaris `share` replaces SunOS `exportfs`

Student notes