Western Region Technical Attachment
No. 91-25
June 25, 1991

PARALLEL PROCESSING IN FORECAST SYSTEMS LABORATORY (FSL)

[Editor's Note: This Technical Attachment (TA) was originally printed in the June 1991 issue of *<u>FSL Forum.</u>* *The authors, Bernardo Rodriguez and Roldan Pozo, are graduate students at the University of Colorado Department of Engineering and Computer Sciences. They have served as consultants for FSL.]*

Because of technological advances mostly related to circuit integration and materials, the speed of computers has increased rapidly in the last 30 years. There has been a decrease in the supercomputer cycle time from 100 nsec in the CDC 6600 in 1964, to 27.5 nsec in the CEDC 6700 in 1969, to 12.5 nsec in the Cray-1 in 1976, to 4 nsec in the Cray-2 in 1985, and there is a projected decrease to the 2- to 3-nsec range in the next generation of machines (i.e. Cray-3, Horizon). However, this increase in processor speed is not expected to continue for long: during the 15-year period from 1955 to 1970, cycle times improved by a factor of 436; recently, the trend is a two-fold improvement every 4 to 5 years. The main limits to faster technology are that information can not travel faster than the speed of light, and that processors can not be made infinitely small. These limits should be reached within the next 20 years.

Parallel processing has been used to further increase computational speed. Instead of executing a single operation at a time, several operations that form part of the same problemare executed simultaneously. For example, a vector sum of length 10,000 should take approximately 40 microseconds on a Cray-2 processor. If we use 10 of those processors, and give a vector of length 1000 to each, the sum will then take approximately 4 microseconds. There are, however, overheads in distributing the work to all the processors and in synchronizing them. Therefore, one should not expect to achieve an N-fold improvement over a single processor implementation when using N processors. The measure of this improvement is called the speed-up of the parallel program.

Currently, there are two styles of parallel computers: those that use a few of the most powerful processors available, and those that use many small processors. Generally the first type is based on vector processors that are especially designed to perform vector operations, and are connected through simple bus architectures: Cray Y-MP, IBM 3090, Fujitsu VP, NEC SX-3. The second type is based on small, inexpensive processors, generally connected through more complex networks: Intel IPSC/860, Alliant FX/2800, Meiko, Ncube. A measure of the maximum raw power of a computing platform (usually stated in MFLOPS: millions of floating point operations per second) could be calculated by simply multiplying the speed of a processor (S) times the number of those processors in the system (N), MFLOPS = S*N. Traditionally there has been the question of whether to increase S or N in order to make a more powerful computer. It is generally accapted that increasing the speed of a processor is more expensive than increasing the number of them. However, it is also accepted that it is more efficient (higher speed-up) for general computing to use fewer processors. The programming and optimization (search for high speed-ups) for each type of machine is quite different: in a computer with few processors, the emphasis is put in the optimization of code according to the individual processor/memory architecture, since the communication between them is inexpensive. In a machine with many processors the problem has to be studied and restructured according to the interconnection network topology.

We believe that in the future the leading supercomputers will have many processors. The limit to processor speed will be reached, and the only choice for building faster machines will be to increase the number of processors, even up to thousands of processors, an approach usually known as massively parallel processing (MPP).

FSL has started an effort that includes the evaluation of MPP architectures and their applicability to weather modeling. An immediate goal of this work is the use of a moderately parallel computer whose architecture will be similar to future massively parallel computers, and therefore will allow FSL to make the first steps in code production and algorithm optimization. For the past year we have been trying to determine what that architecture would be.

Three architectures have been studied: single instruction multiple data (SIMD), shared memory multiple instruction multiple data (shared MIMD), and distributed memory multiple instruction multiple data (distributed memory MIMD). Current SIMD architectures (Thinking Machines CM-2) have thousands of processors that during execution perform the same operation over different data (data parallelism). These machines are very scalable and their price/performance ratio is excellent. Difficulties include computations such as those for microphysics in atmospheric models, which are only necessary over a small portion of space. Since every processor must perform the same instruction, most of the processors could be idle while the rest do microphysics calculations.

MIMD computers can also have many processors. Generally they are more powerful than those used for SIMD machines, and the hardware to connect them is more expensive. In a MIMD machine each processor executes its own program. MIMD computers have a wider range of applications because they can exploit both data and functional parallelism: processors can work on different/same data structures, executing different/same functions on them. In an MIMD architecture, information can be shared between processors by using a common memory space (shared memory) or by passing messages between them containing the information that is needed (distributed memory). Shared memory machines are easier to program but they do not grow in number of processors as naturally as distributed memory computers, since the interconnection network is much more expensive. For this reason, computer scientists agree that by allowing a higher degree of parallelism, the maximum performance on MIMD architectures will be reached through distributed memory systems. Although the original effort in programming and optimizing applications in distributed memory MIMD supercomputers is considerable, by dedicating resources and scientists to the study of the implementation of atmospheric modeling and other algorithms on those architectures, we will gain experience in using the computing environments that will lead to high performance in weather forecast applications.

Our researrch is concentrated in two areas: the optimization of atmospheric modeling code at the system level and at the processor level. The former is mainly related to the issue of dynamic load balancing (making sure at run time that high utilization of the machine is maintained). The latter is concerned with the code restructuring techniques and memory mapping schemes that better utilize the individual processors. We are interacting with different computer manufacturers (Intel, Ncube, Meiko, Alliant, Thinking Machines), compiler designers (The Portland Group, Kuck and Associates), and university groups (University of Colorado, University of Illinois, Stanford University), to guarantee that FSL will be a leader in parallel processing for atmospheric modeling.