ACCELERATOR SIMULATION ON A PARALLEL COMPUTER

A. Jejcic, J. Maillard, J. Silva

Laboratoire de Physique Corpusculaire Collège de France, 75231 Paris Cedex 05, France

M. Auguin, F. Boeri, A. Vincent-Carrefour

Laboratoire des signaux et systèmes LASSY Université de Nice, 06041 Nice Cedex, France

Abstract

Implementation of tracking codes on a parallel SIMD/SPMD computer is described. Performances obtained with both parallelisation schemes are discussed and comparison is made with corresponding figures given by a VAX run. Attempt is made to draw some general conclusions concerning utilisation of parallel computers for accelerator calculations.

1 Introduction

Tracking codes have been used for many years for accelerator design and study. Their efficiency is mainly determined by the likelihood one could attribute to the calculated results on the basis of comparison made with observations at disposal. The operational flexibility is also an important factor, intimately bound to the available computing power. The use of parallel computers could allow significant reduction of execution time of existing codes by increasing the available processing power, and could extend the field of applications by improving the simulation of physical processes and the operational conditions.

Parallel computers are represented by a broad class of systems exploiting the parallelism in processing as the very central implementation feature. The amount of operated concurrency could vary from a large fraction of code in intrisically parallel codes to a sequence of instructions when only vectorisation could be worked out. The computing efficiency is closely related to the matching one could achieve between architecture and the simulated physical process. In the work reported hereafter, two particular examples of tracking codes are considered, in order to illustrate how the performances of a parallel computer can be influenced by the architecture. They correspond to two types of approaches one has to adopt for implementation of particles tracking codes. Either the code can be considered as intrisically parallel and an SPMD (Single Program Multiple Data) scheme with network controlled data stream is used, or only a time consuming DO loop is identified and the SPMD or SIMD (Single Instruction Multiple Data) scheme is worked out. The first case is illustrated by the simulation of coherent beam-beam interaction on SSC[1], the second by a PATRICIA simulation of an early ESRP lattice[2].

2 Description of the utilised computer

The reported results were obtained with OPSILA, a prototype computer developped by LASSY (Université de Nice, France) and SINTRA/ALCATEL under DGA/DRET contract [3,4].

The overall structure of OPSILA corresponds to a classical SIMD scheme [fig.1]. It can be divided into three main parts:

- 16 couples of memory bank (MB) and processing elements (PE). Each PE has all the functionalities of a sequential processor (i.e. own instruction decoder and sequencer), the data are provided by its associated bank or by the interconnection network (IN).
- the interconnection Benes network with an associated controller unscrambles vectorial data before processing and realizes data exchange between PEs.
- the central control unit is composed of two processors; the scalar processor (SP) and the vector instruction processor (IP). The SP fetches instructions in the scalar memory (SM), executes the scalar instruction and sends to IP a description of the vector code sections contained in SM.

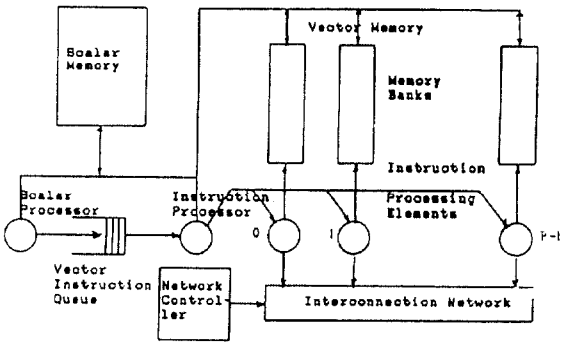

Figure 1: *OPSILA block diagram*

Though its computing power is rather low (5Mips,500kflops), due to the technology which allows only 500 ns clock interval, its architecture enables efficient exploitation of different kinds of parallelism usable for numerical algorithms. Two operating modes are provided:

- synchronous mode (SIMD): the machine acts like a monoprocessor operating upon vectors.
- asynchronous mode (SPMD): the machine configuration is a set of 16 completely independent processors.

The computing set up is completed by a micro VAX used as a frontal computer.

The implemented software comprises an assembler and a high level compiled programing language HELLENA matched to OPSILA architecture[5]. The software appears as rather poor if experimental high energy physics applications are considered, it is sufficient for calculations requiring only elementary mathematical functions as it is the case for the reported work[6].

3 Tracking code implementation

Tracking codes perform simulation of particles motion in accelerators. They operate by transforming vectors, describing particles dynamics, conforming to the repeated action of electromagnetic forces localised in the different elements composing the machine. They pro-

duce numerical values for phase space portraits from which characteristics of investigated accelerator structures are deduced.

Tracking codes parallelisation take advantage on the following two features:

- calculations are carried out on vectors
- repeated execution of the same computing patterns

Either the first characteristic is exploited and a SIMD parallelisation is worked out or the second and a SPMD parallelisation has to be achieved. In fact the choice is rather subtle if optimal performances are sought. The parallelisation efficiency results from an interplay between the computer (a set of 16 processors in the case of OPSILA), code and data structure.

3.1 Coherent beam-beam effect simulation

The parallelisation potential of OPSILA has been examined through the implementation of a code written by A. Chao and M. Furman[1]. This code calculates, for the particular case of the SSC, the perturbation produced by successive beam-beam frontal and long range interaction in the linear approximation for the electromagnetic forces and examines its influence on the stability of one-dimensional betatron oscillation of rigid beams. It was considered as intrinsically parallel and an SPMD parallelisation scheme with network control has been worked out.

A rather simple application was considered for working out the parallelisation procedure. In the machine (the SSC), 8 bunches were assumed to circulate in each opposite direction, their time distribution in the lattice allows 4 frontal collision points and 4 longe range collision points per turn. The original code was entirely redrafted. The implementation was based on the choice following which two processors were affected to each collision point and the data streaming was worked out by the network. So eight processors handle the positron beam and the other eight the electron beam. They calculate the perturbation on the betatron motion resulting from beam-beam interaction and its transfer to the next collision point. The data flow from one processor to the other one is realised by the network, for which an appropriate controlling sequence has been written and implemented as an operating primitive. Thus data fetching, known to be a time consuming operation, is eliminated. One could notice the similarity between the data flow from one processor to the other and the longitudinal motion of bunches from one collision point to another.

Concerning the benchmarking a comparison of execution time was made with figures obtained on a VAX 785, a gain factor of 8 was reached. This figure has to be appreciated by taking into account that for other applications [6] OPSILA exhibits performances similar to that of a VAX 785.

3.2 PATRICIA implementation

3.2.1 SIMD processing mode

PATRICIA, worldwide known chromaticity correction code, implementation was achieved by partitioning the code in two parts[7]. One part representing about 90 percent of the code and devoted to transfer matrices and particle amplitude calculations was implemented on the frontal micro VAX. The second, the time consuming part representing the remaining 10 percent of the code and acomplishing the tracking was HELLENA recoded for SIMD processing and installed on OPSILA. At this level the code was sligthly modified so as to allow the tracking of any number of particles over a given number of machine revolution. The possibility to exhibit the computed stability domain was also worked out.

On figure 2 the results concerning the parallelisation efficiency are brought together . The data for code running were provided by an early lattice of the ESRF project.

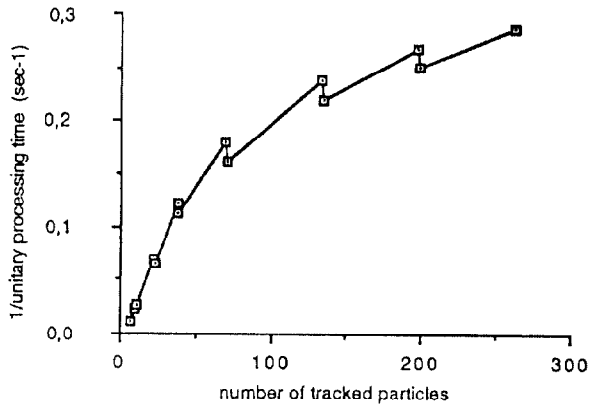

Figure 2: *Unitary processing time versus the number of tracked particles*

The efficiency is expressed by the inverse of the processing time per particle for 400 turns tracking as a function of the number of tracked particles. The plot displays the usual behaviour exhibiting an asymptotic value for the unitary processing time met for very long vectors (for 2048 tracked particles one has encountered 2.92 for the unitary processing time). An efficiency decrement is observed for vector dimensions not being equal to a multiple of 16, it is another obvious feature of SIMD parallelisation.

Figures 3 and 4 represent two examples of output one has obtained by tracking 2048 particles over 400 turns. One shows the stability domain for a given set of hexapolar corrections, the other is an example of a phase space plot.

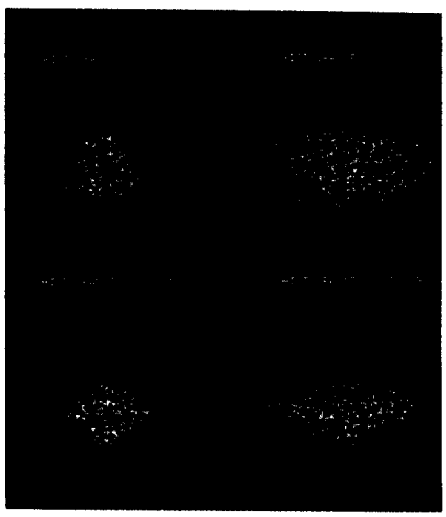

Figure 3: *Phase space plot at 400 turns*

The asymptotic unitary processing time could be compared to the 8 s obtained by running PATRICIA on a micro VAX.

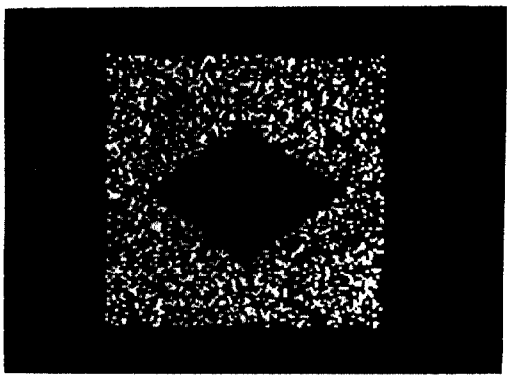

Figure 4: *Stability domain displayed in the xy plane in term of multiples of beam dimensions (in the vertical direction one has 80 and horizontal 100 times σ)*

3.2.2 SPMD processing mode

The SPMD parallelisation was worked out by HELLENA recoding the 10 percent portion of original code implemented on OPSILA.

Figure 5 represents the unitary processing time versus the number of tracked particles, the processing efficiency displays a monotonous growth toward an asymptote, no efficiency decrement is noticed as the number of tracked particles is increased.

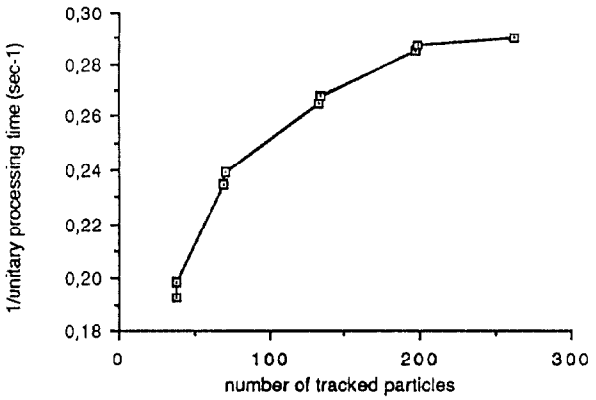

Figure 5: *Unitary processing time versus the number of tracked particles*

3.2.3 Comparison of SIMD and SPMD parallelisation

On the table below the results for the two parallelisation modes are gathered together

Nb part	τ SIMD (sec)	τ SPMD (sec)
32	8.75	5.25
33	9.45	5.09
64	5.75	4.31
65	6.52	4.25
128	4.31	3.81
129	4.71	3.78
192	3.83	3.54
193	4.10	3.52
256	3.58	3.48

Table 1: **Unitary processing time for the two parallelisation schemes.**

One notices that the two modes converge to the same asymptotic value and that for lower number of particles the SPMD parallelisation is more efficient. This is explained by the fact that for the SIMD mode the processing charge of the instruction processor generating instruction and data adresses is independent on the vector size.

4 Conclusion

The reported results allow to draw the following conclusions:

- Tracking codes performances could be significantly improved by implementation on parallel computers, gain factors of one and even two orders of magnitude could be foreseen. The choice of the parallelisation mode has to take into account the computer and code structure and code implementation problems.
- The architecture flexibility of OPSILA represents a real advantage when optimal performances are sought for different operational conditions. The computing potential of OPSILA is important if one takes into account that the component microprocessors are more than 5 years old as are the utilised manufacturing methods.
- The parallel architecture could rapidly provide consistent improvements in computing power and thus enable to work out more precise tracking codes based on a better description of involved electromagnetic fields.

Professor M.Froissart is gratefully aknowledged for help and support.

References

[1] M.Furman,
Results of coherent dipole beam-beam interaction studies for SSC lattices,
SSC Central Design Group, May 1986

[2] B.Buras and S.Tazzari,
European Synchrotron Radiation Facility,
ESRP c/o CERN, October 1984

[3] M.Auguin and F.Boeri,
The OPSILA Computer,
International Workshop on parallel algorithm and architecture.
Marseille-Luminy, France April, 14-18, 1986

[4] M.Auguin and F.Boeri,
Presentation of the LASSY's work in the field of parallel architecture,
Seminaire International sur les Supercalculateurs Scientifiques.
Paris, France. February, 2-6, 1987

[5] Y.Jegou,
Le langage vectoriel HELLENA,
IRISA/INRIA, Rennes France 1988

[6] A.Jejcic and al.,
Use of an SPMD computer for simulation and
pattern recognition
Workshop on detector simulation for the SSC.
Argonne, USA August, 24-28, 1987

[7] H.Wiedeman,
Users guide for PATRICIA,
PEP Technical Memo,PTM-230,(1981)