CMOST 7.3 Release Notes

Part I: Overview

1 Introduction

CMOST Version 7.3 provides partial support for the CM-5E and full support for the SPARCstation 10 as a control processor. New I/O products supported are the Integrated Tape System (ITS) and the CM5-HIPPI, which use the new CMIX operating system.

This version of CMOST also supports a new job management system, DJM, and several enhancements to the PN kernel.

2 Hardware and Software Requirements

2.1 Hardware Required

CMOST Version 7.3 runs on all models of the CM-5, including the CM-5E, CM-5 Scale 3, and CM-5 Scale 3 with CM-5E components.

CMOST Version 7.3, if loaded with all its subproducts, requires approximately 402 Mbytes of disk space.

Copyright © 1994 Thinking Machines Corporation

2.2 Software Required

CMOST Version 7.3 is based on SunOS Version 4.1.3 (the previous CMOST release, Version 7.2, was based on SunOS Version 4.1.2). As such, all Sun hardware in the CM-5 system (CPs, DPs, etc.) should be upgraded to/installed with SunOS 4.1.3.

Note that CMOST Version 7.3 uses a new version of DPAS, Version 1.1. Appendix D contains the release notes for DPAS Version 1.1.

2.3 Software Supported

Versions of CM-5 software compatible with CMOST Version 7.3 systems are as follows:

- CMMD 3.2 or later
- CMRTS 8.0 or later
- CMF 2.1.1−2 or later
- PNDBX 1.2 Patch 6
- C* 7.1 Patch 2
- CMSSL 3.1 or later
- CM/AVS 1.1 Beta for 7.3, or later
- CMX11 1.6
- Prism 2.0 Beta and Prism 1.2, or later
- CMAX 1.0
- CMview 1.0 or CMview 1.1
- *Lisp 7.6

Copyright © 1994 Thinking Machines Corporation

3 Porting Information

3.1 From Earlier Versions of CMOST

Users must relink with the new CMNA library.

Also, to take advantage of the performance enhancements made to the CMFS library for this release (see Section 4.2.1), users must relink their code.

4 New Features

4.1 New Functionality

4.1.1 Support for New Hardware

SPARCstation 10 Support

CMOST Version 7.3 includes support that enables the SPARCstation 10 (model 40) to be a CM-5 control processor (acting as a partition manager, IOCP, system administration console, etc.).

CM-5E Support

CMOST 7.3 features support for the CM-5s that contain CM-5E SuperSPARC processors, new NIs, and new Data Network (DR) hardware.

Any particular CM-5 may be upgraded in stages to CM-5E status; CMOST 7.3 supports partially upgraded machines. As such, the machine may contain a mixture of new and standard components, within certain boundaries; see Table 3 in Section 11.6.

New I/O Hardware

CMOST Version 7.3 supports the Integrated Tape System (ITS) and the CM5-HIPPI. The nodes that control and make up the ITS and CM5-HIPPI run an operating system called CMIX. These new Thinking Machines hardware products and their software support are described in the *CM5-HIPPI User's Guide* and the *CM5-HIPPI System Administrator's Guide*,

Copyright © 1994 Thinking Machines Corporation

and the *CM-5 ITS User's Guide* and the *CM-5 ITS System Administrator's Guide*.

4.1.2 Support for New Software

I/O

Parallel I/O operations (file ↔ PNs) on network sockets is now supported.

DJM

CMOST Version 7.3 supports DJM (Distributed Job Management). CMOST DJM is based on Minnesota Supercomputing Center's DJM Version 0.9.14. (To run MSC's version of DJM rather than the CMOST version, recompile the MSC sources against the CMOST Version 7.3 header files and libraries.)

DJM is designed to

- manage the flow of jobs through a Connection Machine CM-5 system
- avoid resource conflicts
- provide a load balancing capability among the various control processors for a Connection Machine CM-5

To balance system load, DJM handles all application processes running on the CM-5 — interactive jobs as well as batch jobs. It allows jobs for which there are sufficient resources to execute immediately; it queues all other jobs for later execution.

- Interactive jobs execute "in the foreground" and allow terminal input and output.
- Batch jobs execute "in the background." They allow input and output only to specified files.

More information on the use and administration of DJM is described in Appendix C of these release notes and in the manual *Using DJM on the CM-5*.

Copyright © 1994 Thinking Machines Corporation

PN Kernel

Several enhancements to the PN kernel have been made. In particular,

- Profiling is now supported.
- User alarms are now fully functional.
- Global asynchronous interrupt capabilities are now provided.
- The C library routines **_setjmp** and **_longjmp** can now be called by users.

These and other PN kernel enhancements are described in Sections 10.1 and 11.1.

4.2 Performance Enhancements

4.2.1 CMFS Library

The performance of the CMFS library, for small parallel I/O operations to or from SDA files, has been increased by a factor of approximately twenty. Such code will see this performance enhancement if it is relinked with the CMFS library.

4.2.2 CM-5E

To utilize the higher clock speed available on CM-5E (new NI, new PN) partitions, after executing **cmreset**, execute a **cmdiag**-shell command, as shown below:

1. Execute **cmreset** from the system administration console, as usual:

```
syscon# cmreset
```

2. Execute the **cmdiag** command to enter the **cmdiag** shell:

```
syscon# cmdiag
<CM-DIAG>
```

3. At the **<CM-DIAG>** prompt, execute the following command:

```
<CM-DIAG> fm e 40
```


Copyright © 1994 Thinking Machines Corporation

4. Exit the **cmdiag** environment:

```
<CM-DIAG> exit
syscon#
```

5. Continue with the documented bringup procedure (execute **cmreset -s** on each CP and IOCP, etc.).

The **fm e 40** **cmdiag**-shell command sets the clock speed to 40 MHz.

As a side effect, the **fm e 40** command might cause some PNs to crash, with **find-cm-error** showing IU errors; there is no need to intervene if this happens because the PNs will be rebooted (and the IU errors cleared) when **cmpartition start** is executed.

(Note to CM-5 system administrators: A recent technical bulletin, *CM5_hw_019_PN_clock_on_CLKDN* (January 12, 1994), asked you to execute **fm d 32** from the **cmdiag** shell after executing **cmreset** on the machine. It is no longer necessary to execute this **cmdiag**-shell command at system bringup because **cmreset** now sets the correct clock source automatically for CM-5 (non-CM-5E) systems.)

5 Changes from Previous Versions

5.1 Changes to Interface

The behavior of some **ts-daemon** switches has changed. See Section 11.3.

5.2 Obsolete Features

Do not use the **sdafs** command or the **sdafsmk** command. These commands are obsolete and will be removed from CMOST.

Do not use the routine **CMOS_get_time**. Instead, use the new routine **CMOS_get_wall_time** (see Section 10.1).

Copyright © 1994 Thinking Machines Corporation

6 Limitations and Restrictions

6.1 Partial Support for CM-5E

CMOST Version 7.3 provides only partial support for the CM-5E. Support for the following CM-5E features is forthcoming:

- MNI, the SPARCstation 10's M-bus network interface board
- long messages, both within a partition and between a partition and its PM

6.2 Swapping Not Supported

Swapping — context-switching in a new process, with the SDA storing the memory of a non-running process — is not supported at Version 7.3.

7 Documentation for This Release

CMOST and I/O documentation available with CMOST 7.3 include:

- User's guides:
  - *CM-5 User's Guide*, CMOST Version 7.3.
  - *CM-5 I/O System Programming Guide*, CMOST Version 7.3
  - *ITS User's Guide*, CMOST Version 7.3
  - *CM5-HIPPI User's Guide*, CMOST Version 7.3
  - *Using DJM on the CM-5*, Version 2.0
  - *VU Programmer's Handbook*, CMOST Version 7.2
  - *DPEAC Reference Manual*, CMOST Version 7.1 (see also the *DPAS Version 1.1 Release Notes* in Appendix D)
- Administrator's guides:
  - *CM-5 System Administrator's Guide*, CMOST Version 7.3
  - *ITS Administrator's Guide*, CMOST Version 7.3

Copyright © 1994 Thinking Machines Corporation

- *CM5-HIPPI Administrator's Guide*, CMOST Version 7.3
- *CM-5 Field Service Guide*, CMOST Version 7.3
- *Using DJM on the CM-5*, Version 2.0

Manual pages are available in printed form as well as online.

Please note that some Version 7.3 manuals may lag the release.

8 Errors, Feedback, and Assistance

8.1 Bugupdate Files

For the list of fixed and outstanding bugs, please see the file **`cmost-7.3-final.1.bugupdate`** (in **`/usr/doc`** by default).

8.2 Request for Feedback

Thinking Machines Customer Support encourages customers to report errors in Connection Machine operation and to suggest improvements in our products.

When reporting an error, please provide as much information as possible to help us identify and correct the problem. A code example that failed to execute, a session transcript, the record of a backtrace, or other such information can greatly reduce the time it takes Thinking Machines to respond to the report.

Internet Mail: **`customer-support@think.com`**

uucp Mail: **`ames!think!customer-support`**

U.S. Mail: Thinking Machines Corporation
Customer Support
245 First Street
Cambridge, Massachusetts 02142-1264

Telephone: (617) 234-4000

Copyright © 1994 Thinking Machines Corporation

9 Installation and Customization Information

9.1 New SunOS

As previously noted, CMOST Version 7.3 is based on SunOS Version 4.1.3. The previous CMOST release, Version 7.2, was based on SunOS Version 4.1.2. Therefore, all Sun hardware in the CM-5 system (CPs, DPs, etc.) should be upgraded to, or installed with, SunOS Version 4.1.3.

9.2 Explicit Autosize Required

After upgrading to CMOST Version 7.3, perform an explicit autosize: execute **`cmdiag -Z`** from the system control processor before executing **`cmreset`** and partitioning the machine.

9.3 Loading the CPROOT Subproduct

The CPROOT subproduct on the CMOST Version 7.3 tape is bigger than it has been in previous CMOST versions — it now contains three different versions of **`vmunix`** rather than two, taking up approximately 10 Mbytes in total. Loading CPROOT will probably exceed the remaining capacity of the root file system on most control processors.

To work around this problem, installers should load the tape into an alternate directory with more capacity and then copy the required **`vmunix`** over to the root directory and then either rename it **`vmunix`**, or make a link to the correct **`vmunix`**.

Copyright © 1994 Thinking Machines Corporation

9.4 Correcting Parity on the SDA

After upgrading to CMOST Version 7.3, please verify the data on your SDA's parity disk by executing

```
syscon% sda_disk_util -p -d logical_device_name
```

If the output of this command line indicates bad parity, correct the parity by executing

```
syscon% sda_disk_util -c -d logical_device_name
```

9.5 The netstat Program

CMOST 7.3 requires a new version of the **netstat** and **w** programs, which were previously installed in **/usr/ucb**. The new versions are shipped in the system-specific **kvm** directory (**/usr/kvm**), but they are not correctly installed in **/usr/ucb**. (Running the old versions causes incorrect results.)

To install the **netstat** and **w** programs correctly, do the following:

```
# rm -f /usr/ucb/w /usr/ucb/netstat
# ln -s "/usr/kvm/w /usr/ucb/w
# ln -s "/usr/kvm/netstat /usr/ucb/netstat
```

9.6 The hardware.install File

Please ensure that the **/etc/cm/configuration/hardware.install** file matches the actual configuration of the CM-5 or CM-5E: The **cmsa-server** program now verifies the **hardware.install** entries against the hardware that it knows to be present in the machine, and if **hardware.install** is not up to date, **cmsa-server** will not start. For more information, see Section 11.6.2.

(Note that in CMOST versions previous to 7.3 Beta 2.7, an out-of-date **hardware.install** file would not always cause **cmsa-server** to fail to start. From CMOST Version 7.3 Beta 2.7 onward, however, **cmsa-server** does always care about the veracity of **hardware.install**.)

Copyright © 1994 Thinking Machines Corporation

Part II: Detailed Information about New and Changed Features

10 Information for Users

10.1 PN Kernel Enhancements

- The PN kernel now handles some user-caused errors more gracefully than before: Previous to CMOST Version 7.3, the PN kernel panicked if the timesharing daemon issued it a request to do something illegal, such as to copy data from a non-existent memory address (this scenerio could be caused by a user passing a bad argument to a system call). The PN kernel now handles such errors more gracefully by returning an error code to the timesharing daemon, which then decides how to handle the error.
- User mishandling of stacks no longer causes IU errors to be generated.
- Most bad traps generated by users no longer cause the PN kernel to panic.
- Users can now more seamlessly set up their own stacks; the following provisos are now in effect:
  - The user-defined stack pointer must always point to an area of memory mapped writable to the user.
  - The user-defined stack must provide space for CMOST to save register windows: CMOST assumes that the area between **`[%sp]`** and **`[%sp + 60]`**, inclusive, is available for this purpose.
  - CMOST does not automatically grow a user-defined stack (it does automatically grow a standard stack segment).

Copyright © 1994 Thinking Machines Corporation

- CMOST uses the value of the stack pointer itself to differentiate between a stack pointer that is part of the standard stack segment and one that is user-defined:
  - If the stack pointer is below the constant **USER_STACK_BOT**, currently defined in **cmsys/cm_add.h** to be **0xf4000000**, CMOST assumes that the stack is user-defined.
  - If the stack pointer's value is between **USER_STACK_BOT** (currently **0xf4000000**) and **USER_STACK_TOP** (currently **0xf8000000**), CMOST assumes that the stack is part of the standard stack segment.

- The CMNA timers now more accurately measure user busy time: They no longer charge as user time the PN kernel ↔ timesharing daemon communication required by a system call.
- The emulation of certain multiply instructions is now faster than before.
- A new system call, **CMOS_get_wall_time**, returns the number of seconds and microseconds that have elapsed since timesharing was brought up on the partition on which it is called. The syntax of **CMOS_get_wall_time** is

```
int
CMOS_get_wall_time(struct timeval *tp)
```

10.2 Interrupt Handlers

Users can now register an interrupt handler to be invoked when the global async line is set. See the **CMOS_signal**(3) manual page for more information.

10.3 User Alarms

The user alarms **CMOS_alarm**, **CMOS_ualarm**, **CMOS_setitimer**, and **CMOS_getitimer** are now fully functional. These functions work in conjunc-

Copyright © 1994 Thinking Machines Corporation

tion with **CMOS_signal** to set up a timer interrupt handler that is invoked whenever an alarm is signaled. For more information, see the **CMOS_signal** manual page and the individual alarm function manual pages.

10.4 The ST_FLUSH_WINDOWS Trap

The trap **ST_FLUSH_WINDOWS** is now fully functional. This trap enables users to use the C Library routines **_setjmp** and **_longjmp** (see the SunOS manual page **setjmp**(2)) and the new node-level system calls **CMOS_message_handler_longjmp**, **CMOS_timer_handler_longjmp**, and **CMOS_async_handler_longjmp** on the PNs.

Use **CMOS_message_handler_longjmp**, **CMOS_timer_handler_longjmp**, and **CMOS_async_handler_longjmp** to **longjmp** out of the message interrupt handler, timer interrupt handler, and async interrupt handler, respectively; these system calls each take the same arguments as the SunOS **_longjmp** system call (see **_longjmp**(2)).

10.5 Profiling

Users can now profile their PN codes. In support of this, profiled versions of all the **libcmna** libraries are now available. Also, the CMOST command **cmld** now supports building profiled CM-5 executables; see the **cmld** manual page.

10.6 The pndbx Program's Memory Access

The **pndbx** program can now access all vector-unit memory.

10.7 User Access to Tags

For messages to or between PNs, users can now use tags 4−9 in addition to 0−3. (Do not send messages with tags 4−7 to the CP; this will be addressed in a future release.)

Copyright © 1994 Thinking Machines Corporation

10.8 The cmcmp Program

A new user command, **cmcmp**, performs a byte-by-byte comparison of two files. For command syntax and other information, see the manual page.

10.9 The ptrace Program

CMOST now supports full **ptrace** capability on each PN: The Prism debugger can run as a separate process on each PN, timeshared with the process being debugged.

Prism code can call the following functions:

- **ptrace**
- **kill**
- **wait** (**waitpid**, **wait4**, **wait3**, and **wait**)

Each of these calls operates on the CM process that is being debugged.

10.10 CMIX Nodes (ITS and CM5-HIPPI)

- The **CM_read()** and **CM_write()** system calls are now supported. These routines allow C* users read or write with more fine control than is possible with the CMFS library's read and write routines. The **CM_read()** and **CM_write()** system calls also enable C* users to use the CM5-HIPPI.

  For more information, see Chapter 7 of the *CM-5 I/O System Programming Guide*; the *C* Release Notes*, Version 7.1.1; and the *CM5-HIPPI User's Guide*.

- Do not attempt to log in to a CMIX node that has crashed. Doing so may crash the entire CM-5 system.

- CMOST Version 7.3 provides ITS-only versions of some I/O commands:
  - **itsdd**: supports StorageTek 3480-type and EXABYTE drives
  - **itsdump**: supports StorateTek 3480-type and EXABYTE drives

Copyright © 1994 Thinking Machines Corporation

- **itsrestore**: supports StorateTek 3480-type and EXABYTE drives
- **itstar**: supports StorateTek 3480-type and EXABYTE drives
- **itsibmtape**: supports EXABYTE drives only

For more information, please see the *ITS User's Guide*.

10.11 Enhanced Parallel I/O Support

CMOST Version 7.3 supports parallel I/O operations on all UNIX file descriptors and network sockets. Parallel I/O operations use parallel system services such as the SDA or the CM5-HIPPI whenever possible, or transparently serialize data through the CP when parallel transport facilities are not available. Parallel processes on the same partition can communicate through the use of pipes, UNIX domain sockets, or Internet sockets, with all data remaining on the CM processors at all times.

CMOST now supports the following system calls:

- **CM_read()**
- **CM_write()**
- **CM_send()**
- **CM_sendto()**
- **CM_sendmsg()**
- **CM_recv()**
- **CM_recvfrom()**
- **CM_recvmsg()**

These newly supported system calls are documented in the *CM5-HIPPI User's Guide*.

10.12 New CMFS Library Routines

Three new CMFS library routines, **CMFS_get_cmio_busid**, **CMFS_set_cmio_busid**, and **CMFS_get_cmio_busids**, are concerned with

Copyright © 1994 Thinking Machines Corporation

the CMIO bus ID of the device on which the CMFS file (represented by the *fd* argument) resides:

- **CMFS_get_cmio_busid** returns the current CMIO bus ID of the file's device.
- **CMFS_set_cmio_busid** selects the CMIO bus that will be used in subsequent operations involving the given file descriptor.
- **CMFS_get_cmio_busids** gives a list of CMIO bus IDs that connect the given file to the device that called the routine.

Note that these routines do not operate with SDA or UNIX file descriptors. For syntax information, please see the **CMFS_get_cmio_busid**(3) manual page.

11 Information for System Administrators

11.1 PN Kernel Enhancements

- The **memory_status** command can now retrieve the actual bad address from VU-based memory controller chips that have single-bit errors logged.
- The PN kernel now sends a signal to the **ts-daemon** when it detects an excessive number (greater than 256) of single-bit memory errors being corrected by any one memory controller. The PN kernel's signal produces a **ts-daemon** log message urging the site administrator to extract the details of these errors by running the **memory_status** command.

Copyright © 1994 Thinking Machines Corporation

11.2 Tests

- The following CMOST tests are new:
  - **node_setjmp**: tests node-level **_setjmp**/**_longjmp** calls
  - **dispatch_test**: verifies that CMNA correctly dispatches functions
  - **user_async**: tests user asynchronous interrupt handler functionality
  - **get_time_test**: sanity checks low-level timer calls
- The following CMOST tests have been enhanced:
  - **user_alarm**: now checks **longjmp** functionality
  - **user_ints**: now checks **longjmp** functionality
  - **errors**: now includes a test to generate a bad trap

11.3 The ts-daemon Program: Switch Changes

11.3.1 Context Switching

The maximum time quantum for context switching, setable via **ts-daemon**'s **-q** switch, is now 15 seconds (formerly there was no maximum).

11.3.2 Router Reinjection

The **-R** switch no longer does anything. (In CMOST Version 7.2, **-R** turned off router reinjection — a feature that reduces timing fluctuations. At CMOST Version 7.3, router reinjection is on by default and because it must remain on, there is no way to turn it off.)

11.3.3 Parallel Networking

The **-M** *number_of_pages* switch allows you to set the number of 4096-byte pages to set aside for use by parallel networking. The default argument to **-M** is calculated by an algorithm (noted in the **ts-daemon** manual page) and is dependent on the number of PNs in the partition; see Table 1.

Copyright © 1994 Thinking Machines Corporation

Table 1. The `ts-daemon -M` switch's default argument.

Partition Size	Default Number of Pages
16 PNs	128
32 PNs	128
64 PNs	64
128 PNs	64
256 PNs	32
512 PNs	32
1024 PNs	32
2048 PNs	32

To turn off parallel networking, include **`-M 0`** on the **`ts-daemon`** command line.

11.4 The ts-daemon Program: New Switches

11.4.1 Verification

The new **`-V`** (capital V) switch allows a site to verify whether each user who registers with timesharing daemon does, in fact, have permission to run programs on the CM-5. The **`-V`** switch takes one argument, *verifier_script_pathname*, which is the pathname of the site-supplied script that does the verification. The script file is called with three arguments, supplied by **`ts-daemon`**:

script_filename *user's CM_ACCOUNT_ID* *user's uid*

The script must be executable. If it is not, the timesharing daemon prints out **`ts-daemon:`** *filename* **`is not an executable.`**

The executable is expected to return statuses via **`exit`**:

`0`: Valid (the process is allowed to register). Do not log the registration in the **`ts-daemon`** log file.

`1`: Valid (the process is allowed to register. Log the registration in the **`ts-daemon`** log file, and continue.

Copyright © 1994 Thinking Machines Corporation

2: Invalid (the process is not allowed to register — the request is aborted). Log the registration attempt in the **ts-daemon** log file.

all others: Error (the process is not allowed to register — the request is aborted). Log the registration attempt in the **ts-daemon** log file.

Note that because the timesharing daemon waits for a response from the verification program, a hanging verification program will cause the timesharing daemon to hang.

11.4.2 CM-5E Support (See Also Section 11.6)

CMOST 7.3 supports two new switches to **ts-daemon**, **-v** and **-f**, either or both of which may be required for a particular partition, depending on the hardware it contains:

- **ts-daemon -v**: must be used to set the cache for the new SuperSPARC chips.

  The **-v** switch ensures that code does not attempt to execute instructions out of data space on CM-5E systems. Such attempts cause no problems on CM-5 systems equipped with standard SPARC microprocessors; on CM-5E systems, however, they may cause cache consistency problems and thus crash the system. If the **-v** switch is used, the **ts-daemon** prevents the attempt; rather than crashing the system, the user's code receives an illegal instruction trap and is aborted.

  (Technical note: The CM-5E has an internal instruction cache, so it actually invalidates I$ entries when it encounters **IFLUSH** instructions. It then relies on the Mbus L2 cache consistency protocol to fetch a copy of an instruction when it needs to execute it. It offers no hooks for trapping on **IFLUSH** instructions.)

- **ts-daemon -f**: must be used when the partition's PNs contain the new CM-5E NI.

Copyright © 1994 Thinking Machines Corporation

11.5 I/O Device Administration

It is no longer necessary to run the **cmdiag** boot scripts — **isolate-sda-scale3**, **isolate-sda-cm5-generic**, and **isolate-hippi-cm5-generic**. The situation in CMOST Version 7.2 that had required you to run these scripts has been remedied.

If you continue to run these scripts at CMOST Version 7.3, no harm will result, however.

11.5.1 IOCA (CM5-HIPPI) Administration

The nodes that control and make up the CM5-HIPPI run an operating system called CMIX. Please note the following (for information on administering the CM5-HIPPI and CMIX, consult the *CM5-HIPPI System Administrator's Guide*):

- Upon installation and first system bringup of the CM5-HIPPI, you must execute **cmdiag -z** to get the IOCA backplane(s) into the CM-5 autosize database.
- In order to run functional diagnostics on an IOCA, a partition containing the IOCA must be created using **cmpartition create**'s **-iop** switch. If **cmpartition list -l** does not show an entry for the IOCA in question, then diagnostics will *not* be able to be run on that particular IOCA.

  For more information about running IOCA diagnostics, see Appendix A, "CM5-HIPPI Diagnostics."
- Do not attempt to log in to a CMIX node that has crashed. Doing so may crash the entire CM-5 system.
- All CMIX nodes (ITS and CM5-HIPPI) must boot off of the same CP.
- Table 2 shows the error codes that may be returned by the CM5-HIPPI system calls **CM_read()**, **CM_send()**, **CM_sendmsg()**, and **CM_sendto()**:

Copyright © 1994 Thinking Machines Corporation

Table 2. Error codes that may be returned by the CM5-HIPPI system calls.

errno	Cause
53 (ECONNABORTED)	Packet aborted by system (e.g., due to application interrupt).
54 (ECONNRESET)	Output packet/connection was aborted by destination (i.e., destination dropped connect).
60 (ETIMEDOUT)	Timeout waiting for connect or ready from destination.
61 (ECONNREFUSED)	Output packet/connection was rejected by destination.
80 (ENONET)	Source interconnect lost during packet transmission.

- If parallel networking is enabled, **cm5ftp** can act as a complete replacement for serial **ftp**. To replace serial **ftp** with **cm5ftp**, change a line in **inetd.conf**; replace

  ```
  ftp    stream  tcp    nowait  root /usr/etc/in.ftpdin.ftpd
  ```

  with

  ```
  ftp    stream  tcp    nowait  root   /usr/etc/cm5ftpd \
  cm5ftp  -c /usr/bin/sdacat
  ```

  This change activates the proper **ftp** daemon. To make this change visible to **inetd**, either reboot the CP or execute

  ```
  # kill -HUP pid of inetd
  ```

- Mounting a file system via NFS over the CM5-HIPPI is not supported at CMOST Version 7.3.

11.5.2 SDA Administration

NOTE: Do not use the **sdafs** command or the **sdafsmk** command. These commands are obsolete and will be removed from CMOST.

Copyright © 1994 Thinking Machines Corporation

- When bringing up a CM-5 with an SDA for the first time after installing CMOST Version 7.3, you must perform an explicit autosize: execute **cmdiag -Z** from the system control processor before executing **cmreset** and partitioning the machine.

- **sda_disk_util** can now be used to verify parity on a per-disk basis (with the **-p** switch) or on a block-range basis (with the **-P** switch). The **-c** switch does a destructive rebuild of the parity disk.

- If the SDA is mounted, run **sdafsck** only if its **-n** switch included on the command line.

- The following tests are new within **cmdiag**'s SDA group. The first three run only under field mode (that is, when **-f** or **f** is specified on the command line); the last three tests run under manufacturing mode (**-m** or **m** on the command line) or field mode:

```
test_serial_timers();
test_double_bufmem_ni_loop();
test_async_bufmem_ni_loop_router_clogged();
test_scn_buffer_done_interrupt();
test_iop_done_interrupt();
test_scn_send_done_interrupt();
```

- Currently, a heal operation requires that the IOCP/CP managing the logical device to be healed is acting as the partition manager of a partition comprising 16 (valid for a Scale 3 only), 32, or 64 PNs. The heal runs fastest with 64 PNs, an amount that enables 256 Mbyte writes to the SDA, approximately the maximum for this operation.

  The heal procedure involves allocating an IOCP partition and running the heal command from the IOCP. An alternate heal procedure involves booting the SDA from a PM and running the heal command from the PM. Both heal procedures are listed in Chapter 13 of the *CM-5 System Administrator's Guide*.

11.5.3 NFS-Mounted File Systems

If **mmap** fails on an NFS-mounted file system, the timesharing daemon will crash.

Copyright © 1994 Thinking Machines Corporation

11.6 CM-5E (See Also Section 11.4)

Table 3, below, lists the CM-5, CM-5E, and CM-5/CM-5E−combination partition configurations supported at CMOST Version 7.3.

Table 3. Configurations supported at CMOST Version 7.3.

PN CPU / NI	CP	Height 1−2 DR	Height 3−6 DR
standard / standard	SPARCstation 2	standard	standard
standard / standard	SPARCstation 10	standard	standard
standard / standard	SPARCstation 10	standard	new
SuperSPARC / new	SPARCstation 10	new	new
SuperSPARC / new	SPARCstation 2	new	new

11.6.1 Partition Configuration Rules

1. All the PNs in a partition must contain the same type of NI: either standard (CM-5) or new (CM-5E). (In either case, the NI on the PM must be the standard version, because the new version is not yet supported on CPs.)
2. If a partition has the new SuperSPARC PNs, the PNs must also be equipped with vector units.
3. A partition manager (PM) can be either a SPARCstation 2 or a SPARCstation 10 (with the S-bus NI, SNI).
4. All Height 3/4, 5, 6 DR boards use the same Data Network within a CM system.

11.6.2 The hardware.install File

If a backplane contains all CM-5E components (SuperSPARC PNs, the new CM-5E NIs, and the new CM-5E Data Network (DR) hardware, change the backplane's **`hardware.install`** entry from **`PN`** to **`SSPNFGT`**.

Copyright © 1994 Thinking Machines Corporation

If you change the **PN** entry to **SSPNFGT**, remove that backplane's **PNMEM** entry (for backplanes without vector units) or **PNDP** entry (for backplanes with vector units) from **hardware.install**.

Also, if the boards in a backplane have the new DR, change the boards' entries as follows:

DR to **DRGT**
SPI to **SPIGT**
FILLER to **FILLERGT**
DR5 to **DR5GT**

The **cmsa-server** program now verifies the **hardware.install** entries against the hardware that it knows to be present in the machine: If **hardware.install** is not updated correctly, **cmsa-server** will not start, and an explanatory message will be logged in **cmsa-server.log**.

(Note that in CMOST versions previous to 7.3 Beta 2.7, an out of date **hardware.install** file would not always cause **cmsa-server** to fail to start. From CMOST Version 7.3 Beta 2.7 onward, however, **cmsa-server** does always care about the veracity of **hardware.install**.)

11.6.3 Setting the Clock Speed

To utilize the higher clock speed available on CM-5E (new NI, new PN) partitions, after executing **cmreset**, execute a **cmdiag**-shell command, as shown below:

1. Execute **cmreset** from the system administration console, as usual:

```
syscon# cmreset
```

2. Execute the **cmdiag** command to enter the **cmdiag** shell:

```
syscon# cmdiag
<CM-DIAG>
```

3. At the **<CM-DIAG>** prompt, execute the following command:

```
<CM-DIAG> fm e 40
```

4. Exit the **cmdiag** environment:

```
<CM-DIAG> exit
syscon#
```


Copyright © 1994 Thinking Machines Corporation

5. Continue with the documented bringup procedure (execute **cmreset -s** on each CP and IOCP, etc.)

The **fm e 40** **cmdiag**-shell command sets the clock speed to 40 MHz.

As a side effect, the **fm e 40** command might cause some PNs to crash, with **find-cm-error** showing IU errors; there is no need to intervene if this happens, because the PNs will be rebooted (and the IU errors cleared) when **cmpartition start** is executed.

(Note to CM-5 system administrators: A recent technical bulletin, *CM5_hw_019_PN_clock_on_CLKDN* (January 12, 1994), asked you to execute **fm d 32** from the **cmdiag** shell after executing **cmreset** on the machine. It is no longer necessary to execute this **cmdiag**-shell command at system bringup because **cmreset** now sets the correct clock source automatically for CM-5 (non-CM-5E) systems.)

11.7 The cmdiag Program

Do not run the test group **nipn** within a partition that contains both standard and new (CM-5E) NIs on the PNs, because false errors may result. (Such a partition is legal for diagostic purposes only. We do not support mixed-NI partitions for running user code; see Section 11.6.) Run **nipn** on a partition with only one kind of NI on the PNs.

Copyright © 1994 Thinking Machines Corporation

Appendix A: CM5-HIPPI Diagnostics

A.1 Running IOCA JTAG Diagnostics

At the system prompt, type

```
% cmdiag
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rgroups m JTAG_group_name
```

where *JTAG_group_name* is one of the following:

- **JTAG-IOCACTRL**
- **JTAG-IOCABUF**
- **JTAG-IOCADR**
- **JTAG-IOCHDST**
- **JTAG-IOCHSRC**

A.2 Running IOCA Functional Diagnostics

At the system prompt, type

```
% cmdiag -p partition_name
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rgroups {m | f} functional_group_name
```

where *functional_group_name* is one of the following:

Copyright © 1994 Thinking Machines Corporation

- **IOCA-CTRL**
  - local XBUS register tests
  - NI functional tests
  - XBUS functional tests
  - XBUS interrupt tests
- **IOCA-BUF**
  - local XBUS register tests
  - NI functional tests
  - memory tests
  - address table tests
  - NIFSM send tests
  - NIFSM receive tests
  - NIFSM send/receive tests
  - address engine static/functional tests
  - XBUS functional tests
  - XBUS interrupt tests
  - exception-handling tests
  - simultaneous board-function tests
- **IOCA-BUFMFG**
  - local XBUS register tests
  - NI functional tests
  - extensive memory tests
  - extensive address table tests
  - static send tests
  - static receive tests
  - NIFSM send tests
  - NIFSM receive tests
  - NIFSM send/receive tests
  - address engine static/functional tests

Copyright © 1994 Thinking Machines Corporation

  - static PBUS tests
  - XBUS functional tests
  - XBUS interrupt tests
  - exception-handling tests
  - simultaneous board-function tests
- **IOCA-DEST**
  - local XBUS register tests
  - memory tests
  - FIFO tests
  - static PBUS tests
  - static CDP tests
  - CDP/HIP tests
  - XBUS functional tests
  - XBUS interrupt tests
  - exception-handling tests
- **IOCA-SRC**
  - local XBUS register tests
  - memory tests
  - FIFO tests
  - static PBUS tests
  - static CDP tests
  - CDP/HIP tests
  - XBUS functional tests
  - XBUS interrupt tests
  - exception-handling tests
- **IOCA-PBUS**
  - single channel PBUS read tests
  - single channel PBUS write tests
  - PBUS exception handling tests
  - multiple PBUS channel tests

Copyright © 1994 Thinking Machines Corporation

- **IOCA-HIPPILOOP**
  - HIPPI internal loopback tests
  - HIPPI external loopback tests support
- **IOCA-SYS**
  - simultaneous board-function tests

A.3 Debugging the IOCA DRs

At the system prompt, type

```
% cmdiag
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rgroups m JTAG_IOCADR
<CM-DIAG> rgroups m bsr
<CM-DIAG> rgroups m dr
<CM-DIAG> rgroups m IOCA-CTRL
<CM-DIAG> rgroups m IOCA-BUF
```

A.4 Debugging the IOCA PBUS

At the system prompt, type

```
% cmdiag
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rgroups m IOCA-PBUS
<CM-DIAG> rgroups m IOCA-SYS
<CM-DIAG> rgroups m IOCA-BUF
<CM-DIAG> rgroups m IOCA-SRC
<CM-DIAG> rgroups m IOCA-DEST
```


Copyright © 1994 Thinking Machines Corporation

A.5 Debugging the IOCA HIPPI

At the system prompt, type

```
% cmdiag
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rgroups m IOCA-SRC
<CM-DIAG> rgroups m IOCA-DEST
<CM-DIAG> rgroups m IOCA-HIPPILOOP
<CM-DIAG> rgroups m IOCA-SYS
```

A.6 Debugging the IOCA XBUS

At the system prompt, type

```
% cmdiag
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rgroups m IOCA-CTRL
<CM-DIAG> rgroups m IOCA-SRC
<CM-DIAG> rgroups m IOCA-DEST
<CM-DIAG> rgroups m IOCA-BUF
```

A.7 Debugging the IOCA External Cables

At the system prompt, type

```
% cmdiag
```

At the resulting **<CM-DIAG>** prompt, type

```
<CM-DIAG> rs
<CM-DIAG> lsb
<CM-DIAG> cpm
<CM-DIAG> load-file iocahiplooptests
<CM-DIAG> ept 2 0 1
```


Copyright © 1994 Thinking Machines Corporation

```
<CM-DIAG> sm hippi_external 1
<CM-DIAG> lps iSrcUtilIFieldToUse
              # output will be formatted as follows
              # PE Adress <variable address X> is
              # iSrcUtilIFieldToUse
<CM-DIAG> write (ioca-pn dr address) variable address X 0 \
ioca-ifield
              # where x = 1 if source board is in IOCA
              # backplane slot CH0. x = 2 if source
              # board is in IOCA backplane slot CH1.
<CM-DIAG> eapt
```

NOTE: If you are performing these tests through a network switch, you must use the assigned **`Ifield`** for the device under test.

A.8 Run Times for the Functional Groups

Table 4. Run times for the `cmdiag` CM5-HIPPI functional groups.

Test Group	Minutes for `-f`	Minutes for `-m`
`IOCA-CTRL`	1.0	1
`IOCA-DEST`	1.5	4
`IOCA-SRC`	3.0	3
`IOCA-BUF`	9.0	20
`IOCA-BUFMFG`	13.0	80
`IOCA-PBUS`	7.0	7
`IOCA-SYS`	11.0	11
`IOCA-HIPPILOOP`	3.0	3

Copyright © 1994 Thinking Machines Corporation

A.9 IOCA hardware.install Entries

The **/etc/cm/configuration/hardware.install** file requires the following entries for the IOCA:

```
PN_Cabinet    0
{

  .
  .
  .
  .

   IO_Backplane    1
   {
         IOCACTRL         0
         IOCADR           0-1
         IOCABUF          0-3
    }
  .
  .
  .
}
```

In this example, the IOCA is backplane 1 and there are two IOCA DR boards and four IOCA buffer boards.

Copyright © 1994 Thinking Machines Corporation

Appendix B: Executing cmreset −s

The command **cmreset -s** resets the NI chip on the CP on which it is executed. This can cause any data being moved through the NI at the time it is reset to be lost. There are specific times when it is okay to reset the NI and other times when it should not be reset, depending on how the NI is being used.

- On an IOCP, never execute **cmreset -s** when the SDA file system is mounted. It is okay to execute **cmreset -s** on an IOCP after unmounting the SDA file system.
- On a CP, never execute **cmreset -s** when either the **rifc** (HIPPI) or **dr** (CMIX DR) interfaces are active. It's okay to execute **cmreset -s** on a CP when the **rifc** and **dr** interfaces are down.
- On a PM only (a machine that is not an IOPC and not a CMIX CP), it is okay to execute **cmreset -s** whenever timesharing is *not* running. Unmounting the SDA prior to executing **cmreset -s** is not necessary in this case.

Besides the timesharing daemon and the CP portions of user programs running on the CM, the CM-5 I/O system uses the NI for communication between the various CPs and the I/O devices listed in Sections B.1−B.3.

B.1 The NI and the SDA

An IOCP for an SDA file system uses the NI extensively to communicate with the SDA controllers. This communication is driven not by the timesharing daemon, but rather by the SFS file system software within the UNIX operating system.

Copyright © 1994 Thinking Machines Corporation

A CP using the SDA communicates with the IOCP via Ethernet for all serial I/O operations. Parallel I/O, however, does involve direct communication between the timesharing daemon and the SDA controllers via the DR. Thus, as long as timesharing is down, a partition manager will not use the DR.

Unmounting the SDA file system from the IOCP will stop all DR activity over that IOCP's NI that is related to that file system.

B.2 The NI and CMIX

Although CMIX isn't a product by itself, it is important to remember that both the CM5-HIPPI and ITS are based on CMIX running on the CM5-HIPPI or ITS controller nodes.

CMIX implements a network of UNIX systems using the CM-5 Data Network (DR) like an Ethernet. Any CMIX node can talk to any other CMIX node (or CMIX-configured CP) at any time. The CMIX interface to the DR on a CP is called **dr0**. This is the pseudo-network interface on which CMIX sends and receives DR messages.

If a system's DR interface is up and running, don't reset the NI on that system. If the system's timesharing daemon is down, as shown below, and it is not an IOCP for a SDA file system, it can be safely reset.

```
% ifconfig dr0
dr0: flags=0<>
% cmreset -s
```

B.3 NI and the CM5-HIPPI

The UNIX networking software communicates with the IOCA system through the use of DR messages. Again, the timesharing daemon is not involved.

CPs communicate with the CM5-HIPPI through **rifc** (the remote interface client interface). The status of the **rifc** interface is also available via the **ifconfig** command. In the example below, the RIF interface is down.

```
% ifconfig rifc0
rifc0: flags=1000<>
%
```


Copyright © 1994 Thinking Machines Corporation

Appendix C: DJM

DJM is now being made available on the CM-5. For detailed information about using and administering DJM, see *Using DJM on the CM-5*. The text below provides an overview of DJM.

DJM stands for Distributed Job Management, a system initially developed at the Minnesota Supercomputer Center. DJM was designed to

- manage the flow of jobs through a Connection Machine CM-5 system
- avoid resource conflicts
- provide a load balancing capability among the various control processors for a Connection Machine CM-5

In order to balance system load, DJM handles all application processes running on the CM-5, interactive jobs as well as batch jobs. It allows jobs for which there are sufficient resources to execute immediately; it queues all other jobs for later execution.

- Interactive jobs execute "in the foreground" and allow terminal input and output.
- Batch jobs execute "in the background." They allow input and output only to specified files.

DJM ensures that it handles all jobs by trapping jobs that are invoked directly on the CM-5, rather than going through DJM's job-submission commands. It has the ability to penalize such trapped jobs (which it calls *foreign* jobs); whether it does so is up to the system administrator at each site.

DJM allows users and administrators alike a great deal of flexibility. Users can request either dedicated time or multi-user access. They can specify many parameters for their job, or only a few — or, they can accept DJM's default parameters. They can ask that DJM send them mail when a job

Copyright © 1994 Thinking Machines Corporation

begins or ends execution. They can move a job from interactive execution ("the foreground") to batch execution ("the background"), or vice versa.

Administrators can choose how many queues to set up and what restrictions to put on various queues (which users can use the queue, what resource limits are placed on jobs using this queue, etc.). They can choose whether to allow dedicated access to the system, and when to allow it. They can determine how "foreign" jobs (jobs not submitted via DJM commands) are to be handled.

The CM system administrator is in charge of configuring DJM to meet the needs of each site. In general, DJM relies on a small number of queues to manage the job flow. At a minimum, there will be two queues at a site:

default: A general-purpose queue. Unless otherwise specified, all DJM jobs will be submitted to this queue.

foreign: A queue that handles jobs that are submitted from outside of DJM. While it is possible to bypass DJM and run jobs directly, they will still be managed by DJM, and end up in the foreign queue. The foreign queue usually has severe limitations imposed on it to force the use of DJM queues, so unless you are running a small fast job, you should submit the job for DJM to handle.

C.1 Compatibility with NQS

Sites currently using the Network Queuing System (NQS) should note that DJM is compatible with NQS, at the user level, in its handling of requests for batch job submissions. Indeed, the system administrator can configure the system in such a way that user-level NQS commands (**`qsub`**, **`qlimit`**, **`qdel`**, and **`qstat`**) will continue to work as they did previously.

Copyright © 1994 Thinking Machines Corporation

C.2 DJM Commands

Table 5, below, lists the DJM user commands and their NQS equivalents. For more information about these commands, consult the on-line manual pages. For more information about DJM in general, see the manual *Using DJM on the CM-5*.

Table 5. DJM and NQS user commands.

DJM Command	NQS Command	Meaning
`jsub`	`qsub`	Submit a job for background processing.
`jrun`	----	Submit a job for interactive processing.
`jdel`	`qdel`	Delete or signal a DJM job.
`jstat`	`qstat`	Display status of DJM system.
`jconnect`	----	Connect a job to a terminal (tty).
`jmsg`	----	Send a message to a job's output buffer.
`js`	----	Display system-wide status of DJM system.
`jwait`	----	Wait for a DJM job to start/finish.

Copyright © 1994 Thinking Machines Corporation

Appendix D: DPAS Version 1.1 Release Notes

These are the release notes for DPAS Version 1.1, which is shipped with CMOST Version 7.3. Please note the fifth item in Section D.2, which represents a possible source incompatibility.

D.1 New Features

- Adds **`-B`** switch, which allows path to CPP to be given. The syntax is **`-B`***path*. The default path is **`/usr/lib/cpp`**.
- Adds **`NOEXCHANGE`** modifier, which selects the default obtained by not saying **`EXCHANGE`**. If both are given on the same line, the rightmost takes effect. In fact, cancelling the effect of a previous **`EXCHANGE`** is the only thing that **`NOEXCHANGE`** actually does. This switch is implemented because it appears in the documentation.

Copyright © 1994 Thinking Machines Corporation

D.2 Fixed Bugs

- Allows spaces before `]` in memory address syntax **`[%reg+%reg]`**.
- Fixes the following code-generator bug in DPAS: incorrect handling of a VU instruction, where the arithmetic operation specifies both an **`Rs1`** stride override (but not zero or unit stride) and memory stride override. For example, these did not generate correct code:

  ```
  uandv V2:2,V2,V2 ; iloadv [%i0]:4,V8
  uandv V2:2,V2,V2 ; iloadv [%i0]:8,V8
  ```

  The effect was as if the memory stride override were not given.
- When passing a long line, untranslated, to the assembly output, DPAS now passes the entire line rather than just part of it.

D.3 Changed Source Files

The following **`usr.bin/dpas/`** source files have changed:

```
Makefile
as-parse.c 1.8
common.h 1.10
dpas.c 1.23
express.c 1.5
instruct.c 1.25
lexical.c 1.11
list.c 1.6
long.c 1.3
macro.c 1.7
object.c 1.42
```

D.4 Changed Header Files

The **`dpeacdesc.h 1.12`** header file, in **`sys/cmsys/`**, has changed.

Copyright © 1994 Thinking Machines Corporation