Chapter 1

Introduction to the CM-5 System

CM-5 user application programs run in part — or, sometimes, all — of the CM-5 system, configured from available hardware resources by the system administrator. In addition to managing the machine's configuration, the system administrator brings the system up and takes it down, maintains the system's I/O devices, runs diagnostic tests, manages the error logging system, and performs software upgrades. These tasks are discussed in other chapters; the requisite background information and CM-5-specific vocabulary are presented in this chapter.

The administrative procedures described in this manual pertain to all models of the CM-5, including the CM-5E and CM-5 Scale 3, running CMOST (Connection Machine Operating System — Timeshared) Version 7.4. (Administratively, these three CM models are synonymous, unless otherwise noted.) CMOST Version 7.4 is based on SunOS 4.1.3_U1 (Solaris 1.1.1 Revision B).

1.1 Hardware Components

Figure 1 shows two typical CM-5 systems — a full-sized CM-5 with five cabinets, and a CM-5 Scale 3 (henceforth referred to as a *Scale 3*).

- Each cabinet in the CM-5 / CM-5E is either a *device cabinet* (see Section 1.1.2) or a *network cabinet* (see Section 1.1.1). Theoretically there is no limit to the number of cabinets in a system; a minimal system contains one network cabinet and one device cabinet (there must be one network cabinet for each four device cabinets). There is no difference in outward appearance between a network cabinet and a device

Copyright © 1995 Thinking Machines Corporation

cabinet. The cabinets contain some or all of the following modules and interfaces:

- processing node (PN) modules
- Control Network, Data Network, and Diagnostic Network modules
- control processor (CP) modules and/or interfaces (see Section 1.1.3)
- optionally, one or more I/O devices or interfaces to them (see Section 1.1.4)

The CM-5E contains upgraded PNs and network hardware and can operate at a faster clock speed.

- A Scale 3 consists of only one cabinet, which contains network hardware, processing nodes, CP interfaces, and possibly a Scalable Disk Array and/or an Integrated Tape System (I/O devices described in Section 1.1.4).

The components of the CM-5 and Scale 3, described in the following subsections, are logically configured into one or more *partitions*, a complete task-execution system that runs application programs. Chapter 3 discusses partitioning in detail.

1.1.1 Network Hardware

The three networks serve the following purposes:

- The Data Network (also known as the data router) handles point-to-point (PN ↔ PN and PN ↔ I/O device) transfers. The CM-5E has a newer version of the Data Network. This network and the Control Network are invisible to users except when using a low-level communications package such as CMMD; even then, the interaction is indirect.

  The Network Interface (NI) is the interface between the Data Network and the Control Network (see below) and the PNs. The CM-5E has a newer version of the NI.

- The Control Network handles multi-PN data transfers.

Copyright © 1995 Thinking Machines Corporation

- The Diagnostic Network is used by system administrators to isolate any hardware component and to test both the component itself and all connections to other components. The Diagnostic Network is also used to set up partitions and boot the partition's processing nodes. This network is completely invisible to users.

The three networks pervade the entire Scale 3 cabinet, and they pervade all the cabinets in the CM-5 system. However, in a full-sized CM-5, an entire cabinet — a *network cabinet* — contains the network modules that form the *uppermost* levels of their respective trees.

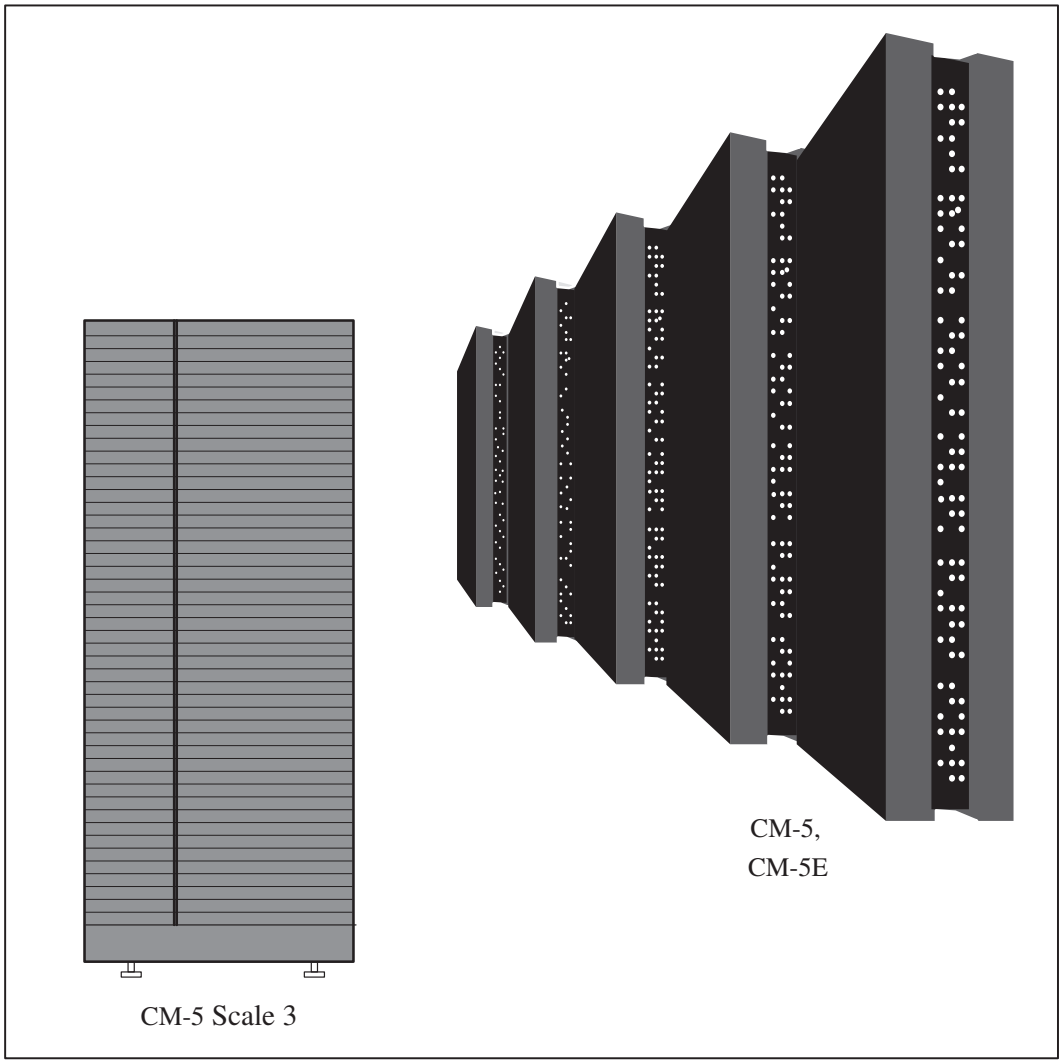

Figure 1. Typical CM-5 systems.

Copyright © 1995 Thinking Machines Corporation

The number of backplanes in a CM-5 network cabinet depends on how large the machine is; that is, it depends on how many processing nodes the machine contains — and, therefore, on how many branching levels are required to network them together.

1.1.2 Processing Nodes

Each processing node is a general-purpose computer that can fetch and interpret its own instruction stream, execute arithmetic and logical instructions, calculate memory addresses, and perform interprocessor communication. The PNs can perform independent tasks or collaborate on a single problem.

Each PN contains the following components:

- One microprocessor, which executes the operating system environment and the scalar portion of the user application code, and manages communication with other system components via the network interface. The microprocessor runs a small CMOST kernel.

  The CM-5 implementation of the PN uses a SPARC microprocessor; the CM-5E implementation uses a SuperSPARC microprocessor.

- A memory-controller function unit. This unit may be implemented as a set of 4 vector units (VUs) to enhance high-performance arithmetic operations (parallel vector, integer, floating-point, bit, and logical instructions).

- Local memory (minimum of 16 Mbytes).

- One interface to the Data Network and one to the Control Network.

In a Scale 3, the processing nodes reside all together in a single backplane. In a CM-5, the processing nodes reside in the many backplanes that make up a *device cabinet*. There are two kinds of device cabinets, each containing 10 backplanes:

- Some device cabinets contain (for all practical purposes) nothing but circuitry for processing nodes and the networks.

Copyright © 1995 Thinking Machines Corporation

- Some device cabinets may contain, in addition to circuitry for processing nodes and networks:
  - an interface to the control processor(s)
  - Scalable Disk Array (SDA) hardware (e.g., disks and SCSI channels)
  - Integrated Tape System (ITS) hardware
  - CM5-HIPPI hardware
  - an IOBA, the interface to a CMIO bus and its associated I/O devices (e.g., the DataVault)

1.1.3 Control Processors

The control processors are SPARC-2 or SPARC-10 (model 40) computers running CMOST. Both the SPARC-2 and the SPARC-10 have 64 Mbytes of memory. A SPARC-2 has 848 Mbytes of disk storage, while a SPARC-10 has 1.05 Gbytes. A typical CM-5 has between two and six CPs, depending on the configuration needs of the site. A Scale 3 supports up to two CPs.

Each CP can serve as a partition manager (PM) and/or an I/O control processor (IOCP):

- A PM manages user-code execution, coordinates interpartition communication, and oversees the partition's timesharing and batch operations. There must be one PM for every CM-5 or Scale 3 partition; each PM can support only one partition. The system administrator assigns a CP to be a partition's PM when configuring the partition (described in Chapter 3).
- An IOCP either performs I/O control for one or more CM-5 file systems (see Section 1.1.4) or provides an interface to tape or FDDI. The CM-5 system can support any number of IOCPs. The SDA requires one IOCP per logical device; if your system has no SDA, no IOCPs are required.

The System Control Processor

One of the CPs is connected at installation by Thinking Machines Corporation to the root node of the Diagnostic Network. This CP is called the *system*

Copyright © 1995 Thinking Machines Corporation

control processor, often abbreviated as *syscon* (the term *system administration console* is synonymous). Certain system programs must run on the system control processor — **`jtagserver`** (a diagnostic program), in particular.

In addition, certain administrative tasks must be performed from the system control processor, including:

- configuring and managing partitions
- executing diagnostic tests
- resetting the system
- running the error-logging system
- managing system software loading

The system control processor can also serve as a PM and/or IOCP.

This manual uses the prompt **`syscon#`** in sample command lines to denote that the command must be executed on the system control processor.

1.1.4 I/O Devices

The CM-5 supports a variety of I/O devices, including both disk-based and tape-based devices and support for VME, HPPI, and FDDI protocols. Each of the CM-5 I/O devices — the Scalable Disk Array (SDA), CM5-HIPPI, CM-HIPPI, Integrated Tape System (ITS), DataVault, VMEIO host computer, and CM-IOPG — is described later in this section. Figure 2 shows a typical CM-5 system that includes I/O peripherals. (To learn about these devices from a user's perspective, see the *CM-5 I/O System Programming Guide*.)

The I/O peripherals use the Control Network and the Data Network or a combination of the networks and external buses to communicate with the CM-5 and each other:

- The SDA has a direct interface to the CM-5 processors via the CM-5's Control Network and Data Network.
- The CM5-HIPPI and ITS have a direct interface to the CM-5 processors via the Data Network.

Copyright © 1995 Thinking Machines Corporation

- The DataVault, VMEIO host computer, CM-IOPG, and CM-HIPPI use Thinking Machines Corporation's proprietary high-speed multidrop CMIO bus to convey data among themselves and to and from the PNs; an Ethernet conveys control and status information. The I/O bus adapter (IOBA) board is the interface between the CM-5 and the CMIO bus.

Note that although the Scale 3 supports the Scalable Disk Array (SDA), it does not support the CMIO bus, so it does not support the DataVault or other devices requiring an IOBA connection. The Scale 3 also does not support the CM5-HIPPI at this time.

Part II and Part III of this manual describe in detail how to configure and manage the SDA, DataVault, VMEIO host computer, and CM-IOPG.

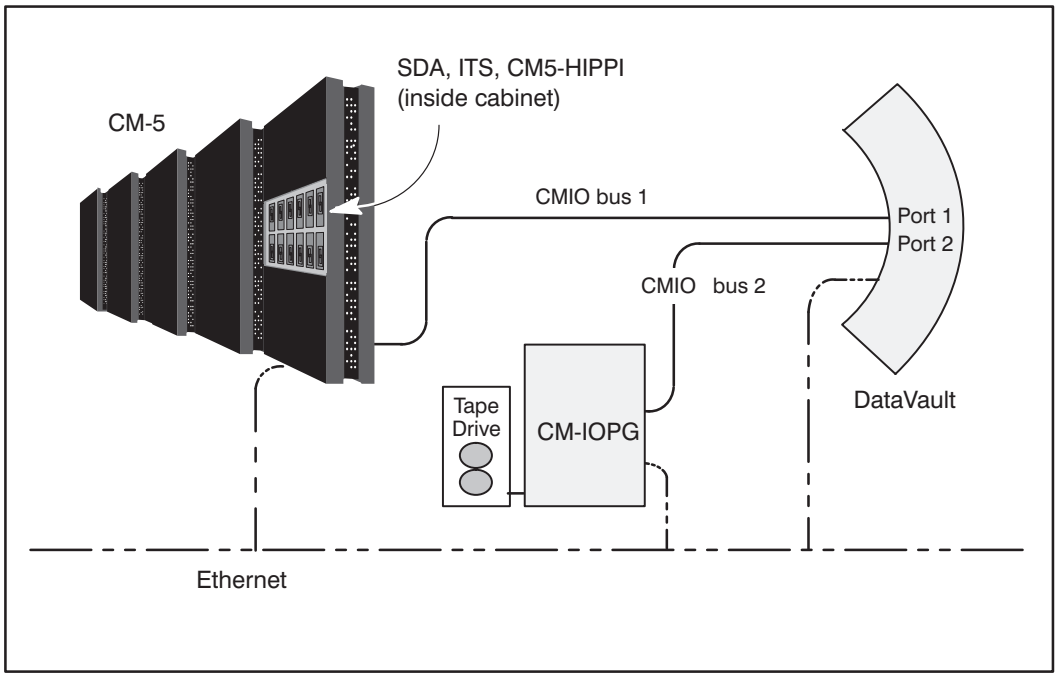

Figure 2. A typical CM I/O system for the CM-5.

Copyright © 1995 Thinking Machines Corporation

Scalable Disk Array (SDA)

The Scalable Disk Array (SDA) is the CM-5 system's main storage device, typically providing 25–200 Gbytes[1] of disk storage at I/O bandwidths of 33–264 Mbytes/second. The SDA, which resides within the CM-5 cabinets, is a collection of sets of high-speed commodity disks directly connected to the CM-5 networks. The direct connection to the networks enables each set of disks to contribute not only to storage capacity but also to I/O performance; the number of the disk sets in an SDA system can be increased or decreased to achieve an I/O system matched to the performance and capacity needs of CM-5 applications.

The SDA is made up of Disk Storage Nodes (DSNs); note that the term *DSN* is synonymous with the term *SCSI Controller Node* (*SCN*). Each DSN (or SCN) contains eight 1.2-Gbyte SCSI disk drives, four SCSI channels, an 8-Mbyte data buffer, a 20-Mbyte/second network interface, and a SPARC processor with its own 8-Mbyte memory. Each DSN (or SCN) runs a controller microkernel that mediates the I/O between the CM-5 and the drives.

DSNs/SCNs are packaged in units of three, making up a backplane. Therefore, systems have disks in quantities of 24, 48, 72, etc. Typical configurations might use 22 or 46 or 70 data disks, one parity disk, and one spare disk. By adding more DSNs/SCNs, you add a balanced amount of bandwidth from the disk through the network interface. By increasing the number of Data Network addresses, which happens automatically when you add more DSNs/SCNs, you increase the overall Data Network bandwidth as well.

The SDA's UNIX-compatible file system is called SFS (Scalable File System). Its file server resides on an IOCP. A single SDA is configurable to support any number of separate logical devices, each of which can support a separate SFS file system. Data is striped over all the disks in a file system, transparently to the user. Parity is checked (and single-bit errors are corrected, if necessary) on each read operation.

The SDA Command Center (SDACC) is a complete, easy-to-use interface to the Scalable Disk Array's administrative/diagnostic command set. The SDACC and the SDA administrative and diagnostic tasks are discussed in Part II of this manual.

1. The capacity of a Scale 3's SDA can be either 25 Gbytes (using 1.2-Gbyte drives) or 50 Gbytes (using 2.0-Gbyte drives).

Copyright © 1995 Thinking Machines Corporation

DataVault

The *DataVault*, another of the Connection Machine system's main storage devices, provides up to 60 gigabytes of disk storage. The DataVault is dual-ported to allow high-bandwidth data transfer and system flexibility. The *DataVault file server computer*, a microcomputer inside the DataVault, hosts the DataVault's UNIX-like file system, CMFS.

The DataVault's administrative/diagnostic commands are very similar to those used by the SDA. For information about DataVault administration, see Part III of this manual.

Integrated Tape System

The CM-5 Integrated Tape System provides fast, cost-effective tape backup, archiving, and data transfer. The ITS is frequently used for staging data from tape storage onto the SDA for subsequent use by CM-5 applications. The ITS supports a variety of tape drives, including 3480-compatible and 8-mm units.

Like the SDA, the ITS resides within the CM-5 cabinets (in fact, SDA and ITS modules reside in the same backplane). Data transfers between the ITS and SDA media are carried out directly via the CM-5 networks under control of I/O software running on the ITS; no CM-5 partition resources are involved in the transfers. Thinking Machines provides several performance-enhanced CMOST utilities for transferring files between the SDA and tape devices that use various standard tape formats. In addition, the ITS supports the use of standard SunOS tape commands.

This manual does not describe ITS administration; please refer to the *ITS System Administrator's Guide*.

CM5-HIPPI and CM-HIPPI

The CM-5's High Performance Parallel Interface (CM5−HIPPI), fully compliant with the ANSI HIPPI standard, provides a high-speed link between the CM-5 and other computing systems, sending and receiving information at peak rates approaching 100 Mbytes/second. The CMOST operating system supports both low-level HIPPI primitives, including HIPPI-FP and HIPPI-PH, and the TCP/IP protocol suite, modified for parallel processing and HIPPI. CM5-HIPPI programming is done through calls to CMOST I/O system functions. This manual does not address CM5-HIPPI administration; see the *CM5-HIPPI System Administrator's Guide*.

Copyright © 1995 Thinking Machines Corporation

The CM-5 also supports the CM-HIPPI, which resides on a CMIO bus; it is programmed through calls to the CMFS library. This manual does not address CM-HIPPI administration; see the *CM-HIPPI User's Guide for the CM-2*, Version 6.1.

VMEIO Host Computer

The VMEIO host computer typically is a Sun computer that contains a VME interface board connecting the Connection Machine systems to VME-based I/O devices, such as video frame grabbers and magnetic tape drives. Such devices provide convenient means of generating graphical data, and backing up and retrieving data, respectively. The CMFS file system that resides on the VMEIO host computer treats each device as a file and allows the host computer itself to provide secondary disk storage for files.

CM-IOPG

The CM-IOPG, the Connection Machine general I/O processor, also supports the VME protocol. Typically, though, the CM-IOPG connects storage devices that have SCSI drives — such as an IBM-3480−compatible tape drive — to the Connection Machine systems. As on the VMEIO host computer, the CMFS file system that resides on the CM-IOPG treats each device as a file and allows the CM-IOPG itself to provide secondary disk storage for files.

1.2 CMOST Software

The CMOST software is installed from a set of CM software release tapes. The installation is performed by a representative of Thinking Machines Corporation.

Table 1 identifies the software modules available; the specific modules included on your CM-5 system depend on which have been purchased. The CMOST base software resides in **`/`** and **`/usr`** on each CP. Note that the CMOST kernel replaces the SunOS **`/vmunix`**. The compilers and the program support libraries are also installed on each control processor.

Copyright © 1995 Thinking Machines Corporation

Table 1. Software modules available from Thinking Machines Corporation.

Module	Description
CMOST Base Software	CM Operating System.
CMRTS	CM Run-Time System (always shipped with CMOST).
CMview	The CM-5 on-line documentation viewing system.
CMX11	CM graphics routine library (X11-compatible); Motif libraries and window manager.
CM Fortran Compiler	Thinking Machines Fortran (F90-compatible).
CMMD	Connection Machine Message-Passing Library.
CMSSL	Connection Machine Scientific Software Library.
*Lisp	Common Lisp with extensions to support data parallel programming on the CM system.
Prism	Interactive graphical programming/debugging environment.
pndbx	Processing node debugger.
C* Compiler	C* is an extension of the C programming language designed to support data parallel programming.
CMAX	CM Automatic Translator (F77 to F90).
CM/AVS	Advanced Visualization System for the CM-5.

Unsupported Software:

Thinking Machines X Window System Support	Precompiled version of X Window System R4 (base); WCL.
X Window System R4 Source	Sources for X Window System and WCL software.
GNU Bin and Source	Binary and sources for EMACS, MAKE, GCC.

Copyright © 1995 Thinking Machines Corporation

1.3 Job Management Systems

The CM-5 and the CM-5 Scale 3 support two job management systems:

- Network Queueing System (NQS)

  NQS is a standard batch system — programs are not executed directly, but instead are submitted to a queue. The batch system in turn submits the requests in the queue for execution.

  NQS supports two kinds of queues. Batch queues submit requests directly to the CM-5 for execution, while pipe queues pass requests along to batch queues. Pipe queues can be associated with several different batch queues; if one is unavailable, the pipe queue can try the next, and then the next, until it finds one that will accept the request.

  For specific information about NQS, please see the Thinking Machines manual, *NQS for the CM-5*.

- Distributed Job Manager (DJM)

  DJM was initially developed at the Minnesota Supercomputer Center to manage the flow of jobs through the CM-5. It is designed to avoid resource conflicts and provide load balancing among the partitions.

  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. DJM ensures that it handles all jobs by trapping jobs that are invoked directly instead of through DJM. It has the ability to penalize such trapped jobs (which it calls *foreign* jobs).

  For specific information about DJM, please see the Thinking Machines manual, *Using DJM on the CM-5*.

Both systems allow both users and administrators much 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 default parameters. They can ask that the system send them mail when a job begins or ends execution. They can move a job from interactive execution to batch execution, 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

Copyright © 1995 Thinking Machines Corporation

allow dedicated access to the system, and when to allow it. They can determine how foreign jobs are to be handled.

DJM is compatible with NQS at the user level in its handling of requests for batch job submissions. Indeed, the system administrator can configure DJM so that user-level NQS command will work with it. Table 2 gives some examples of how DJM and NQS commands provide similar functionality.

Table 2. DJM and NQS user commands.

DJM Command	NQS Command	Description
`jsub`	`qsub`	Submits a job for background processing.
`jrun`	---	Submits a job for interactive processing.
`jdel`	`qdel`	Deletes or signals a DJM job.
`jstat`	`qstat`	Displays status of DJM system.
`jconnect`	---	Connects a job to a terminal (tty).
`jmsg`	---	Sends a message to a job's output buffer.
`js`	---	Displays system-wide status of DJM system.
`jwait`	---	Waits for a DJM job to start or finish.

Copyright © 1995 Thinking Machines Corporation