Chapter 2

The Basic Components of the CM-5

At its best, parallel processing brings many processors, working in close coordination, to bear on large quantities of data. An effective parallel-processing system must provide a large amount of memory to hold this data and must provide effective access to the data for hundreds or thousands of processors. The CM-5 system meets this goal. Moreover, it allows its memory and processor resources to be applied equally effectively to a single large problem or to job requests from dozens of simultaneous users.

Traditional computer architectures, such as the generic system diagrammed in Figure 4, link one or a few processors to a shared memory via a system bus. This worked well when processing speeds were slower and the number of processors was small. Nowadays it is much more cost-effective to use many processors than to try to make the processors faster. With many processors, a simple bus is a bottleneck, and the complex switches that can provide fast access to a shared memory for every memory reference are both expensive and complicated. Two more changes to the early model are therefore needed to balance communication speed with processing speed: memory must be distributed, rather than shared; and a high-bandwidth network, rather than a bus, must be used. Figure 5 diagrams this second architecture as it appears in the CM-5.

2.1 Processors

A CM-5 system may contain tens, hundreds, or thousands of parallel processing nodes. Each node has its own memory. Nodes can fetch from the same address in their respective memories to execute the same (SIMD-style) instruction, or from individually chosen addresses to execute independent (MIMD-style) instructions.

Copyright © 1993 Thinking Machines Corporation

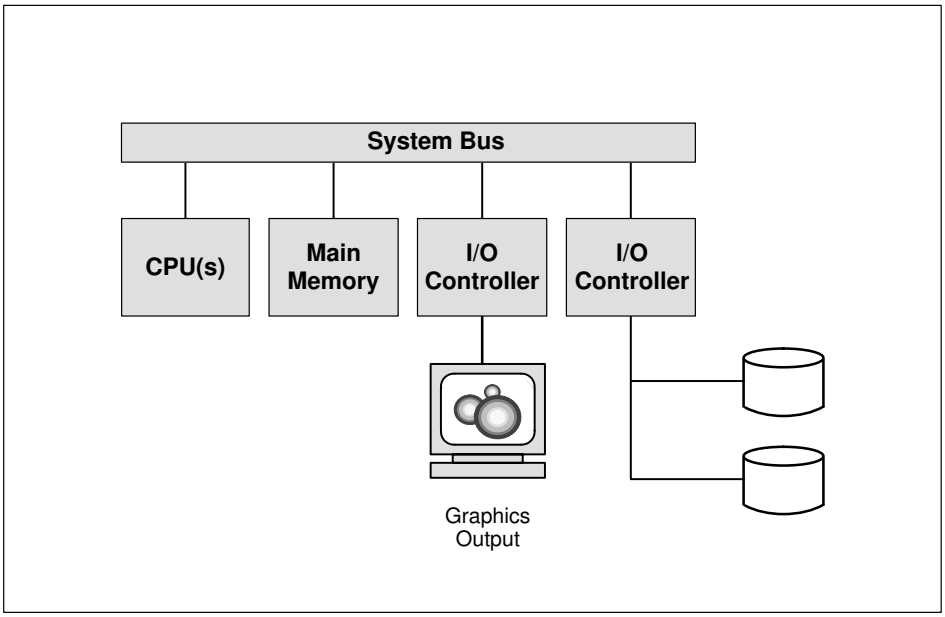

Figure 4. Organization of a traditional computer.

The processing nodes are supervised by a control processor, which runs an enhanced version of the UNIX operating system. Program loading begins on the control processor; it broadcasts blocks of instructions to the parallel processing nodes and then initiates execution. When all nodes are operating on a single control thread, the processing nodes are kept closely synchronized and blocks are broadcast as needed. When the nodes take different branches, they fetch instructions independently and synchronize only as required by the algorithm under program control.

To maximize system usefulness, a system administrator may divide the parallel processing nodes into groups, known as *partitions*. There is a separate control processor, known as a *partition manager*, for each partition. Each user process executes on a single partition, but may exchange data with processes on other partitions. Since all partitions utilize UNIX timesharing and security features, each allows multiple users to access the partition while ensuring that no user's program interferes with another's.

Other control processors in the CM-5 system manage the system's I/O devices and interfaces. This organization allows a process on any partition to access any I/O device, and ensures that access to one device does not impede access to other

Copyright © 1993 Thinking Machines Corporation

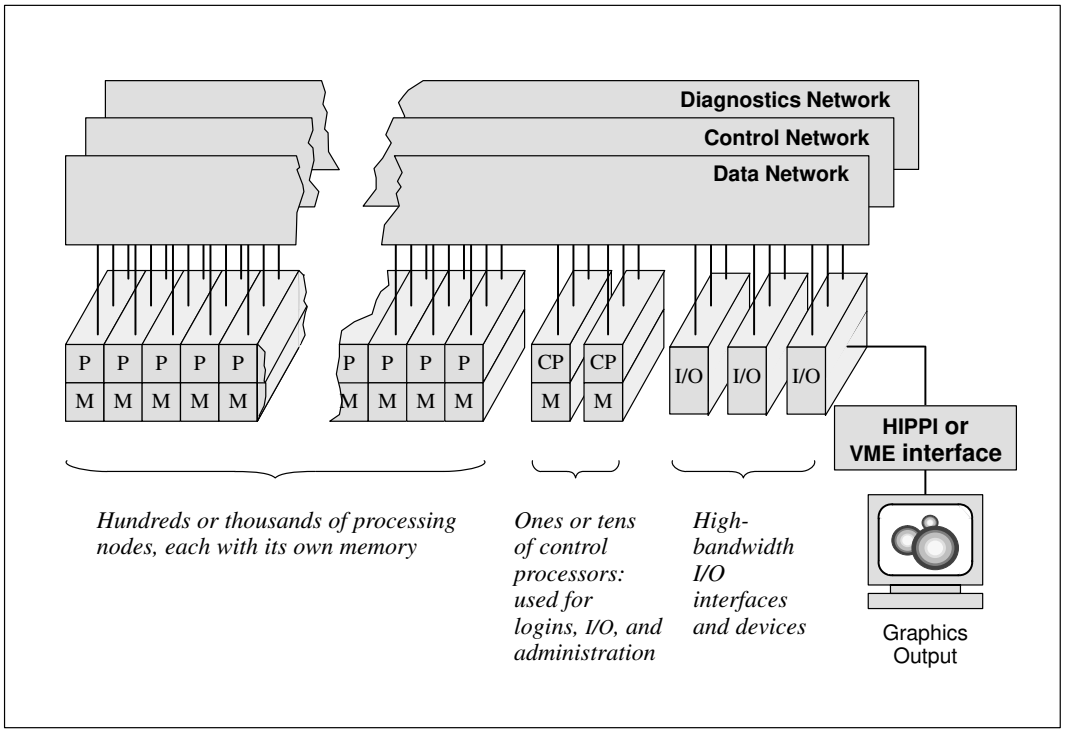

Figure 5. Organization of the Connection Machine system.

devices. (Figure 6 shows how this distributed control works with the CM-5's interprocessor communication networks to enhance system efficiency.)

2.2 Networks

Every control processor and parallel processing node in the CM-5 is connected to two scalable interprocessor communication networks, designed to give low latency combined with high bandwidth in any possible configuration a user may wish to apply to a problem. Any node may present information, tagged with its logical destination, for delivery via an optimal route. The network design provides low latency for transmissions to near neighboring addresses, while preserving a high, predictable bandwidth for more distant communications.

The two interprocessor communications networks are the Data Network and the Control Network. In general, the Control Network is used for operations that involve all the nodes at once, such as synchronization operations and broadcasting;

Copyright © 1993 Thinking Machines Corporation

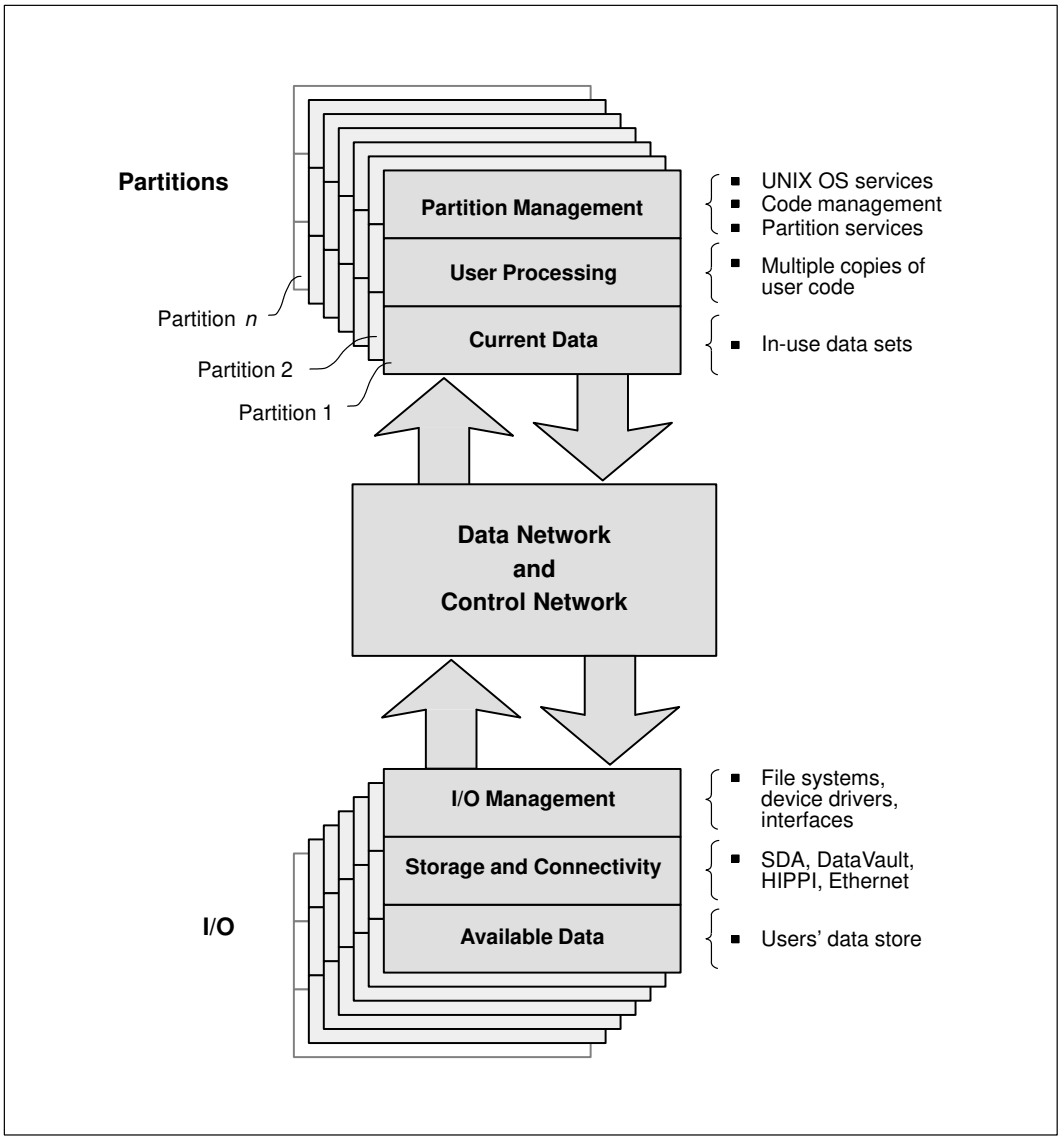

Figure 6. Distributed control on the CM-5.

Functionally, the CM-5 is divided into three major areas. The first contains some number of partitions, which manage and execute user applications; the second contains some number of I/O devices and interfaces; and the third contains the two interprocessor communications networks that connect all parts of the first two areas. (A fourth functional area, covering system management and diagnostics, is handled by a third interprocessor network and is not shown in this drawing.)

Because all areas of the system are connected by the Data Network and the Control Network, all can exchange information efficiently. The two networks provide high bandwidth transfer of messages of all sorts: downloading code from a control processor to its nodes, passing I/O requests and acknowledgments between control processors, and transferring data, either among nodes (whether in a single partition or in different partitions) or between nodes and I/O devices.

Copyright © 1993 Thinking Machines Corporation

the Data Network is used for bulk data transfers where each item has a single source and destination.

A third network, the Diagnostics Network, is visible only to the system administrator; it keeps tabs on the physical well-being of the system.

External networks, such as Ethernet and FDDI, may also be connected to a CM-5 system via the control processors.

2.3 I/O

The CM-5 runs a UNIX-based operating system; it provides its own high-speed parallel file system, and also allows full access to ordinary NFS file systems. It supports both HIPPI (HIgh-Performance Parallel Interface) and VME interfaces, thus allowing connections to a wide range of computers and I/O devices, while using standard UNIX commands and programming techniques throughout. A CMIO interface supports mass storage devices such as the DataVault and enables sharing of data with CM-2 and CM-200 systems.

I/O capacity may be scaled independently of the number of computational processors. A CM-5 system of any size can have the I/O capacity it needs, whether that be measured in local storage, in bandwidth, or in access to a variety of remote data sources. Communications capacity scales both with processors and with I/O. Customers may choose both the processing power and the I/O capabilities that meet their needs, and the CM's communications capacity is automatically scaled to match.

Just as every partition is managed by a control processor, every I/O device is managed by an input/output control processor (IOCP), which provides the software that supports the file system, device driver, and communications protocols. Like partitions, I/O devices and interfaces use the Data Network and the Control Network to communicate with processes running in other parts of the machine. If greater bandwidth is desired, files can be spread across multiple I/O devices: a striped set of eight DataVaults, for example, can provide eight times the I/O bandwidth of a single DataVault.

The same hardware and software mechanisms that transfer data between a partition and an I/O device can also transfer data from one partition to another (through a named UNIX pipe) or from one I/O device to another.

Copyright © 1993 Thinking Machines Corporation

2.4 A Universal Architecture

The architecture of the CM-5 is optimized for data parallel processing of large, complex problems. The Data Network and Control Network support fully general patterns of point-to-point and multiway communication, yet reward patterns that exhibit good locality (such as nearest-neighbor communications) with reduced latency and increased throughput. Specific hardware and software support improve the speed of many common special cases. Chapter 3 outlines the nature of this support, which is discussed in even greater detail in later chapters.

Two more key facts should be noted about the CM-5 architecture. First, it depends on no specific types of processors. As new technological advances arrive, they can be moved with ease into the architecture. Second, it builds a seamlessly integrated system from a small number of basic types of modules. This creates a system that is thoroughly scalable and allows for great flexibility in configuration.

November 1993
Copyright © 1993 Thinking Machines Corporation