CMOST 7.4 Release Notes

Part I Overview

1 Introduction

The principal features of the CMOST Version 7.4 release are summarized below and described more fully throughout the release notes.

- Improved reliability, availability, and serviceability (RAS)
- CM-5E support for long messages
- Improved performance for I/O peripherals
- Support for checkpoint/restart (CPR)
- Improved serial I/O performance via MNI support for SS10- and SS20-based control processors

2 Hardware and Software Requirements

2.1 Hardware Required

CMOST Version 7.4 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 7.4, in aggregate with all its subproducts, requires approximately 330 Mbytes of disk space.

Copyright © 1995 Thinking Machines Corporation

2.2 Software Required

CMOST Version 7.4 is based on SunOS Version 4.1.3_U1 (Solaris 1.1.1 Rev B). This means all Sun hardware in the CM-5 system (CPs, IOCPs, etc.) must be installed with SunOS 4.1.3_U1. The SunOS 4.1.3_U1 CDROM and installation instructions are included with this CMOST 7.4 shipment.

Note that CMOST Version 7.4 uses a new version of DPAS, Version 1.2. Appendix A contains the release notes for DPAS Version 1.2.

2.3 Software Supported

The following versions of CM-5 software are compatible with the CMOST Version 7.4 release:

- CMMD 3.3
- CMRTS 8.1
- CM/AVS 1.1.1
- Prism 2.0 and up (all 7.4 features except CPR; see Section 4.1.2)
- Prism 2.2 (all 7.4 features including CPR)
- CMF 2.1.1, CMF 2.1.2, or CMF 2.2
- C* 7.1.1 or 7.2
- CMSSL 3.2 or 3.2.1
- CMX11 1.6 or 1.7 Beta
- CMAX 2.0
- *Lisp 7.5.1
- DJM 2.0

Copyright © 1995 Thinking Machines Corporation

3 Porting Information

Programs that work under CMOST Version 7.3 continue to work at CMOST Version 7.4. Relink your program with CMOST Version 7.4 to take advantage of the performance enhancements and features new to Version 7.4.

NOTE: Code linked with any beta verson of CMOST 7.4 must be relinked with CMOST 7.4 final.

4 New Features

4.1 New Functionality

4.1.1 Support for New Hardware

Complete CM-5E Support: MNI on SPARCstation 10/20

CMOST Version 7.4 supports use of the MNI board in SPARCstation 10 and 20 CPs. The MNI board's Mbus architecture means faster NI accesses. This hardware speedup is augmented by improvements in data network protocol software (`drlp`), resulting in significantly faster serial I/O in both CM-5 and CM-5E systems.

In a CM-5E system, the MNI board also enables CPs to send and receive long messages of up to 18 words across the data network. This contrasts with the 5-word message length that non-MNI CPs are restricted to.

NOTE: Long-message support requires DR chips of the FAXGT type throughout the message path.

4.1.2 Support for New Software/OS Features

Checkpoint/Restart (CPR)

Checkpoint/Restart (CPR) allows the entire state of a CM-5[E] process (both on the CP and PNs) to be saved out to an SFS file system on an SDA in order to be restarted at a later time. CPR will allow long-running jobs to save their state and rollback in the case of a crash or other problem. CPR also provides system administrators (and batch systems such DJM) flexibility in killing jobs that might have exceeded their resource limits.

Copyright © 1995 Thinking Machines Corporation

You can use CPR to checkpoint CM Fortran and C* programs, and hostless CMMD programs written in C. CMMD programs using CPR require CMMD Version 3.3 or later.

For more information about CPR, see the *CM-5 User's Guide,* Version 7.4, or the manual pages **chkpnt**(1), **restart**(1), and **rmchkpnt**(1).

Swapping

Swapping — context-switching in a new process, with the SDA storing the memory of a non-running process — is supported by CMOST Version 7.4.

C++

Under CMOST Version 7.4, CMOST header files do not interfere with compilation or execution of C++ programs on the CM-5.

DJM

CMOST Version 7.4 supports a new release of DJM, V0−9−17. In addition, DJM is now uncoupled from CMOST and is separately installable.

Runtime Allocation of Context−Switching Buffers

Under CMOST Version 7.4, the timesharing daemon automatically determines whether it can use the CM-5E's new larger context-switching buffers. See Section 9.7 for details.

4.1.3 Reliability-Availability-Serviceability (RAS) Improvements

RAS improvements touch many areas of CMOST 7.4:

- MTBS is increased by
  - more robust and faster cable connectivity tests
  - more robust VU tests
  - more sensitive memory tests
  - more robust SuperSPARC tests

Copyright © 1995 Thinking Machines Corporation

- MTTR is increased by
  - faster access to the Diagnostic Network
  - faster system reset (**cmreset**)
  - faster SDA boot time
  - faster **sdafsck**
  - more user control over core dumping
  - more robust context-switching code

In addition, CM-5 system administration (CMSA) has been simplified: fewer **ts-daemon** command-line switches are now required (see Section 9.6). CMSA now automatically sets and margins clock speed on a per-partition basis, sanity checks partition hardware for incompatibility, and supports a new diagnostic partition type.

4.2 Performance Enhancements

4.2.1 General Application Performance

CMOST Version 7.4 support of long messages (see Section 4.1.1) greatly improves the performance of most applications running on the CM-5E.

4.2.2 I/O Performance

SDA

CMOST Version 7.4 improves the performance of CM-5 disk subsystems by as much as 50% through increases in

- transfer rates
- transaction speed
- **sdafsck** speed
- boot speed
- drive capacities

Copyright © 1995 Thinking Machines Corporation

CM5-HIPPI

CMOST Version 7.4 improves CM5-HIPPI FTP (**cm5ftp**/**cm5ftpd**) speed and small-packet transfer rates.

Serial Transfer Rates

Improvements in the CP's Network Interface driver have increased serial (CP) transfer rates by as much as 700% in an unloaded system.

4.2.3 Core OS

This release supports faster interrupt dispatching and faster broadcasting.

5 Changes from Previous Versions

5.1 Changes to Interface

5.1.1 Output of cmps

The header portion of the **cmps** command's output has changed slightly, which may affect scripts that use **cmps**. Under CMOST 7.3, the output looked similar to this:

```
16 PN System, 27784K mem. free, 6948K VU mem. free, 1 procs,
TS-10/6/94-11:42 (CMOST 7.3 Final 1 Rev 4) Daemon up: 14 seconds

USER   PID     CMPID   TIME   TEXT   ILH   ILS   IGS   IGH   VUS   VUH   COMMAND
root   * 5317  2       0:08   64K    2152K 48K   0K    0K    0K    OK
router_msgs
```

Under this release, CMOST 7.4, the header of the **cmps** output has the following format:

```
16 PNs, 32 MHz, 27656K/6916K VU mem. free, 1 procs, TS-10/4/94-11:47
(CMOST 7.4 CMNA_VERSION 9) Up: 55 seconds

USER   PID     CMPID   TIME   TEXT   ILH   ILS   IGS   IGH   VUS   VUH   COMMAND
root   * 5317  2       0:08   64K    2152K 48K   0K    0K    0K    OK
router_msgs
```


Copyright © 1995 Thinking Machines Corporation

5.1.2 Report of NI Error on a CP

A change has been made to the way NI errors on the CP are reported to the user's terminal. In CMOST Version 7.3, the message had this form:

```
CMOST: Software error (CMNA_interrupt_cause = hex value)
```

where *hex value* represents the contents of the **CMNA_interrupt_cause** register.

CMOST now reports the error specifically as a CP error and attempts to decode the interrupt cause; for example,

```
CMOST CP ERROR: Bad memory access (bus error)
```

If CMOST is unable to decode the error, the message will be in this form:

```
CMOST CP ERROR: Unknown error (CMNA_interrupt_cause = hex value)
```

5.1.3 Uniform NI Status Reporting

With CMOST Version 7.4, the same format is used for all three types of NI status dump operations:

- **ni-info** shell command
- **nistatus kpndbx** command
- SBC driver on **ts-daemon** crash (logged to **/var/adm/messages**)

Previously, each output had a different format. Now a single library function is used, resulting in the same format for all.

In addition, the reporting code has been modified to decode only those fields that are relevant to the applicable register. For example, no "rec tag" field is reported for the BC Status Register.

As a result of these changes, all three output formats have changed from their CMOST 7.3 counterparts. Therefore, any scripts that parse any of these report outputs should be updated for CMOST 7.4.

Copyright © 1995 Thinking Machines Corporation

An example of the new format follows.

```
NI status:
  Configuration:    3  Serial number: 2008073  Interrupt lvl:    0
  Time:      2d41b069  Bad address:  42100004  Rec Int mask:  fc00
  User tag mask: fc00  DR count mask:     3ff  DR msg count:     0
  Chunk size:       1  Partition size:      4  Partition base:  18
  Self address:    10  Hodgepodge:       4f3e  User Rec Int mask:0
  Longest DR Msg:   5
  Interrupt cause:  0x0 < >
  Interrupt cause green:  0x0 < >
RDR:
  Status:    0x205f <sspace 31; rlen 1; rlen left 0; rtag 0> <SOK>
  Private:   0x0
LDR:
  Status:    0x205f <sspace 31; rlen 1; rlen left 0; rtag 0> <SOK>
  Private:   0x0
DR:
  Status:    0x20df <sspace 31; rlen 1; rlen left 0; rtag 0> <SOK RDONE>
  Private:   0x0
COM:
  Status:    0x1067 <sspace 7; rlen 2; rlen left 0> <EMPTY SOK>
  Private:   0x19100
  Control:   0x1
SBC:
  Status:    0x64 <sspace 4; rlen left 0> <EMPTY SOK>
  Private:   0x10
  Control:   0x1
BC:
  Status:    0x68 <sspace 8; rlen left 0> <EMPTY SOK>
  Private:   0x10
  Control:   0x1
```

NOTE: The “User Rec Int mask” and “Longest DR Msg” registers are specific to CM−5E NI chips and are not reported when run on CM−5 NIs.

Copyright © 1995 Thinking Machines Corporation

5.1.4 Accounting

Some changes have been made to the CM-5 accounting interface at CMOST Version 7.4. These are summarized in bulleted list below.

- Starting a partition's accounting is less complicated. Rather than executing **`cmacct -o`** and **`cmacct -d &`** on separate command lines to open the partition's accounting file and start the accounting daemon, respectively, now simply execute **`cmacct -o`** *acctfile*. The **`-o`** switch both opens the accounting file and starts the daemon, backgrounding it automatically. The **`-d`** switch is still operational and is used if the accounting daemon must be started explicitly.

- The old **`-c`** switch has been replaced by the new **`-k`** switch, which stops accounting on the partition and kills its accounting daemon.

- The old **`-m`** switch, which facilitated log file rotation, is now obsolete. Log file rotation is now facilitated by the **`-o`** switch; see Section 10.5 for more information.

- The old **`-w`** switch, which set the file system's warning percentage, has been replaced by the **`-F`** switch. The argument taken by **`-F`**, *warning_percentage*, works differently than the old **`-w`** switch's argument; the old *warning_percentage* was the percentage of space remaining on the file system, whereas the new *warning_percentage* is the percentage of space used up on the file system. See Section 10.5 for more information.

- The **`cmacct`** program has two more tuning flags: **`-a`** and **`-M`**.

- The **`-a`** *bytes* switch specifies the amount of memory to allocate for use by the accounting system. The size is specified in bytes and can be in the range 150 Kbytes to 1500 Kbytes, inclusive. The default is 500 Kbytes. **`cmmact`** interprets a suffix of **`k`** or **`K`** to mean to multiply the preceding number by 1024 and a suffix of **`m`** or **`M`** to mean to multiply by 1024 x 1024.

  NOTE: In the release notes for CMOST 7.4 Beta, the size argument to **`-a`** was incorrectly described as given in Mbytes — as stated above, the correct unit of measure is bytes.

Copyright © 1995 Thinking Machines Corporation

- The **-M** *warning_percentage* switch sets the accounting memory's warning percentage: when the memory allotted to accounting has been consumed to the *warning_percentage* percent mark, a message is sent to the system console. The default *warning_percentage* is 90.

In addition to the changes described above, accounting's error handling has been enhanced. Please see Section 10.5 for details.

Finally, the information kept by the process and I/O accounting records has changed significantly with CMOST Version 7.4.

NOTE: CMOST Version 7.4 changes the **cmacct** extract and report output formats. This may affect the behavior or any scripts that use this output.

CAUTION: The binary files kept by **cmacct** under CMOST Version 7.3 are incompatible with CMOST Version 7.4. Therefore, if you might need to analyze old accounting files after CMOST 7.4 is installed, you should save a copy of the Version 7.3 accounting program(s).

For **cmacct** record structures, see the typedefs **CMAC_Kernel_Account_Data** and **CMAC_Kernel_Account_IO_Data** in the file **/usr/include/sys/cmacct.h**.

5.1.5 New Switches to cmreset and cmpartition start

The **cmreset** command has a new switch, **-f** *fifo_size*, which sets the default FIFO size for the CM-5E Data Networks at Height 3 and above. The **cmpartition start** command's new switch, **-fifo** *fifo_size*, can override the system *fifo_size* default for the CM-5E H3+ DRs (if there are any) within a particular partition. See Section 10.7 for more information.

5.1.6 User Verification

The timesharing daemon's user verification feature, enabled by including **-v** *verifier_script_pathname* on the **ts-daemon** command line, allows your site to verify whether or not a user registering with the **ts-daemon** has permission to run programs on the CM-5. *verifier_script_pathname* specifies the pathname of the site-supplied script that does the verification. The script file is called with three arguments supplied by **ts-daemon** in this order:

```
<script_filename> <user's CM_ACCOUNT_ID> <user's uid>
```


Copyright © 1995 Thinking Machines Corporation

5.2 New PN Memory Allocation Step for SDA Use

Under CMOST 7.3, PN memory was allocated for parallel I/O when the first-parallel I/O operation began. Ordinarily, allocation would be successful and the I/O operation would proceed successfully. This approach did have the potential for crashing **ts-daemon**, however. If PN memory happened to be already fully allocated at the time the SDA allocation was attempted, the timesharing daemon would crash.

Under CMOST 7.4, a new **ts-daemon** capability provides a dependable way to avoid this risk. Simply mount any SDA file system(s) on a CP before its timesharing daemon is started. Then, when **ts-daemon** starts, it will automatically allocate the requisite memory space needed. When this step is executed, an entry like the following is made in **ts-daemon.log**.

```
SDA_update_entry: allocated 0x40000 bytes at 0xf89db6d0
    for SDA parity buffer per PN.
SDA_update_entry: allocated blk buffer at 0xf8a1b6d0,
    0x800 bytes on 64 PNs
SDA_init: 1 SDA device(s) configured successfully.
```

If a file system is not mounted before **ts-daemon** is started, PN memory will be allocated as part of the first parallel I/O operation, just as under CMOST 7.3. If there is sufficient PN memory available at that time, I/O will proceed, just as before.

5.3 Requirement for Power Cycling CM-5E CPs

The commands **cmpoweroff** and **cmpoweron** must be used before power-cycling control processors on the CM-5E. If these commands are not used, errors will be generated, which might cause other partitions to crash.

Control processors on CM-5 systems without “E” attributes can still be power-cycled without using these commands.

Copyright © 1995 Thinking Machines Corporation

5.4 Changes to Diagnostics

5.4.1 Expanded ATPG Coverage

This release increases ATPG (Automatic Test Pattern Generation) coverage of revision 2 vector units. Previously, only about 10% of VU gates were tested (no ALU coverage). Now, approximately 99% of all testable VU gates are covered by ATPG vectors; this represents approximately 52% of all VU gates.

NOTE: In most cases 32 MHz VUs (i.e., ROSS-version) are not revision 2 units and, in most cases, 40 MHz VUs are.

5.4.2 Changes to IOBA Error Code

This release includes four changes to the IOBA error code list. These changes are summarized below.

Error Number	New Definition
37	The program attempted to unlock the CMIO bus, but the lock was already in use by another program.
38	The program attempted to unlock the CMIO bus, but the lock was held by another process.
50	A timeout occurred while waiting for the Data Transfer phase to begin. The DataVault may have timed out.
56	A timeout occurred while trying to send data from the DataVault to the PNs using FIFO READ mode on the IOBA buffer boards. The problem may be one of the following: 1. One or more PNs may be broken. 2. The I/O hardware may be broken. Run the IO tests from CMDIAG.

Error numbers 37, 38, and 50 have entirely new definitions; they had previously been reserved. Error number 56 is a modification of an earlier

Copyright © 1995 Thinking Machines Corporation

definition; the first possible cause listed in the previous definition regarding the **cm.conf** file has been deleted.

5.4.3 On-Line Diagnostics Removed

The on-line exerciser program, **olx**, which was released with CMOST 7.4 beta, has been removed. Beta experience with this software has disclosed bugs capable of causing **ts-daemon** crashes.

5.5 Obsolete Features

5.5.1 Removed ts-daemon Switches -f, -F, -v

The **ts-daemon** switches **-f**, **-F**, and **-v** to are obsolete.

Please remove them from any scripts that invoke **ts-daemon**.

5.6 Obsolete Features Removed

The SVME interface to the Sun 4/330 CP is no longer supported. None are currently in use at customer sites.

The **sdafsmk** command and the **sdafs** command (not to be confused with the **sdafs** file), which were obsolete at CMOST Version 7.3, have now been removed from CMOST; use **sdafsmake** instead.

The **cmboot** command, obsolete in CMOST 7.3, is removed as of Version 7.4.

6 New Information

6.1 Clarification of sdafsck Behavior

If, when a system crash occurs, file systems unmount cleanly (i.e., they are successfully sync'ed), a subsequent **sdafsck -p** will not check them.

Copyright © 1995 Thinking Machines Corporation

Instead, it simply reports that they "unmounted cleanly." This behavior conforms with **fsck** in similar circumstances.

7 Documentation for this Release

Other CMOST and I/O documentation (in addition to these release notes) available on the CMOST 7.4 tape includes

- The *CM-5 User's Guide,* Version 7.4, (postscript). Of note, see Chapter 5, "Checkpointing."
- The *CM-5 System Administrator's Guide,* Version 7.4 (postscript). Of note, see Chapter 7, "Accounting on the CM-5 System."
- CM-5 hardware troubleshooting modules.

Manual pages are available on-line as part of the release.

8 Errors, Feedback, and Assistance

8.1 Bugupdate Files

For the list of fixed and outstanding bugs, please see the file **cmost-7.4.bugupdate** (in **/usr/doc** by default).

8.2 Request for Feedback

Thinking Machines Customer Support encourages customers to report errors in Connection Machine system and software 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.

Copyright © 1995 Thinking Machines Corporation

If your site has an applications engineer or a local site coordinator, please contact that person directly for support. Otherwise, please contact Thinking Machines' home office customer support staff:

Internet Electronic Mail:	`customer-support@think.com`
uucp Electronic Mail:	`ames!think!customer-support`
U.S. Mail:	Thinking Machines Corporation Customer Support 245 First Street Cambridge, Massachusetts 02142-1264
Telephone:	(617) 234-4000

9 Installation and Customization Information

The installation procedure for CMOST 7.4 is documented in the *CMOST 7.4 Installation Guide*, which is included with this release shipment.

After you install CMOST Version 7.4, please check the system's **`/etc/cm/configuration/hardware.install`** file to see whether it needs to be updated with the two new entries described in Section 10.4.

Copyright © 1995 Thinking Machines Corporation

Part II: Detailed Information about New and Changed Features

10 Information for System Administrators

10.1 Revised SDA NFS-Mounting Procedure

At CMOST 7.4, SDA file systems should be NFS-mounted interruptible — include the **`-o intr`** switch on the mount command line. (In the previous release of CMOST, Version 7.3, we recommended that NFS-mounted SDA file systems be mounted uninterruptible because of a software restriction, which has now been removed.)

10.2 cmpartition −l and cmsa−server

The command **`cmpartition -l`** no longer waits for a connection to **`cmsa-server`**, as it did in CMOST Version 7.3. Consequently, you may receive the following message:

```
% cmpartition list -l
<date> <time> cmpartition: Can't connect to cmsa-server
```

Using a script, you can still force **`cmpartition -l`** to loop until it exits successfully.

10.3 cmpartition create −diag

A **`-diag`** switch has been added to the **`cmpartition create`** command. This switch disables configuration uniformity checking when the partition is created. Its purpose is to allow system-wide diagnostic partitions to be

Copyright © 1995 Thinking Machines Corporation

created even when the full system contains a mix of CM-5 and CM-5E component types.

10.4 New Entries in the hardware.install File

There are two new **/etc/cm/configuration/hardware.install** entries that may apply to your CM-5E system:

- **SPI_Backplane** *n*

  This entry is analogous to the **CPI_Backplane** and **PN_Backplane** entries. **SPI_Backplane** identifies backplanes that have SPI/SNI/FILLER/CLKDN/CN boards, but no PN/SSPN boards.

  When CMOST Version 7.4 is installed, check the **hardware.install** file to see whether there are any **PN_Backplane** entries that should be updated to say **SPI_Backplane**. For example, the following module should be updated:

```
PN_Cabinet 0
  {
     PN_Backplane    0    ;change to 'SPI_Backplane'
     {
       SPI      0
       SNI      0
       FILLER       1-3
       SPI      4
       SNI      4
       FILLER       5-7
       CLKDN         0
       CN       0
            }
...
```

If this change is not made, the **cmreset** command will return an invalid clock source error.

- **MNIF** *range*

  The **MNIF** entry applies to the slot-type entries of a backplane module; it is analogous to the **SPI**, **SNI**, **FILLER**, etc., entries. **MNIF** identifies the new CM-5E network interface (NI) board.

Copyright © 1995 Thinking Machines Corporation

Please refer to the **hardware.install**(5) manual page for more information.

10.5 Accounting Changes

10.5.1 New cmacct Syntax

The syntax of the **cmacct** program has changed since CMOST Version 7.3. The **cmacct** syntax at CMOST Version 7.4 is the following:

```
cmacct [operations_flags] [tuning_flags]
```

where *operations_flags* are

-d Starts the accounting logging daemon explicitly, as would be needed if the daemon crashed. The accounting logging daemon is backgrounded automatically.

-k Stops accounting on the PM on which it is executed and kills the accounting logging daemon.

-i Reports the current status of the accounting system. This is the only **cmacct** flag that can be executed without root privileges. See Section 7.3.2 for more information.

-o *acctfile* Activates accounting facilities by opening the log file **acctfile** and starting the accounting logging daemon (backgrounded automatically).

If **cmacct -o** is executed when accounting is already running, and the *acctfile* argument is a filename different from the one currently being used, the log file is rotated; that is, logging goes to the new *acctfile*, and the previous one is not deleted.

If **cmacct -o** is executed when accounting is already running, but the *acctfile* argument is the same filename as the one currently being used, there is no effect (information continues to be appended to *acctfile*).

Copyright © 1995 Thinking Machines Corporation

-s Generates an accounting snapshot of the processes that are currently running, for the purpose of including accounting records for processes that run long-term. Records generated by a sync are flagged so that they can be differentiated from a record generated when a process exits.

and *tuning_flags* are

-a *bytes* The amount of memory to allocate for use by the accounting system. The size is specified in bytes and can be in the range 150 Kbytes to 1500 Kbytes, inclusive. The default is 500 Kbytes. **cmacct** interprets a suffix of k or K to mean to multiply the preceding number by 1024 and a suffix of **m** or **M** to mean to multiply by 1024 x 1024.

-M *warning_percentage*
Sets the accounting memory's warning percentage: When the memory allotted to accounting has been consumed to the *warning_percentage* percent mark, a message is sent to the system console. The default *warning_percentage* is 90.

-F *warning_percentage*
Sets the file system's warning percentage: When the file system to which accounting writes becomes full to the *warning_percentage* percent mark, a message is sent to the system console. The default *warning_percentage* is 90.

-l Limits reporting to CM-5 processes. The default is to report all processes, including UNIX-only ones.

-u Record accounting information for all processes, including UNIX-only ones. This is the default behavior.

Copyright © 1995 Thinking Machines Corporation

10.5.2 New cmacct Error Handling

All errors generated by the accounting system are logged to the console and to **/var/adm/messages**. The accounting error messages and their meanings are as follows:

`CM_accounting:` *acctfile* `unexpectedly truncated to` *integer* `bytes.`

This message appears if the accounting file is too short to contain the header, e.g., an empty file.

`CM_accounting: version mismatch` *version1* `(`*version* `in` *acctfile*`).`

This message appears if the accounting software and the accounting log file are of different versions.

`CM_accounting: log file` *acctfile* `more than` *percent*`% used.`

This message appears when the log file has reached its warning percentage of available disk space remaining.

`CM_accounting: archive-log file system full.`

This message appears when the file system in which the accounting log resides is 100% full.

`WARNING! accounting daemon unable to access` *acctfile*`.`

This message appears when the CP kernel is unable to write on the daemon's behalf. This message usually indicates that NFS is down. This message is repeated every time a process exits until the situation is remedied.

`CM_accounting enabled [ruid` *real_user_id*`].`

This message appears when the accounting system is turned on. The real user ID of the person issuing the **cmacct -o** command is logged.

Copyright © 1995 Thinking Machines Corporation

`CM_accounting disabled [ruid` *real_user_id*`].`

This message appears when the accounting system is turned off. The real user ID of the person issuing the **`cmacct -k`** command is logged.

`CM_accounting: reached highwater mark of` *integer* `bytes[`*memory_allocated*`]`

This message appears when the accounting memory limit (shown by *integer* value) is reached. The *memory_allocated* value in square brackets is the amount of memory that has actually been allocated.

`CM_accounting: cannot allocate` *integer* `bytes.`

This message appears if the kernel heap has been exhausted.

`CM_accounting: using free list.`

This message appears when a memory-crunch situation (see the highwater mark message) has been rectified; that is, memory has become available to the accounting system.

`CM_accounting: memory more than` *percent*`% used.`

This message appears when the resource-warning percentage has been reached.

10.6 New ts−daemon Switches

10.6.1 ts−daemon −b

User programs may access an NFS-soft-mounted (or NFS-hard-mounted interruptible) file system. If this is the case, administrators may want to specify the **`-b`** *integer* switch to the **`ts-daemon`** command line, where *integer* is the number of allowed attempts to talk to an object file in the face of NFS-server non-response (in the range −1 to MAX_INT). If the **`-b`** switch is not on the **`ts-daemon`** command line, the timesharing daemon will retry ten times before giving up.

Copyright © 1995 Thinking Machines Corporation

If the **ts-daemon** eventually times out the request because it has tried — unsuccesfully — *integer* times to achieve a response, the process will die with a bus error and an explanatory message will be logged in the **ts-daemon** log.

If **-1** is supplied as the argument to **-b**, the timesharing daemon will keep attempting to obtain a response until it succeeds. In this case, a process that encounters an NFS-server non-response could die with no explanation, with the **ts-daemon** giving another process a chance to run. This was the default behavior in CMOST versions prior to 7.4. (In some cases, **ts-daemon** would hang indefinitely.)

10.6.2 ts−daemon −u

The **ts-daemon -u** switch limits the number of processes that **ts-daemon** will run concurrently, and proportionally reduces the amount of memory set aside on the PNs for the context-switching buffers.

The argument, *integer*, can be 1−30, inclusive. The default value is 30, when **-u** is supplied on the command line without an argument. The value of the argument cannot be changed while **ts-daemon** is running; to change it, stop **ts-daemon** and restart it with a new value for *integer*.

When *integer* processes are already running on the partition and another process tries to start, the invoking user will receive a message that, because the maximum number of processes is already running, he/she should try to run again later.

10.7 Larger Data FIFOs in Data Network on CM-5E Systems

10.7.1 cmreset −f <fifo_size>

On CM-5E systems only, administrators can execute **cmreset -f** *fifo_size* to set the system-wide default FIFO size for the CM-5E Data Networks (DRs) at Height 3 and above. (The FIFOs for all DR nodes at Height 1 and Height 2 are automatically set to the smallest value to avoid latency issues.) Valid values for *fifo_size* are 16, 64, 128, and 256.

If **-f** is not specified on the **cmreset** command line, the system-wide default FIFO size for the CM-5E Height-3+ DRs is 256 words.

Copyright © 1995 Thinking Machines Corporation

Note that the larger the FIFO size, the longer context switching will take; so, with small timesharing quanta, the time to do the context switch can become prohibitive.

10.7.2 cmpartition start −fifo <fifo_size>

Use the **cmpartition start -fifo** *fifo_size* switch to override the default system-wide FIFO size for the CM-5E H3+ DRs (if there are any) within an individual partition.

10.7.3 New PN Kernels for the CM-5E Data Network

Under CMOST Version 7.4, there are two new PN kernel executable files: **pn_kernel.faxgt**, and **pn_kernel.unopt.faxgt**. These kernels have larger context-switching buffers, which can be used by the CM-5E. Two new CMNA libraries correspond to the new PN kernels: **libcmna_sup_faxgt_pe.a** and **libcmna_debug_sup_faxgt_pe.a**.

The timesharing daemon for 7.4 automatically determines whether the Connection Machine is a CM-5E (that is, whether it contains the new Data Network); if so, **ts-daemon** appends **.faxgt** to the name of the PN kernel executable to be booted. So, the new PN kernel executable files are, with one exception, transparent to administrators and users. Except when running **kpndbx** on a CM-5E processing node, continue to specify the old kernel name (i.e., either **pn_kernel** or **pn_kernel.unopt**). When running **kpndbx** on a CM-5E, however, you must include the **.faxgt** suffix to specify the appropriate kernel.

NOTE: Because of this implementation, CM-5 users do not unnecessarily pay the penalty for the larger PN context-switching buffers.

10.8 Clock Speed Set Automatically

For a CM-5E system, CMSA now automatically sets the clock source to 40 MHz — administrators no longer have to execute **fm 3 40** from a <CM-DIAG> prompt after resetting the machine. The clock speed of a CM-5 system is automatically set to 32 MHz.

Copyright © 1995 Thinking Machines Corporation

10.9 New cmstatus Command

The new **cmstatus** command displays CM system hardware information:

```
/usr/etc/cmstatus [[-d] | [-p partition_name]] [-l] [-m]
```

cmstatus displays a list of system-level information:

- The system name
- The system-level hardware flags
- The default system fifo size

In addition, for each ALLOCATED or ACTIVE partition, the following information is presented:

- Partition name
- PM name
- PN ranges
- Type (either timeshared, boot, or diagnostic)
- State (either allocated or active)
- Partition DR fifo size on H3+ DR nodes
- PN backplane clock source (either Half PLL, External Osc, or Default Osc)
- PN backplane clock speed
- The parsed partition hardware flags

For information about the **cmstatus** command's arguments and the output format, please refer to the **cmstatus** manual page.

10.10 False Partition Test Group Errors

Prior to this release, the partition group in the **cmdiag** verifier test collection would fail in configurations in which the control processor clock rate was faster than the node clock rate — for example, if a partition contained a SPARC20 control processor and ROSS processing nodes. In such cases, errors reported by the partition test group were not reliable.

Copyright © 1995 Thinking Machines Corporation

This bug is fixed in CMOST Version 7.4.

10.11 New /man/mancm/mount.8 Man Page

An adapted version of the SunOS **mount**(8) manual page has been added to the CM man pages. The CM version adds **sdafs** as a recognized file system type and identifies the options that are valid for **sdafs** file systems. In all other respects, the man page is identical to the Sun OS version.

11 Information for Users

11.1 Naming Functions and Variables

The following undocumented internal functions and variables in the CMNA libraries do not follow Thinking Machines Corporation's naming convention:

```
BASE_OF_NI
connect_cmni
init_mem_map
interrupt_occured
ni_init_pe
ni_init_sp
pe_report_timer
svme_send_first_double
table_entry
write_register
write_register_d
```

In your programs, please avoid defining any functions or variables with these names. In a future release, these symbols may be made static or be renamed to adhere to the CM-system naming convention of starting with "**CM**" or "**cm**".

Copyright © 1995 Thinking Machines Corporation

11.2 New CMOS _longjmp Man Pages

This release adds man pages for the three CMOS versions of **_longjmp**: **CMOS_timer_handler_longjmp**, **CMOS_async_handler_longjmp**, and **CMOS_message_handler_longjmp**. They are in **<mancm/man3>**.

These routines were released under CMOST Version 7.3, but man pages for them were not released until now. The **CMOS_signal**(3) man page has also been modified to include references to these functions.

Copyright © 1995 Thinking Machines Corporation

Appendix A: DPAS Version 1.2 Release Notes

New Features:

1. The new **-W** switch gives a warning when padding instructions are inserted.

2. Output line directives into **as** source (as long as input is not **stdin**) so that **as** will report correct DPEAC source line numbers in its error messages.

3. Pad using **TRAP** rather than **FNOPS**; **TRAP** is safer. This relies on CMOST guaranteeing that the debug trap is always disabled. **FNOPS** is not a good padding instruction because it is not completely a **NOP** (it shifts the Vector Mask by one), and so sometimes had to be replaced by the much slower **DPSYNC**. The **TRAP** instruction is the same number of cycles of padding and does not have the side effect. Some code speedup may be observed as a result.

4. Implement post-padding (padding instructions inserted after certain instructions). See item 5 for default postpad cases. A new modifier, **;NOPOSTPAD**, has been added, which disables any default postpadding for the DPEAC instruction in which it is given.

5. Insert default postpadding instructions in the following cases:

    a. After all microcoded instructions. This fixes a bug where such microcoded instructions caused hazards (TMCaa03264).

    b. After arithmetic operations accompanied by single stores. When such instructions are followed by double stores (even if they are pre-padded), hazards could result. Note that this probably represents a fairly serious performance penalty in single-precision pipes where stores are piggy-backed on arithmetic operations. Use of the **;NOPOSTPAD** modifier is recommended where such instructions are not followed by double-precision stores.

Copyright © 1995 Thinking Machines Corporation

6. The old **-d** switch is obsolete and is not present in DPAS Version 1.2. (DPAS Version 1.1 had warned that **-d** would go away.)

7. The new modifier **;NOWARN** suppresses warnings for that DPEAC instruction.

Bugs fixed:

1. Fixed the DPEXECD bug, which generated "`ld [%reg+(+user_)expression+4`", which sometimes would not assemble because user_expression could be a link-time expression containing **%hi** or **%lo** (TMCaa04734).

2. Undefined symbols in preprocessor statements, like CPP, are now handled correctly in the internal preprocessor (TMCaa03954).

3. DPAS now warns when an open-comment delimiter ("/*") is encountered within a comment (TMCaa04497).

4. Fixed bug TMCaa04507: In the CPP preprocessor, if a symbol (macro) is defined using the **-D** switch, but the "**=value**" syntax is omitted, i.e., **-Dfred** (rather than **-Dfred=3**), the symbol is now defined as "**1**". DPAS Version 1.1 defined the symbol as a blank string, causing incompatibility. (The DPAS preprocessor is intended to be upwardly compatible with CPP, so this was considered a bug, and DPAS has been changed to act like CPP.)

   NOTE: This subtle change in behavior could affect the way existing programs compile.

5. Fixed the bug that occurred when a SCALAR double-precision POPCOUNT is used alone, i.e., not in conjunction with a double-precision arithmetic operation (TMCaa04270). Bad code had been generated because DPAS did not realize that the source register is double precision. Source example:

   ```
   depcs(S4)=S8
   ```

   In this case, the **rLS** field of the instruction should be 2, taken from the source register number, R4, shifted right by one because of the double precision. Because DPAS did not realize that the source register is double precision, it was not shifted, and thus would cause R8 to be used as the source.

Copyright © 1995 Thinking Machines Corporation

Sometimes an error message would result. For example, if the source was given in the range S16 through S30, DPAS would give an error message. (Because DPAS thought that the operand was single precision, it would not recognize these as as valid registers.)

6. Fixed the bug wherein certain triadic (3-operand) instructions do not always generate the correct rLS stride field in the generated code (TMCaa04982). The instructions that have this problem have the following characteristics:
   - They use the true triadic form (*op* **`rS1,rLS,rS2,rD`**).
   - They use the Register Stride format.
   - They do not specify the **`rLS`** stride (which is defined to default to 1 for single-precision operations, and 2 for double-precision operations),
7. Fixed bug TMCaa04302: DPAS no longer gives a segmentation fault when the vector length is given
   - explicitly (which implies the Mode Set instuction format)
   - as a VU register, and
   - on BOTH the arithmetic and memory parts of the instruction

   For example:

   ```
   faddv*%i1 V1,V2,V3; floadv*%i1 [%l3],V1
   ```

8. DPAS now accepts `#INCLUDE` file names with any legal UNIX syntax. Previous DPAS versions did not handle special characters correctly (TMCaa04273).
9. DPAS now gives a friendlier error message when 8 or 9 is used in an octal constant (constant with leading zero or 0o at the front), or 2−9 is used in a binary constant (0b at the front).
10. Fixed bug TMCaa03996: In lexical statements, DPAS no longer recognizes "`!`*text*" as a comment. This change may cause an incompatibility with previous versions of DPAS because it was fairly common in previous versions to use "!" as a comment character, especially in `#define` lines:

    ```
    #define nbits 3     !three-bit field
    ```

    "`!`" comments are not compatible with CPP (which the internal preprocessor strives to be). Also, "`!`" is syntactically ambiguous, because

Copyright © 1995 Thinking Machines Corporation

"`!`" is a legal expression operator — logical NOT. Thus, with this bug, the following could not be coded:

```
#if !debug
```

Such comments must be removed or changed to C-style comments:

```
/* text */
```

11. DPAS no longer dies mysteriously if a macro name is either missing or not a symbol in the `#macro` directive offered by the internal preprocessor (TMCaa03953).

12. Fixed bug TMCaa04160: DPAS now specifically disallows immediate as a first argument in a convert operation, e.g., **`itodfs 100,S2`** is now illegal. DPAS previously just silently generated bad code.

13. Fixed bug TMCaa03950: In evaluation of expressions, DPAS does not simply treat symbols (usually the result of a missing `#DEFINE`) as zero, but remembers that the result is indeterminate. However, DPAS handles logical OR properly, so that "`#if (1 || smergle)`" is the same as "`#if 1`" even if `smergle` is undefined.

14. Fixed bug TMCaa05401: If a double-precision triadic was used with an explicit `MEMNOP`, an erroneous error would be given. For example,

    ```
    dfmsrtv S2:0, V10, V4, V10; memnop;
    ```

    Odd **`rLS`** stride for triadic operand (stride defaults to 1 from mem operation). The CMF compiler could generate this message, so this error could have been given during a CMF compilation.

15. DPAS now detects the use of an invalid keyword to the `VMMODE` modifier. Previous versions of DPAS did not detect the case where a particular keyword was not valid in that context.

Copyright © 1995 Thinking Machines Corporation