CM-P00056654

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN/ECP 90-4
10 August 1990

**PARALLEL COMPUTING:
SOME ACTIVITIES IN HIGH ENERGY PHYSICS**

Ian Willers
CERN, Geneva, Switzerland

Submitted to the International Journal of Modern Physics C
(devoted to Computational Physics and related subjects)

1. INTRODUCTION

The aim of this paper is to give an overview of some of the current activities involving parallel computing in High Energy Physics. A parallel architecture provides an explicit, high-level framework for the development of parallel programming solutions by providing multiple processors, whether simple or complex, that cooperate to solve problems through concurrent execution. In this paper emphasis is placed on explaining the motivation rather than the detailed technical ideas behind each project. We start by looking at the dilemma facing the computer centers and how they are influenced by new developments in the supercomputer arena. Then we look at the success story of physics event off-line farming and finish by taking a glimpse at new ideas that are being exploited in the field of on-line support for experiments. The references at the end of the paper should be pursued by the reader who requires more precise information on each project.

2. WHY IS PARALLELISM INTERESTING?

2.1 Parallelism and Supercomputers

For some years the name Cray has been synonymous with the supercomputer. The latest model, Cray Y-MP, is a shared memory multi-processor with 8 processors which still relies on vectorization to attain its peak speed. There were announcements in the financial and business press recently of the successor to the popular iPSC/80. It has 128 processors arranged in a hypercube, is known as the iPSC/860 and is based on Intel's new i860 chip. It was also noted that the Connection Machine with up to 65,535 simple processors had obtained approximately 10% of the supercomputing market. What was once a well understood segment of the computing scene now seems to lack direction.

The Cray Y-MP has a peak performance (with perfect vectorization on 8 processors) of 2.7 Gigaflups and costs between \$20m and \$30m. The Intel iPSC/860 has 128 i860 processors with distributed memory. Each processor has its own memory and communication is via messages passing over a communications structure known as a hypercube. The peak performance of the Intel iPSC/860 is 7.6 Gigaflups and costs about \$3m! The Connection Machine has 65,535 processors obeying a single stream of instructions putting it in the SIMD category. It has a peak performance of 10 Gigaflups and a price in the \$8m range.

An obvious question at this point is 'Why should I buy a Cray instead of an Intel iPSC/860 or a Connection Machine?'. One answer to this question is based on the relative difficulty of

programming each of these machines in order to obtain maximum performance. All three computers have distinctive features that make it difficult to obtain 50% or even 25% of their peak performance when running many programs. On the Cray, physicists have been unsuccessful in obtaining large gains by vectorizing large numerical codes for analyzing experimental data. However, in general, the software on the Cray is more mature, vectorization is relatively well understood, the CRAY compilers are very smart and the machine excels in problems that can be vectorized. While the iPSC/860 and the Connection Machine use less advanced software tools, the 'hard-nosed' business man believes that he can buy a lot of programming for the difference in price between them and the CRAY. The iPSC/860 has been used to successfully solve problems where the data can be distributed over many processors. And the Connection Machine has already proved itself in situations where large databases are involved. It permits its many processors to simultaneously examine database records that are stored across many disks and then accessed in a parallel fashion.

2.2 Cost Effectiveness of RISC and Superscalar High Speed Servers

It was once the pride and joy of the computer center manager to point to the Cray in the machine room. This was a status symbol that was not questioned until recently. Some scientists without computer center budgets started stringing together high performance personal workstation servers, such as the Apollo DN10000, Silicon Graphics 240 or DEC MIPS machines, and attained the same performance as they got from the Cray (see [14]). Their problems did not lend themselves to vectorization but were easily parallelized over a distributed system. So another question was asked; 'Do I need a supercomputer?'. You certainly benefit from a Cray style machine if your application is vectorizable, time critical and requires high I/O throughput. Bluntly put, if you need to predict tomorrow's weather you had better not take 24 hours to do it.

The top of the range DN10000 and Silicon Graphics servers cost in the range of \$300,000 and contain 4 processors with shared memory and have a peak performance in the 100 MIPS range. These servers can be connected by a fast local area network and work can be distributed across them. CERN has a proposal for a pilot scheme using DN10000's connected by UltraneT in the computer center. Simulation studies show very encouraging results for event analysis.

2.3 Success of Vectorization and Parallelization of Algorithms

Much of the success of matching the problem to the computer depends on the programmer's ability in vectorizing or parallelizing the code. The solution can be finding the algorithm that can be matched to the capabilities of the computer. There have been some extraordinary successes and some clear failures in this quest. Compilers and other tools aiding vectorization are now mature. It appears that the research aspect of vectorization is ending and the search for parallel solutions is commencing.

The levels at which code can be made parallel include instruction execution, small grain parallelism, large grain parallelism, tasking and sessions. Vectorization is just one example of instruction parallelization. The new superscalar processors, including the DN10000's PRISM and IBM 6000's POWER architecture, will execute unrelated instructions in parallel (see [22]). Superscalar processors speed up vectorizable code and scalar code. Small grain and large grain parallelism are based on the size of the chunks of code that can be run separately. Small grain parallelism is used effectively on shared memory systems, and large grain parallelism on distributed memory machines, where the message passing overhead has to be taken into account. 'Farming' is the execution of identical tasks running with independent data, and is a good example of task level concurrency. Finally, session level parallelism refers to users being allocated a varying number of processors, as required by their work, and is handled at the operating system level.

2.4 Physics need for computer power

A statement from Fermilab reads 'High Energy Physicists have always wanted more computer power than they could afford to buy in the commercial market place'. CERN has predicted that 600 CERN units¹ will be needed within the next two years (see [2]). This is to be divided up as 150 units in the CERN computer center, 300 units from outside organizations and 150 units from parallel farms.

The profile of computing from experiments using the LEP accelerator at CERN is shown in the graph in figure 1 which is taken from the so-called CERN Green Book (see [2]). This will take up about half of the available computing power. Also note that when the Z⁰ physics has been well understood there will probably be more demanding requirements from new LEP physics. The LHC (planned Large Hadron Collider) will continue to increase demand.

1: A CERN Unit corresponds roughly to 1/8 th. of the power of a single processor in the CRAY X-MP.

Figure 1: Total CPU needs of the LEP experiments

3. THE FARMING PARADIGM IN OFF-LINE PHYSICS

3.1 Farming in its Simplest Form

The farming paradigm has been extremely successful in applying parallelism to the analysis of physics events. The fact that every event is quite independent of any other event creates a situation which is known in computing circles as 'embarrassingly parallel'. In its simplest form a *master* program reads an event, allocates the event data to an available *slave*, collects the results from the *slave* and writes the results to tape (see figure 2). Each slave runs an identical analysis program which will handle one event after another.

Figure 2: Farming – Master controlling slaves each receiving a single physics event.

3.2 The IBM Emulators – The Original

The first farm of processors was based on IBM instruction set emulators. The idea was pioneered at SLAC and CERN (see [11]). The original motivation did not come from ideas of parallelism but from the need to preserve software investment. At the end of the 1970's it was realized that software costs were escalating while hardware costs were diminishing rapidly. The idea of the emulator was to construct an inexpensive processor which exactly emulated a subset of some popular processor's instruction set. This subset of the instruction set was chosen so that physics code could be run. Questions of I/O were handled effectively with data being supplied by a host machine. The effectiveness of this approach is demonstrated by the large number of IBM emulator farms now in use in the physics community.

The recent arrival of fast RISC processors has provided an alternative to the IBM emulator farm, if the user gives up compatibility with the IBM 3090 mainframe. CERN did not want to manufacture processors when industry was clearly capable of producing far better results using modern chip technology. CERN approached IBM to see if a joint solution could be found. The result was the Parallel Processing Computer Server, PPCS (see [9]). This machine consists of 32 IBM microprocessors that execute the IBM 370 instruction set. These processors are inter-connected using a 'crossbar' switch

which gives efficient inter-processor communication. CERN has connected the machine to a standard VME bus and from there to an IBM computer. The future of the system within IBM remains unknown and its future availability as a commercial product uncertain.

3.3 Fermilab ACP

The idea of farming was further explored in Fermilab (see [21]). In figure 3 we can see the separation of the input task from the output task. This enables parallel input and parallel output to support a larger number of slave processors arranged using a flexible crossbar switch. The separate master program is replaced by a queue manager. When data is ready to be processed it is placed on a queue. A slave processor removes this from the queue. When the analysis task terminates the results are placed on the output queue. The tasks that are writing the results to tape then take their data from this output queue.

Figure 3: Farming – the general solution

The project embarked on producing two versions of the ACP machine, each of which had an architecture that mapped well onto this more general farming scheme. The first machine used Motorola 68000 processors and the second machine was planned using the MIPS R2000 RISC processors. The ACP project was successful but, in a similar way to the emulators, the group decided to abandon the building of computers and operating systems in situations where this was clearly within the capabilities of industry to produce a cost effective solution.

The Fermilab group have more recently worked on the Advanced Computer Program Multi-Array Processor System, ACPMAPS (see [20]). This is a massively parallel floating point computer designed for lattice gauge theory and other grid orientated problems. In 1989 a 16 node machine was constructed using single board array processors based on the Weitek XL chip set arranged in a hypercube. A 256 node system is being constructed.

3.4 L3 Lepics – The Compromise

The L3 experiment, running at CERN and using the LEP accelerator, has a project which aims to balance the efforts to produce custom solutions while tracking current technology. The system, Lepics (see [6]), is shown in figure 4 and consists of Apollo DN10000's connected to an IBM 3090 mainframe. The IBM is used for its excellent data storage and I/O capabilities, and the Apollo DN10000's for their relatively cheap computing power. The hardware for the connection to the IBM channel was made by industry and the software to support data transfer for farming has been written by L3 programmers.

Figure 4: L3 Lepics

3.5 Aleph FALCON – The System Builders

Finally in this overview of farming, the ALEPH experiment at CERN using the LEP accelerator has adopted the approach of pure system building with no hardware development (see [19]). This FALCON system is an Ethernet based local area cluster of VAXstations used as a parallel event processor. In figure 5 we see the preliminary configuration with 6 VAXstation 3200's running VMS and a MicroVAX 3600 for data storage. The disks are switchable between the data acquisition VAX and the FALCON system. Once the data have been acquired, FALCON is given access to it and the events are distributed to the slaves for processing. A server runs on the MicroVAX which controls the allocation of events to the data analysis programs running in the rack of VAXstation 3200's.

By using the simplest farming concepts and commercially available hardware, the FALCON system is considered by many as 'the way to do farming'. An attempt to enhance FALCON by including DEC's UNIX based workstations using the MIPS processors complicated the software and demonstrated the problems of mixing operating systems. But FALCON has proved to be an efficient and cost effective solution to the farming problem.

Figure 5: ALEPH - Falcon

4. PARALLELISM IN EXPERIMENTS (ON-LINE)

4.1 Example of a Data Acquisition System

The descriptions of present and future data acquisition systems use a consistent set of computing layers which are the detector electronics, the event builder, a processing farm, an on-line data recording VAX and off-line data analysis (see figure 6). The technology to connect this together includes Fastbus, VME, Ethernet and Optical Links. We have been looking so far at the off-line analysis of experimental data. We shall now turn our attention to the lower levels where interesting data are collected and uninteresting data rejected.

Figure 6: Example Data Acquisition System

4.2 IBM Emulators

We have already discussed the IBM emulators and the latest Parallel Processing Computer Server, PPCS. As can be seen in figure 6, there is a processor farm connected to a bus such as VME for collecting data and to a host for data recording. The IBM emulators have been used in this on-line role for a long time with great success, including the famous UA1 experiment. CERN has developed a VME connection to the IBM, called VICI, and to the crossbar switch for the PPCS which will enable CERN to use the PPCS in an experiment.

4.3 RISC Sparc on FASTBUS

A CERN group is developing a board which will contain 4 Sparc RISC processors and connect to Fastbus via a Sparc co-processor chip (see [15]). This ambitious project involves CERN gaining experience in the art of building chips using the latest technology in software and hardware. The idea behind the project is to move impressive amounts of computing power closer to the detector.

4.4 INMOS Transputers

INMOS transputers are unusual since they are processing chips which include on chip support for communication with other similar processors. This enables a large number of them to be assembled together to form a parallel computer. The T800 range of transputers was designed foremost to be incorporated into embedded systems. This makes it suitable for incorporation into experiments such as UA6 and ZEUS, at DESY (see [7]). UA6 will contain about 64 transputers while ZEUS will contain an impressive number of more than 500.

Attempts to use transputers to build general purpose computers has met with varying success since the transputer is not a general purpose processor built to support the UNIX operating system. Meiko Computer supplied CERN with an early version of their machine which was connected to a VAXstation (see [17]). The results (see [1]) were initially disappointing since it was awkward to program, I/O speeds were disappointing and the resulting performance was not as high as was hoped. CERN plans to examine the latest version which incorporates an interface based on Sparc processors, either a SUN SparcStation or a Sparc board (see [16]). The development around Sparc will give the whole system a cheaper and more user-friendly front-end. Another additional board will contain the fast i860 chip making the system very fast for some applications.

Making transputers easier to use and integrating them with a workstation host was one of the driving forces behind the developments at Cresco-Data (see [10]). Cresco-Data developed an advanced user interface to the development software and a remote procedure interface. CERN uses the Apollos for training purposes and will have 12 such workstations containing transputer boards at this year's CERN School of Computing.

CERN has developed a board which integrates a transputer with a fast digital signal processor, DSP32C (see [4] and [8]). The idea being to move the transputer close to the data source while using the computing power of the DSP to prepare the raw data.

The present T800 range of transputers are no longer considered fast by the standard of today's modern processors. However, INMOS will soon introduce a new range of transputer known as the H1 family. CERN is collaborating with INMOS to use the H1 transputer in a large scale machine.

4.5 ASPEX

At the level of detector electronics much expense occurs in moving the raw data from the detector mounted electronics to the first level of trigger processing. Since the first level trigger is simple, the LAA project has pursued the idea of moving simple processors and their inter-processor communication closer to the experiment (see [5] and [13]). This has resulted in the MPPC project for designing very fast front end detector electronics using the ASP processors from the ASPEX company in England (see [12] and [18]). The ASP processors are associative processors in the SIMD tradition. They obey a single instruction stream and contain very fast communication channels for handling the instructions and data. Instructions can be obeyed conditionally on flags which are set according to the contents of the processor's relatively small memory. In this way the associative processors resemble associative memories.

4.6 AMT DAP

The AMT DAP also contains a large number of simple processors arranged in a 2 dimensional array. Within the CERN and DEC Joint Project, a DAP was connected to a VAX and used to histogram large amounts of data (see [3]). The results were impressive, with selected cuts being changed and displayed in real time. The large amount of time required to load the machine with the data and its very expensive memory, means that it has not been used as an on-line event filter and the present DAP is viewed as being unsuitable for general use.

5. CONCLUSION

We have seen that farming is a well developed science. The original development of extensive hardware around emulators will be replaced by the PPCS and the system builder will use more commercially available hardware. The use of extensive parallelism in experiments is developing at all levels including the application of simple SIMD parallel processing at the level of the detector electronics. With future HEP experiments expected to be even more demanding there will be advances in pipe-lining and the use of parallelism to attain the processing speeds required.

The art of good computing, as described in this review, is to build on top of existing systems which are available from industry in a cost effective way. This minimises the amount of development work and enables people to track the fast moving technology currently present in computing. As one moves from the computing centers to the front end electronics of an experiment, the need for special solutions grows. Unfortunately, figures that would enable us to compare the cost effectiveness of commercial versus home-made solutions are never available. There remains a number of relatively unexplored fields, for example, data flow machines and neural nets are exciting new areas of work yet to be exploited.

REFERENCES

1. S. Booth, R.W. Dobinson, D.R.N. Jeffery, W. Lu, K.M. Storr, A. Thornton, An Evaluation of the Meiko Computing Surface for HEP FORTRAN Farming, 1989;
2. CERN, Computing at CERN in the 1990s, 1989, CERN DD;
3. CERN DEC Joint Project, AMT DAP, CERN;
4. D. Crosetto, Fast Digital Parallel Processing module, FDPP, 1989, CERN DD report;
5. J. Dhatt, A. Krikelis, R.M. Lea, A. Sandoval, Performance of the ASP on the LAA Benchmark, 1989, CERN LAA project;
6. Fridolin Dittus, Parallel Processing with Attached Processors in a Computer Center Environment, 1989, Computing in High Energy Physics, Oxford;
7. R.W. Dobinson, J.L. Pages and J.C. Vermeulen, Transputers in Particle Physics Experiments, 1990, submitted to Particle World;

8. M.L. Fuccio, R.N. Gadenz, C.J. Garen, J.M. Huser, B.Ng, S.P. Pekarich, The DSP32C: AT&T's Second-Generation Floating-Point Digital Signal Processor, December 1988, IEEE Micro;
9. IBM CERN Joint Project, PPCS, 1989, CERN;
10. Henrik Kristensen, Evaluation of the CD-TSE Transputer Development Environment for Apollo Workstations, 1990, CERN CN report;
11. P.F. Kunz, M. Gravina, G. Oxoby, P. Rankin, Q. Trang, P.M. Ferran, A. Fucci, R. Hinton, D. Jacobs, B. Martin, H. Masuch, K.M. Storr, The 3081/E Processor, 1984, CERN DD report;
12. R.M. Lea, ASP: A Cost-effective Parallel Microcomputer, October 1988, IEEE Micro;
13. S. Lone, R.K. Bock, Y. Ermolin, W. Krischer, C. Ljuslin, K. Zografos, A. Sandoval, Fine-grain parallel computer architectures in future triggers, 1989, CERN EP report;
14. George J. Luste, Supercomputing Myth and Reality, 1989, Supercomputing Symposium, Toronto;
15. Alessandro Marchioro, Architectural Design of a new VLSI FASTBUS integrated master, 1989, CERN EF notes;
16. Meiko, In-Sun Computing Surface, 1989;
17. Meiko, The Computing Surface, 1988;
18. MPPC project node: The Massively Parallel Processing Collaboration: the MPPC project, 1989;
19. M. Delfino, E. Fernandez, S. Orteu, A. Pacheco, The ALEPH Event Reconstruction Facility: Parallel Processing using Workstations, 1989, Computing in High Energy Physics, Oxford;
20. T. Nash et al., The ACP Multiprocessor System at Fermilab, 1987, Computer Physics Communications;
21. T. Nash et al., A Site Oriented Supercomputer for Theoretical Physics: The Fermilab Advanced Computer Program Multi Array Processor System (ACMAPS), 1989, Fermilab;
22. Peter Wayner, VLIW: Heir to RISC, August 1989, BYTE;