ADVANCED COMPUTERS AND SIMULATION*

Robert D. Ryne
Los Alamos National Laboratory
Los Alamos, NM 87545 USA

Abstract

Accelerator physicists today have access to computers that are far more powerful than those available just 10 years ago. In the early 1980's, desktop workstations performed less than one million floating point operations per second (Mflops), and the realized performance of vector supercomputers was at best a few hundred Mflops. Today vector processing is available on the desktop, providing researchers with performance approaching 100 Mflops at a price that is measured in thousands of dollars. Furthermore, advances in Massively Parallel Processors (MPP) have made performance of over 10 gigaflops a reality, and around mid-decade MPPs are expected to be capable of teraflops performance. Along with advances in MPP hardware, researchers have also made significant progress in developing algorithms and software for MPPs. These changes have had, and will continue to have, a significant impact on the work of computational accelerator physicists. Now, instead of running particle simulations with just a few thousand particles, we can perform desktop simulations with tens of thousands of simulation particles, and calculations with well over 1 million particles are being performed on MPPs. In the area of computational electromagnetics, simulations that used to be performed only on vector supercomputers now run in several hours on desktop workstations, and researchers are hoping to perform simulations with over one billion mesh points on future MPPs. In this paper we will discuss the latest advances, and what can be expected in the near future, in hardware, software and applications codes for advanced simulation of particle accelerators.

Introduction

Today accelerator physicists are finding it possible to perform particle simulations and electromagnetic calculations of unprecedented precision and complexity. Such calculations would have been impractical only two to three years ago, but they are possible now because of two major developments: (1)the availability of very high performance, affordable workstations and (2)the availability of Massively Parallel Processors (MPPs) and distributed computing environments. These topics will be discussed in the first part of this paper.

In the second part of this paper we will discuss simulations we have performed using the Connection Machine 5 at the Advanced Computing Laboratory of Los Alamos National Laboratory. We are developing computer codes that will support efforts related to the design of next-generation, high current accelerators. We will show preliminary results of beam dynamics calculations with 16,777,216 ($=2^{24}$) particles.

ADVANCED COMPUTERS

Workstations

One of the most significant developments of the late 1980's to early 1990's is the emergence of very high performance workstations. During the early 1980's, desktop workstations generally performed at less than 1 Mflop. The concept of a pipelined machine (which can result in an effective throughput of one result per machine cycle for vector operations) was reserved for multi-million dollar vector supercomputers. Today vector processing and superscalar implementations can be obtained on the desktop for a price measured in thousands of dollars. Typical characteristics of a moderate to high end workstation (in 1993) are as follows:

1. RISC CPU
2. RAM: up to 512 MB (16 SIMMS @ 32 MB/SIMM)
3. Hard Disk: 1 GB to a few GB
4. Performance: 10–40 Mflops
5. Price: $10,000 to $80,000

Workstations of this caliber are now in widespread use at many laboratories, universities and institutions. Such machines are often dedicated to running 3D electromagnetics simulation programs such as ARGUS and MAFIA. It is not uncommon for these jobs to require several hours on a high end workstation. Even so, for many problems the turn-around time is faster on a dedicated workstation than that obtained running on a time-shared vector supercomputer. The performance of workstations will likely increase significantly in the near future, with the advent of multiprocessor machines, 64–bit CPUs (for which double precision calculations will be unnecessary), and more CPUs operating around 200 MHz.

Table 1 shows performance benchmarks for a sample of widely used workstations:

* Work supported by the U.S. Dept. of Energy, Office of Energy Research: Office of High Energy and Nuclear Physics, Office of Superconducting Super Collider, Office of Fusion Energy, and Office of Scientific Computing

0-7803-1203-1/93$03.00 © 1993 IEEE

Table 1 Benchmarks for a sample of workstations

Machine	Spec92Int	Spec92FP	MFLOPS
Intel 486/50	30.1	14.0	
Intel 486/66	32.4	16.1	
Intel P5/66	64.5	56.9	
DEC 3k/400 (Alpha)	65.3	112.2	18.7
DEC 3k/500 (Alpha)	74.3	126.0	
HP 710 (PA RISC)	32.7	56.4	
HP 730 (PA RISC)	52.0	86.7	
HP 735 (PA RISC)	80.0	150.6	40.8
Sun Sparc IPC	26.4	21.0	
Sun Sparc 10/30	44.2	52.9	
Sun Sparc 10/41	52.6	64.7	
IBM RS/6000 320H	20.9	39.4	11.7
IBM RS/6000 350	35.4	74.2	
IBM RS/6000 580	59.1	124.8	38.1
SGI Indigo (R4000)	57.6	60.3	16.0

When choosing a workstation, benchmarks like that shown above should be regarded along with several other factors. For example, it is worth keeping in mind that the performance of workstations from various manufacturers often leap-frog one another in a time span on the order of a year. Long term stability of a workstation manufacturer, and the ability to incorporate a specific machine into one's local network are also important considerations. Finally, benchmarks obtained by running frequently used applications programs are a more valuable indicator of expected performance than SPECmark results, particularly when comparing machines of comparable sophistication from different manufacturers. Table 2 shows the performance of some workstations in running a MAFIA eigenvalue problem [1]:

Table 2 Benchmarks based on a MAFIA eigenvalue problem

Machine	MFLOPS
DEC 3k/500 (Alpha)	22.1
HP 730 (PA RISC)	12.3
Sun Sparc 1	1.36
Sun Sparc IPC	1.71
Sun Sparc 2	3.9
Sun Sparc 10/30	10.4
IBM RS/6000 320	10.1
IBM RS/6000 550	13.4
IBM RS/6000 580	38.1

Massively Parallel Processing and Distributed Processing

Another major advance that has occurred in recent years is the development of computer platforms, software and algorithms for massively parallel processing and distributed processing. These environments are aimed at solving problems that are on the (present) leading edge of high performance computing, and which require performance on the order of 10 Gflops to 1000 Gflops. Typical applications include: global climate modeling, quantum chromodynamics, molecular dynamics, self-consistent electronic structure calculations, 3D seismic calculations, and high resolution simulations in computational fluid dynamics, electromagnetics, and structural analysis, to name a few.

Large scale MPPs typically fall into one of two categories: SIMD (Single Instruction/Multiple Data) and MIMD (Multiple Instruction/Multiple Data). The SIMD model is the least complex and the easiest to utilize from a user standpoint. In this case, all the processing elements operate in synchronism, performing the same operation on different sets of data. Typically, programmers can write code using array constructs like that found in Fortran 90 (for example, `a=b+c`, where `a`, `b` and `c` are arrays); the compiler will recognize this as an operation that can be done in parallel. Thus, from the standpoint of a scientific programmer, it is easy to use MPPs for problems that are inherently *data parallel*. The drawback of this approach is that it is restrictive with respect to the types of calculations that can be implemented (in a straightforward manner). In contrast, in the MIMD approach all the processing elements can be performing different instructions, asynchronously, on different sets of data. While this approach is flexible, it is also far more difficult to make use of from a programming standpoint. In MIMD-style programming, the programmer specifies operations at the level of individual processors and makes use of message passing routines for interprocessor communication.

The implementation of massively parallel processing is complicated by many factors. One factor is the type of memory requirement for a given problem: Do the processors require only local memory, nearest-neighbor communication, or increasingly global communication? If the communication is global, are the required patterns regular or random? Another important factor is the issue of load balancing: It is pointless to allocate a large number of processing elements to a job if only a few processors are working while many others are idle. There are many operations, common to large applications programs, that are not obviously parallelizable and do not lend themselves easily to load balancing. For such operations, it is necessary to devote a great deal of effort to circumvent the difficulties (if it is possible at all). An example of this is the operation of laying down a distribution of charge or mass onto a regular mesh. If the distribution is highly nonuniform (such as a Gaussian distribution), then processors associated with the tails of the distribution will cease to be utilized long before processors associated with the core of the distribution, unless an effort is made to deal with this issue. Researchers in

the field of plasma particle-in-cell (PIC) simulation have made significant progress in this area [2].

For many problems, computational physicists can obtain moderately high performance without having to program at the level of individual processing elements. Two things make this possible: First, many MPP manufacturers are developing scalable software libraries to perform operations of widespread applicability to scientific programmers. Secondly, as mentioned above, many MPP compilers recognize Fortran 90 constructs as operations than are to be performed in parallel. Just as programmers in the 1980's learned to write "clean code" with vectorizable do-loops to achieve high performance on vector supercomputers, scientific programmers of the 1990's will learn to write "clean code" in Fortran 90 to obtain high performance on MPP's. (In addition to Fortran 90, some MPP manufacturers also provide support for the C programming language.) Of course, the very highest performance will still be obtained by researchers who have the time, energy, and need to program at the individual processor level.

Today there are several MPPs available, and many are under development. One of the early successful MPPs was the Intel Touchstone Delta: It consists of 540 i860 CPUs and is capable of a peak performance of 32 Gflops. It has been used very successfully for large scale astrophysical simulations [3]. One of the leading MPPs at this time is the Thinking Machines Corporation Connection Machine 5 (CM-5): It is a tree-connected MIMD machine that consists of a number of Sparc-based CPUs each with 4 vector units. A 1024 node machine is capable of a peak performance of 128 Gflops. A third example of an MPP is the T-3D, which is under development by Cray Research Incorporated. The T-3D is planned to consist of a large number of Digital Equipment Corporation Alpha CPUs, each running at 150 MHz. A 2048–processor system is expected to have a peak performance of 300 Gflops. Many MPP manufacturers believe that performance of a teraflop will be achieved soon after mid-decade.

It is worth pointing out that advances in the workstation market have had a significant impact on the development of MPPs. The goal of an MPP manufacturer is not just to build a high performance machine, but also to build a cost effective, affordable machine. As should be evident from the previous paragraph, MPP manufactures are now building systems around mass produced, inexpensive RISC CPUs. In addition, they make use of inexpensive DRAM memory. A typical MPP will have 16MB to 64 MB per node.

Another approach to high performance computing is to use collections of computers to perform parallel computations. In the simplest case, it consists of a homogeneous collection of machines, such as a cluster of workstations (or workstation "farm"). In a more sophisticated approach, it consists of a heterogeneous collection of computers, including workstations, vector supercomputers, and even the nodes of an MPP; all the processors involved in the computation are connected by a variety of networks. Software such as PVM (Parallel Virtual Machine) is now available to support such calculations. [4]. Distributed processing has been aided by the development of high speed networks that operate at nearly a gigabit per second. Many of these are based on the High Performance Parallel Interface (HIPPI), which operates at 0.8 Gbit/sec. Also, 5 Gbit/sec networks are under development. Using high speed networks, it will soon be possible to coordinate the high performance computing platforms at several geographically separated supercomputer centers to run Grand Challenge simulations.

ADVANCED ACCELERATOR SIMULATION

Today there are several important problems in computational accelerator physics that cannot be handled without making use of high performance computing platforms. These include:

1. Self-consistent beam dynamics problems involving very large numbers of particles ($\gtrsim 10^8$ particles)
2. Electromagnetics simulations involving very large numbers of mesh points ($\gtrsim 10^8$ points)
3. Long term tracking in circular machines
4. Optimization of beamlines and electromagnetic structures, where each iteration in the simulation is itself difficult or impossible to perform on a single workstation or vector supercomputer

Below we will discuss the simulation of intense beams using large numbers of simulation particles, motivated by the need to predict beam halo in next-generation, high intensity CW accelerators.

During the past several years, many technical advances have been made in the field of Accelerator Technology that are opening up new areas of research and application. Several projects, of major importance to the environment and to international competitiveness, now seem feasible. These include the Accelerator Transmutation of Waste (ATW), Accelerator Production of Tritium (APT), accelerator-based production of 14 MeV neutrons for fusion materials testing, Accelerator Based Conversion of Plutonium (ABC) and the development of a next-generation, high intensity spallation neutron source. These projects are also extremely challenging: They rely on very high intensity, continuous wave accelerators (as opposed to the moderately low intensity, pulsed machines of the past). Adding to the difficulty is the fact that these projects are considered acceptable only if hands-on maintenance of the accelerator is possible. (Remote handling increases the complexity and the cost of these projects significantly.) In order to meet scheduled maintenance requirements and Occupational Safety and Health requirements, the radioactivity level must be a few millirem/hour shortly after shutdown of the accelerator. This corresponds to very strict beam loss requirements: the allowed beam loss is on the order of 0.1 nanoampere/meter, or equivalently, about 1 part in 10 to 100 million. These projects are expensive (some are expected to cost over 1 billion dollars). It will be difficult to get approval for any of them unless one can demonstrate clearly and convincingly that it is possible to meet the beam loss requirements.

Beam losses at this low level are caused by particles in the very sparsely populated beam halo, at a distance of 6 or more standard deviations from the beam axis. Beam halo and resulting beam loss are due to space charge effects coupled with effects such as beam mismatch, machine errors and field nonlinearities. Understanding and predicting beam halo has become a critical issue for many advanced accelerator-based projects. In order to have confidence that we can meet these ultra-low loss requirements, we need to perform particle simulations with on the order of 100 million particles. Consider, for example, that if one simulated a round Gaussian beam with 100 million particles, only about 4 particles would be located in the region beyond six standard deviations. Accelerator simulations of this magnitude are impossible with vector supercomputers, but with the advent of MPPs such simulations are within reach.

We are developing simulations using the CM-5 at the Los Alamos National Laboratory Advanced Computing Laboratory in order to better understand and predict beam halo in high intensity, ultra-low loss accelerators. Ultimately, we expect to be able to perform simulations with on the order of 100 million particles. Accelerator simulations of this magnitude (in which we would propagate a beam in fine detail through up to a kilometer of accelerating structures) are unprecedented in the field and represent a quantum leap beyond what has been accomplished in the past. We hope to be able to accurately predict beam halo, and thereby help demonstrate the soundness of advanced accelerator designs.

In order to learn to utilize the CM-5, we have begun by developing a simulation for a simple test problem. Our program models an intense beam in a periodic focusing channel (in two dimensions). The beam and the external fields are assumed to have cylindrical symmetry (the transport system consists of magnetic solenoids.) Our simulation includes nonlinearities from the beam self-fields and from the external fringe fields of the focusing magnets. The particles are propagated using a 4th order symplectic integration algorithm [5].

Figures 1 and 2 show the initial and final beam density, respectively, from a typical run of an initially mismatched, Gaussian beam propagating in the focusing channel. Figures 3 and 4 show the initial and final horizontal phase space, respectively. The parameters of this CM-5 simulation are shown below:

# of particles	16,777,216 (=2^{24})
# integration steps	100
# processors	512
CPU time	26 min

These results were obtained using beta versions of the operating environment and run time library (CMOST 7.2, Beta 1, patch 4), the CM Fortran compiler (CMF 2.1, Beta 0.1) and the CM Scientific Software Library (CMSSL 3.1, Beta 3).

In the future, we plan to extend our model to perform simulations of 3–dimensional beams propagating through systems of quadrupole magnets and accelerating structures.

Figure 1 Initial Beam Density

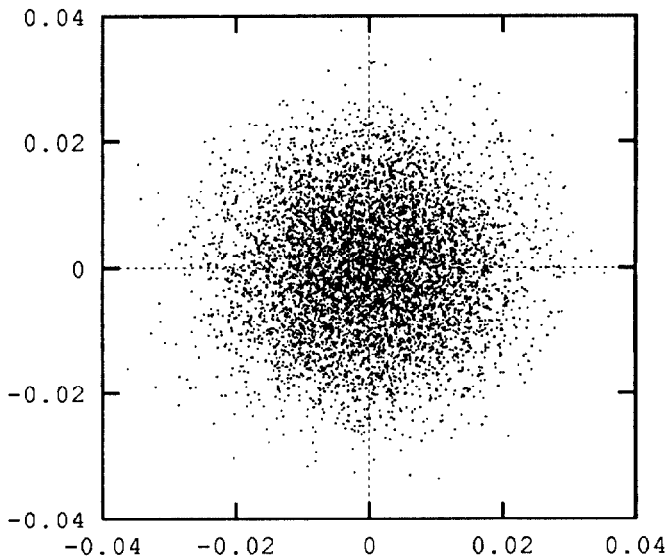


Figure 2 Final Beam Density

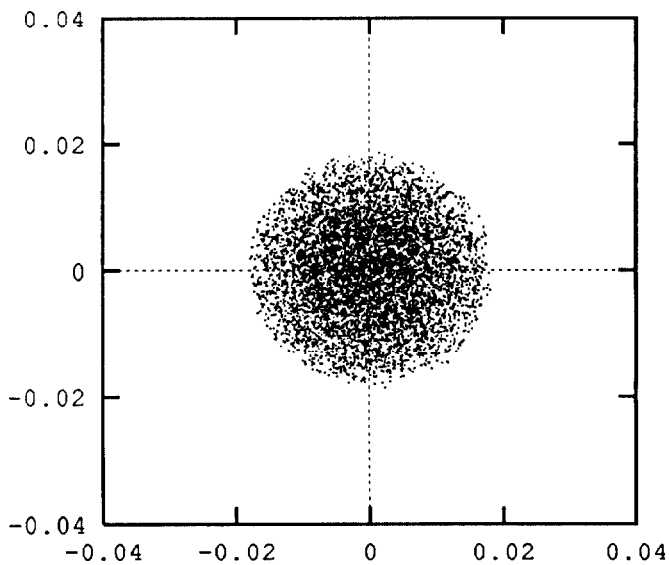


Figure 3 Initial Horizontal Phase Space

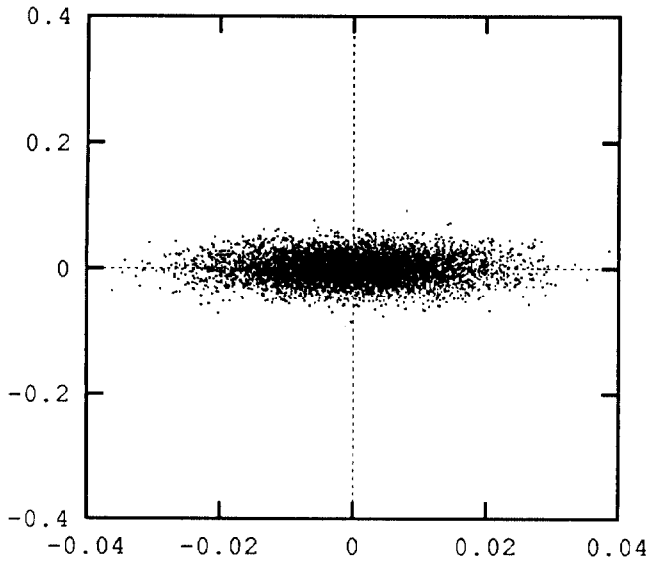
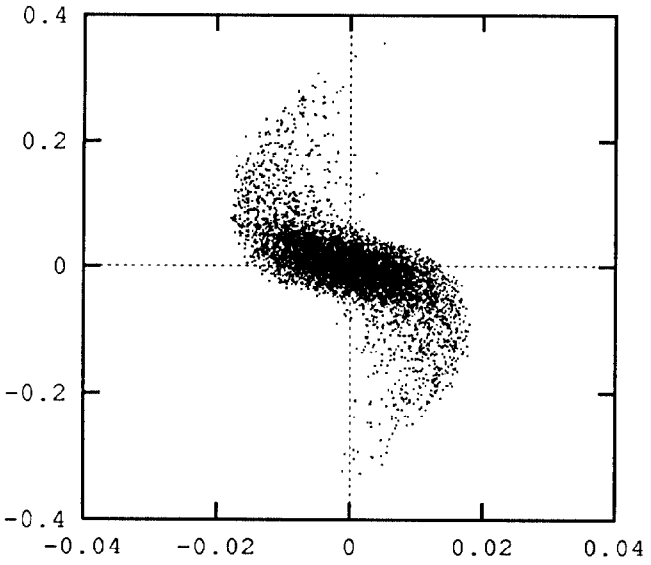


Figure 4 Final Horizontal Phase Space



CONCLUSION

Accelerator physicists presently have access to resources that are far superior to what was available just a decade ago. The performance of desktop workstations now approaches that of the early Cray supercomputers; with the advent of massively parallel and distributed processing, gigaflop performance is becoming increasingly common. By the time of the 1995 Particle Accelerator Conference, many of the participants will have begun to use MPPs, and most will probably program them in a language that resembles Fortran 90. These advances will have an especially profound impact in the area of 3–d modeling. At present, simulations on high end workstations may require many hours, making optimization of structures a difficult task; and though some electromagnetics codes have been ported to MPPs, few researchers have access to these platforms. In the near future, with the advent of 64 bit CPUs operating at very high clock speeds, and with the introduction workstations and several thousand node MPPs based on these structures, researchers will have access to machines of unprecedented performance.

Acknowledgments

This research was performed in part using the resources located at the Advanced Computing Laboratory of Los Alamos National Laboratory, Los Alamos, NM 87545.

REFERENCES

[1] F. Krawczyk, Private Communication

[2] P. Liewer and V. Decyk, "A General Concurrent Algorithm for Plasma Particle-in-Cell Simulation Codes," *J. Comp. Phys.* **85**, 302 (1989).

[3] M. Warren and J. Salmon, "Astrophysical N-Body Simulations Using Hierarchical Tree Data Structures," Proceedings of Supercomputing '92, September 11, 1992

[4] J. Dongarra et al., "Integrated PVM Framework Supports Heterogeneous Network Computing," *Computers in Physics* **7**, 166 (1993).

[5] E. Forest and R. Ruth, "Fourth-Order Symplectic Integration," *Physica D* **43**, 105 (1990).