Chapter 4

Monitoring the System

A computer system consists of processors, memory, and I/O devices. Users log in to the system and run business jobs or applications. Network software and interface cards connect users over the network to the computer system. This chapter focuses on monitoring the processors and memory components, as well as the user activity on the system. Disks, networks, databases, and applications are covered in separate chapters.

Identifying Important System Monitoring Categories

This chapter identifies important system resources to monitor, so that you can detect faults, avoid problems, and ultimately ensure availability. Many important system resources can be monitored for events and faults, and many system management tools are available with which to monitor them. Instead of categorizing based on specific hardware components, this chapter relates its descriptions of tools to the different ways of monitoring your system. For example, your focus as an operator may be on watching for system faults and failures, software or hardware configuration changes, system resource usage, performance management, or security. This chapter tries to show a tool's role, if any, in each of these monitoring categories.

Monitoring System Configuration Changes

This category includes monitoring for changes in hardware and software configurations that can be caused by an operating system upgrade, patches applied to the system, changes to kernel parameters, or the installation of a new software application, for example. The root cause of system problems can often be traced back to an inappropriate hardware or software configuration change. Therefore, it is important to keep accurate records of these changes, because the problem that a change causes may remain latent for a long period before it surfaces.

Adding or removing hardware devices typically requires the system to be restarted, so configuration changes can be tracked indirectly (in other words, remote monitoring tools would notice system status changes). However, software configuration changes, or the installation of a

new application, are not tracked in this way, so reporting tools are needed. Also, more systems are becoming capable of adding hardware components online, so hardware configuration tracking is becoming increasingly more important.

Monitoring System Faults

After ensuring that the configuration is correct, the first thing to monitor is the overall condition of the system. Is the system up? Can you talk to it, ping it, run a command? If not, a fault may have occurred. Detecting system problems ranges from determining whether the system is up to determining whether it is behaving properly. If the system either isn't up or is up but not behaving properly, then you must determine which system component is having a problem.

This chapter addresses monitoring various components of the system for faults or events. The fault category generally covers system hardware components, including the Central Processing Unit (CPU), memory, and system buses, as well as peripherals, such as tape drives and printers. (Disks are covered in Chapter 5.) CPU and memory faults may cause system failures or degraded performance. Tape faults may result in a backup failing, a bad backup, or delays in completing a backup in a timely manner. With proactive monitoring, you can find out that a tape drive is having problems before backup is actually scheduled to begin.

Monitoring System Resource Utilization

For an application to run correctly, it may need a fixed amount of system resources. Some resources are renewable, such as the amount of CPU or I/O bandwidth an application is entitled to use during a time interval. The resource category refers to those system resources that an application acquires and then releases at its own discretion. For example, an application can allocate a segment of shared memory or launch a group of processes. Other examples included in the resource category are the number of open files or sockets, message segments, and system semaphores that an application has. The system has fixed limits for each of these resources, so monitoring their use is important. If these system tables are exhausted, the system may no longer function properly. You may want to set up alarms to notify you when the available resources in a given system table are below a certain threshold, which will give you time to react before the problem becomes critical.

Another aspect of resource utilization is studying the amount of resources that an application has used. You may not want a given workload to use more than a certain amount of CPU time or fixed amount of disk space. Some resource management tools, such as quota, can help with this.

Monitoring System Security

One way that a system's availability can be impacted is through unauthorized use. Performance and resource controls are not useful if the system is used for the wrong purposes. You need to prevent unauthorized use of system resources by using password files, network firewalls, and so forth. In addition to setting up access rights and policies, you need to monitor the system, so that

you know when security has been compromised. This chapter briefly mentions some of the security tools that are available.

Monitoring System Performance

Knowing both that system resources are available and that your application is performing well is important. Eliminating bottlenecks or, even better, preventing them, allows the system to provide its intended services. Monitoring the performance of system resources can help to indicate problems with the operation of the system. Bottlenecks in one area usually impact system performance in another area. CPU, memory, and disk I/O bandwidth are the important resources to watch for performance bottlenecks. You should monitor during typical usage periods to establish baselines. Understanding what is "normal" operation helps you to identify when system resources are not behaving well. Resource management tools are available that can help you to allocate system resources among applications and users.

In this and each of the next four chapters, performance issues are contained in a separate section and described after the other tools are discussed.

One way to check for system problems is to watch the system's front panel of lights. Any change from normal (for example, color changes from green to red or a light starts flashing) could be indicative of a hardware or firmware problem. Of course, to monitor the system in this way, you need an operator to watch the front panel of the system manually. If an operator isn't always available to watch the front panel, a delay in detecting a problem may occur. Many other, more sophisticated tools are available to help you detect system problems that may occur in your system. These tools, which are covered in the following sections, range from standard UNIX commands to sophisticated add-on monitoring software suites.

Using Standard Commands and Tools

Many UNIX commands exist to check configuration, status, and resource information. These tools generally report on only a snapshot in time. You can write or use custom scripts that incorporate these or other commands and run them periodically so that you can track configuration changes or test the status of system resources over time.

The more commonly used commands are described in this section. Note that they are organized alphabetically. You may also want to check the online man pages for additional information about each command. Unless otherwise noted, the commands listed in this section are available on multiple UNIX platforms. (Tools that are specific to networking, such as netstat and nfsstat, are discussed in Chapter 6.)

In addition to these commands, you may want to check the system log file, /var/adm/syslog /syslog.log, for error messages if your system is experiencing problems. Messages written to this log file include information regarding the module experiencing the problem and the time that the event occurred, which can be very valuable when troubleshooting.

bdf and df

The bdf and df commands are commonly used to show the amount of disk and swap space used and available. bdf – i reports the number of used and free filesystem structures (inodes) in the kernel.

By default, bdf shows information for all mounted filesystems. If this information is too lengthy, you can also run the command and specify a filesystem as a command-line option. An example is shown in Listing 4-1.

ioscan

The ioscan command is used to discover and display the system hardware, usable I/O system devices, or kernel I/O system data structures. The results displayed list the default hardware path to the device, the class of hardware, and a brief description. ioscan includes information on the following hardware: processors, memory, network interface cards, and I/O devices. Listing 4-2 shows how you can check the number of processors on your system by using ioscan.

ioscan is a good tool to use to get a complete picture of your system hardware layout. It reports the status of the installed software, indicating whether the proper drivers are loaded. By storing the command output in files, you can maintain a history of the hardware configuration changes to your system.

iostat

iostat reports CPU statistics and I/O statistics for disks and terminals. For disks, it lists the device name, number of bytes transferred per second (bps), number of seeks per second (sps), and milliseconds per average seek (msps). For terminals, it shows the number of characters read and the number of characters written. For the CPU, it shows the percentages of time that the system has spent in user mode, nice mode (low-priority user processes), and system mode. Listing 4-3 shows sample output for a system with only one physical disk.

Listing 4-1 bdf output for a specific filesystem.

```
# bdf /dev/vg00/lvol3
Filesystem          kbytes    used    avail   %used Mounted on
/dev/vg00/lvol3     126976    33003   93912    26%  /
#
```

Listing 4-2 Output from ioscan for a two-processor HP-UX system.

```
# ioscan |grep processor
32          processor              Processor
34          processor              Processor
#
```

Listing 4-3 Output from the iostat command showing performance measures for disks, terminals, and CPU.

```
# iostat -t
                  tty             cpu
                tin tout        us  ni  sy  id
                  0    1         1   0   2  97

  device    bps     sps    msps

  c0t5d0      0     0.0     1.0
```

You may want to use iostat to compare the activity on different disks, to see whether a load imbalance exists. It is normal for the system disk to have more activity.

ipcs

The ipcs command shows the status of active message queues, shared memory, and system semaphores. Listing 4-4 shows example output from using ipcs. You may want to consult the online manpage to see all the available options for this command.

mailstats

If your system is being used as a mail server, you may want to use mailstats to check mail statistics. The mailstats command shows the number of messages and amount of data sent or received for each mailer running on the system.

ps

The ps command is used to display information about all processes on the system. The metrics provided by ps include: Process Identifier (PID), parent PID, process start time, cumulative execution time, process state, priority, physical size (in pages), and the command with its command-line options.

ps is a quick way to get a profile of the processes on your system. It is useful for checking whether a specific application or process is running. For example, Listing 4-5 shows an easy way to display the Network File System (NFS) daemons running on your system. This listing can be used to identify runaway processes, both in CPU time and size. Numerous processes in the wait state may be an indication of a system bottleneck.

sar

sar is the System Activity Reporter. It is useful for monitoring system activity and can be used to identify memory, CPU, and kernel bottlenecks. It enables you to specify the polling interval and has

Listing 4-4 Output from ipcs showing active message queues, shared memory, and semaphores.

```
#ipcs
IPC status from /dev/kmem as of Sun Mar 14 17:47:20 1999
T      ID     KEY        MODE        OWNER     GROUP
Message Queues:
q       0 0x3c1c0330 -Rrw--w--w-      root      root
q       1 0x3e1c0330 --rw-r--r--       root      root
Shared Memory:
m       0 0x2f180002 --rw-------        root       sys
m     201 0x411c031b --rw-rw-rw-     root       sys
m     402 0x4e0c0002 --rw-rw-rw-     root       sys
m     403 0x41201219 --rw-rw-rw-     root       sys
Semaphores:
s       0 0x2f180002 --ra-ra-ra-     root       sys
s      65 0x411c031b --ra-ra-ra-     root       sys
s     130 0x4e0c0002 --ra-ra-ra-     root       sys
s     131 0x4120121a --ra-ra-ra-     root       sys
s       4 0x00446f6e --ra-r--r--       root      root
s       5 0x00446f6d --ra-r--r--       root      root
s       6 0x01090522 --ra-r--r--       root      root
s       7 0x411c1f3a --ra-ra-ra-     root      root
s       8 0x410c319a --ra-ra-ra-     root      root
#
```

Listing 4-5 Finding your NFS daemons.

```
#ps -ef |grep -E 'nfs|PPID'
     UID   PID  PPID  C  STIME   TTY       TIME COMMAND

    root   681     1  0  Dec 22  ?         0:00 /usr/sbin/nfsd 4
    root   682   681  0  Dec 22  ?         0:00 /usr/sbin/nfsd 4
    root   686   681  0  Dec 22  ?         0:00 /usr/sbin/nfsd 4
    root   688   681  0  Dec 22  ?         0:00 /usr/sbin/nfsd 4
    root 16761 16718  1 12:14:48 pts/0     0:00 grep nfs
#
```

the ability to log data to a file (in binary format). It can report on activity from many system resources, including CPU utilization by processor, buffer cache, swapping, disks and tape, run and swap queues, and several system tables. Refer to the online man page for the command-line options.

For CPU activity, sar shows CPU utilization by user mode, system mode, idle time waiting for I/O to complete, and idle time either on a per-processor level or averaged for all processors. Sample output is shown in Listing 4-6.

Listing 4-6 sar output showing system activity.

```
# sar 5 5

HP-UX cadbury B.10.20 A 9000/871    03/15/99

20:36:32    %usr    %sys    %wio   %idle
20:36:37       0       1       0      98
20:36:42       0       1       0      99
20:36:47       1       1       0      99
20:36:52       0       1       0      99
20:36:57       0       1       0      99

Average        0       1       0      99
#
```

By using sar -q, you can look at the average lengths of the run and swap queues, and the percentage of times the queues were occupied. This is shown in Listing 4-7. High CPU utilization and a large run queue may indicate a CPU bottleneck. A large swap queue is one sign of memory contention.

sar can be used to check the effectiveness of buffer cache use. It reports the rates of reads and writes between a disk and the buffer cache. It also reports the rates of logical reads and writes to and from the buffer cache, as well as buffer cache hit ratios.

For swapping activity, you can monitor swap-in rates, swap outs per second, and context switch rates.

sar -v reports the current size, maximum size, and number of overflows of various system tables, including the process table, inode table, and system file table.

Listing 4-7 Output from sar showing queue lengths.

```
# sar -q 5 5

HP-UX cadbury B.10.20 A 9000/871    03/15/99

20:44:03 runq-sz %runocc swpq-sz %swpocc
20:44:08     1.0      20     0.0       0
20:44:13     0.0       0     0.0       0
20:44:18     0.0       0     0.0       0
20:44:23     1.0      10     0.0       0
20:44:28     0.0       0     0.0       0

Average      1.0       6     0.0       0
#
```

swapinfo

swapinfo reports system paging or swapping activity, and memory utilization. On some implementations of UNIX, it is called swap. This command is useful for showing swap space usage and configuration. It displays for each swap type and device the kilobytes (K) available, kilobytes used, kilobytes free, and percentage used. If you have insufficient memory, you may see lots of pages being swapped or high utilization of the swap device. An example using swapinfo is shown in Listing 4-8.

For device swap areas, reserve is the number of 1K blocks reserved for filesystem use by ordinary users. For device swap areas, this value is always "-". Checking swapinfo periodically may help you to schedule additions to your swap capacity.

sysdef

The sysdef command, available on HP-UX, reports on a system's tunable kernel parameters. For each kernel parameter, this command shows the current value, value at boot time, and minimum and maximum values allowed for the parameter, as demonstrated in Listing 4-9. This command can be used both to monitor whether the system kernel is configured properly and to track whether certain kernel resource usage is at or approaching its configured limit. You can also use this command, together with ioscan, to track kernel configuration changes.

Listing 4-8 Output from the swapinfo command shows system paging activity.

```
# swapinfo
         Kb      Kb      Kb   PCT  START/      Kb
TYPE  AVAIL    USED    FREE  USED   LIMIT RESERVE  PRI NAME
dev  524288  12488 511800    2%       0       -    1 /dev /vg00/lvol2
reserve       -  246876 -246876
memory   404396  207844  196552   51% v
```

Listing 4-9 Showing current values of kernel-tunable parameters.

```
#sysdef
NAME                    VALUE     BOOT     MIN-MAX    UNITS     FLAGS
acctresume                  4        -    -100-100                   -
acctsuspend                 2        -    -100-100                   -
allocate_fs_swapmap         0        -           -                   -
bufpages                10714        -          0-       Pages       -
create_fastlinks            0        -           -                   -
dbc_max_pct                50        -           -                   -
dbc_min_pct                 5        -           -                   -
default_disk_ir             0        -           -                   -
```

continued

Listing 4-9 (continued)

```
dskless_node              0        -       0-1                    -
eisa_io_estimate        768        -         -                    -
eqmemsize                15        -         -                    -
file_pad                 10        -       0-                     -
fs_async                  0        -       0-1                    -
hpux_aes_override         0        -         -                    -
maxdsiz               16384        -     256-655360 Pages         -
maxfiles                120        -      30-2048                 -
maxfiles_lim           1024        -      30-2048                 -
maxssiz                2048        -     256-655360 Pages         -
maxswapchunks           256        -       1-16384                -
maxtsiz               16384        -     256-655360 Pages         -
maxuprc                  75        -       3-                     -
maxvgs                   10        -         -                    -
msgmap              2555904        -       3-                     -
nbuf                   5772        -       0-                     -
ncallout                316        -       6-                     -
ncdnode                 150        -         -                    -
ndilbuffers              30        -       1-                     -
netisr_priority          -1        -      -1-127                  -
netmemmax          14356480        -         -                    -
nfile                  1034        -      14-                     -
nflocks                 200        -       2-                     -
ninode                  500        -      14-                     -
no_lvm_disks              0        -         -                    -
nproc                   300        -      10-                     -
npty                     60        -       1-                     -
nstrpty                  60        -         -                    -
nswapdev                 10        -       1-25                   -
nswapfs                  10        -       1-25                   -
public_shlibs             1        -         -                    -
remote_nfs_swap           0        -         -                    -
rtsched_numpri           32        -         -                    -
sema                      0        -       0-1                    -
semmap              4128768        -       4-                     -
shmem                     0        -         0-1                  -
shmmni                  200        -      3-1024                  -
streampipes               0        -           0-                 -
swapmem_on                1        -           -                  -
swchunk                2048        -  2048-16384   kBytes          -
timeslice                10        --1-2147483648   Ticks          -
unlockable_mem         2158        -           0-    Pages        -
```

timex

The timex command can be used to measure and report, in seconds, the elapsed time, user CPU time, and system CPU time spent executing a given command. The command to be executed is given on the timex command line. This command reports process accounting data for the command and all of its children, as well as the total system activity during execution of the command. The timex command can give you a crude idea of the impact of a command on the rest of the system.

top

The top command is useful for monitoring the system CPU and memory loads. It also lists the most active processes on the system. top output is displayed in the terminal window and is updated every five seconds, by default.

top shows CPU resource statistics, including load averages (job queues over the last 1 minute, 5 minutes, and 15 minutes), the number of processes in each state (sleeping, waiting, running, starting, zombie, stopped), the percentage of time spent in each processor state (user, nice, system, idle, interrupt, and swapper) per processor on the system, as well as the average for each processor in a multiprocessor system.

For memory utilization, top shows virtual and real memory in use, the amount of active memory, and the amount of free memory.

At the process level, top lists the top processes, based on their CPU usage. The process data displayed by top includes the PID, process size (text, data, and stack), resident size of the process (K), process state (sleeping, waiting, running, idle, zombie, or stopped), the number of CPU seconds consumed by the process, and the average CPU utilization of the process. This command can be used to identify processes that may be using large amounts of CPU or memory. Note that top can also be a quick way to check the number of processors on your system. Listing 4-10 shows the output for a four-processor system.

Listing 4-10 Output from the top command showing process activity.

```
System: gsyview1                Fri Feb 12 13:40:24 1999
Load averages: 0.08, 0.11, 0.16
616 processes: 614 sleeping, 2 running
Cpu states:
CPU LOAD USER  NICE  SYS   IDLE  BLOCK SWAIT  INTR  SSYS
 0  0.30 0.0%  0.0%  1.3% 98.7%  0.0%   0.0%  0.0%  0.0%
 1  0.00 0.0%  0.0%  0.7% 99.3%  0.0%   0.0%  0.0%  0.0%
 2  0.01 0.0%  0.0%  0.2% 99.8%  0.0%   0.0%  0.0%  0.0%
 3  0.02 0.4%  0.0%  7.9% 91.8%  0.0%   0.0%  0.0%  0.0%
—- —— ——  ——  —— ——-  ——   ——  ——  ——
avg 0.08 0.0%  0.0%  2.6% 97.4%  0.0%   0.0%  0.0%  0.0%
```

```
Memory: 25754K (2356K)real, 27864K (6144K)virtual, 27838K free  Page# 1/42

CPU TTY    PID USERNAME PRI NI   SIZE    RES STATE    TIME %WCPU %CPU COMMAND
 3 pts/4 12555 jsymons  187 20 25992K   568K run      0:02  7.84  5.48 top
 0 rroot    19 root     100 20     0K     0K sleep 1449:04  1.05  1.05 netisr
 1 rroot   494 root     154 20   216K   284K sleep 1479:50  1.03  1.02 syncer
 0 rroot     3 root     128 20     0K     0K sleep  960:56  1.00  0.99 statdaemo
 0 rroot  1432 root      20 20  8120K  6956K sleep  842:38  0.61  0.61 cmcld
 3 rroot    38 root     138 20     0K     0K sleep  336:22  0.32  0.31 vx_iflush
 1 rroot     7 root     -32 20     0K     0K sleep  321:07  0.25  0.25 ttisr
 3 rroot   934 root     154 20  6100K  1436K sleep  297:15  0.22  0.22 rpcd
 1 rroot    40 root     138 20     0K     0K sleep  193:38  0.16  0.16 vx_inacti
 0 rroot 26626 root     154 20   868K   880K sleep  245:15  0.15  0.15 opcle
 2 rroot 26587 root     154 20  2580K  1348K sleep  125:22  0.07  0.07 opcmsga
 1 rroot    39 root     138 20     0K     0K sleep   88:15  0.07  0.07 vx_ifree_
 3 rroot    22 root     100 20     0K     0K sleep  159:58  0.06  0.06 netisr
 1 rroot 26586 root     154 20  8468K  1752K sleep   53:47  0.06  0.06 opcctla
```

uname

The uname command can be used to display configuration information about your system. This information includes the operating system name, machine model, and operating system version.

You may want to gather this information and store it for later use. This may be useful if you are trying to keep all of your systems on the same release of the operating system, for example.

uptime

The uptime command is probably the most commonly used command to check system resources. This command shows the current time, length of time the system has been up, number of users logged on, and the average number of jobs in the run queue for the last 1, 5, and 15 minutes.

Using uptime with the -w option shows a summary of the current activity on the system for each user. As shown in Listing 4-11, you can see the login time, CPU usage, and command activity for each user.

vmstat

The vmstat command provides good information about system resources, including virtual memory and CPU usage, and is useful for detecting whether you are low on memory or swap space.

Listing 4-11 Output from the uptime command showing paging activity.

```
uptime -w
 12:49pm  up 3 days, 2:19, 5 users, load average: 0.49, 0.56, 0.56
User     tty           login@  idle  JCPU PCPU  what
jsymons  console      12:32pm 74:17             /usr/sbin
jsymons  ttyp7        12:18pm                   uptime -w
```

For monitoring real and virtual memory, vmstat shows page faults and paging activity, including reclaimed pages and swapping rates.

For the CPU, you can see more detailed information with vmstat than that provided by iostat. vmstat shows faults, including device interrupts, system calls, and context switches. vmstat also includes the breakdown of CPU utilization by user, system, and idle time.

For processes, vmstat shows the number of processes in various states, including the following: currently in the run queue, blocked on an I/O operation, and swapped out to disk.

The statistics that you see vary depending on the command option that you specify. By specifying a time interval, you can have vmstat run continuously, so that you can see how the values vary over time. As shown in Listing 4-12, using the -s option prints paging-related activity.

who

The who command tells you who is logged in to the system, and how long each user has been connected. This command can be useful if a performance problem arises, because you can

Listing 4-12 Output from the vmstat command showing paging activity.

```
$ vmstat -s
5431 swap ins
5431 swap outs
1376 pages swapped in
426 pages swapped out
9704169 total address trans. faults taken
2159795 page ins
9236 page outs
136606 pages paged in
21451 pages paged out
2064504 reclaims from free list
2097094 total page reclaims
773 intransit blocking page faults
6040874 zero fill pages created
3925703 zero fill page faults
1457303 executable fill pages created
76804 executable fill page faults
0 swap text pages found in free list
735656 inode text pages found in free list
185 revolutions of the clock hand
105428 pages scanned for page out
16850 pages freed by the clock daemon
50286274 cpu context switches
90662460 device interrupts
2732863 traps
229976779 system calls
```

quickly determine whether an increase in the number of concurrent users has occurred. It can also be useful in checking for security intrusions, because you may notice an unexpected user.

Using System Instrumentation

Standards for network and system management, such as the Simple Network Management Protocol (SNMP) and Desktop Management Interface (DMI), were developed to help make management easier. They provide industry-standard ways to build instrumentation and interface into the instrumentation, respectively. SNMP is used to access Management Information Bases (MIBs), and DMI is used to access Management Information Formats (MIFs).

Standard MIBs and MIFs define the metrics that can be used by any vendor's instrumentation. Vendor-specific MIBs and MIFs provide vendor-specific instrumentation. This section looks at some of the system instrumentation available through each of these standards.

Many tools already exist for accessing this instrumentation. Several vendors offer browsers and monitoring capabilities that use a common interface to access instrumented objects from different hardware platforms and operating systems. For example, the common enterprise management frameworks, such as the HP Network Node Manager, include a MIB Browser tool to access MIB data. They may also include tools that can be used to monitor MIB data on remote systems from the enterprise management platform. Toolkits are available that provide an interface with which people can write their own tools to monitor or track this information. Furthermore, toolkits exist for creating your own instrumentation.

Many valuable system resources can be monitored via these standard interfaces, to detect system events or faults. Some of the resources that you may be interested in are reviewed in this section.

SNMP

A MIB is a standard way of representing information of a certain category. For example, MIB-II provides useful information about a system, such as the number of active TCP connections, system hardware and version information, and so forth. OpenView IT/Operations (IT/O), discussed later in this chapter, provides a MIB Browser. The MIB Browser tool helps you to discover which MIBs are available, and to see the information being provided by each MIB. The MIB Browser tool can check the value of anything contained in a MIB. If you find a MIB that contains some useful fields, you can use the MIB Browser to gather that data from the target system. The resulting data is displayed in the MIB Browser's output window on the screen. By browsing through available MIBs, and by querying values of selected MIB fields, you can gather specific information needed to monitor systems and troubleshoot problems.

The SNMP interface provides access to objects stored in various MIBs. MIB-II is a standard MIB that has been implemented on most UNIX systems. On HP-UX systems, the HP-UNIX MIB defines various metrics for monitoring system resources. Other vendors, such as Sun, have vendor-specific MIBs that provide similar information. Appendix A includes complete MIB definitions.

MIB-II, the "System MIB," is a standard repository for information about a computer system, and is supported on a variety of platforms, including UNIX and Windows NT. MIB-II contains information about a computer system, such as its name, system contact, and the length of time that it has been running. It also contains statistics from the key networking protocols, such as TCP, UDP, and IP. Statistics include packet transmission counts and error counts. Table 4-1 lists several variables in MIB-II that will help you to monitor system resources effectively. Both the actual MIB variable name and a description are provided for each variable.

The HP-UNIX MIB contains important information about the users, jobs, filesystems, memory, and processes of a system. The number of users logged in to the system and number of jobs running are both indications of how busy the system is. Reduced amounts of free swap space or filesystem space can serve as warnings of potential problems. The process status can be checked to see whether a particular application is still running normally on the target system. Table 4-2 contains some of the interesting metrics from the HP-UNIX MIB for monitoring system resources.

Table 4-1 Important MIB II Fields to Monitor

MIB Variable Name	*Description*
`sysDescr`	System description
`sysObjectID`	Unique identifier for the system
`sysUpTime`	Amount of time since the last system reboot
`sysContact`	System contact person
`sysName`	System name
`sysLocation`	System location
`sysServices`	The network services performed by this system

Table 4-2 Important HP-UNIX Variables to Monitor

MIB Variable Name	*Description*
`computerSystemUsers`	Current number of users on the system
`computerSystemAvgJobs1`	Average job queue length over the last minute
`computerSystemAvgJobs5`	Average job queue length over the last 5 minutes
`computerSystemAvgJobs15`	Average job queue length over the last 15 minutes
`computerSystemMaxProc`	Maximum number of processes allowed in the system
`computerSystemFreeMemory`	Amount of free memory
`computerSystemPhysMemory`	Amount of physical memory
`computerSystemMaxUserMem`	Maximum user memory
`computerSystemSwapConfig`	Amount of swap space configured
`computerSystemEnabledSwap`	Amount of swap enabled via swapon
`computerSystemFreeSwap`	Amount of free swap space
`computerSystemUserCPU`	Amount of CPU used by users
`computerSystemSysCPU`	Amount of CPU used by the system
`computerSystemIdleCPU`	Amount of idle CPU

DMI

System resource information can also be retrieved by using the Desktop Management Interface (DMI), which is another standard for storing and accessing management information. Management information is represented in a text file in the Management Information Format (MIF). Management information is divided into components. Each component has a Service Provider (SP) that is responsible for providing DMI information to the management applications that request it.

Several system platforms, including HP-UX, provide instrumentation for the System MIF and the Software MIF. Appendix A contains a complete listing of these MIFs.

Similar to MIB-II, the System MIF can be used to get generic system information, such as how long it has been running, and system contact information. It includes the system name, boot time, contact information, uptime, the number of users, as well as some information about the filesystem and disks.

The Software MIF provides information about the software products and product bundles installed on the system. The Software MIF can be a useful tool after a problem with a product has been discovered. By using a MIF Browser, you can examine the Software MIF to see whether the problem might be caused by a bad patch installation or a modified file. The MIF contains revision information for each product, and its creation and modification times. Version information can be checked to see whether a compatibility problem exists. Finally, the product's vendor information is provided in case product support personnel needs to be contacted.

Using Graphical Status Monitors

The graphical status monitoring tools described in this chapter are also referred to as *enterprise management frameworks*. These tools monitor multiple systems from a central location and display status information graphically.

Because you may not be able to sit at a console or watch the front panel of each system for which you are responsible, you need to be able to monitor system faults from a tool that is external (or remote) to the system. Besides, even if the console or front panel doesn't indicate any problems, network problems can make the system inaccessible to the end-user. Graphical status monitors can detect connectivity problems because they rely on network polls to gather status information.

Your server may depend on services provided by other systems. For example, network services such as the Domain Name Service (DNS), Dynamic Host Configuration Protocol (DHCP), Network File System (NFS), and e-mail are critical to the server, but are unlikely to be running on the server that you're monitoring, and consequently aren't tracked by any local monitors. This is critical information, because if the DNS is down, other systems may not be able to reach the system being monitored. Because the enterprise framework products are gathering status information about multiple systems at a central site, it is more likely that both the server and its service providers are being monitored.

Graphical status monitors provide many features that can help you to detect system faults, especially hardware or software faults. Most graphical status monitors provide hierarchical maps or visual displays that indicate status information. This saves you the time of correlating event

data from logs to determine status. The graphical status monitors provide remote management capabilities, so you aren't required to have a physical console for each system. Furthermore, they can automatically discover the systems in your enterprise, so you don't have to remember system names and manually configure them. A graphical view typically can be customized by setting up filters, so that you see only those systems that you are responsible for managing.

This section describes only a couple of the available graphical status monitors. Others with similar system monitoring capabilities, such as HP's IT/O, Sun's Enterprise SyMON, or BMC PATROL, are mentioned in other parts of the book.

OpenView Network Node Manager

Network Node Manager (NNM) is a management product based on the HP OpenView platform. NNM is used primarily to view and monitor the status of network and system resources. Information is displayed graphically through a window-based display. A hierarchical set of submaps is available, enabling the customer to navigate and drill down through complex network topologies. Network and system components are represented graphically in maps as icons, which are color-coded to indicate the health of the objects represented. Events are propagated, based on severity, to higher-level submaps to indicate events, such as failures, at lower levels. Through pull-down menus available in NNM, the operator can run tools to get additional real-time status information, or remotely log in to the system and execute diagnostic commands.

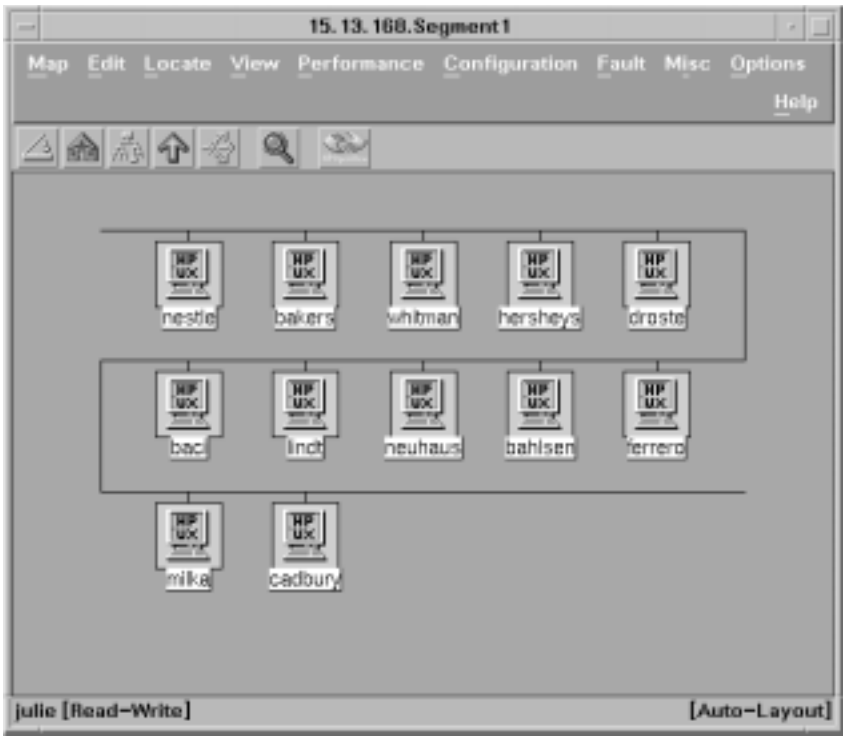
One key feature of NNM is its ability to discover automatically network-addressable components, such as routers, hubs, and computer systems. Because the network discovery activity uses noticeable network resources, you may want to limit it to just the networks that you manage. This can be done by using discovery filters, a configurable option in NNM; or, alternatively, you can schedule discovery process(es) to run during off-peak hours.

After discovery is complete, the network topology information is then displayed on submaps, with colors used to indicate status. An operator can navigate through the submaps to find a particular LAN segment to monitor. In addition to viewing systems, an operator can also drill down to see system information, such as configured network interfaces. Figure 4-1 shows a segment map with icons indicating the health of each system in the segment.

NNM provides mechanisms to collect statistics and generate reports on individual network devices, including systems. NNM periodically checks the status of systems and devices by sending an ICMP echo request (ping). If no reply is received, a Node Down event is sent as an SNMP trap and logged in the Event Browser. NNM also listens to SNMP traps from SNMP supported devices. For instance, if NNM gets a Node Down trap, it changes the color of the icon representing the node that just went down.

The NNM Event Browser is a graphical display of the events that have been received from systems on the network. Events are sent to the management station as SNMP traps. The trap handler receives these traps and stores them in a database. The events can be viewed through the Event Browser, and filters can be used to prevent the operator from being flooded with noncritical information. Filters can be configured based on the sending system or event criticality, for

Figure 4-1 Network Node Manager segment submap shows the health of the systems in the segment.



example. NNM can also process Common Management Information Protocol (CMIP) events for multivendor interoperability. You can use filtering to get history events from a particular system when troubleshooting a problem. After an event is handled, you can use the Event Browser to acknowledge the event.

By using the NNM Event Configuration utility, you can configure how specific SNMP traps should be handled, including the following:

- Logging and display options
- Event severity
- Message format for display in the Event Browser
- Automatic actions

Events can be configured to automatically display a pop-up notification, or to run a command on the management station to send e-mail, call a pager, change an icon color, or generate an audible alert.

After you recognize that a problem exists, the NNM menu interface provides many tools to troubleshoot problems or monitor the system in more detail. NNM provides a performance menu that you can use to check network activity, CPU load, and disk space, or to graph SNMP data. A configuration menu is provided so that you can check network configuration, system statistics, or the SNMP trap configuration. From the fault menu, you can try to reach the system through the network connectivity poll, a ping from the management station, or a ping initiated from another remote system. If you suspect that the route to a system is down, you can test that from the fault menu as well. A terminal window, the SAM interface (HP-UX only), and a MIB Browser are all available from the pull-down menus.

NNM also provides several utilities to help you gather and process data provided in MIBs. You can configure data collection of MIB objects and define thresholds for when to generate an event. You can build your own MIB application to collect MIB objects for graphing or generating tabular output.

As this section has described, OpenView NNM can provide help for numerous system monitoring categories. Faults can be detected and shown graphically, with failure events sent to the Message Browser. You can monitor network performance by using the performance menus. You can check some of the system resource limits by using predefined tools or the MIB Browser. NNM, however, is typically used only if you have many systems to monitor.

NNM is a building block for other HP OpenView applications. Application integration is provided through developer's kits and registration files. More than 300 applications are integrated today with HP OpenView. HP IT/O, discussed later, is one product that extends NNM's capabilities. The most commonly used partner applications are CiscoWorks, Bay Networks Optivity, 3Com Transcend, Remedy ARS, and HP NetMetrix.

OpenView NNM runs on NT and UNIX platforms. Both versions can be used to monitor UNIX systems.

ClusterView

ClusterView is a graphical monitoring tool integrated with OpenView NNM and IT/O. It provides monitoring of systems and other resources in MC/ServiceGuard environments. MC/ServiceGuard is a Hewlett-Packard high availability software product that detects system failures, network or LAN card failures, and the failure of critical applications. While MC/ServiceGuard can be configured to handle these failures automatically, it is through ClusterView that you can capture these high availability events and graphically view the health of systems that are part of MC/ServiceGuard clusters. MC/ServiceGuard is supported only on HP 9000 Series 800 systems running HP-UX 10.*x* or later operating systems.

MC/ServiceGuard is most commonly used in a cluster environment. Software on each system monitors the other systems. When system failures occur, MC/ServiceGuard software can detect the problem and automatically restart critical applications on an alternate node. Monitoring of failures is done automatically, but without ClusterView, you may need to use MC/ServiceGuard commands to verify that the cluster software itself is working.

MC/ServiceGuard detects numerous cluster events, such as the failure of a critical application. These events can be forwarded to a management station by using either SNMP traps or opcmsg, a proprietary communication mechanism used by IT/O. Information about an MC/ServiceGuard cluster is stored in the HP Cluster and HP MC/ServiceGuard Cluster MIBs. These Cluster MIBs are listed in Appendix A. The following is a list of the MC/ServiceGuard events that trigger SNMP traps from the MC/ServiceGuard subagent:

- MC/ServiceGuard subagent was started
- MC/ServiceGuard cluster is reforming
- MC/ServiceGuard cluster is up on this node
- MC/ServiceGuard cluster is down on this node
- MC/ServiceGuard cluster configuration has changed
- MC/ServiceGuard package is starting
- MC/ServiceGuard package is running
- MC/ServiceGuard package is halting
- MC/ServiceGuard package is down
- MC/ServiceGuard service is down
- MC/ServiceGuard package switching flags have changed
- MC/ServiceGuard relocatable IP address added
- MC/ServiceGuard relocatable IP address removed
- MC/ServiceGuard network interface local switch
- MC/ServiceGuard subnet is up
- MC/ServiceGuard subnet is unavailable
- MC/ServiceGuard node joined the cluster
- MC/ServiceGuard node has halted
- MC/ServiceGuard node has failed

MC/ServiceGuard software detects a variety of error conditions, but it does not have a sophisticated notification mechanism for customers to learn what happened. Errors often are written to the system log, which can be used to help retrace what occurred.

Whereas MC/ServiceGuard can monitor the system, network, and processes and provide automatic recovery, ClusterView provides you with event notification of these recovery events. For example, if MC/ServiceGuard detects a local LAN card failure, it can reconfigure the IP connectivity on a backup LAN card on the local system transparently. Using ClusterView, you will see an event indicating that MC/ServiceGuard has performed a local switch to a backup LAN card. The bad LAN card should be replaced, to eliminate the LAN card as a single point of failure.

ClusterView can be used to help you with diagnosis in MC/ServiceGuard environments. ClusterView is an OpenView application with custom monitoring capabilities for MC/ServiceGuard and MC/LockManager clusters. An SNMP subagent that can be used to send events to an OpenView management station is included with the MC/ServiceGuard and MC/LockManager

products. These events, sent as SNMP traps, can actually be received by any management station that understands SNMP (for example, Computer Associates' Unicenter product). These events are received in OpenView's event browser.

ClusterView provides automatic discovery and real-time status and event notification via the Event Browser and graphical displays of MC/ServiceGuard clusters, systems, and packages. Templates are provided to map the cluster events to readable text. Without these templates, events are unrecognized or unmatched traps in OpenView. With these templates, the traps are formatted in the NNM Event Browser or IT/O Message Browser when ClusterView is installed.

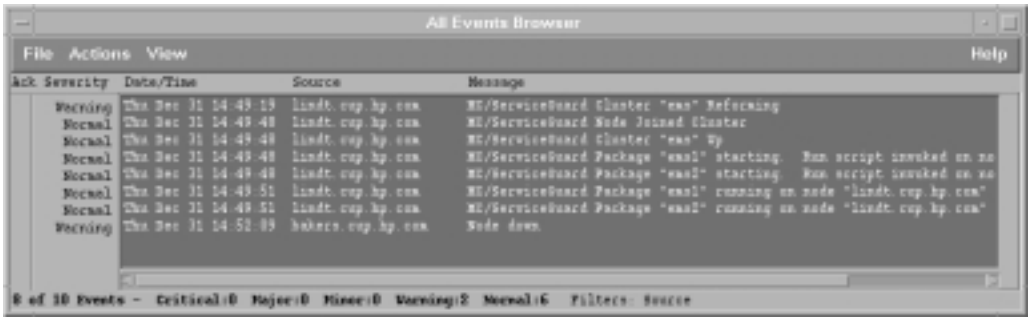
When a system failure is detected by MC/ServiceGuard, it moves all critical resources to an alternate node. A series of SNMP traps are generated by the MC/ServiceGuard subagent as an event occurs. Figure 4-2 shows the events in the NNM Event Browser after the system "bakers" fails. MC/ServiceGuard first detects the failure and starts the two packages, ems1 and ems2, on the alternate system, lindt. The Node Down trap is generated by NNM when it detects that system "bakers" is down.

ClusterView provides additional capabilities when used with IT/O. SNMP events are sent to the Event Browser, where ClusterView provides special troubleshooting instructions and recommends actions to help resolve the problems. Some data collection activities are done automatically. For example, in response to a package failure, ClusterView automatically retrieves the system's system log file entries from the time of failure to aid in diagnosis. Common HP-UX monitoring tools, such as netstat and lanscan, are included by ClusterView in IT/O's Application Desktop, along with MC/ServiceGuard-specific tools, such as cmviewcl.

In addition to high availability clusters, ClusterView can monitor user-defined clusters. ClusterView provides a configuration tool that enables the administrator to create a cluster, and then displays that cluster on its cluster submap. The operator can then monitor all the cluster systems at a glance, because they are all in the same OpenView window. Also, the operator can launch monitoring tools, such as HP PerfView, on the cluster, avoiding the need to select each system manually when running each tool.

ClusterView can be a useful extension to the capabilities of NNM if you are managing MC/ServiceGuard clusters, MC/LockManager clusters, or groups of systems. You can view detailed

Figure 4-2 NNM Event Browser showing MC/ServiceGuard events after a system failure.



screens containing high availability configuration information about your cluster. In addition to processing faults, ClusterView provides recovery actions and troubleshooting help for these events.

ClusterView runs on HP-UX and NT systems and requires OpenView NNM or IT/O. The ClusterView software for either platform can also be used to monitor Microsoft's NT Cluster Servers, its high availability clusters. Both NT and MC/ServiceGuard clusters can be monitored concurrently from the same ClusterView software.

Unicenter TNG

Computer Associates' Unicenter TNG is an enterprise management platform that provides graphical status monitoring and provides system and network management for a heterogeneous enterprise. Unicenter TNG provides monitoring and management for all the resources in your environment, including system resources, networks, databases, and applications.

Unicenter TNG provides the framework for an integrated management solution to manage all IT resources via a common infrastructure. The TNG framework itself includes the following components: auto discovery, object repository, Real World interface, event management, calendar management, reporting, virus detection, and desktop support. Together with vendor, third-party, and custom-built applications, Unicenter TNG provides increased management and maintenance capabilities for the enterprise.

Unicenter TNG provides automatic discovery of networked objects, including systems and other resources within the enterprise. Information is stored in the Common Object Repository and can be displayed topographically in the Real World interface. Discovery filters can also be used, to limit discovery to a specific subnetwork or to specify which types of resources Unicenter TNG should discover.

A Common Object Repository stores the information used to create the Real World graphical views. You can browse the repository by using the Class Browser, Object Browser, or Topology Browser. Using ObjectView, you can get details on the performance of devices, and you can even graph the data.

The Real World interface provides graphical views that can be organized based on business functions, geographical location, or any logical groupings. The views can show the topology of the enterprise in two or three dimensions. These views can be used to see the status of the systems and resources in your environment.

Unicenter TNG provides management by using a distributed management approach. Distributed agents are responsible for monitoring and control. Centralized managers provide core management throughout the enterprise, including data correlation from one or more agents, workload management, and job management. The agents monitor and control based on policies provided by managers.

The agents run on managed nodes and gather data, apply filters, and report when necessary. Some provide control or execution on behalf of the managers. The agents send notifications, and can be polled. The agents can also collect performance data or be configured to send events and perform actions based on thresholds.

To view agents and get information about them, Unicenter TNG provides a MIB-II agent display to view MIB-II information, a node view, Distributed State Machine (DSM) view, and an Event Browser. The DSM tracks the status of objects across the network. It gathers information from the repository and agents to maintain the state of objects based on configured policies. The node view displays detailed state information about the system objects that are watched by the DSM.

Finally, managing events is one of the core capabilities provided by Unicenter TNG. The hub of event management is at the Unicenter TNG Event Console. You can configure policies to respond automatically to specific events, send SNMP traps based on events, forward events, filter out unimportant events, correlate events from several agents, or feed events into the DSM. In conjunction with the calendar management provided by Unicenter TNG, you can change or set event policies based on the time. For example, you may want to apply different policies during the weekend.

Using Event Monitoring Tools

This section covers various event monitors that are available for monitoring system resources. You can configure event monitors to generate a message when a change in status occurs or when a predefined threshold condition is met. This is different from commands, which give you status reports only when asked, and performance monitoring, which is generally studied over a long period of time. Event monitors generate a notification message soon after faults and events occur.

Event Monitoring Service

Several monitors discussed in this section are integrated into the Event Monitoring Service (EMS) framework. The EMS framework, available only on HP-UX systems, enables monitors to be provided in a consistent manner for a system. Although the EMS framework itself is freely available, some monitors are delivered with HP-UX Online Diagnostics, some are sold separately, and others are bundled with the individual products for which they provide monitoring.

EMS provides a consistent GUI for the discovery and configuration of resources that can be monitored. Using EMS, you can define conditions that indicate when notification events should be sent, which can be at periodic intervals, when a component's state changes, or when a threshold condition is met. EMS also enables you to configure where events should be sent. You can configure EMS to send events to OpenView IT/O, directly to any SNMP-capable management station, to a network application listening on a TCP or UDP port, to an e-mail address, or locally to the console, system log file, or a regular log file. Furthermore, you can configure MC/ServiceGuard to make packages dependent on these EMS resources.

EMS monitors provide help primarily with fault and resource management. Performance monitoring generally requires more sophisticated tools. Some system fault and resource monitoring capabilities are provided by the EMS HA Monitors product, discussed in the next section. Other EMS monitors allow you to detect when you are getting low on system resources, such as file descriptors, shared memory, and system semaphores.

Templates for formatting EMS SNMP traps into various enterprise management platform Event Browsers, including OpenView, CA Unicenter, and other freely available EMS tools, are

available to download from the Internet at http://www.software.hp.com, under the High Availability Software product category. A developer's kit is also available so that customers and system management software providers can integrate their own EMS monitors. EMS manuals are available at http://docs.hp.com/unix/ha.

EMS High Availability Monitors

The HA Monitors product contains several EMS monitors for monitoring filesystem space, network interface status, disk status, and MC/ServiceGuard cluster status. HA Monitors also detects changes in the number of users and jobs. This product has been extended to include database monitoring capabilities as well.

Available filesystem space can be monitored for any mounted filesystem. The operational status is monitored for each configured network device. For disks, you can monitor physical volume status, logical volume status, the number of mirrored copies, and summary information.

The MC/ServiceGuard cluster monitor, included with HA Monitors, reports on cluster events, such as the failure of a cluster node, and provides monitoring that is similar to the events reported in ClusterView. The ClusterView product provides more complete monitoring of MC/ServiceGuard clusters, but it requires the purchase of HP OpenView. Here are the resources monitored by the cluster monitor:

- Cluster status
- Node status
- Package status
- Service status

Monitoring node status using EMS can be done to provide notification when MC/ServiceGuard detects problems with the system. Whereas MC/ServiceGuard's job is to detect a system failure and move the configured application package(s) to an alternate system, the EMS cluster monitor's job is to notify you of such an event.

EMS Hardware Monitors

The EMS Hardware Monitors provide the ability to detect and report problems with system hardware resources, including system memory, tape devices such as SCSI, Digital Linear Tape (DLT), and Digital Data Storage (DDS), tape libraries, and autoloaders. These monitors detect device errors, component failures, page deallocation errors, and other faults. They poll the hardware at regular intervals and most notify of hardware errors in real time. These monitors are delivered with HP-UX Online Diagnostics, which are freely available for HP-UX. The EMS Hardware Monitors provide monitoring for the following system components and Hewlett-Packard products:

- System memory
- SCSI tape devices

- DDS-2 Autoloader (A3400A)
- DDS-3 Autoloader (A3716A)
- DLT 4000 4/48 Library; HP-UX; Differential SCSI (A3544A)
- DLT 4000 2/48 Library; HP-UX; Differential SCSI (A3545A)
- DLT 4000 2/28 Library; HP-UX; Differential SCSI (A3546A)
- DLT 4000 & 7000; 2/28; Drives Differential; Robotics SE/Diff (A4850A)
- DLT 4000 & 7000; 15 slot; Deskside/Rack; Differential (A4851A)
- DLT 4000 & 7000; 4/48; Drives Differential; Robotics SE/Diff (A4855A)
- DLT 4000 & 7000; 588 slot; Drives Diff; Robotics SE (A4845A)
- DLT 4000 & 7000; 100 slot; Drives Diff; Robotics SE (A4846A)
- DLT 4000 & 7000; 30 slot; Differential (A4853A) Channel Adapters

These EMS Hardware Monitors are designed to provide consistency in the configuration interface, event detection, and message formats that provide a detailed description of a problem and a recommended recovery action.

The EMS Hardware Monitors can report low-level device errors that are encountered during an I/O session with a device. They detect and report component and Field Replaceable Unit (FRU) failures, including fan and power supply problems. Protocol errors are also detected.

For monitoring tape devices, events include problems reading or writing data, bad tapes, wrong tapes, temperature problems, tape loader errors, tape changer problems, and incorrect firmware. For monitoring system memory, the monitor checks the page deallocation table and reports an event when the table is 60, 90, or 100 percent full. This indicates that a new memory SIMM (Single In-line Memory Module) should be added to replace a failed memory chip. These threshold values are configurable.

The monitor assigns hardware events severity levels, which reflect the potential impact of an event on system operation. Table 4-3 provides a description of each severity level.

EMS Hardware Monitor configuration is done by using the Hardware Monitoring Request Manager. Notification conditions can be configured in a consistent way for all supported hardware resources on the system. As hardware is added to the system, monitoring can be enabled automatically. Figure 4-3 shows an example of using the Hardware Monitoring Request Manager to send SNMP traps of all critical and serious tape and memory events. To configure with MC/ServiceGuard, you need to use the MC/ServiceGuard configuration interface.

The EMS Hardware Monitors provide fault information only. No performance-related events are included.

When an EMS Hardware Monitor detects an event, a notification message is sent to the designated target locations. The message contains a full description, including the system on which the event occurred, the date and time when the event was detected, the hardware device on which the event occurred, a description of the problem, the probable cause, and recommended action. The event message contains detailed information, including product/device identification information, I/O log event information, raw hardware status, SCSI status, and more.

Table 4-3 Description of Hardware Event Severity

Severity	*Description*
Critical	An event that will or has already caused data loss, system downtime, or other loss of service. Immediate action is required to correct the problem. System operation will be impacted and normal use of the hardware should not continue until the problem is corrected. If configured with MC/ServiceGuard, the package will experience failover.
Serious	An event that may cause data loss, system downtime, or other loss of service if left uncorrected. The problem should be repaired as soon as possible. System operation and normal use of the hardware may be impacted. If configured with MC/ServiceGuard, the package will experience failover.
Major Warning	An event that could escalate to a more serious condition if not corrected. The problem should be repaired at a convenient time. System operation should not be impacted and normal use of the hardware can continue. If configured with MC/ServiceGuard, the package will not experience failover.
Minor Warning	An event that will not likely escalate to a more serious condition if left uncorrected. The problem can be repaired at a convenient time. System operation will not be interrupted and normal use of the hardware can continue. If configured with MC/ServiceGuard, the package will not experience failover.
Information	An event that occurs as part of the normal operation of the hardware. No action is required. If configured with MC/ServiceGuard, the package will not experience failover.

Most EMS Hardware Monitors are “stateless.” In other words, events of the designated severity are forwarded as soon as they are detected; no aspect of history or correlation with other data is involved, except that the monitor limits repeated messages by using a repeat frequency. Determining the current status of a device is difficult, because messages are sent only when an event occurs.

To monitor for hardware device state changes, you can use a Peripheral Status Monitor (PSM), which maintains the state of monitored hardware devices and reports state changes. The PSM gathers events from the other EMS Hardware Monitors, but does not send its own notification unless a state change has occurred. By default, critical or serious events cause the PSM to change a device’s status to Down.

For example, critical tape events from a tape monitor would cause the PSM to change the device’s status to Down. The PSM would then send a single “Tape device status = Down” event if the administrator had requested to be notified of such an event. This may be the only message visible to the administrator. Additional disk failure messages would not be forwarded because they are not the result of a status change (in other words, the status remains Down). This reduces the number of events that need to be processed by the user. The last event received should reflect the current device status.

Figure 4-3 Configuring EMS Hardware Monitors using the Hardware Monitoring Request Manager.

Most monitors cannot automatically detect when a device has been fixed. When a problem is solved, the set_fixed command must be used manually to alert the PSM to reset the device status to Up.

When using an enterprise management tool, such as IT/O, which receives messages from multiple systems, you should use the PSM to reduce information overload. However, make sure that the stateless events are also configured to go somewhere (such as the system log file), because they provide valuable diagnostic information when a component has failed.

Monitoring hardware device status is done through the EMS Configuration GUI.

You can learn more about Hewlett-Packard's diagnostic tools on its Web site at http://docs.hp.com/hpux/systems/.

Enterprise SyMON

Sun's Enterprise SyMON is a system management platform for monitoring and managing the Sun systems in your enterprise. Enterprise SyMON provides administrators the capability to manage all of their Sun systems remotely from a common interface. Enterprise SyMON can automatically discover the Sun systems in the environment. Intelligent SyMON agents run on each system, to provide monitoring and remote management capabilities.

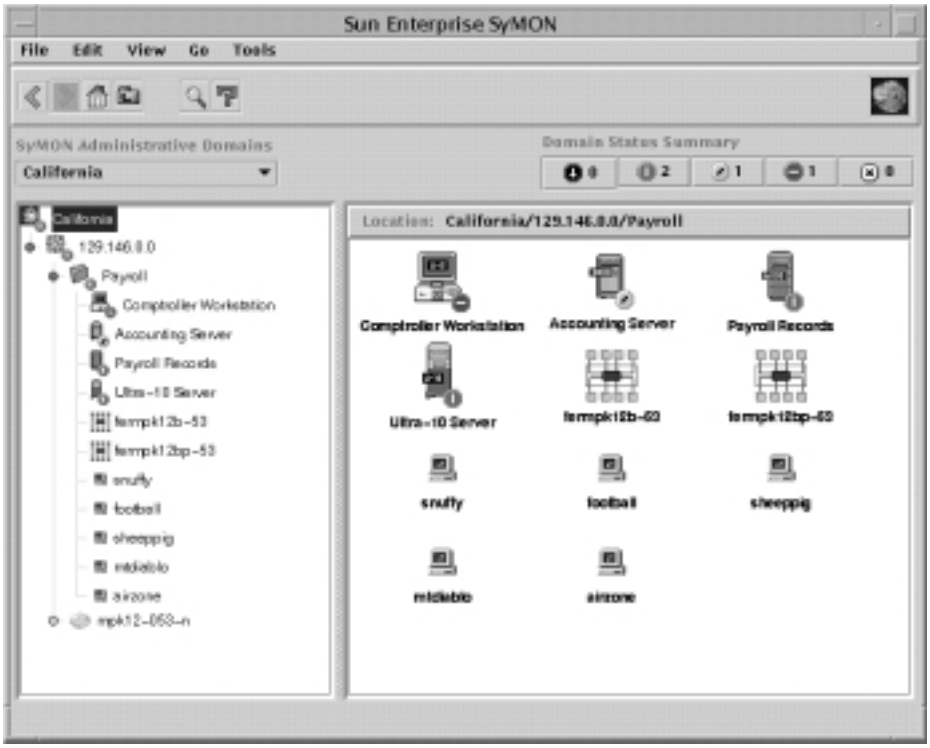
The console layer of Enterprise SyMON provides a visual representation of all managed objects. The console layer provides several views of the enterprise, including logical views and topological views. The Logical View window provides a hierarchical representation of the systems being managed. Indicators are used on system icons to indicate the alarm status of the system. As shown in Figure 4-4, you can see the status of the systems in the payroll domain. Badges on each system icon indicate the alarms for the node. The Domain Status Summary at the top of the window shows how many alarms are outstanding in each category. Figure 4-4 shows no down alarms, two critical alarms, one alert, one caution, and no disabled. So, Comptroller Workstation has a caution-level alarm outstanding.

When a critical event occurs, such as a hardware component failure, it is indicated in the Logical View window. As the event occurs, the failed hardware component is also highlighted in the Physical View window. You can use this photo-like view of the system to detect and isolate failed or failing components. As shown in Figure 4-5, the Physical View indicates that board 3 is disconnected, and highlights the back panel of the server to show you where the board is.

Enterprise SyMON provides event and alarm management. Alarms and actions can be configured so that events are sent via SNMP traps to the SyMON console when certain conditions or thresholds are met. Event-based actions and notifications, such as e-mail, can also be configured. Recovery actions can be associated with an event. Additional events can be defined and generated by placing rules written in the TCL scripting language in a special directory. SyMON provides features for correlating events and filtering based on priority and severity.

SyMON provides intelligent agents, which run on the systems being monitored. The agents are configured with intelligence to detect abnormal conditions, to generate alarms based on default or customized thresholds, and to perform actions automatically, based on certain predefined events.

Figure 4-4 Viewing the status of systems in the payroll domain from the SyMON console.

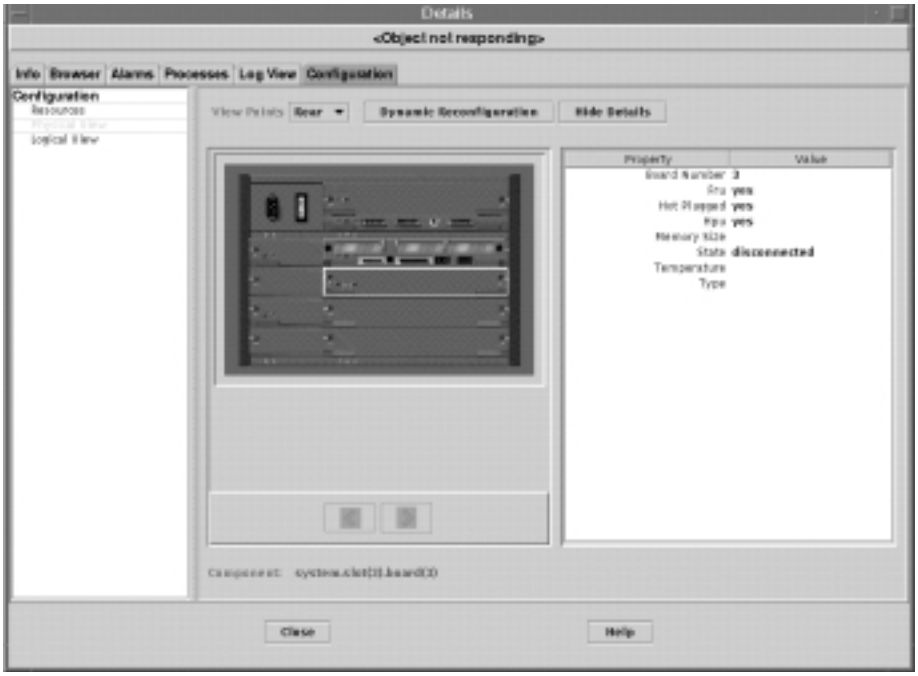


The agent architecture consists of several modules. For example, the Config-Reader module is responsible for monitoring all hardware components. The agents are extensible such that new modules can be dynamically loaded from the console without disrupting service. If you don't need certain modules, you can save resources by unloading them.

A browser window, shown in Figure 4-6, shows the different statistics that can be monitored on the left panel. The panel in the right shows current System Load Statistics. Many of the resources available to be monitored are mentioned in this section.

The Hardware Config-Reader module provides configuration management by tracking the hardware and firmware configured on the system, down to the serial number. This information is used to create logical and physical views. The SyMON agent provides hardware fault monitoring, as well as predictive failure analysis for memory and disks. The Config-Reader module monitors hardware and alerts you at the console when a problem exists. If it is a predicted memory failure, you can configure actions to do a dynamic reconfiguration to remove the bad memory. The Config-Reader reports on many hardware faults, including temperature problems and power supply status. It monitors CPU and memory board status, controllers, I/O devices, and tape devices.

Figure 4-5 Using the SyMON Physical View to identify a failed hardware component.



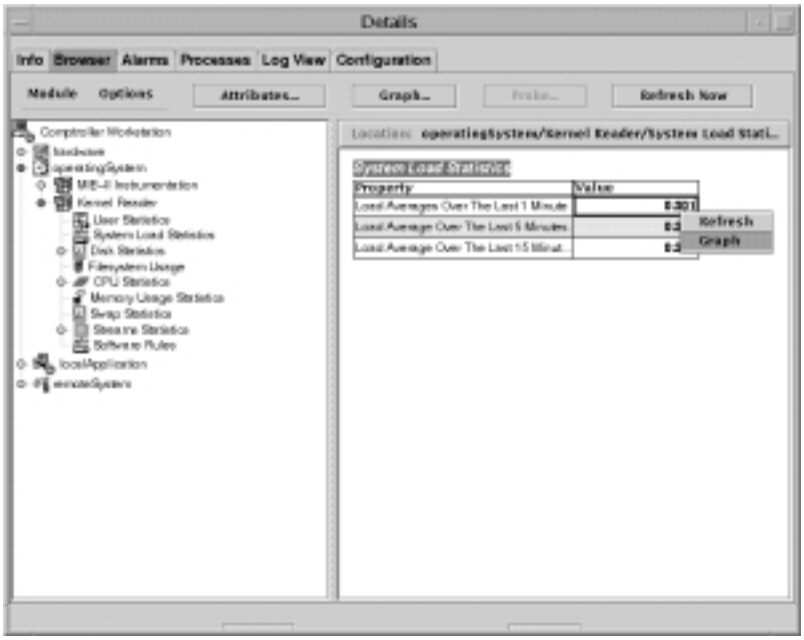
For the operating system, the agent includes modules to monitor CPU utilization, memory usage, directory size, file size and file modification time, MIB-II objects, NFS activity, inode usage, swap statistics, filesystem usage, and disk rates and service times. A file-scanning module also is available that can be used to check log files, such as the system log, for errors or specific patterns.

The Health Monitor uses rules based mostly on performance metrics to correlate the metrics to detect when alarm conditions exist. It sends an alarm when alarm conditions occur, along with suggested steps on how to improve system performance. The Health Monitor has rules to detect swap space conditions, kernel contention, CPU, disk, or memory bottlenecks, printer problems, and filesystem conditions.

SyMON agents provide active management, including active configuration management controls for dynamic reconfiguration, system domain management, and an “alternate pathing” feature for redirecting disk I/O in the event of a controller failure. With this capability, administrators can take care of repairs, such as replacing a failed memory board, without a service interruption.

Sun Enterprise Servers Models E3000 through UE10000, with Solaris 2.6 or greater, support dynamic reconfiguration. This feature enables you to replace boards online without taking down the system. You can have backup boards on standby and available for immediate use. Or, if a CPU or memory board fails, you can unconfigure and disconnect the failed board online via SyMON,

Figure 4-6 The SyMON Browser showing the various system resources that can be



replace the board, and then connect and configure the new board, making the resource available to the system without a reboot. This feature is also available for hot-pluggable disk devices.

As previously described, SyMON can help you with configuration, fault, and resource monitoring on your system. However, SyMON is available only for monitoring Sun systems.

OpenView IT/Operations

IT/Operations (IT/O) is an OpenView application providing central operations and problem management. NNM is included as part of the IT/O product. IT/O uses intelligent agents that run on each managed system to collect management information, messages, and alerts, and send the information to a centralized console. After receiving events, IT/O can initiate automatic corrective actions. When an operator reads an individual message, guidance is given and actions may be suggested for further problem resolution or recovery.

IT/O comes with predefined monitors and templates, including monitors for e-mail, CPU utilization, and swap utilization, among other things. Using log file templates allows you to monitor system log files for system errors, switch user events, logins, logouts, and kernel messages.

IT/O enables you to define and customize your own monitors and templates so that you can monitor arbitrary MIB variables, such as the system uptime MIB variable mentioned earlier in this chapter. IT/O periodically queries the MIB object to determine whether or not a message should be generated. You can write a program or script that can be periodically invoked by an

IT/O agent, and you can modify templates and message conditions so that an operator is paged under certain conditions.

Many other tools plug in to IT/O to provide additional monitoring and management capabilities. IT/O is useful when an operator needs to manage numerous systems consistently. Templates can be modified and then downloaded to a set of systems, enabling multiple systems to be monitored identically. This way, monitoring can be set up in a consistent way for all systems.

IT/O has four main windows:

- **Node Bank:** Displays the systems managed by an operator as icons, and enables them to be organized into node groups.
- **Message Groups:** Displays logical message groups, such as Performance, Oracle, and backup. The message groups serve as one way to organize messages in the Message Browser window.
- **Message Browser:** Shows the events that have been received by the management server.
- **Application Desktop:** Provides access to commonly used diagnostic and administrative applications.

You can see an example of the Node Bank window in Figure 4-7. In this window, the node color reflects the color of the most critical event that has been received but not yet acknowledged.

The Message Browser can filter out messages from systems that you don't care about. If you are responsible for only a specific system function, such as performance, you can configure the Message Browser to show only those messages from a specific message group.

The IT/O Application Bank provides some other tools to monitor your system, and it also provides remote access tools to diagnose problems further. From the Application Bank, you can

Figure 4-7 IT/O Node Bank window showing node status.

bring up a telnet window or, for HP-UX systems, run SAM on the system having problems. You can run PerfView or GlancePlus (discussed later in this chapter), check the print status, or check the CPU load on any UNIX system.

As previously described, IT/O provides assistance with multiple aspects of system monitoring, especially faults, and resource and performance management. IT/O can also help with security monitoring, with its predefined template for monitoring root login attempts.

GlancePlus Pak 2000

Hewlett-Packard also includes a preconfigured, single-system version of IT/O with its GlancePlus Pak 2000 product. In addition to displaying performance data, the product includes a Java-based GUI that presents diagnostic applications and an Event Browser. The product enables you to connect to information from multiple systems, as long as you connect to one system at a time.

GlancePlus Pak 2000 includes the intelligent agent technology from its enterprise version, enabling it to collect events from a variety of sources and execute automated actions. After events are received in the Event Browser, an operator can trigger some predefined recovery actions.

Security Monitoring

The whole area of security is a huge subject. Entire books are dedicated to security alone. Security is usually broken down into two categories: network security and host-based security. This section focuses on host-based (or system) security, so that you can monitor and detect activities that could compromise system, application, or data availability. Host-based security intrusions usually are the most problematic. As a system administrator, you should monitor the system for activities that would prevent the system from doing what it is intended to do.

Security Overview

The level of security needed for your system depends on what you are trying to protect. Both the US government and European Information Technology Security Evaluation Criteria (ITSEC) have defined sets of security levels. The most common level is C2, which is the de facto standard for secure UNIX systems. Level B1 security, which is more secure than C2, is often required in government, military, and commercial applications. HP-UX, for example, operates in two modes of security: standard mode, which has no security, and trusted mode, which is C2-level compliant. Each level of security has different requirements.

Regardless of the level of security you are trying to provide in your environment, several categories for system security apply to all levels. The requirements in each category are more restrictive as you increase the security level. These categories are authentication, authorization, access control, data security, and physical security.

Before describing each of these categories, you need to know that implementing a host-based security plan will include defining policies for preventing intrusions and for monitoring to detect intrusions. Password policies are an example of the many policies that should be defined

and enforced to try to prevent intrusions. When monitoring to detect intrusions, detection systems need to be told what to monitor.

After you implement intrusion-prevention policies, you need to put security monitoring and intrusion-detection monitors in place. You want these monitors to tell you when an intrusion was attempted, is occurring, or has occurred. The following are the system security categories that apply to all security levels, along with a description of what can be done to prevent and detect intrusions:

- **Authentication:** Usually done to verify a user's ID prior to allowing access to a system or resource. Authentication is usually accomplished with a password, which serves as proof that a user is who they claim to be. Password length and complexity restrictions, as well as password lifetime limits, are some of the devices that can be used to make getting past authentication checks more difficult for unauthorized users. More secure measures include hiding the password file, which contains encrypted passwords.
- **Authorization:** The process of granting privileges to individual users. UNIX has two main classes of users. *Root users* (or superusers) have authorization to do almost anything on a system, including administer the system, perform backups, and bypass security controls. *Regular users* have ordinary access to programs and data. Authorization can be controlled by using time-based authorization, whereby users are restricted to certain hours of the day. Fine-grained authorization allows root access to be restricted to more narrow tasks, giving users only as much power as they need to accomplish their tasks. This provides more control over system security. HP-UX has a special version of SAM, called restricted SAM, which allows restricted use by authorized users, allowing you to delegate limited authority. Monitoring for failed login attempts and super-user logins is important, to see whether anyone is gaining or attempting to gain unauthorized access to the system. IT/O is one product that provides this capability.
- **Access control policies:** Used to define which users have access to various system resources, including files, programs, and printers. Access control is generally handled through UNIX file permissions, which define read, write, and execute permissions by user, group, and everyone. Access Control Lists (ACLs) are also used to grant file access to users on a list. Or, ACLs can be used to list those users who don't have access rights to a file. You can check a file's access rights by using the ls command. You may want to monitor access rights for changes that allow other users to access these restricted files.
- **Data security:** Helps you to protect critical data. This includes backups, which can protect against accidental data loss, and data encryption, which can protect the privacy of information.
- **Physical security:** Covers the physical protection of system resources against deliberate or accidental threats. This includes ensuring against even simple threats, such as someone tripping on an exposed power cord.

Security Monitoring Tools

Auditing is a way to log security-related events on a per-user basis. It can be set up to monitor system calls, specific users, password policies, logins, superuser logins, failed login attempts, and so forth. Because auditing incurs lots of system overhead, you should try to limit it to the most critical security-related events. On HP-UX, auditing is available only in trusted mode. Auditing can be enabled on HP-UX using SAM. Although the system itself provides the library routines for auditing, data reduction and analysis tools are useful for extracting relevant information from audit logs.

Some of the common tools for looking at audit logs are:

- **OmniGuard/ITA (Axent Technologies):** Used to detect intruders and abuse. It uses data from log files and listens for SNMP traps to feed into its rules engine when detecting intruders. It can also monitor file-level accesses.
- **Stalker (Haystack Labs):** Analyzes and compares audit logs to its database, to detect system misuse, attacks, and known system vulnerabilities. It can collect and store audit logs from multiple UNIX systems at a centralized server.

MEMCO Software provides SeOS Access Control, which provides more granular root capabilities. Narrow capabilities can be granted to users to perform specific tasks. This means that you don't need to grant a user full root capabilities, which can be dangerous, just to perform system backups, for example.

You can make some simple checks to help protect your system. Check the /etc/hosts.equiv and /.rhosts files to ensure that the remote host systems listed are authorized to access the system. Also, the optional file /var/adm/inetd.sec can be used to explicitly deny or allow access to specific network services, so you should verify that this file is configured correctly.

Using Diagnostic Tools

Various support tools monitor errors and faults, configuration information, and troubleshooting for hardware components, including the CPU, system memory, and tape devices. Some of these support tools also monitor software configurations, to track changes.

Support Tool Manager

HP's Support Tool Manager (STM) provides access to a set of tools for verifying and troubleshooting HP-UX system hardware. These online diagnostic tools provide the ability to determine device status, get configuration information, and diagnose hardware problems. These tools are available by using a GUI or through commands, and have the flexibility to be invoked automatically at periodic intervals.

STM discovers the hardware devices on a system and can diagnose memory errors, Low-Priority Machine Check (LPMC) errors, I/O driver errors, Logical Volume Manager (LVM) errors, and over-temperature events. Memory errors include single-bit errors and page deallocation events.

STM includes the Automatic Configuration Mapper, shown in Figure 4-8, which gives a graphical view of your hardware configuration using color-coded icons, showing device status as well as logical relationships, such as the peripherals connected to an I/O card. Each icon on the map represents a hardware device. These icons display the device type, device identifier, device path, last active tool, and test status (from last active tool). You can launch other STM tools from this view as well.

The Information tool provides product identifier information, product description, hardware path, vendor name, firmware revision, and error log statistics, including read errors, which

Figure 4-8 STM Configuration Mapper showing the latest status of the CPU and memory.

can be used to trend and anticipate problems. This tool also checks onboard log information, and can be used to track configuration changes.

Several other tools under STM perform varying levels of testing to stress a device or determine and diagnose problems:

- **Verifier tool:** Can be invoked on a particular device to verify quickly that it is connected and functioning properly.
- **Exerciser tool:** Stresses a device, to help reproduce and troubleshoot intermittent problems by stressing the hardware to the maximum point expected in a customer environment.
- **Diagnose tools:** Perform a complete test of the hardware, to help isolate failures down to the component or FRU level.
- **Expert tools:** Are sophisticated troubleshooting tools for expert users.
- **Logtool tool:** Helps you to format, filter, and extract error information from raw data contained in system logs. You can monitor recoverable errors detected by the computer, such as I/O device errors. This data can be used to troubleshoot and trend historical information, so that you can fix failures before they become critical. The errors that you see here are automatically forwarded to the EMS Hardware Monitors, which generate an event if an error is serious enough.
- **Firmware Update tools:** Provide a customer-usable way to update the firmware on hardware devices.

STM enables an operator to run a module on several devices simultaneously. In addition, the operator can start diagnostic tests running on more than one system from within the user interface.

STM provides both configuration and fault monitoring capabilities for the system. STM tools detect the same errors as the EMS Hardware Monitors, but the EMS Hardware Monitors report them in real time. After getting an EMS event, you can run STM to further diagnose a problem.

STM is used to diagnose local or remote systems. It is available on HP-UX releases 10.01 and later. STM replaces the Sherlock diagnostics. The software (product number is B4708AA) is being distributed on the HP-UX Diagnostic/IPR Media.

HP Predictive Support

HP Predictive Support detects and predicts system-related faults. When problem conditions are detected, notification is sent to the HP Response Center. This level of care is meant for customers with special support contracts with Hewlett-Packard. The Predictive Support software proactively monitors the system and automatically reports information back to the HP Response Center via modem access. Because the HP Response Center is available 24 hours a day, 7 days a week, this procedure can lead to a quick response to problems.

The Predictive Support software focuses on system event information for memory and I/O devices. Error logs are analyzed daily, with potential problems diagnosed. By proactively warning

of potential problems, scheduled maintenance can replace the unplanned downtime associated with a failed component.

Predictive Support uses a set of rules on a managed node to determine when events should be sent to the HP Response Center. These conditions can be updated periodically by downloading new rules from HP. Event correlation ensures that duplicate messages are suppressed and that the Response Center is not repeatedly warned of the same root problem.

Predictive Support analyzes on-board logs, system logs, and memory logs. The software can automatically dial the HP Response Center to transmit error data and logs, or the system administrator can initiate modem transmission. Similarly, Predictive Support software updates, to include new rules for generating predictive events, can be triggered automatically or controlled by the administrator. Configuration and administration is controlled through a menu-driven interface.

System logs are scanned for I/O errors and LPMCs. Logged data is analyzed for trends associated with specific disk or tape devices, such as correctable errors. LPMC records are analyzed for internal cache errors. Memory logs are also scanned to look for error rates exceeding specified thresholds.

The Response Center determines where a failed device is located, its model number, its manufacturer, and its serial number, so that repairs can be made. This information is sent in the failure notification messages.

HP Predictive Support does not help with other areas of system monitoring, such as resource and performance management. Also, the software runs only on HP-UX systems.

HA Observatory

HA Observatory is a suite of tools used to detect and quickly diagnose system problems. The products include the Configuration Tracker, which keeps track of the server's software configuration, Network Node Manager, and HP Predictive Support. A support system and network router are also maintained at the customer site.

HA Observatory relies on HP Predictive Support to report hardware failures. In addition, configuration information collected by the Configuration Tracker is available. The Configuration Tracker generates and maintains a snapshot of the configuration so that it can detect software configuration changes.

HA Observatory uses a secure network link to HP's High Availability Support Center from a special system at the customer site. This support system, an HP 9000 Series 700 workstation, collects system configuration information from key servers and can be used to view network status and topology information. Hardware failure notifications and configuration information can be sent to HP. When permitted by the customer, HP support engineers can access the customer servers over the secure link to gather additional information.

HA Observatory is supported only on HP-UX systems and is available only to customers with BCS and CCS support contracts.

Monitoring System Peripherals

When monitoring your UNIX server, you should not forget about the system peripherals, such as disk, tape, and printer devices.

Disks

Disk devices store your corporate data, so ensuring their correct operation is critical. Disk monitoring should be included as part of any effort to monitor a system. Chapter 5 is dedicated to covering the tools for monitoring disk devices because of the importance and complexity of the subject.

Tapes

By checking for tape hardware failures periodically, problems can be found before the tape drive is needed. For example, early detection and repair can result in the evening backup application running without delays.

You should use hardware monitors, such as the EMS Hardware Monitors, to ensure that your tape drives are healthy, so that backups may occur as scheduled. As mentioned earlier, a variety of tape-related errors can be detected, such as failed I/O operations to the tape, tape loader errors, and tape changer errors.

Printers

This chapter discussed earlier how you can use MIB instrumentation to look for important system information. A Printer MIB specification typically is supported by modern network printer devices. The Printer MIB stores the status of the printer (online, offline, and so forth), as well as information about printer subcomponents, such as an empty printer tray. A tool such as IT/O can be used to enable monitoring of individual fields in the MIB, such as printer status.

Using the lpstat command, you can check the status of jobs in the print queue and verify that the print scheduler is running. Using lpstat with no options shows you information for all the printers connected to the system.

Some printer management software is also available. One example is HP JetAdmin software, which can be used to manage printers and monitor the status of jobs printing remotely. This host-based application provides real-time printer and job status information, as well as status updates, remote diagnostics and troubleshooting, an optional status log, and a remote front panel. The utilities are accessible through a graphical interface. Web JetAdmin software adds additional capabilities, such as the ability to view customized maps that show printer locations and status information from a Web browser.

Web JetAdmin 4 is an HP software application that is used to monitor all peripheral devices on the network from a Web browser. Maintenance and troubleshooting can be done on any MIB-compliant printer device, including ScanJet 5 scanners and HP SureStore CD-ROMs.

Collecting System Performance Data

Users call their IT department when they have delays in accessing data or applications. Good tools are needed to help an operator pinpoint the source of the problem. This section covers some of the interesting performance and resource-utilization metrics, and the tools available to collect data about these metrics.

A wide range of conditions may result in resource and performance problems. Running out of available memory may be caused by a failure of a memory component or by a memory leak in an application. A sudden rise in CPU utilization could be an indication of processor failure or the introduction on the system of a CPU-intensive application. Analysis is needed to determine whether resource problems can be fixed with a configuration change, hardware repair, or other techniques.

Many important system resources have configured limits. The following system resource metrics are important to monitor:

- Number of named pipes
- Number of messages and message queues
- Number of system semaphores
- Amount of shared memory
- Number of open files
- Number of processes

Earlier, this chapter discussed some of the tools that can be used to check system resource usage. The sar and sysdef commands can compare current usage to configured limits. An EMS monitor is available to detect thresholds being exceeded for the following resources:

- Callout table
- Process table
- File descriptor table
- File lock table
- Shared memory
- System semaphores
- Message queues and message segments

The performance tools discussed in this section can also detect resource usage problems.

Some system performance monitoring is available from the SAM Performance Monitors, with which an administrator can obtain information on system, disk, and virtual memory activity, for example. Text-based information is displayed in a Motif window when one of the desired metrics is selected.

Having historical information is important, to understand how the system performance has varied over time. Knowing how your system behaves under normal conditions helps when trying

to troubleshoot system performance problems. Note that the performance tools themselves impact the performance of the system, so you need to find a tool with low overhead.

This section describes some common tools for measuring and monitoring system performance. Here are some of the key metrics discussed in this section:

- **Buffer cache queue length:** Refers to the number of processes blocked that are waiting for updates to the buffer cache. If this value is high, it could be an indication of a memory bottleneck.
- **Context switches:** How often processes are being swapped out of the run queue.
- **CPU utilization:** Expressed as a percentage of time spent in various execution states. Low utilization indicates that the CPU spent the majority of its time in the idle state.
- **CPU run queue length:** The average number of processes in the run state waiting to be scheduled.
- **Memory utilization:** Usually expressed as a ratio of the amount of memory in use versus the total memory available.
- **Paging:** Refers to the transfer of data between virtual memory (disks) and physical memory.
- **Swapping:** Refers to the transfer of data between physical memory and a special virtual memory area reserved for swapping.

Performance tools, such as BMC PATROL and MeasureWare, don't always provide the same set of metrics on all platforms. For simplicity, this section focuses on the Sun Solaris and HP-UX platforms only. Also, these products are continually being enhanced, so the actual metrics available for use in your environment may not precisely match the information presented in this section.

MeasureWare

HP MeasureWare Agent is a Hewlett-Packard product that collects and logs resource and performance metrics. MeasureWare agents run and collect data on the individual server systems being monitored. agents exist for many platforms and operating systems, including HP-UX, Solaris, and AIX.

The MeasureWare agents collect data, summarize it, timestamp it, log it, and send alarms when appropriate. The agents collect and report on a wide variety of system resources, performance metrics, and user-defined data. The information can then be exported to spreadsheets or to performance analysis programs, such as PerfView. The data can be used by these programs to generate alarms to warn of potential performance problems. By using historical data, trends can be discovered. This can help address resource issues before they affect system performance.

MeasureWare agents collect data at three different levels: global system metrics, application, and process metrics. Global and application data is summarized at five-minute intervals, whereas process data is summarized at one-minute intervals. Important applications can be defined by an administrator by listing the processes that make up an application in a configuration file.

Table 4-4 Categories of MeasureWare Agent Information

Category	*Metric Type*
System	CPU, disk, networking, memory, process queue depths, user/process information, and summary information
Application	CPU, disk, memory, process count, average process wait states, and summary information
Process	CPU, disk, memory, average process wait states, overall process lifetime, and summary information
Transaction	Transaction count, average response time, distribution of response time metrics, and aborted transactions

The basic categories of MeasureWare data are listed in Table 4-4. Also included are optional modules for database and networking support. MeasureWare agents also collect data provided through the DSI interface.

The following lists the global system metrics that are available from MeasureWare on HP-UX and Sun Solaris. Additional metrics provided by MeasureWare are covered in other chapters.

- CPU use during interval
- Number and rate of physical disk inputs/outputs
- Maximum percent full of all disk file sets
- System CPU use during interval
- User CPU use during interval
- CPU use at nice priorities
- CPU idle time during interval
- Rate of system procedure calls during interval
- Main memory use
- Swap space use on disk
- Number and rate of memory page faults during interval
- Number of process swaps during interval
- Percentage of virtual memory currently in active use
- Number of processes in run queue during interval
- Number of processes waiting for a disk during interval
- Number of processes waiting for memory during interval
- Number of processes currently in sleep state during interval
- Number of processes waiting for other reasons during interval
- Number of user sessions during interval
- Number of processes alive during interval
- Number of processes active during interval
- Number of processes started during interval

- Number of processes completed during interval
- Average runtime of completing process during interval
- Operating system version
- Number of processors in the system
- Number of disk devices and their device IDs
- Main memory size
- Swapping space allocated
- Disk I/O information (see Chapter 5)
- Networking statistics (see Chapter 6)

Note that, in addition to performance metrics, MeasureWare provides useful configuration information, such as number of processors and the number of disk devices.

The following additional global system metrics are available on HP-UX:

- CPU use at real-time priorities
- CPU use for context switching during interval
- CPU use for interrupt handling during interval
- Number of processes waiting for interprocess communications during interval
- Number of processes waiting on network transfers during interval
- Number and rate of terminal transactions during interval
- Average terminal transaction "think" time
- Average terminal transaction first response time
- Average terminal response to prompt time
- Distribution of transaction first response times
- Distribution of transaction response to prompt times

You can have alarms sent based on conditions that involve a combination of metrics. For example, a CPU bottleneck alarm can be based on the CPU use and CPU run queue length.

MeasureWare agents provide these alarms to PerfView for analysis, and to the IT/O management console. SNMP traps can also be sent at the time a threshold condition is met. Automated actions can be taken, or the operator can choose to take a suggested action.

MeasureWare's extract command can be used to export data to other tools, such as spreadsheet programs. Additionally, Application Resource Measurement (ARM) APIs (described in detail in Chapter 7) can be used to instrument applications so that response times can be measured. The application response time information can be passed along to MeasureWare agents for analysis.

Although MeasureWare provides extensive performance and resource information, it provides limited configuration information and no data about system faults. For further information, visit the HP Resource and Performance Management Web site at http://www.openview.hp.com/solutions/application/.

GlancePlus

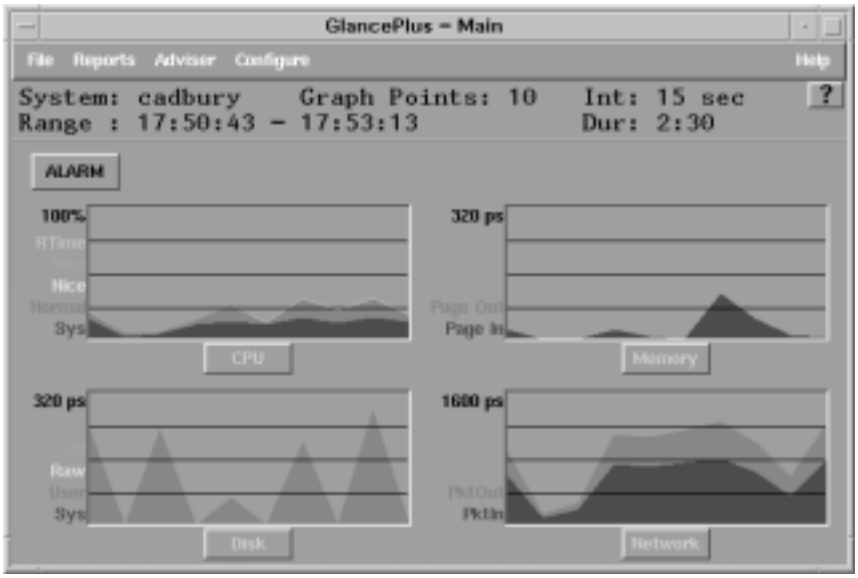
GlancePlus is a real-time, graphical performance monitoring tool from Hewlett-Packard. It is used to monitor the performance and system resource utilization of a single system. Both Motif-based and character-based interfaces are available. The product can be used on HP-UX, Sun Solaris, and many other operating systems.

GlancePlus collects information similar to the information collected by MeasureWare, and samples data more frequently than MeasureWare. GlancePlus can be used to graphically view the following:

- Current CPU, memory, swap, and disk activity and utilization (see Figure 4-9)
- Application and process information
- Transaction information, if the MeasureWare Agent is installed and active
- Alarm information, color-coded to reflect severity
- CPU utilization, with per-processor information available for multiprocessor systems
- Memory utilization, split among cache, user, and system memory
- Disk utilization, with the I/O paths of the top disk users indicated
- I/O activity, by filesystem or logical volume

GlancePlus is also capable of setting and receiving performance-related alarms. Customizable rules determine when a system performance problem should be sent as an alarm. The rules are managed by the GlancePlus Adviser. The Adviser menu gives you the option to Edit

Figure 4-9 The GlancePlus main screen showing system utilization.



Listing 4-13 Defining alarms in GlancePlus.

```
alarm CPU_Bottleneck > 50 for 2 minutes
  start
    if CPU_Bottleneck > 90 then
      red alert "CPU Bottleneck probability= ", CPU_Bottleneck, "%"
    else
      yellow alert "CPU Bottleneck probability= ", CPU_Bottleneck, "%"
  repeat every 10 minutes
    if CPU_Bottleneck > 90 then
      red alert "CPU Bottleneck probability= ", CPU_Bottleneck, "%"
    else
      yellow alert "CPU Bottleneck probability= ", CPU_Bottleneck, "%"
  end
    reset alert "End of CPU Bottleneck Alert"
```

Adviser Syntax. When you select this option, all the alarm conditions are shown, and you can then modify them.

Alarms result in onscreen notification, with the color representing the criticality of the alarm. An alarm can also trigger a command or script to be executed automatically. Instead of sending an alarm, GlancePlus can print messages or notify you by executing a UNIX command, such as mailx, using its EXEC feature.

To configure events, you need to edit a configuration file. The GlancePlus Adviser syntax file (/var/opt/perf/adviser.syntax) contains symptom and alarm configuration. Additional syntax files can also be used. A condition for an alarm to be sent can be based on rules involving different symptoms. Listing 4-13 shows an example of how you can set up an alarm for CPU bottlenecks that is based on CPU utilization and the size of the run queue.

You can also execute scripts in command mode. To execute a script, type:

```
glance -adviser_only --syntax <script file name>
```

In this example, a yellow alert is sent to the GlancePlus Alarm screen if a CPU bottleneck is suspected. As a bottleneck becomes more likely, the alarm changes to red. You can define the threshold for when the alarm should be sent. The symptoms are re-evaluated at every time interval.

Here is a sampling of some of the useful system metrics that can be monitored with GlancePlus:

- CPU utilization
- CPU run queue length
- Number of processors
- Filesystem buffer cache queue length
- Disk utilization and queue length
- Physical memory capacity

- Amount of physical memory available
- Memory page fault rate
- Total swap space
- Amount of swap space available
- Filesystem I/O rates
- Amount of buffer cache available
- Available shared memory
- Available file table entries
- Available process table entries
- Most active processes
- Wait states
- System table resources
- Open file information

More than 600 metrics are accessible from GlancePlus. Some of these metrics are discussed in other chapters. The complete list of metrics can be found by using the online help facility. This information can also be found in the directory /opt/perf/paperdocs/gp/C.

GlancePlus allows filters to be used to reduce the amount of information shown. For example, you can set up a filter in the Process view to show only the more active system processes.

GlancePlus can also show short-term historical information. When selected, the alarm buttons, visible on the main GlancePlus screen, show a history of alarms that have occurred.

GlancePlus also shows Process Resource Manager behavior, if PRM is installed, and allows the PRM process group entitlements to be changed.

For further information, visit the HP Resource and Performance Management Web site at http://www.openview.hp.com/solutions/application/.

PerfView

PerfView is a graphical performance analysis tool from Hewlett-Packard. It is used to graphically display performance and system resource utilization for one system or multiple systems simultaneously, so that comparisons can be made. A variety of performance graphs can be displayed. The graphs are based on data collected over a period of time, unlike the real-time graphs of GlancePlus. This tool runs on HP-UX or NT systems and works with data collected by MeasureWare agents.

PerfView has the following three main components:

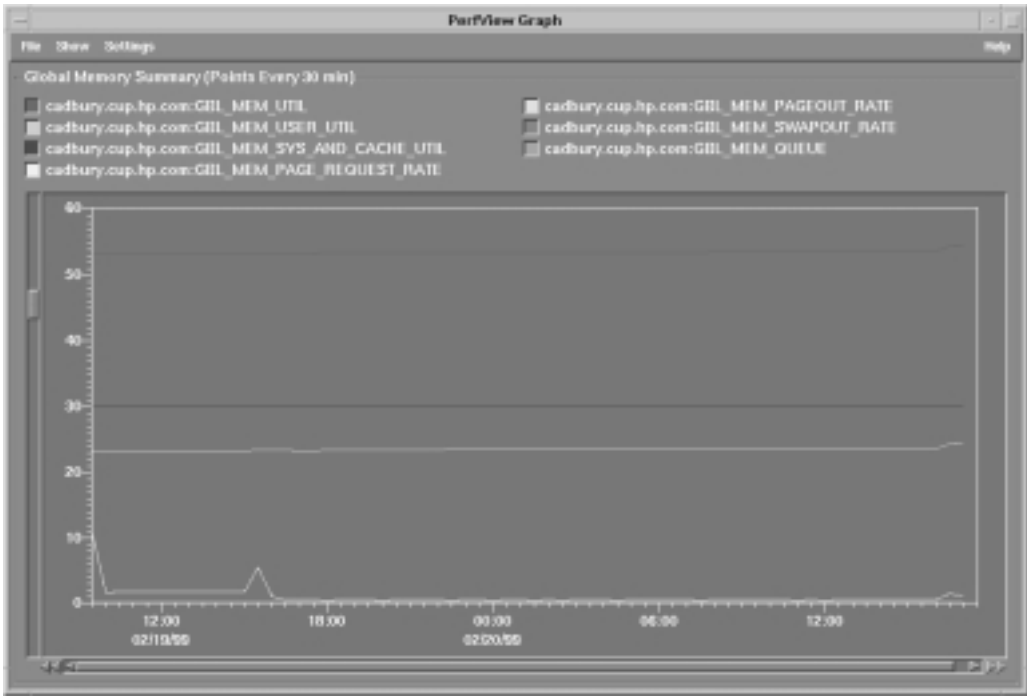
- **PerfView Monitor:** Provides the ability to receive alarms. A textual description of an alarm can be displayed. Alarms can be filtered by severity, type, or source system. Also, after an alarm is received, the alarm can be selected to display a graph of related metrics. An operator can monitor trends leading to failures and then take proactive actions to avoid problems. Graphs can be used for comparison between systems and to show a history of

resource consumption. An internal database is maintained that keeps a history of alarm notification messages.

- **PerfView Analyzer:** Provides resource and performance analyses for disks and other resources. System metrics can be shown at three different levels: process, application (configured by the user as a set of processes), and global system information. It relies on data received from MeasureWare agents on managed nodes. Data can be analyzed from up to eight systems concurrently. All MeasureWare data sources are supported. PerfView Analyzer is required by both PerfView Monitor and PerfView Planner.
- **PerfView Planner:** Provides forecasting capability. Graphs can be extrapolated into the future. A variety of graphs (such as linear, exponential, s-curve, and smoothed) can be shown for forecasted data.

PerfView can be used to monitor critical system resources. Figure 4-10 shows the PerfView Analyzer graphing memory utilization and paging rates. Other predefined graphs exist for history, CPU, memory, and queue information. For example, the history graph shows CPU, active processes, disk utilization, memory pageout rates, and swapout rates.

Figure 4-10 PerfView graph showing memory utilization and paging rates.



The PerfView Analyzer graph shown in Figure 4-11 compares the performance of two systems simultaneously. Up to eight systems can be compared in one graph. Comparing system utilization can be useful when determining where to deploy new applications, or when adding new users.

PerfView's ability to show history and trend information can be helpful in diagnosing system problems. Graphing performance information can help you to understand whether a persistent problem exists or if an anomaly is simply a momentary spike of activity.

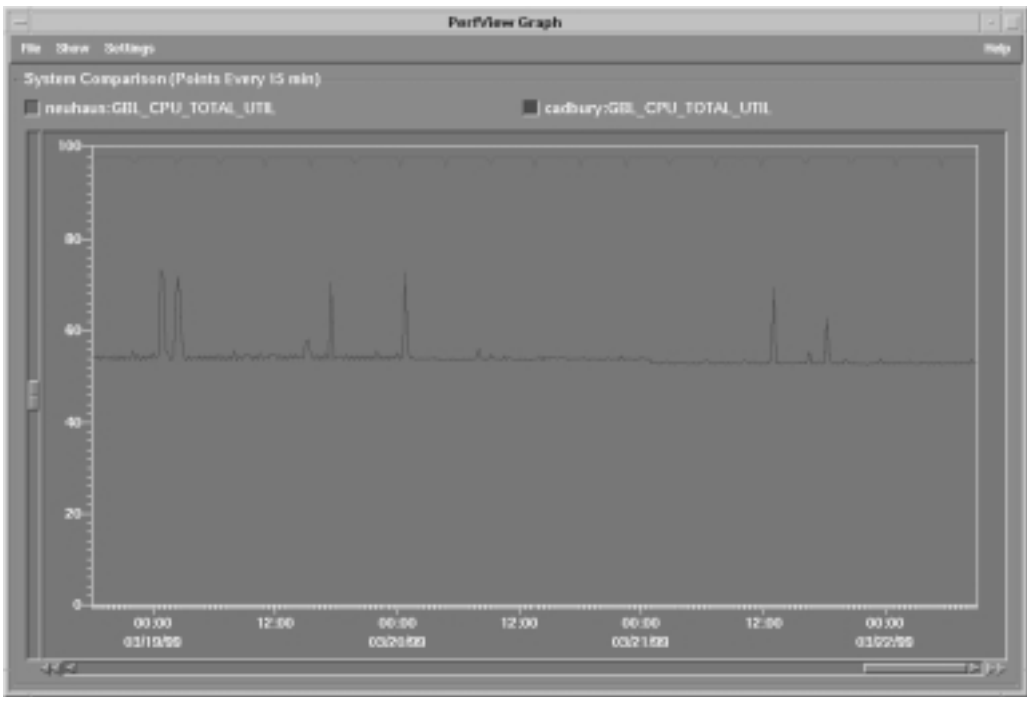
To diagnose a problem further, PerfView Monitor can allow users to change time intervals, to try to find the specific time a problem occurred. The graph is redrawn showing the new time period.

PerfView is integrated with several other monitoring tools. You can launch GlancePlus from within PerfView by accessing the Tools menu. PerfView can be launched from the IT/O Applications Bank as well. When troubleshooting an event in the IT/O Message Browser window, you can launch PerfView to see a related performance graph.

PerfView Monitor is not used with IT/O. Instead, the IT/O Message Browser is used. When an alarm is received in IT/O, the operator can click the alarm and a related PerfView graph can be shown.

PerfView can show information collected from multiple systems in a single performance graph. The PerfView and ClusterView products have also been integrated to enable the operator

Figure 4-11 PerfView graph comparing two systems.



to select a cluster symbol on an HP OpenView submap and launch the PerfView application. This quickly shows a performance comparison between all systems in the cluster.

For further information, visit the HP Resource and Performance Management Web site at http://www.openview.hp.com/solutions/application/.

BMC PATROL for UNIX

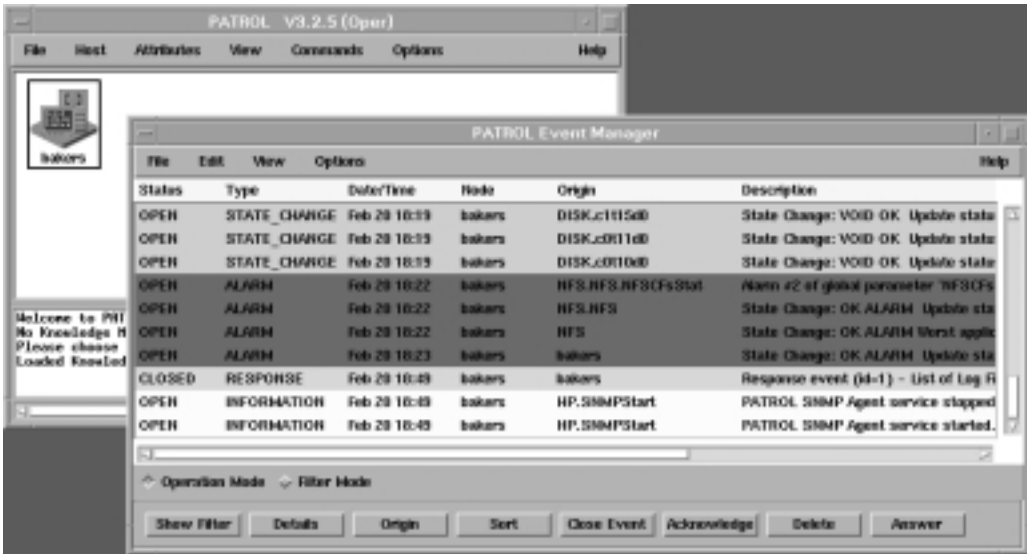
BMC Software provides monitoring capabilities through its PATROL software suite. PATROL is a system, application, and event management suite for system and database administrators. PATROL provides the basic framework for defining thresholds, sending and translating events, and so forth. Optional products, called Knowledge Modules (KMs), are capable of monitoring specific components. For example, BMC PATROL includes KMs for UNIX, SAP R/3, Oracle, Informix, and other applications. In fact, more than 40 KMs are available from BMC for use with PATROL.

With the PATROL KM for UNIX, managed components include the CPU, memory, users, kernel, processes, printers, security, and filesystems. These components are discovered automatically and represented on the PATROL console with status icons. System utilization can be shown as graphs, to capture trends, and data can either be displayed in real time or saved in log files.

Like other graphical monitoring tools, PATROL provides an Event Manager window, which can show received events. Figure 4-12 highlights disk and NFS events received at the console.

For memory and swap resources, PATROL can show total real memory available, total virtual memory available, a list of swap devices, the number of processes swapped, and swap space utilization.

Figure 4-12 PATROL Event Manager showing disk and NFS events.



For the CPU, PATROL can show bottlenecks and utilization information, along with a variety of statistics, such as CPU idle time, run queue length, and swap queue length. Information about the operating system itself is also maintained, such as the name, version, and creation date.

PATROL can display the total number of processes, the number of zombie processes, and heavy CPU users. Through the PATROL console, you can perform administrative tasks, such as reprioritizing processes.

PATROL also can display the total number of users and sessions, and can check security by monitoring the number of failed user and privileged logins. You can check the printer queue to see how many jobs are in the queue and to determine the state of the printer.

PATROL can monitor the filesystem and can automatically determine the effectiveness of the buffer cache. Regular reports can be generated to check disk usage per user, to create a list of the largest files, or to list files that have not been accessed in a long time. Corrective actions, such as removing core files, can also be configured.

In addition to the system metrics monitored by PATROL, the KM for UNIX includes a set of tools to provide additional system monitoring, including tools to monitor CPU usage, paging activity, I/O caching, swap activity, and system log files, tools to check filesystem and kernel file resources, and tools to monitor printer queues.

The following list shows some of the parameters available for monitoring from the PATROL KM for UNIX:

- `CPUCpuUtil`
- `CPUIdleTime`
- `CPUInt`
- `CPULoad`
- `CPUProcsWaiting`
- `CPUProcSwch`
- `CPURunQSize`
- `CPUSysTime`
- `CPUUserTime`
- `KERSysCall`
- `MEMActiveVirPage`
- `MEMFreeMem`
- `MEMPageAnticipated`
- `MEMPageFreed`
- `MEMPageIn`
- `MEMPageOut`
- `MEMPageScanned`
- `PRNQlength`
- `PROCAvgUsrProc`

- PROCCpuHogs
- PROCNoZombies
- PROCNumProcs
- PROCProcWait
- PROCUserProcs
- SWPSwapFreeSpace
- SWPSwapIn
- SWPSwapOut
- SWPSwapSize
- SWPSwapUsedPercent
- USRNoSession
- USRNoUser

The BMC PATROL KM for UNIX is supported on Bull, DG AViiON, DEC Alpha, DEC Ultra, Hewlett-Packard, NCR, Olivetti, OSF/1, Pyramid, RS/6000, SCO, Sequent, SGI, Sun Solaris, SunOS, Unisys, and UNIXWare systems.

Candle

The Candle Corporation provides software for mainframes and distributed systems. The Availability Command Center is a suite of integrated performance monitors and availability management solutions. The Candle Command Center for Distributed Systems is used to manage the performance and availability of computer systems and applications. Command Center solutions are available for UNIX, NT, IBM AIX, and MVS platforms. The Command Center for Distributed Systems can monitor many systems from a single console.

Candle's management agents provide detailed performance and availability metrics. The OMEGAMON Monitoring Agent for UNIX provides system information standardized across multiple UNIX platforms (IBM AIX, HP-UX, Sun Solaris, and SunOS). Available metrics include OS and CPU performance, process status, and disk performance. Disk performance is expressed as kilobytes per second, percent busy, and transfers per second. Disk performance and other tools can be launched from the Command Center console.

The Command Center provides some predefined threshold conditions for sending alerts. You also can change these conditions. If you decide to change the threshold conditions, they are automatically redistributed to the appropriate systems. Different alarm severity levels can be used.

The Command Center's event correlation engine and Visual Policy Editor can be used to create rules that automatically recognize the symptoms of problems and develop automated responses.

Candle has performed additional testing of the Command Center with MC/ServiceGuard to ensure that its Command Center for Distributed Systems product runs in that environment. More information about Candle Corporation's products can be found on the Web at http://www.candle.com.

Using System Performance Data

This section provides a brief introduction on how you can use performance monitoring tools to avoid, identify, and address system performance problems. An extensive tutorial on system performance is beyond the scope of this book.

A *bottleneck* in one system resource can render other system resources unusable. You need to ensure that all system components have sufficient capacity to operate at their optimal level. You can use performance data to avoid bottlenecks, by detecting trends to establish appropriate resource entitlements for each application, and to help eliminate problems when they occur.

This section does not discuss how to troubleshoot network performance issues, which is covered in Chapter 6.

Note that performance monitoring itself can create problems in your environment. Sending regular performance data from each system to a central location could result in hundreds of megabytes per day of network traffic and data storage for medium-sized companies. You should make sure that all the data you are collecting is going to be used. Instead of sending all data, you should send only the unusual or exceptional information. However, enough data should be sent to be able to identify trends for capacity planning. You should store a fixed amount of detailed performance data locally on each system so that you can troubleshoot problems when they appear.

Avoiding Performance Issues

The first step that you can take toward avoiding performance problems is to establish baselines for your environment. Collect performance data when your system is performing well, for long enough to get a valid representation of your system's workload, so that you have something to contrast with a poorly performing system.

Next, you should see whether the CPU, memory, and I/O resources are well-balanced. You should also do capacity planning, to ensure that your system has sufficient headroom to support any additional users and applications that you may be expecting. If excess capacity is not available, you should develop a plan for addressing future growth.

Another area to check is the allocation of system resources. Use the sar and sysdef commands, for example, to see whether any resources are at their configured limits. Check the available swap space and entries in the file and process tables to see whether these are sized appropriately. Use EMS to set up early warnings as the usage of other system resources increases. Because changing these limits often requires that you restart the system, early detection can allow you to plan for the time when the system will be unavailable.

Another way to protect system resources is to use the Process Resource Manager (PRM), a resource management tool used to balance system resources among PRM groups. PRM groups are configured by the administrator and consist of a set of HP-UX users or applications. PRM is then used to give each PRM group a certain percentage of the CPU, real memory, or disk I/O bandwidth available on the system. PRM ensures that each PRM group gets a minimum percentage of the system's resources, even during heavy loads.

PRM can be used in conjunction with HP GlancePlus to adjust system configuration. For example, if an administrator detects unwanted system load for a PRM group, GlancePlus can be used to lower that group's entitlement dynamically.

Normally, if one PRM group doesn't need its system resources, PRM allocates them to other groups that may need them. However, PRM can also help with capacity planning, by allowing resource maximums to be specified. Thus, if an administrator knows that a system will soon have 25 percent more users, the administrator can allocate a maximum of 80 percent of system resources to simulate the upcoming load.

Although PRM can ensure that users get a certain percentage of CPU resources, it can't prevent all system performance problems. For example, an application sending large network packets but using very little CPU resources can starve a more critical application, because network bandwidth is not controlled by PRM.

PRM can also be used to adjust workload dynamically in a high availability environment. For example, if three MC/ServiceGuard packages are each running with similar PRM entitlements, and one package fails to another system, this can be automatically detected, and a new PRM configuration can be applied, giving the two remaining packages higher entitlements.

Despite these efforts, you still are likely to have some performance problems to investigate. The next sections describe how to use the data collected by the various performance monitoring tools to address performance issues.

Detecting CPU Contention

UNIX commands, such as top and uptime, and performance monitoring tools, such as GlancePlus, provide CPU utilization information. CPU utilization and run queue length can be used together to determine whether a CPU bottleneck exists. High CPU utilization alone may not be indicative of a problem; batch jobs may be consuming the CPU remaining from interactive users. However, if interactive users are getting poor response times, that indicates a problem, such as a system bottleneck.

If the run queue is greater than one, the likelihood that a CPU bottleneck exists increases as the CPU utilization gets closer to 100 percent. Make sure that the high utilization and large run queue are sustained for a period of time.

If a CPU bottleneck is identified, recovery may depend on the applications and processes consuming large amounts of CPU. This can be determined by using performance monitoring tools such as GlancePlus. Applications spending the majority of their time in system code may need to be changed. In some cases, an application can be recompiled, optimized, or restructured to improve its performance. If batch processing is causing a problem, a job scheduler can be used to route jobs to less utilized systems. Less important applications, such as batch processes, can also be reconfigured to run at a lower priority by using nice. An application may need to be aborted or moved to another system if it continually thrashes with other applications. Tools such as PRM can be enabled or reconfigured to handle resource allocation among applications or users. PRM can keep applications within configured CPU limits.

Checking System Resource Usage

This chapter has described a variety of tools to monitor system resource usage. System table utilization can be checked by using tools such as GlancePlus. Using EMS monitors to set up thresholds is another useful approach.

The number of processes allowed and the number of concurrent open files allowed are two parameters that should be checked and that can be reconfigured using SAM.

Many actions to correct this type of problem require restarting the system, but if the problem is due to a runaway application, you may be able to detect the problem before other applications are affected. You can abort the application before system resources are depleted.

Detecting Memory and Swap Contention

To check for a real memory bottleneck on the network server, you can first check the amount of free memory. It should not drop below 5 percent of the total available. If the system cannot keep up with the demands for memory, it will start paging and swapping. Excessive paging and swapping, viewed from GlancePlus, may be a sign of a memory bottleneck. Two other signs that may indicate a memory bottleneck are a high percentage of processes blocked on virtual memory and large disk queues on swap devices.

To lower the swap rate, you may want to configure a higher percentage of available disk space for swapping. Increasing the capacity of the system by adding more memory or disk space may also eliminate the bottleneck. PRM can be used to ensure that the most important applications get a sufficient percentage of the memory.

If the amount of memory being used seems unusually high, you can use performance tools to determine which processes are using the most memory. A program may need to be redesigned to use memory more efficiently. A program may also need to be examined for memory leaks.

Detecting Disk and File System Bottlenecks

System, application, and disk information should be studied together to resolve disk performance issues. MeasureWare provides a lot of information about an application's disk utilization, which may need to be correlated with system data.

To avoid disk bottlenecks, you need to balance I/O across filesystems, disk spindles, and disk controllers to reduce uneven queuing and delays. Performance monitoring tools such as GlancePlus can be used to find the process with the highest I/O rate, and also the busiest physical disk. Checking the I/O rate only is insufficient, because a slower device has a higher utilization than a faster disk with the same I/O rate. If a single disk has greater than 50-percent utilization for an extended period of time, it may be an indication of an I/O bottleneck. The percentage should be compared with that of other disks, to see whether a severe load imbalance exists. However, a high utilization is not sufficient to identify a problem. The disk may still be capable of handling more I/O. A continually long disk queue length is also needed to indicate a problem. Heavily used disks are likely to have large disk queue lengths as well.

Both BMC PATROL and MeasureWare collect read cache hit ratio information. Determining how many logical reads are satisfied by the system's buffer cache is an indication of whether the cache size was configured correctly. Because increasing the cache size negatively affects the system memory available for other purposes, the appropriate cache hit ratio depends on the type of workload being run on the system. For I/O-intensive applications, you may want to configure your system such that this ratio is as high as 90 or 95 percent. Similarly, you may want to ensure that your write cache hit ratio is at least 75 percent. If your hit rates are too low, the system buffer cache may be too small.

After you determine that the system buffer cache is too small, you can increase its size on HP-UX by using SAM. Select the Configurable Parameters option from the Kernel Configuration functional area. The appropriate parameter to modify depends on whether a static or dynamic buffer cache is being used, which can also be checked on this screen. Fixed-size buffer caches are most effective if the environment and workload are static. Dynamic buffer caches fluctuate in size based on the demands for I/O or virtual memory, and are useful when workloads vary. If the nbuf and bufpages system parameters are set to 0, a dynamic buffer cache is in use. When using a dynamic buffer cache on systems with greater than 1GB of real memory, you should lower the maximum size below 50 percent, because caches greater than 500MB actually cause performance degradations.

Detecting disk contention is discussed in Chapter 5. If no problem seems to exist with the CPU, memory, or disk, other possibilities include networking or system table utilization. Checking network utilization is discussed in Chapter 6.

Avoiding System Problems

To avoid system problems related to misconfigurations, you need to have appropriate product documentation and business policies in place. The administrators making the changes should have access to caveats and a history log of past changes. Changes should be logged and a revision control system should be used so that you can quickly revert an old configuration.

System components will fail, but you can reduce consequential problems by investing in high availability or resiliency products and features. In the 1997 D. H. Brown survey of high availability providers, Hewlett-Packard was rated above average in its ability to detect and recover from failures. HP-UX provides dynamic memory resiliency, dynamic processor resiliency, and dynamically loadable kernel modules. Single-bit CPU cache errors can be corrected automatically. Memory Error–Correcting Code (ECC) and checksums reduce memory problems, but don't eliminate the need to monitor the memory subsystem. HP supports error thresholds for memory and disks, and its Memory Page Deallocation feature enables dynamic memory deselection for failing memory locations.

As vendors improve the resiliency of their operating systems, CPU failures become less likely to cause a system to fail. In some cases, if diagnostic tools detect a problem with a CPU, the processor can be deallocated while the operating system continues to run. For example, this

can be done if the rate of corrected single-bit CPU cache errors exceeds a predefined threshold. The processor can also be deallocated if a problem is found in the self-test during boot.

For companies with HP support contracts, HP Predictive Support can be used to detect trends that might lead to system problems. An engineer can then be sent to the customer site to make repairs before a problem becomes serious.

You also must back up your data regularly to prepare for any problems. The backups should be tested regularly to ensure that they are working properly.

You should also try to avoid performance and resource management problems by closely monitoring how your system is being used. Techniques for accomplishing this are described in the previous section.

Recovering from System Problems

When a server fails and can't be immediately repaired, high availability cluster software (such as MC/ServiceGuard) can be used to reduce the downtime associated with the situation and to keep services available. MC/ServiceGuard detects the failure of an application and automatically restarts the application on another system. This automatic detection and recovery can save you downtime. MC/ServiceGuard can detect a failure and restart an application on another system in under one minute.

However, even with the kernel's capability to mask certain failures and high availability software's capability to move applications to redundant servers, ultimately you still need to repair the failed components. For hardware problems, Support Tool Manager can provide fast diagnosis on HP-UX systems. SyMON can be used for Solaris environments.

You may need to find a software or firmware patch to fix your problem. For HP-UX, you can obtain patch information from the HP Web site by following links to support information. Customized patch bundles are available for customers with HP software support agreements. For Sun Solaris, you can access patch information over the Web through a service called SunSolve Online.

Recovering from a security violation by a malicious intruder may be more difficult. The system administrator may need to revert to system backup tapes from a known good system state.

Comparing System Monitoring Tools

This section provides a brief summary and comparison of the tools discussed in this chapter. The summary is organized by the key focus areas of system monitoring: configuration, faults, and resource and performance management.

Limited tools are available for monitoring configuration changes. The burden is largely on you to run tools such as iostat and STM to gather configuration data, and then to store that data in your own defined areas for comparison purposes later. The DMI Software MIF contains some software configuration information, but DMI management tools are not yet available. The Configuration Tracker is one tool that is available for monitoring software configuration changes.

ClusterView can track changes made to high availability cluster configurations. More tools are needed that can track changes to hardware and software configuration information. This will be more important as it becomes easier to add or remove components while a system stays active.

System component failures can be reported through EMS for HP-UX, and SyMON for Sun Solaris systems. These products, with their automatic failure notifications, are preferred to using tools manually to probe the state of hardware components. Hardware monitors and monitor developer's kits are available for both products. Both can send notifications by using a variety of methods, including SNMP, but EMS has tighter integration with NNM and IT/O. SyMON provides integrated recovery actions with its product. Note that detecting failures of the system requires a remote monitoring tool, such as NNM.

If you don't have access to a performance management product, some low-budget performance monitoring can be done with UNIX commands, principally iostat, vmstat, top, uptime, sar, and swapinfo.

GlancePlus provides performance monitoring in real time for a large number of metrics on a single system. SyMON, by contrast, has very little performance data. For historical performance data, sampled over a longer time period, you should use a tool such as MeasureWare or BMC PATROL. If you want the tool to be integrated with a performance management product, then you may want to use MeasureWare, which provides tight integration with PerfView. Both products can report information via SNMP.

Some amount of resource monitoring can be done with performance monitoring tools or EMS. Additional resource usage data is available from ipcs, sysdef, and the HP-UNIX MIB.

Case Study: Recovering from Memory Faults

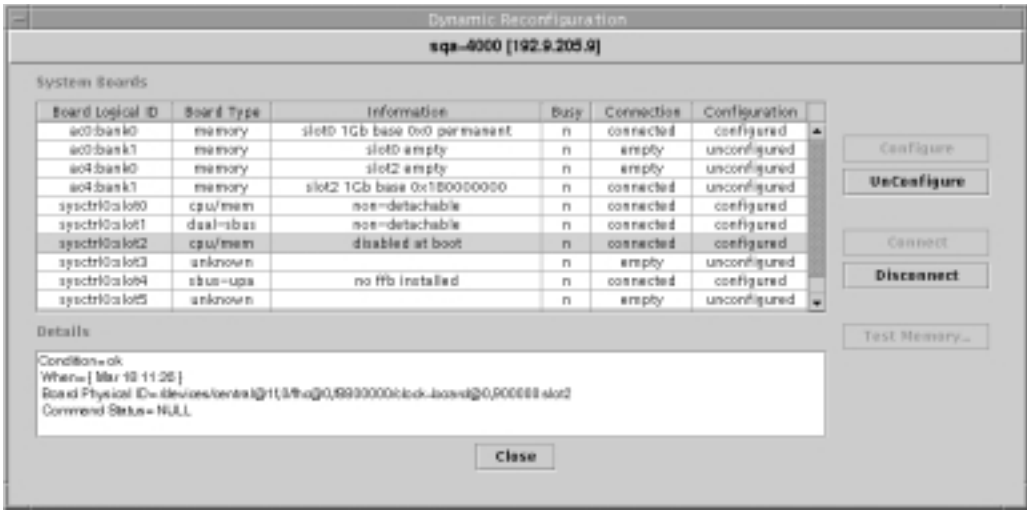
In this case study, the system administrator, Bill Landis, is responsible for maintaining system availability 24 hours a day, 7 days a week for the Silicon Valley Hospital's billing system. Based on past experience, he wants a system for which system components, such as CPU and memory, can be replaced without the need to bring down the system. Bill has a Sun Enterprise 10000 server, which has dynamic reconfiguration capabilities. To take advantage of Sun's dynamic reconfiguration, Bill is configuring his system memory so that it can be taken offline and replaced in the event of a board failure.

Verifying Configuration

The system's real memory is divided into memory banks, which become ineligible for dynamic reconfiguration when they contain kernel pages. With Sun's dynamic reconfiguration features, you can configure kernel pages to use certain memory banks. Once configured, you use the Dynamic Reconfiguration screen (shown in Figure 4-13) in SyMON to verify that these banks aren't "permanent" and are available to be unconfigured.

As Figure 4-13 indicates, the memory board in slot 2 is configured, but it isn't assigned to a permanent memory bank. As a result, Bill can use this screen in SyMON to take the memory

Figure 4-13 Using SyMON dynamic reconfiguration to replace a failed memory board.



offline. In contrast, slot 0 is associated with a permanent memory bank and can't be disconnected while the system is running.

Setting Up Monitoring and Reconfiguration

From the Enterprise SyMON console, Bill loads the Config-Reader and Dynamic Reconfiguration modules to ensure that he will be notified of hardware faults, so that he can handle the faults without having to take the system down. The Config-Reader module is located under Hardware and the Dynamic Reconfiguration module is located under local applications (shown in Figure 4-7, earlier in this chapter).

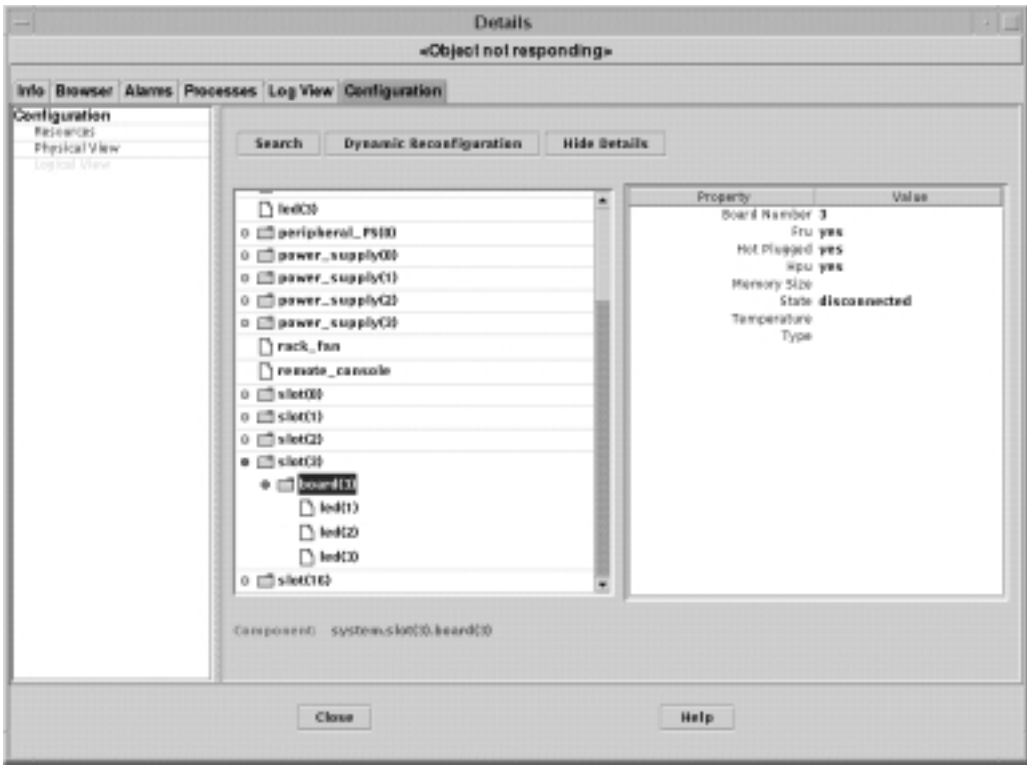
Memory Board Failure Occurs

When a critical memory fault occurs, the icon for the system on the SyMON console indicates the alarm. Bill looks at the Alarm window to see more details about the event. He notices that a memory board has failed. Using the Logical View, like the one shown in Figure 4-14, he locates the failed memory board. Using the Physical View, like the one shown earlier in Figure 4-5, he locates the exact location of the physical board in the system.

Fixing the Failure and Restoring Service

Bill accesses the Dynamic Reconfiguration screen from the SyMON console. First, he selects the failed memory slot and clicks the Disconnect button to unconfigure and disconnect the board. Next, he replaces the failed board and then connects the board by clicking the Connect button; he leaves it temporarily unconfigured, however, while he performs a memory test using

Figure 4-14 Using the SyMON Logical View to locate a failed memory board.



the Test Memory button, to ensure that the new board is functional. Finally, Bill clicks the Configure button to make these memory resources available to the system.

Bill was able to handle a failed memory board in this environment with very little impact to the system. Sun's dynamic reconfiguration capabilities, available from SyMON, provide a powerful feature that allows failed memory, CPU, and I/O resources to be fixed without having to bring down the system.