Multiple Minicomputers: Inexpensive and Reliable Computing

Large computers are expensive to buy, and they have the unfortunate habit of breaking down, often when only a small part of the overall system fails. However, some computer scientists are now saying that for a large class of problems cheaper and more reliable computers with a computational power equivalent to that of large machines can be constructed by connecting a number of minicomputers together. Problems that may be handled by multiple minicomputer systems range from such specialized jobs as controlling the flow of information in a communications network or controlling airport traffic to such general purpose jobs as managing large amounts of data.

The idea of having separate processors (the part of the computer in which programs are executed) for carrying out separate tasks is already well entrenched. For example, in some existing computers a large central processor unit (CPU) handles all computation and oversees operation of the overall system and a smaller peripheral processor is responsible for controlling the various input-output devices, such as card readers, teletypes, printers, or alpha-numeric display terminals.

In the multiple minicomputer systems, however, there are many identical processors, each one able to fulfill any function it is called upon to perform, and no one processor exerting overall control of the system at all times. Each processor in computers of this type, while it is carrying out a task, has its own instructions to execute and its own data on which to operate—a so-called multiple instruction stream–multiple data stream computer. Research involving multiple minicomputers tends to fall into one of two groupings. Multiprocessors is the term used when many computers at the same location are connected in such a way that all memory is shared between all the processors. Distributed computing involves many computers located at physically separated sites with no shared memory.

As usual, economics is one of the primary motivations for considering these kinds of multiple minicomputer systems. Minicomputers are manufactured in larger volumes than large computers and hence partake of some of the cost reducing benefits of mass production. Reliability may become an even more important incentive for some applications. With many processors (as well as other components), the loss of one unit would not cause the entire system to fail. Moreover, programs that are responsible for the operation of the computer (operating system) can be written so that failed components can, in effect, be amputated from the overall system.

Other advantages conceived of by researchers include modularity and increased efficiency. A small system could be expanded, in increments, by adding more processor and memory modules. And, for the user who has many small jobs or has a job that lends itself to being divided up among the many processors, the number of jobs finished in a time period could be increased as compared with a conventional computer of comparable performance.

Researchers under the direction of Severo Ornstein at Bolt Beranek and Newman Inc. (BBN), Cambridge, Massachusetts, have adopted the multiprocessor approach in their design for an improved interface message processor (IMP) for the Advanced Research Projects Agency computer network (Arpanet). An IMP stands at each of the more than 50 nodes (sites where computers belonging to institutions participating in Arpanet are located) in the network and is responsible for organizing and routing communications between the computers at its node and those in the rest of the system. The problems faced by an IMP are like those found in any time-sharing situation. Communications that need attention are constantly arriving either from the host computer or from IMP's at other nodes. The message processor must allocate its resources so that all requests for attention are honored without too long a delay. At present, this job is being handled by a minicomputer; but, as the Arpanet has grown and as the size and needs of each computer site differ, it has become clear that faster and more reliable IMP's of different sizes are needed.

The two major problems in building a multiprocessor are designing the mechanism by which the components are connected and developing programs to operate the resulting, connected system.

At BBN, an entity called a bus coupler is used as the connection mechanism. A bus is a high speed pathway for information to flow from point to point within an electronic system and is, in this case, a collection of electrical cables or other conductors together with associated circuitry for attaching the components of the computer. Several Pluribus, as the BBN multiprocessor is called, systems have been built consisting of seven processor buses (typically two Lockheed SUE miniprocessors are attached to each bus), two memory buses (up to, but usually less than, 500,000 words of memory can be attached to each bus), and two input-output buses (to which various sorts of devices associated with input-output functions are attached). A bus coupler is used to connect buses together. One coupler is used for each processor to memory, processor to input-output, and memory to input-output connection.

Multiprocessors are also parallel processors because parts of the program are executed concurrently by several processors. No general program has been devised that can decompose an arbitrary user's program into parallel subprograms. Thus, the user has to explicitly consider the parallel organization of the computer and must switch from thinking sequentially to thinking in parallel.

In Pluribus, the program is divided into short segments called strips, none of which can take longer than a predetermined maximum time for any one processor to execute. These program segments are assigned a priority and are queued up by mechanisms called pseudo interrupt devices that are attached to the input-output buses. The pseudo interrupt devices resemble a bulletin board containing jobs to be done, and each processor checks the bulletin board for a new job when it has finished its old one. Division of the program into strips that cannot run beyond a certain time ensures that a processor will always be available to handle urgent tasks.

A slightly different multiprocessor organization is being explored by researchers under the direction of William Wulf at Carnegie-Mellon University (CMU). The Carnegie-Mellon multiprocessor design, known as C.mmp (for multi-miniprocessor), was dictated in part by the special needs of an artificial intelligence (speech understanding) project and by the desire to build a general purpose computer usable by not-too-sophisticated programmers.

The CMU researchers have avoided the proliferation of cables associated with the bus coupler concept by connecting processors (Digital Equipment Corporation PDP-11's) with memory modules (of 64,000 words each) through a central electronic switch. However, this reduces the modularity of the computer slightly because the switch itself is not entirely modular. In addition, the PDP-11 processors require that an input-output device be associated with a specific processor rather than with a common bus. Up to 16 processors are planned for the finished system, but at present only 5 are running.

Because it is intended for timesharing of several programs whose nature is not known in advance, the C.mmp cannot take advantage of two features of the Pluribus multiprocessor. In the Pluribus, since there is only one program to be run and it is well understood in advance, a small amount of local memory associated with each processor is used to store portions of the program that are known to run frequently; thus its operation is considerably speeded up.

For the same reason, in Pluribus, the program strips never need to be interrupted. But at CMU programs are divided into one or more segments of varying length called processes. Processes are allowed to run on a given processor only for a certain amount of time. These segments are parceled out to the various processors by a piece of the operating system known as KMPS (kernel multiple processing system). If a new process to be done arrives with a higher priority or if time runs out, a process in operation may be interrupted before it has been completed. The KMPS keeps track of how far the old process had been completed, sends it to be stored somewhere, and brings it back when a processor is available.

At the University of California, Berkeley, Herbert Baskin and his associates are testing a third approach to multiprocessing that emphasizes program security and continuous availability in a time-sharing system. The Prime system, as it is called, begins with sets of identical components, including processors (Digital Scientific Corporation Meta-4), primary memory, and secondary memory (mass storage such as a disk). Then connections are made between an individual processor from the processor set and a number of primary and secondary memory elements, so that a self-contained computer subsystem is constructed around each processor.

At any one instant, each subsystem is isolated and operating by itself; but in the next instant, all the parts may be exchanged through the switching connections, so that new subsystems are formed. One of the processors acts as a control to direct the activities of the others, which are busy executing programs. But, just as the composition of a given subsystem is dynamic, so also can the control of the overall system migrate among the processors.

Prime Is Fail-Soft

The dynamically changing subsystem composition and migrating control set the Prime system apart from Pluribus and C.mmp and are also responsible for the security. Since subsystems do not cooperate, there can be no security leaks. In both the Pluribus and C.mmp, all processors have access to all of the common memory, and control is distributed throughout the system. Prime is "fail-soft" as are the other multiprocessors—that is, it continues to run when a component, such as a processor, fails. For example, any one subsystem can cause the control to migrate to an uninvolved processor in the event the control processor fails.

The Berkeley researchers have been testing a multiprocessor system consisting of 3 processors, 26 memory modules of 4000 words each, and 6 disks.

It is sometimes desirable to have processors physically distributed over wide geographical areas, as, for example, when collecting experimental data or when maintaining data banks at several separated sites. A form of distributed computing that has some features of both computer networks and multiprocessors is being exploited by David Farber and his colleagues at the University of California, Irvine. Computer networks consist of a number of computers dispersed over several sites; but generally programs are executed only by one computer in the network, each computer is responsible for controlling its own operation, and there is no shared memory. In distributed computing, processors can be located far apart and there is no shared memory; but programs are partitioned into processes that may be executed by different processors within the system and control of the overall system is distributed among all the processors.

Cost could be an especially important advantage of the distributed computing concept, because sites in the distributed system would belong to different users. Each user would only have to invest in a miniprocessor, a primary memory, maybe a secondary memory, and one or more input-output devices, but the resources of the entire system would be available to him. Furthermore, the user could drop out of the network at will and later reconnect.

Each miniprocessor in the distributed computer network is connected around a high speed data ring (which is essentially an electrical cable capable of transmitting digital data at a high rate) by a special device called a ring interface. As in multiprocessors like C.mmp since all communications address the names of processes rather than their locations, care need not be taken to keep track of which processor a process is in.

When processes residing in different processors need to interact, messages are sent out around the data ring, checked by each ring interface to determine whether the message is intended for a process currently residing in its processor, and returned to the point from which it started. Because of the somewhat slower communication between processors in distributed computing than in multiprocessors, there is a tendency to minimize interprocessor communication.

Experiments are now under way at Irvine on a system consisting of five processors (three Lockheed SUE's and two Varian 620's) and a number of terminals, disks, and printers.

It is an open question as to how soon multiprocessors or distributed computer networks will appear for use by the unsophisticated. Skeptics point out that there are a number of formidable problems yet to be solved before a truly general purpose computer of this type can be claimed to have been built. Moreover, minicomputers, no matter how they may be connected, will never be able to handle certain types of problems that only a large computer can touch. However, there is no doubt that for problems that can be decomposed into parallel processes, that need highly reliable computers, or for which cost is a consideration, multiple minicomputer systems of these types may emerge as the best answer.

—ARTHUR L. ROBINSON