SSC RC-004
10 PAGES

ANL-HEP-TR-89-85
September 29, 1989

PROPOSAL

BENCHMARK DEVELOPMENT AND PERFORMANCE TESTING OF COMPUTERS FOR SSC DETECTOR DESIGN*

Submitted to
the SSC Laboratory
Physics Research Division
by

Edward N. May
Principal Investigator

Argonne National Laboratory
High Energy Physics Division
Argonne, IL 60439

Approved for the Argonne High Energy Physics Division:

Thomas B.W. Kirk
Director
High Energy Physics Division

*Work supported by the U.S. Department of Energy, Division of High Energy Physics, Contract W-31-109-ENG-38.

Benchmark Development and Performance Testing of Computers for SSC Detector Design

(A proposal submitted to the SSC Laboratory by the
Argonne National Laboratory High Energy Physics Division)
September 29, 1989

Goals:

1. Assemble a set of benchmark codes for SSC Detector Simulation based on existing HEP community developed computer codes; document performance of these codes on a wide class of currently available computers.

2. Develop a set of benchmark computer codes for detector simulation based on the codes noted above modified to exploit event-level parallelism; document the performance of these parallelized codes on a wide class of currently available parallel processor based computers.

Introduction

Detector simulation with computer programs is an important phase in the development of designs for HEP experiment detectors to be used at the SSC. The computing requirements for such simulations are large. These requirements have been reviewed in the most recent Snowmass Summer Workshop. We summarize here, from the working group report "Computing For High Energy Physics in the 1990's"[1], material of relevance to detector simulation. The working group estimated the effort and CPU requirements over a 3 year period as:

	Effort Level FTE	CPU-Usage VAX-780 Equivalents
FY1989	10	70
FY1990	15	840
FY1991	30	5000

The cost of providing the computing in FY1989 has been estimated by SSC Task Force on Computing [2] as $1.1M for 30 VAX 780-equivalent of conventional VAX/VMS computers and 20 (VAX 780-equivalent units) of CRAY XMP computers respectively, from existing laboratory computing facilities. Both references noted that newer, more cost-effective, computing facilities must be utilized in order to meet the needs for computing beyond 1989. These facilities could include, but are not limited to, newer mini-computers, mini-supercomputers, supercomputers, advanced workstations, and parallel processing computers. It is currently expected that the parallel processing approach will be potentially the most cost effective, whether with commercial parallel processors (e.g., Intel HyperCubes or a

network of RISC-based workstations) or with HEP community-developed parallel processors (e.g., FNAL ACP-2). This is because of the inherent coarse-grained parallelism of most HEP event simulations.

Benchmarks and model codes are important elements in judging both the cost effectiveness and general usability of new computing systems. It is generally believed that in order to meet the SSC computing needs, extensive new facilities based on 'farms' of Reduced Instruction Set Computing (RISC) processors will be required. Thus it is important to: a) identify the processors; and b) make them work together in an efficient and user-friendly environment.

The work proposed here is intended as a service to the SSC Laboratory and its user community. The results will be made available in detailed reports, including the tested benchmark codes. The results and reports will be available to interested parties in a network accessible library, as well as in conventional print form.

General Description of Proposed Work

We propose a project in two major parts:

- The first part includes performance evaluation of a broad range of computers using a set of standard benchmark programs of interest to the SSC detector simulation community. The results of the performance measurements will be well documented and broadly distributed. Major results will include:

  1. Identification of cost-effective hardware computing systems on which to run large SSC detector design codes.
  2. Specification of a technical basis and performance standard to be used in the competitive procurement process for SSC Laboratory computer systems.
  3. Development of useful techniques for debugging and maintaining a general user library of codes for SSC detector simulation.

- The second part of the project is the development and testing of portable parallelized benchmark programs, and is based on our experience that codes used for detector simulation are inherently parallel at a coarse-grained level (typically the event level). At this level, each succeeding physics event is independent of all previous or future events during a simulation run with fixed detector characteristics. This type of code is well suited to many Multiple Instruction Multiple Data (MIMD) parallel computers. Associated computing resources are enhanced approximately linearly with each added processor in a parallel MIMD computer. This linear growth is subject to architectural limitations of the computer design (for example the bus and communications designs). Good benchmark programs are fundamental to probing the performance of new parallel architectures.

Statement of Work

Part 1: Serial Benchmark Study

The first work package will identify an appropriate set of benchmark codes and run them on serial-architecture machines to establish the relative performance of each machine on these codes. It will assemble the benchmarking package in a form that can be used externally to benchmark new computers not presently available for evaluation.

Performance period:

October 1, 1989 - September 30, 1990

Personnel and Duties:

0.1 FTE Physicist
- project manager
- definition of goals
- selection of codes
- definition of parameters of standard runs
- standards for correct program operation
- review tabulated results
- documentation

0.6 FTE Scientific Applications Programmer
- assemble and test benchmark code packages on ANL computers
- negotiate with commercial vendors for benchmark runs on their computers
- carry out benchmark runs at several research computer laboratory and university computer centers
- document performance with each computing system
- compile and tabulate benchmark results

Part 2: Parallel Benchmark Study

The second work package will seek to develop a set of codes for parallel-architecture computers and a set of event-level tools to parallelize HEP codes for efficient use by these machines. Details are given below:

Performance Period:

October 1, 1989 to September 30, 1990

Personnel and Duties:

0.1 FTE Physicist
- project manager
- definition of goals
- selection of codes
- definition of parameters of standard runs, standards for correct program operation.
- review tabulated results
- write up results

1.0 FTE Scientific Applications Programmer
- implement selected codes using PARMACs system
- develop parallelized random number generation mechanism
- set up benchmark code package
- assemble and test benchmark code package on ANL machines
- negotiate and interact with commercial vendors to obtain benchmark runs
- negotiate and obtain temporary accounts on appropriate computers at research computer laboratory and university computer centers, make runs
- document experiences with each computing system
- tabulate results

0.4 FTE Systems Applications Programmer
- implement PARMACs system on parallel machines of interest

Detailed Considerations for Benchmarks:

The set of benchmark codes to be used will span the range of typical detector simulation for the SSC. They will include both full simulation and fast approximate algorithm codes. In addition, they will be reasonably close to production codes. The benchmark package will be assembled as a self-contained stand-alone package, which can then be run by project personnel on various computers or submitted to vendors for benchmarking on unannounced computers.

The package will contain source code for all programs and libraries, and will conform, insofar as possible, to the Fortran 77 code standard. Necessary exceptions, such as timing routines, will be individually and carefully documented. Finally, the package should contain sufficiently detailed installation and operation instructions to enable a typical scientific applications programmer to install and run benchmarks in approximately 3 days.

Because the benchmark package is intended to be portable, benchmark runs should be possible with a minimum of system-dependent work (e.g., inclusion of job control language

and timing routines), with very little change to the code. Results will also be tabulated for runs that include the (documented) application of automatic optimization techniques and the use of compiler directives.

Because the intention at this stage is not competitive procurement but rather an evaluation of the suitability of particular computers for detector design simulations, interested vendors will be allowed and encouraged to optimize parts of the benchmark package by hand to exhibit their machines' performance capabilities. Any results so obtained will be documented, along with the techniques used to generate them, in a manner that distinguishes them from hands-off timing figures.

Candidate Codes:

HEP-related candidate codes for benchmarking include the following:

1. GEANT3 in full simulation mode

   This is a CERN developed code for complete detector simulation. It is an example of a "large" code which, when run in full simulation mode is very compute intensive.

2. EGS4 in full simulation mode

   This is a SLAC developed code for simulation of electromagnetic cascades. It is an example of a "small" code which, when run in full simulation mode is very compute intensive.

3. ISAJET

   This is a BNL developed code for generating the kinematics of physics motivated events expected at high energy colliders.

4. ANLSIM

   This is an ANL-developed code (based on GEANT3) which uses a fast approximate algorithm for shower development attempting to speed up the compute intensive portion of GEANT3.

5. HANSON Tracking Code

   This is a SLAC developed code (based on GEANT3) which implements a specific charged particle tracker.

Candidate Computers:

Specific computer models on which the benchmark codes may be run include:

CRAY YMP series
CRAY XMP series
CONVEX C series
Amdahl
IBM 3090 series
Multiflow Trace
Intel RISC-based HyperCube
Alliant FX/8
BBN Butterfly-2
FNAL ACP-2
Advanced Workstations:
Sun, MIPS, Silicon Graphics, Apollo, DecStation 3x00 series

Discussion of Parallel Benchmarking:

There are a large number of commercially available and HEP community- developed MIMD parallel processors. The typical performance claimed for such computers on the type of HEP codes considered here compares well to existing mainframe supercomputers, and the cost is claimed to be substantially lower. Based on previous work [3] we believe it is possible to develop a portable parallel programming environment for FORTRAN codes which makes effective use of most MIMD parallel machines, giving rise to further advances in speed and computing effectiveness.

In this part of the project we will develop a set of event-level parallel codes to benchmark a wide variety of parallel MIMD computers. The development of parallel codes will make use of the PARMACs set of tools developed in the Argonne Mathematics and Computer Science Division to produce a system which is highly portable between different parallel computers and which requires minimum modification of the original single-processor code. The PARMACs system is a set of tools which act as a machine independent language extension to standard FORTRAN 77, which allows the applications programmer to express the natural parallelism inherent in many FORTRAN codes. These new pseudo-fortran elements provide a compact, easily understood abstract model of parallel programming in an architectual and machine independent manner. A standard macro processing utility is used to convert the PARMACs elements into the appropriate data structures, FORTRAN 77 code, parallel programming library and operating system calls to implement the parallel program on the desired target parallel computing system. More details on the PARMACs system and an example application to a small demonstration HEP code will be found in reference [3].

Parallel implementation of Monte Carlo codes can require significant restructuring of pseudo-random number generation mechanisms in order to ensure reproducible results. Reproducibility is even more difficult when the number of processors is allowed to vary. Such considerations are important in benchmark contexts—to compare machine performance, it is helpful to ensure that the tasks performed by various machines are as nearly identical as possible. We will investigate various approaches to the reproducibility problem in our

development of the parallel benchmark package.

We will also investigate alternative methods for producing portable parallelized code. Two approaches will be considered: first, we will use the PARMACs package that can be installed on UNIX based systems; second, we will use a collection of tools designed for use with existing Fortran codes to detect loop-based parallelism.

These tools will be used to produce machine-specific compiler directives to force parallel execution of loops that the compiler may not be able to detect by itself. A particular instance of this situation involves loops in which subroutine calls occur. Because of the general nature of the Fortran language, it is not always clear that subroutine loops can be parallelized. The tools noted above will provide interaction with the user to determine whether loops can be parallelized and (when possible) invoke automatic insertion of the necessary compiler directives.

These alternative two approaches will be investigated and comparisons made of the applicability and efficiency of each.

Personnel

The Principal Investigator and physicist/manager for each of these parts will be Edward May of the Argonne High Energy Physics Division. The work described in both parts can be done in parallel,all support will be provided by ANL. The enthusiastic support and cooperation of the various vendors will be actively pursued. The approximately 2 FTE's of analysis and programmers (probably two or more people) outlined above will be provided by the Argonne Computing and Telecommunications Division.

Cost

Item		**Operating**	**Equipment**
Technical Personnel	2.2 FTE	$185K	
Travel		10	
Computing		5	
Workstation			15K
Other M&S		10	
Subtotal		210	
Argonne Overhead (25%)		52	
Total		$262K	15K

We comment briefly on the estimated budget above. We anticipate some travel to vendors to implement the PARMACs system on newer machines which are not connected via a network. Our experience with INTEL Corp in the past year suggests a week of work per system as not unusual. The base code systems will be developed and tested at ANL. Some computing services and cycles will be obtain directly from CTD on the usual ANL computing cost recovery schedule. In our experience of installing the EGS4 system in the parallel environment of the ACRF [3], we found that an X11 windowing workstation strongly enhances productivity in the development and debugging of parallel codes. Therefore we

propose to acquire a Unix-based workstation (Decstation 3100) to connect to the Unix server system proposed as part of separate proposal for Simulation Software Center Library at ANL [4].

Schedule

This work will be accomplished during the one-year period of October 1, 1989 - September 30, 1990 (FY90). The first item of work will be the serial benchmarking study; this will be followed by (and overlap) the parallel benchmarking work.

References

1. Loken, S.C., "Computing For High Energy Physics In the 1990's", Study on High Energy Physics in the 1990's Snowmass, CO (June 1988), Proceedings.

2. Ballam, J., "Report of the Task Force on Computing for the SSC", SSC internal report SSC-N-579, private communication.

3. May, E., "Portable Parallel Programming in a Fortran Environment", Conference on Computing in High Energy Physics, Oxford Univ. April 1989. Lusk, E., et.al., "Portable Programs for Parallel Processors", Holt, Rinehart and Winston, New York, 1987.

4. Trost, H., "Simulation Software Center for the SSC at ANL", Proposal, private communication.