

An Approach to Using a Parallel Computer Architecture
to Enhance Accelerator Modeling Performance

Vern Paxson and Lindsay Schachinger

SSC Central Design Group

July 1, 1988

ABSTRACT

We evaluate one possible method of using a multiple-CPU computer architecture to greatly increase the performance of accelerator modeling software with relatively little expense. Extrapolating preliminary measurements and projected costs shows that the time needed for TEAPOT^[1] to compute a fundamental quantity such as machine tune could be decreased by a factor of roughly 200 for a cost expenditure of \$75-100K. This compares very favorably with the increased performance given by buying a similarly-priced single processor computer. Such a parallel system should be available within the time frame of one year.

Introduction

Certain components of accelerator modeling software—multi-particle tracking, in particular—are especially well-suited to parallel programming. That is, portions of the modeling computations can be done simultaneously if one has a central processing unit (CPU) available for each portion. Since a computer system's CPU is only a fraction of the total system cost, a specialized computer having multiple CPU's is a much more economical approach to parallel computing than linking together a group of single-CPU computers over a network. Such a multiple-CPU system also has a greater capacity for performance since the inter-CPU communication rate is typically much higher than can be attained over a network. This means that for a given expenditure one can attain much higher performance for "parallelizable" programs running on specialized hardware than would otherwise be possible.

The Advanced Computer Program (ACP) at Fermilab has developed the first generation of such a parallel computer system and is planning to construct a more powerful system, projected to become available in January, 1989 at a cost of \$3-5K per CPU. We give an overview of the architecture of the system, background information on evaluating the processors on which the system is built, and the expected performance of such a system based on measurements made on various types of CPU's.

An Overview the ACP Parallel Architecture

The ACP group has developed two parallel computing architectures^{[2][3]}. The first is optimized for very high speed communication (multiple 20 Mbytes/sec conversations active at any given moment) between processors and is currently used for lattice gauge theory computations. The second trades lower communication bandwidth (one 20 Mbytes/sec conversation active at any given moment) for more flexibility in system configuration and greater ease of connecting the

system to an existing computer network. This latter system is currently used by Fermilab's Central Detector Facility for event trigger and analysis.

When using a parallel computer architecture to improve the performance of multi-particle tracking, the natural way to partition the computation into parallel tasks is to allocate one CPU to each particle being tracked. With this partitioning the amount of information which must be communicated between CPU's during computations is small enough that the lower communication bandwidth of the second architecture is still ample (by several orders of magnitude) for the task. As the second system mentioned above is in most ways simpler to use, it would be the natural choice for us. We limit subsequent discussion to this system.

The basic component of the system is a VME crate populated by up to 16 cards. These cards can be any mix of CPU's, Ethernet connections, disk controllers, and so forth. Processors on different cards can communicate with each other via the VME backplane at 20 Mbytes/sec. One such conversation can be active at any given time. In addition, ACP Branchbus^[4] cards can be installed in the crate to facilitate multiple, simultaneous 20 MByte/sec conversations.

Multiple crates can be connected using either Ethernet (up to 1 MByte/sec) or the ACP Branchbus. Thus, any network node accessible via Ethernet can serve as a CPU.

The current implementation of this system uses Motorola MC68020 CPU's coupled with MC68881 floating point coprocessors and 6 Mbytes of memory. The implementation slated for completion by next January utilizes MIPS R3000 CPU's (with a corresponding floating point unit) and will have 8 Mbytes of memory for program data and 2 Mbytes for program code, ample for Teapot tracking of the SSC.

In addition to the parallel computer hardware, the ACP group provides two software packages for using the hardware. The first,^[5] designed for event analysis, partitions the available CPU's into classes which perform different functions (data gathering, computation, results gathering) and provides a simple set of op-

erations for inter-CPU communication. The second,^[6] CANOPY, is considerably more general. It removes restrictions on CPU functionality, provides more flexible inter-CPU communication, and hides the particular hardware configuration (number of CPU's, which CPU's are in which crates) from the software. Either would be suitable for multi-particle tracking; CANOPY would be preferable since it does not impose a particular parallel-processing model (such as "event analysis") and it is the newer generation of the software.

Theoretical Relative Computer Performance

The parallel architecture discussed above is based on the high-performance MIPS R3000 CPU, which has recently become available. To put its raw (non-parallel) performance in perspective, Table 1 summarizes the most commonly used measurements of CPU performance for some of the computers now (or soon) being used for SSC accelerator modeling and for the MIPS family.

It is worth noting that rating the processing power of a computer is a notoriously slippery task. Today's CPU architectures have enough (often hidden) special cases such as caching and pipelining that one can readily "tune" a benchmark in order to maximize its performance on a particular CPU. The programs which the buyer of a machine will actually be running on it are very often not amenable to such tuning, so the officially rated performance of the machine may well exceed the end-performance which the user ultimately sees. Such official performance figures are colloquially referred to as *Guaranteed Not To Exceed* numbers. They are still useful as an indicator of which machines are roughly comparable.

To provide a more meaningful method of evaluating computer performance, several standardized benchmarks programs have emerged which are run unmodified on the computer being evaluated. These reduce (but do not eliminate, since compilers can be optimized with the benchmarks in mind) the effects of "tuning"

on the benchmark values, but still leave the problem of how closely characteristic is the benchmark to the real applications one wishes to execute on the machine.

System	MIPS	KFLOPS	\$ for 16 MByte system
Sun 3/160 FPA	1.5	405	
Sun 4/280	7-8	1100	
Vax 8650	≈ 6	≈ 970	
MIPS M500	5	720	
MIPS M1000	10	1600	\$57K
MIPS M120	12	2200	\$37K
MIPS R3000	20	4000	\$82K

Table 1. Basic Benchmarks for Several CPU families

Some notes on the table:

- The Sun 3/160 with Floating Point Accelerator (FPA) is the Sun system currently being used by the Accelerator Physics group (this machine uses the Motorola MC68020 CPU and a floating point unit which performs substantially better than the MC68881 mentioned in the previous section); the Sun 4/280 is the model now on order for delivery mid-summer; the Vax 8650 is the processor available on LBL's Computing Cluster; and the MIPS processors (in particular, the R3000) are what are used in the ACP system. For the MIPS processors the table also includes the cost for a 16 MByte (main memory) development system. The curiosity of the M120 system outperforming the higher-priced M1000 system is due to the M1000 system being capable of having additional hardware (disks, memory, tape systems, terminals, etc.) plugged into it, while the M120 system is self-contained.
- "MIPs" stands for Million Instructions Per second (as well as being the name of the MIPS Computer Company), and is a measurement of how fast

the processor performs non-floating-point computations. A Vax 11/780 is nominally equivalent to 1 MIP. However, not all instructions are created equal, so the MIPs rating is at best only a ballpark figure of true CPU performance. The values quoted here are from vendor literature. (The MIPs figure is a *Guaranteed Not To Exceed* number.)

- “KFLOPS” stands for 1000 Floating point Operations Per Second, and is a measure of how well the processor performs on computations which are heavily or exclusively floating point (depending on the benchmark). The numbers above are for the standard LINPACK benchmark, using double precision calculations. This benchmark is heavily but not exclusively floating point, and therefore should be a good indication of expected Teapot performance.
- The values for the Vax 8650 are interpolated from values given for other Vax family computers (8800, 8600) and are upper bounds.
- The scaling between the M500, M1000, M120, and R3000 CPU’s is due to increased clock speeds and pipelining and thus it is sound to similarly scale results for benchmarks run on one of these CPU’s to other members of the family. The floating-point performance actually increases more sharply than the MIPs performance, but in our extrapolations in the next section we assume scaling in line with the MIPs performance.

Measured Computer Performance

We ran two different TEAPOT jobs on the Sun 3/160, Sun 4/280, Vax 8650, and MIPS M500 processors. The results are summarized in Table 2. The first job, “SSC Setup”, entails fitting the tune of the machine and calculations of the closed orbit, tunes, beta functions at every element, and chromaticities (“analysis”). The second generates multipole errors up to ninth order, does an analysis, and then tracks one particle for 512 turns. The final column in the table is the

estimated raw time needed by TEAPOT to compute the tune. This value is the amount of time needed to track 11 particles twice around; it does not include the overhead of going from the tracking information to the tune itself, or that of finding the closed orbit.

	SSC		SSC Analysis		Tune Computation
System	Setup	(ratio)	& Tracking	(ratio)	(est.)
Sun 3/160 FPA	1214	1	2683	1	94.2
Sun 4/280	326	3.72	796	3.37	23.6
Vax 8650	NA	NA	602	4.46	25.0
MIPS M500	319	3.81	566-601	4.74/4.46	18.5
MIPS R3000	80*	15.2	142*	18.9	4.62
11 × R3000	66*	18.4	127* (28.9)	21.2 (92.8)	.42

Table 2. Measured and Expected TEAPOT Performance
(times are in CPU Seconds; * indicates estimate)

Some comments on the table:

- Columns marked “(ratio)” are with respect to the Sun 3/160 system
- Two sets of times are given for MIPS M500 running the Analysis & Tracking job. During one run we measured anomalously high operating system overhead, equal to the difference between the two timings. The source of overhead is unknown; the system on which the timings were made had been subject to some experimental tuning which may have resulted in the occasional presence of the additional overhead.
- The MIPS R3000 times are scaled from the expected factor of four difference between the R3000 and the M500.
- The row labeled “11 × R3000” is for an eleven CPU parallel computer using

R3000 CPU's (the configuration available from the ACP by early-to-mid next year). In the row, the parenthetical entries for Analysis & Tracking and the corresponding ratio are the timings that would be expected if instead of tracking one particle for many turns, 11 particles were tracked for correspondingly fewer turns.

- The measured performance of the Sun 4/280 configuration is lower than would be expected from Table 1. This may be due to our use of an early release of both the operating system and the compiler.

Conclusions

Given a parallel computer architecture such as the one being developed by Fermilab's ACP, it is possible to economically realize large performance increases for multi-particle tracking tasks. In particular, a factor of 200 in tune calculation over our present system (and 50 over the general development system soon to be available) can be had for an expected expenditure of \$75K to \$100K, comprising \$35-\$60K for the parallel computer and \$40K for an accompanying development computer. Such a performance increase has several implications: the ability to track beams of particles where currently only single particles are tracked; interactive study of the behavior of computationally intensive machine functions such as tune; and the possibility of improving the performance of other aspects of accelerator modeling which are conducive to parallel processing.

Acknowledgments

We would like to thank Irwin Gaines and Tom Nash of Fermilab's Advanced Computer Program for valuable assistance in acquiring the information presented in this paper, and Mike Helm of LBL's Advanced Accelerator Studies Department for facilitating our use of their Sun-4 computer for making performance timings.

REFERENCES

1. L. Schachinger and R. Talman, "Teapot: A Thin-element Accelerator Program for Optics and Tracking," **Particle Accelerators**, Vol. 22, 1987.
2. T. Nash, et. al., "The Fermilab Advanced Computer Program Multi-Array Processor System (ACPMAPS): A Site Oriented Supercomputer for Theoretical Physics," Adriatico Conference on the Impact of Digital Microelectronics and Microprocessors on Particle Physics, International Centre for Theoretical Physics, Trieste, Italy, March 28-30, 1988.
3. Irwin Gaines, et. al., "Multi-Processor Developments in the United States for Future High Energy Physics Experiments and Accelerators," Adriatico Conference, 1988.
4. R. Hance, et. al., "The ACP Branchbus and Real Time Applications of the ACP Multiprocessor System," FERMILAB-Conf-87/76, 1987.
5. "ACP Software User's Guide for Event Oriented Processing," Fermilab Advanced Computer Program, Document #GA0001, Revision 3.0, December, 1987. Available via HEPNET as
FNACP::ACPDOCS_ROOT:[DOCS.SOFTWARE]SOFTGUIDE.HQX
6. "CANOPY - ACP MAPS Software - Specifications Document," Fermilab Advanced Computer Program, internal document. To be released.