SPACE-CHARGE SIMULATIONS USING PARALLEL ALGORITHMS

M. Giovannozzi, CERN, Geneva, Switzerland

Abstract

The evaluation of the space-charge effects on the beam dynamics requires CPU-time intensive numerical simulations. To obtain accurate results it is necessary to consider a distribution of particles and to track them through the magnetic lattice. To reduce the real time needed by the simulations, a possible solution is to develop parallel algorithms. In this way, the computational burden can be distributed over many CPUs thus reducing the time to obtain results. In this paper parallel techniques to simulate space-charge effects are presented; in particular we investigate the overall performance of the algorithm and its scalability.

1 INTRODUCTION

The new generation of particle accelerators can be divided into two groups: the first one includes the high energy machines, such as the Large Hadron Collides [1], while the second one includes the high intensity machines such as "drivers" for inertial fusion, neutron sources or nuclear waste incineration. The first class is intended to test the standard model. Due to the high energy, in order to reduce the overall size of the machine, it is mandatory to use superconducting magnets to guide the charged particles along the machine circumference. This generates strong nonlinear magnetic field errors, due to the rather poor field quality of superconducting magnets. Hence the motion becomes highly nonlinear and instabilities are generated, thus reducing the beam lifetime and the overall machine performance.

For the high intensity machines, the main source of concern comes from the strong interaction between particles in the beam. Once again this generates strong nonlinear forces and instabilities.

In recent years, a set of tools has been developed to study these nonlinear phenomena [2]. Analytical tools, such as normal forms, provide powerful techniques to deal with the resonant behaviour of nonlinear systems. Unfortunately, the divergent character of the series involved in the normal form theory shows up at the border of the stability domain, also called dynamic aperture, thus making such a tool almost useless in the analysis and cure of instabilities and particle losses. Therefore, the only way to gain some insight consists in carrying out numerical simulations and to apply sophisticated techniques to reconstruct the phase-space topology from tracking data [3, 4].

2 SIMULATION OF SPACE CHARGE EFFECTS

In a previous paper [5] it has been shown how the study of nonlinear beam motion can be pursued using parallel algorithms. In particular, the issue of evaluating the dynamic aperture for non-interacting particles has been considered and an optimal solution found. This approach has the advantage of reducing the total CPU-time needed in numerical simulations, thus achieving higher accuracy.

In spite of the encouraging results, it is clear that the analysis of space-charge effects on the particle motion represents the real benchmark for the development of efficient parallel algorithms for beam dynamics. Some attempts to use a parallel approach to solve this problem can be found in the literature [7, 8, 9], but there are still many open problems.

In the model considered here, it is assumed that the beam is a continuous unbunched beam. For the sake of simplicity it will be assumed that the motion is restricted to one plane only (in any case the generalisation to the four-dimensional case is straightforward). Under these hypotheses, the charged particles are described by the two-dimensional transverse coordinates. The initial beam distribution is defined in terms of a set of $N_{\rm part}$ particles $x_1(0), \ldots x_{N_{\rm part}}(0)$ called super-particles.

The magnetic channel, a circular machine or a transfer line, is made up by linear elements interleaved with nonlinear elements. The effect of the space-charge forces is equivalent, in first approximation, to a reduced focusing strength all along the magnetic channel. It is then possible to replace this continuous effect by a series of space-charge kicks uniformly placed along the magnetic channel. This approach is equivalent to the one used to represent the thick nonlinear magnetic elements as the composition of a sequence of drifts and nonlinear kicks [2]. This technique produces a symplectic solution to the equation of motion. The evolution of each super-particle is computed via the following tracking scheme

$$\begin{pmatrix} x_i(n+1) \\ p_i(n+1) \end{pmatrix} = \mathbf{M}_n^k \begin{pmatrix} x_i(n) \\ p_i(n) + \tau K(x_i(n)) + \eta F_i(N_{\rm part}) \end{pmatrix}$$

where $\mathbf{M}_n^k$ represents the transfer matrix of the nth slice of the kth magnetic element and τ, η are integration steps related with the number of slices used to represent the element $\mathbf{M}^k$. The function K is a polynomial function describing the nonlinear kick: it is different from zero if the kth element is a nonlinear magnetic element. The function F takes into account the space-charge interaction, namely

$$F_i(N_{\rm part}) = \sum_{1 \le j \le N_{\rm part},\, j \ne i} \mathcal{F}(|x_i(n) - x_j(n)|). \qquad (1)$$

For a model where the super-particles are considered as short charged segments of length $2L$ at a distance r and

with charge density ρ, $\mathcal{F}$ takes the form

$$\mathcal{F} = -\rho^2 \log(1 - \frac{4L^2}{r^2}). \qquad (2)$$

In the limit $L/r \rightarrow 0$ $\mathcal{F}$ tends to the standard form of the Coulomb force. The proposed tracking scheme actually represents a set of $N_{\rm part}$ coupled equations, describing the evolution of the set of super-particles.

The critical point in the definition of an efficient parallel algorithm is the computation of the space-charge interaction term, as this requires the knowledge of the coordinates of all the particles in the distribution. This term is intrinsically non-parallel and some special techniques are necessary to tackle this problem.

3 PARALLEL EVALUATION OF COULOMB POTENTIAL AND FORCE

The main difficulty in defining a parallel algorithm to evaluate the Coulomb potential or force consists in the amount of communication between the various processors. This is a consequence of the pair-wise and long-range character of the electro-magnetic interaction.

The first step consists in defining an efficient parallel decomposition of the problem. Supposing that $N_{\rm local}$ stands for the number of initial conditions stored in a certain processor, then $N_{\rm part} - N_{\rm local}$ coordinates should be communicated to compute the force and $N_{\rm part}$ $N_{\rm local}$ pair-wise interactions should be evaluated. It is possible to estimate both the time spent in communication $T_{\rm comm}$ and in computation $T_{\rm comp}$

$$\begin{aligned} T_{\rm comp} &\approx \max(N_{\rm local}) N_{\rm part}\ t_{pair} \\ T_{\rm comm} &\approx \max(N_{\rm part} - N_{\rm local})\ t_{exch}, \end{aligned} \qquad (3)$$

where $t_{\rm pair}$, $t_{\rm exch}$ stand for the CPU-time needed to compute a single pair-wise interaction and to communicate a single coordinate while $\max(x)$ is the largest value of x in any processor of the system. If $t_{\rm pair} \approx t_{\rm exch}$ then the optimal decomposition will minimise $T_{\rm comp}$ and will be such that $\max(N_{\rm local})$ is minimised, in other words one should store the same number of initial coordinates in each processor. Due to the lack of locality in long-range forces, the way initial coordinates are shared among processors is irrelevant.

The next problem concerns the flow of communication. The previous considerations are valid provided the processors can communicate without bottlenecks. This is certainly not the case in practice, unless the connection topology between processors is carefully chosen. The solution consists in defining a *ring topology*: communication can only occur between nearest neighbours. Each processor receives data from its left neighbour and it sends data to the right neighbour. This mechanism avoids bottlenecks and it allows data to be sent around the ring in only $N_{\rm proc} - 1$ time steps, where $N_{\rm proc}$ is the number of processors. Thanks to this topology the time needed to communicate the particle positions to all processors is simply $N_{\rm part}\ t_{\rm exch}$. This optimal result is based on the assumption that the communication is performed concurrently and that the load imbalance is negligible.

At each communication step, the pair-wise contributions to the force are computed on every processor. It is then necessary to take into account the symmetry of the Coulomb force, to avoid self-interactions, and also to avoid counting pair-wise contributions twice. All these constraints can be fulfilled fairly easily.

These considerations have been gathered together into a parallel algorithm to evaluate Coulomb forces for a given distribution of charged particles. The approach followed in the design of the piece of code consists in using standard FORTRAN 77 instructions for the actual computations combined with calls to the functions of the MPI library [10] to control communications between processors and the communication topology.

4 HARDWARE DESCRIPTION

The MEIKO CS-2 computer used to test the parallel algorithm presented in the previous section, is a distributed-memory, scalable, parallel system using SPARC microprocessors and a MEIKO-developed interconnection which enables programs to read and write memory in remote nodes without context switching. The CERN CS-2 has 64 nodes, each with two 100 MHz HyperSPARC processors (rated at over 100 Specint92 per processor) and 128 MB of memory. Each node has a local disk for temporary data storage and paging or swapping as well as SCSI connections for additional peripheral equipment.

The CS-2 service is at present used for the support of data recording and event reconstruction for high energy physics experiments, and for event-parallel simulation using a specially developed version of the GEANT program. It also provides a Parallel Interactive Analysis Facility (PIAF).

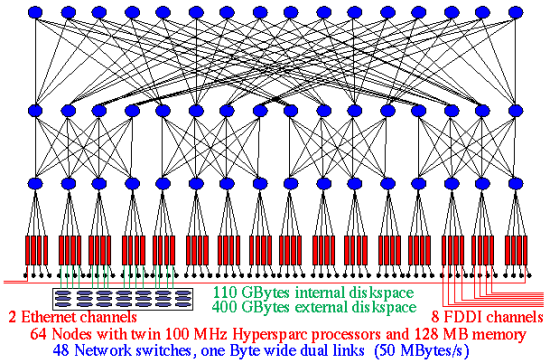

Figure 1: Schematic view of the CS-2 structure. The connections between different processors are shown.

5 NUMERICAL RESULTS

A series of numerical checks have been carried out in order to study the performance and scalability properties of the algorithm. The tests consist of the evaluation of the Coulomb force for a distribution of 50000 particles, whose coordinates and charge are randomly chosen. The algorithm is bench-marked on the CERN MEIKO machine and the total number $N_{\rm proc}$ of processors used in the evaluation of the force is varied in the range $1 \leq N_{\rm proc} \leq 66$. The real time needed for these computations as a function of $N_{\rm proc}$ is shown in Fig. 2. The steady reduction of the real time as a function of $N_{\rm proc}$ is the sign that the algorithm is performing well, and that the real time is not dominated by communications. Furthermore, the real time scales as $1/N_{\rm proc}$ as it should. In Fig. 3 the speed-up of the algorithm as a function of $N_{\rm proc}$ is shown. The speed-up is defined as

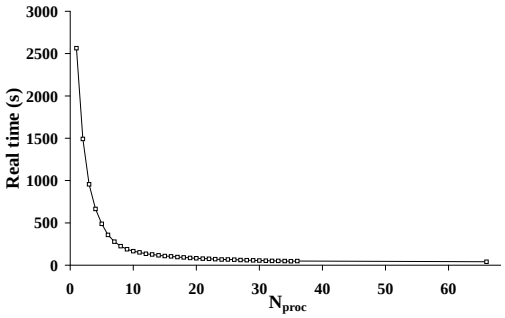

Figure 2: Real time to compute the Coulomb potential as a function of the number of processors $N_{\rm proc}$. The number of particles used in the computation is 50000.

$$\mathcal{S}(N_{\rm proc}) = \frac{T(1)}{T(N_{\rm proc})}, \qquad (4)$$

where $T(N_j)$ is the real time needed to evaluate the force using N_j processors. In the ideal case $T(N_j) \propto T(1)/N_j$ and $\mathcal{S}(N_j) \propto N_j$. This is actually the case as can be seen in Fig. 3. Furthermore, two interesting features can be observed. First of all, starting from $N_{\rm proc} = 5$, the speed-up satisfies $\mathcal{S}(N_j) > N_j$. This can be explained as a *cache effect*: the data set has the right size to fit the cache and the higher access speed improves the overall performance of the algorithm. Secondly, the case $N_{\rm proc} = 66$ shows a saturation effect. This feature is the consequence of communication overhead: as $N_{\rm proc}$ increases, the communication time dominates over the computing time and the performance of the algorithm gets worse (this is the so-called *Ahmdal's law*).

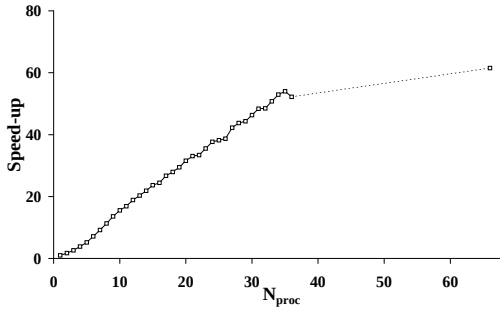

Figure 3: Speed-up due to the parallel evaluation of the Coulomb potential as a function of the number of processors $N_{\rm proc}$. The number of particles used in the computation is 50000.

6 CONCLUSIONS

The numerical tests confirm the assumption that the ring topology is the appropriate solution to avoid bottlenecks due to communication flow. Using this approach it is possible to evaluate efficiently Coulomb forces with a system of concurrent processors. Numerical simulations of space-charge effects could profit from the development of the parallel algorithm presented here.

7 ACKNOWLEDGEMENTS

I should like to express my gratitude to E. McIntosh for valuable help and useful discussions. I thank G. Turchetti for many stimulating discussions. Furthermore, I would like to thank N. Re for computing support.

8 REFERENCES

[1] The LHC Study Group, "The Large Hadron Collider: Conceptual Design Study", *CERN AC (LHC)* **95-05** (1995).

[2] A. Bazzani, E. Todesco, G. Turchetti and G. Servizi, "A normal form approach to the theory of nonlinear betatronic motion", *CERN* **94–02** (1994).

[3] R. Bartolini, M. Giovannozzi, W. Scandale, E. Todesco, "Methods for nonlinear single-particle dynamics: the contribution from the CERN-Bologna collaboration in the framework of the HCM network", *CERN LHC Note* **134** (1998).

[4] R. Bartolini, A. Bazzani, M. Giovannozzi, W. Scandale and E. Todesco, *Part. Accel.* **52**, (1996) 147.

[5] M. Giovannozzi, and E. McIntosh, *Int. J. Mod. Phys. C* **8**, (1997) 155.

[6] M. Giovannozzi, E. McIntosh, G. Franchetti, G. Turchetti, "New computational physics applications: parallel approach to nonlinear effects in beam dynamics", *Unpublished Note Bologna University* (1997).

[7] R. D. Ryne, S. Habib and T. P. Wangler, in *Sixteenth Biennial Particle Accelerator Conference*, edited by LANL (IEEE Operations Center, Piscataway, NJ, 1996) p. 3149.

[8] J-M. Lagniel and D. Libault, in *Sixteenth Biennial Particle Accelerator Conference*, edited by LANL (IEEE Operations Center, Piscataway, NJ, 1996) p. 3235.

[9] G. F. Dell and S. Peggs, in *Sixteenth Biennial Particle Accelerator Conference*, edited by LANL (IEEE Operations Center, Piscataway, NJ, 1996) p. 2327.

[10] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, "MPI: The Complete Reference", (The MIT Press, Cambridge, 1995).