TO BE PUBLISHED IN PROCEEDINGS OF 1984 SUMMER STUDY ON THE DESIGN AND UTILIZATION OF THE SUPERCONDUCTING SUPER COLLIDER, SNOWMASS, COLORADO, JUNE 23 - JULY 13, 1984

COMPUTING REQUIREMENTS FOR S.S.C. ACCELERATOR DESIGN AND STUDIES

BNL--35263

DE85 001231

A. Dragt[*], R. Talman[+], R. Siemann[+], G.F. Dell[++], B. Leemann,[**]
Ch. Leemann[**], U. Nauenberg[*+], S. Peggs[+], and D. Douglas[**]

[+]Cornell University; [++]Brookhaven National Laboratory; [**]Lawrence Berkeley Laboratory;
[*+] University of Colorado; [*]University of Maryland

I. Introduction

We estimate the computational hardware resources that will be required for accelerator physics studies during the design of the Superconducting SuperCollider. It is found that both Class IV and Class VI facilities (1) will be necessary. We describe a user environment for these facilities that is desirable within the context of accelerator studies. An acquisition scenario for these facilities is presented.

II. Calculations Required for Accelerator Design and Studies

An analysis of computer requirements for the S.S.C. accelerator project must investigate the scope of calculations that will be performed and the demands that the computations will place on an S.S.C. computing facility. We have identified several categories of computations which place special demands on any computing facility intended to service accelerator study groups. A listing of these categories and a short characterization of each follows.

A. Single Purpose Programs: short codes intended to perform specific calculations (such as the evaluation of luminosity formulae for various parameters sets) which are neither time nor memory intensive. Typical running times for such programs would be a few VAX-seconds or minutes; these programs would consist of a few hundred lines of code and require only a few tens of kilobytes of memory. The primary demand that such use places on a system is that the system be user friendly and support tools for interactive program development and execution. It should be observed that although such use of a system is often the least c.p.u. intensive, it is, for many (if not most) users the most labor intensive; the highest percentage of the user's connect time is spent on such applications. The criterion of user-friendliness is therefore very important.

B. General Purpose Programs for Accelerator Design: large codes intended to perform a variety of calculations related to machine design. There are at least three types of accelerator design codes (2):

1) Beam Optics Programs
2) Particle Tracking Programs
3) Programs for the Calculation of Magnetic and Electromagnetic Fields, Impedances, etc.

Each type of program places particular demands on a computer and operating system.

1) Beam Optics Programs (Examples: SYNCH, TRANSPORT, TURTLE, MAGIC, parts of MAD, MARYLIE, ALIGN, LATTICE,...) compute lattice functions, tunes, closed orbit corrections, etc. for a given machine design. Computation times are dependent on the calculation being performed, but typically range from a few seconds to a few minutes on a CDC-7600.

Example: To fit tunes and compute matched lattice functions for a 6.5 Tesla lattice (such as that in the Reference Designs Study (3)) SYNCH requires approximately 10 c.p.u. seconds on a CDC-7600. Assuming that the CDC, in single precision, runs 30 times faster than a VAX 11/780 in double precision (double precision is required for sufficient accuracy (4)), this calculation will require 5 c.p.u minutes on a VAX. To perform a momentum scan on the same lattice (compute matched off-energy lattice functions), SYNCH requires 30 c.p.u. minutes on the CDC-7600. The corresponding VAX time (scaling by the factor of 30) will be 15 c.p.u. hours.

Certain lattice calculations may, as the above example indicates, be performed on a VAX-like machine, provided the machine is not heavily loaded. Others, such as momentum scans, are better suited to more powerful Class IV computing environments.

For a large machine (such as the S.S.C.) memory requirements can also be rather extreme. A program to simulate closed orbit errors due to magnet misalignments must, for example, store six random numbers (three displacements, three angular misalignments) for each of the 5000 magnetic elements in an S.S.C. lattice. The required storage (assuming 64 bit words) is then

$$
\begin{aligned}
&(6\ \text{errors/magnet}) \times (5000\ \text{magnets}) \\
&\quad \times (64\ \text{bits/error}) \times (1\ \text{byte}/8\ \text{bits}) \\
&\qquad = 240\ \text{kilobytes}
\end{aligned}
$$

for just the location of magnets! Typical optics codes (such as SYNCH) often require over 500 kbytes memory to be loaded. Finally, in the design phase of the machine, numerous runs of such programs must be made. Convenient facilities for dataset preparation should be available, and the machine should have a high throughput, in the event that time consuming calculations (such as the momentum scan of the above example) are required.

2) Particle Tracking Programs (Examples: PATRICIA, RACETRACK, TEVLAT, MARYLIE, DIMAT, Parts of MAD,...) are perhaps the most time intensive of all design codes.

Example: Dell (5) finds that, on a CRAY-1S, PATRICIA can track 1 particle for 1 turn in the 6.5 Tesla Reference Designs lattice in 91 milliseconds. (this is with no vectorization of the program - pipeline processing features of the CRAY were not exploited). To track a single particle for one synchrotron period (approximately 1000 turns) therefore requires 91 seconds. A typical tracking experiment might require tracking 5 particles for 100 synchrotron periods for 10 settings of some parameter (such as the maximum energy deviation of a synchrotron oscillation). Such an experiment therefore requires

---

[++] Work performed under the auspices of the U.S. Dept. of Energy.

(5 particles) x (100 synchrotron oscillations)
x (91 seconds/synchrotron oscillation/particle)
x (10 parameter settings)
= 450,000 seconds = 126 hours.

That is, this single experiment would require almost 1 week of c.p.u. time on a dedicated CRAY-1S. The requirement of high speed is obvious. Moreover, if a number of such experiments are to be performed (as is expected), the vector processing capabilities of the CRAY must be exploited. Were this to be done, a factor of 7 in the reduction of execution time might be achieved, bringing the total required time to about 18 c.p.u. hours in the example above.

3) Programs for the Calculation of Magnetic and Electromagnetic Fields, Impedances, etc. (Examples: POISSON, SUPERFISH, BCI, TBCI,...) Two aspects of such programs are of interest. The first is that three dimensional field calculations will be necessary. This will create unprecedented demands on any computing facility both in terms of execution time and memory. Futhermore, code development on such programs will require a user-friendly environment supporting extensive software development tools.

Secondly, such codes (even existing two-dimensional field finders) are also quite time intensive. Siemann (6) observes that computation of the impedance of an object in the vacuum chamber may require a mesh of 200,000 points and as much as 8 c.p.u. hours on an IBM 3081. During the design phase of the S.S.C., perhaps 100 such calculations must be performed each year (for development, e.g., of cavity designs), meaning that a total of

(8 hours/computation) x (100 computations/year)
= 800 hours/year

or over 1 c.p.u. - month per year of dedicated time on an IBM 3081 will be required for this type of calculation alone. If we assume that a CRAY-1S is about 4 times faster than the IBM in scalar arithmetic, the availablity of a CRAY would immediately reduce the rather substantial 800 hour/year figure to a more moderate 200 hours/year. In addition, use of the pipelining feature of the CRAY could reduce this number by an additional factor of 7. In this case, only 29 c.p.u. hours/year would be needed. The reduction in computing time from a c.p.u. month to a c.p.u. day is very desirable, as it would release a mainframe computer for other uses.

The large number of mesh points needed for such calculations also places additional demands on the computing hardware. First, a large, fast, real (as opposed to virtual) memory is desirable. Each mesh point requires seven words of storage (six field components and a word that describes the mesh). Therefore the 200,000 mesh point example of Siemann requires 1.4 million words of storage. Secondly, a large address space is desirable to facilitate the exensive initialization of code that is required.

<u>C. Beam Dynamics Codes</u>: fall into at least two classes.

1) Beam-Beam Interaction Simulations
2) Single-Beam Instability Simulations

It will be necessary to initiate a number of beam-beam interaction simulation studies during the design phase of the S.S.C. Such calculations may prove to be very time intensive, especially if effects such as the abort gap in the train of bunches, the long-range beam-beam interaction, and coupling of beam-beam forces to nonlinear forces from the lattice magnets, are properly simulated. The inclusion of nonlinear lattice effects, for example, is the basic problem of particle tracking calculations. Some beam-beam simulations therefore will require as much time as tracking calculations to account for the nonlinearities in the accelerator lattice, and will, in addition, require additional time to simulate the effect of the beam-beam interaction at the crossing point. If the simulation of the beam-beam force is done in some detail, it could as much as double or triple the time required over that demanded by a tracking calculation.

Single beam instability calculations may also be very time consuming. Siemann (7) makes estimates of the time required to perform a typical computer simulation of a single beam instability. He finds that 1 to 2 c.p.u. hours on an IBM 3081 are required to simulate 1 second of real S.S.C. time, if 5000 test particles are used and provided the lattice is modeled using linear beam transport. If nonlinear lattice effects are included, the computation time may increase by a factor of 10 to 50. If we wish to simulate the machine behavior for 10 synchrotron oscillations (on the order of 10 seconds) with nonlinear beam transport, we may optimistically expect to use

(1 hour/experiment/sec)
x (10 synchrotron periods)
x (1 sec/synchrotron period)
x (10 times speed reduction for nonlinear lattice)
= 100 c.p.u. hours/experiment .

Pessimistically we may need as much as

(2 hour/experiment/sec) x (10 synchrotron periods)
x (1 sec/synchrotron period)
x (50 times speed reduction for nonlinear lattice)
= 1,000 c.p.u. hours/experiment .

These times (especially the latter) are extremely long. If we now assume that the calculation is done on a CRAY-1S (which runs a factor of four times faster than the IBM in scalar arithmetic) with a vectorized code (which provides as much as an additional factor of seven in speed) the optimistic time estimate falls to only (100 hrs/4)/7 = 3.6 hrs, and the most pessimistic time estimate falls to 36 hours. Optimistically, the calculation is therefore well within the range of possibility, and even pessimistically the situation is not hopeless, provided only a limited number of such computations need be performed. In either case, the need for high speed computing is clear; in the pessimistic case, vector processing (pipelining) is necessary.

<u>D. Other Possible Simulation Calculations</u>: there will be a class of simulations that will include operational modeling of the S.S.C. to determine the effects of r.f. noise, simulation of feedback systems and calculation of the effects of real (i.e. measured) magnetic field errors on beam behavior. It is not known, at present, the extent to which such calculations will be compute-bound. Experience with various other modeling and simulation calculations, such as those discussed above, implies that a need for Class VI computing power will arise.

Each of the above classes of programs may require extensive database handling facilities. For example, tracking calculations with measured multipole data from actual magnets may require input of 30 or more measured multipoles for each of the 5000 ring magnets. This list of 150,000 numbers must be entered in the appropriate sequence to correctly model the location of each magnet in the ring; such a task is greatly facilitated by database handling facilities. Such facilities have already been employed at CERN, where the design code MAD accesses the engineering database used to define LEP.

III. Conclusions

A. Implied Machine Requirements

Several of the aforementioned categories clearly demand the availability of Class VI computing time. Without Class VI time it will not be possible to perform the necessary tracking calculations, nor will it be possible to perform the required beam dynamics computations. Also, the availability of Class VI computing power will greatly facilitate the electromagnetic field calculations (especially in three dimensions) which must be performed.

However, some categories of calculation require less that Class VI power. In particular, lattice calculations, special purpose codes, database manipulations, etc., as well as job file editing and some aspects of code development (e.g. source file editing) are better suited to a user-friendly Class IV system. Moreover, extensive interactive use of most Class VI machines leads to an appreciable degradation in performance. It is therefore desirable to provide Class IV computing facilities to those users who do not require the special features (high speed and/or vector processing) of a Class VI machine.

B. User Environment

The rapid degradation of Class VI machine performance with increasing numbers of interactive users suggests strongly that such a machine should be primarily used in a batch operation context (on jobs requiring several minutes to several hours c.p.u. time). The number of interactive users on any Class VI machine dedicated to accelerator studies should therefore be limited to a sufficiently small number such that the machine performance is not impaired. Experience at LANL and LLL suggests that the number of interactive users on a CRAY-1S - like machine should be limited to the order of 20. If more interactive use is anticipated, additional Class VI computing power should be provided.

The "20 user" criterion is not unreasonably restrictive if low demand jobs (such as dataset preparation and source file editing) are restricted to a Class IV machine provided for the less time intensive jobs. This implies that suitable networking facilities (including the capability of batch job submission and return between Class VI machines and Class IV machines) are available. The Class VI machine is to be used to run those codes which demand high speed and pipelined processing; the Class IV machine is to be available for all other tasks. Communication between the two machines will be enhanced if the same operating system is employed on both. (This has an added advantage - if one machine is down, the other could be used to carry the entire computing load for a short period of time.) A possible model for such an arrangement is provided by the DC system in use at LBL. In this system, a CDC-6600 serves as an interactive host for a purely batch CDC-7600. The 7600 throughput is maintained at a high level because the machine performance is not degraded by numerous interactive demands. Communication between the two is, however, transparent; both machines employ the same operating system.

Each individual user should have available those facilities required to make effective use of the available computing power. Thus, the system should support interactive high resolution (1024x512 pixel minimum) high speed (19.2 kbaud) graphics, high speed (9.6 kbaud) terminals with screen editing facilities and high speed (several hundred lines / minute) local printing facilities for each user group. The communication between user and machine should be transparent (no network lag time) and low cost (minimal contribution to overhead). There should be adequate disk storage space for each user. (This may mean as much as 250 Mbytes or more for users developing extremely large codes.) Finally, the system (both Class IV and Class VI machines, considered as a networked pair) should be user-friendly, supporting up-to-date code development and database handling facilities.

C. Siting and Acquisition Scenario

The Class IV and Class VI machines should both be sited so as to insure maximum performance. One means to achieve this end would be to site the Class VI machine at an existing Class VI facility which has a history of effective management of such machines and a desire to service a broad user community. This will avoid the (possibly long and costly) "break-in" period which an inexperienced management often encounters upon first exposure to the special demands of the Class VI hardware environment. Possible siting locations for Class VI hardware include the MFECC at LLL, the LANL computing center, and the computing center at the National Center for Atmospheric Research.

Upon siting at an existing Class VI facility, the Class VI machine would be tightly networked to the Class IV system. It is assumed that the Class IV machine will be sited at the S.S.C. design center; the networking between the Class IV and Class VI machines should, however, be of such a quality that both machines can be considered as local (on site) computing to the user community at the design center. Members of the S.S.C. design team who are not located at the design center should have ready, high quality, high speed access to both machines through either direct connection from their home institutions or via one of several national networking facilities. The connections for such users should appear (as for the local users at the design center) transparent.

Computing power of Class IV and VI will be required at the time the S.S.C. design project begins. To insure that it is available, early aquisition is therefore desirable.

IV. Notes and References

(1) A "Class IV" computer is a large main-frame computer with fast cycle time (on the order of 200 nanoseconds or less), large memory and high batch throughput. Examples: CDC-7600, CDC Cyber 850-870 series, IBM 3080 series. A "Class VI" computer is a so-called "supercomputer" with a very fast cycle time (on the order of 30 nanoseconds or less), all other features of Class IV machines, and parallel or vector (pipeline) processing. Examples: CRAY's, CYBER-205.

(2) See, e.g. E. Keil, Proc. 1982 Summer School on High Energy Particle Accel., A.I.P. Conf. Proc. No. 105, M. Month, ed. (1983).

(3) M. Tigner et.al., Report of the Reference Designs Study Group on the Superconducting Super Collider (1984).

(4) If 32 bit words are used, the mantissa of each number occupies 24 bits. Each variable is therefore exact to one part in $2^{24} \cong 2^4 \times 10^6$. Each matrix multiply will produce roundoff errors in the last bit. If these errors accumulate (a typical occurence, e.g. in a VAX), <u>all</u> precision is lost after 2^{24} (i.e. 4×10^6) matrix multiplies. The relative precision after N multiplies is $N/4 \times 10^6$.

The SSC contains of the order of 10^4 elements. The results of a matrix calculation of betatron functions will be exact only to within 1 part in $10^4/4 \times 10^6$ (or about .25%) if the calculation is done in 32-bit single precision. For a machine with a tune of 100, this leads to a tune error of the order of .25. This error is completely unacceptable. Double precision (64 bit word is therefore required.

(5) F. Dell, note on tracking in the SSC (private communication).

(6) R. Siemann, in the Report of the 1983 Ann Arbor Workshop on Accelerator Physics Issues for a Superconducting Super Collider; UM HE 84-1, M. Tigner, editor.

(7) ibid.

DISCLAIMER