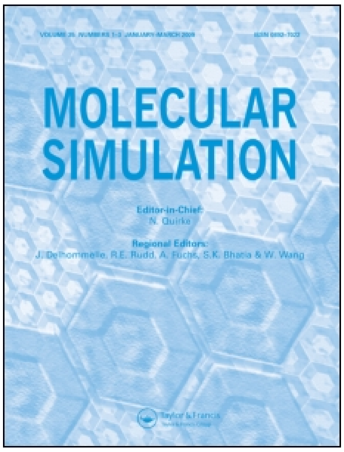

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Molecular Dynamics Simulation on the Connection Machine

A. Windemuth[a]; K. Schulten[a]
[a] Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA

To cite this Article Windemuth, A. and Schulten, K.(1991) 'Molecular Dynamics Simulation on the Connection Machine', Molecular Simulation, 5: 6, 353 — 361

To link to this Article: DOI: 10.1080/08927029108022421

URL: http://dx.doi.org/10.1080/08927029108022421

Molecular Simulation, 1991, Vol. 5, pp. 353–361
Reprints available directly from the publisher
Photocopying permitted by license only

MOLECULAR DYNAMICS SIMULATION ON THE CONNECTION MACHINE

A. WINDEMUTH and K. SCHULTEN

Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

(*Received January 1990, accepted March 1990*)

An algorithm for the efficient calculation of macromolecular force fields on the Connection Machine is described. The full force field is separated into bond interactions and non-bonding interactions. Only the latter are implemented on the Connection Machine, the former, less computationally intensive tasks are performed by an existing, conventional molecular dynamics code on the front end. Parallelization of the evaluation of non-bonding interactions is achieved by the *Replicated Systolic Loop algorithm* introduced in this paper. The algorithm is a variant of the *Systolic Loop* scheme often used for the computation of 2-particle forces for the classical N-particle problem.

KEY WORDS: Connection machine, replicated systolic loop algorithm, molecular dynamics simulation, parallel processing

1. INTRODUCTION

Molecular dynamics simulation methods have become a common tool for the investigation of dynamical properties of large biological macromolecules [20, 16, 15, 14, 5, 7], e.g. proteins, RNA and DNA. These calculations consume considerable amounts of computer time, mostly spent on the evaluation of a large number of identical pair interactions. This makes molecular dynamics simulation a good candidate for parallel processing on a Single-Instruction-Multiple-Data (SIMD) machine, which is potentially the most efficient solution for high throughput numerical calculations.

The Connection Machine is a massively parallel SIMD computer with a large number of very simple processors [11]. The machine presently available to us has 32,768 one-bit processors and an array of 1,024 floating point units theoretically capable of 1.2 billion floating point operations per second. The conservative design of the Connection Machine, i.e. a low clock rate and conventional fabrication technology, promises an increase of its capabilities by orders of magnitude in the near future, making accessible problem sizes and time scales never considered for simulation before.

In molecular dynamics almost all computational resources are spent on the calculation of non-bonding forces. These are composed of the electrostatic Coulomb interaction and the attractive/repulsive van der Waals interaction embodied by the Lennard-Jones-potential. In order to conserve computation time, calculations are often carried out with these interactions being limited to neighbouring atoms, interactions between atoms further apart than a specified cut-off distance being ignored [3]. A cut-off of non-bonding interactions can be a good approximation for van der Waals interactions, which have a limited range. Electrostatic interactions, however, extend over long distances. This is especially important in large protein molecules, where inhomogeneous charge distributions are thought to play an important role for protein stability and function. In addition to being of questionable value for the faithful representation of biopolymer dynamics, the introduction of force cut-off also creates technical problems. The necessity of switching functions to maintain the differentiability of the potential energy function, as well as the need to maintain pair lists to account for the proximity of atoms entail additional computation and memory requirements, partially offsetting the advantage gained through the reduced count of interacting pairs.

There are methods to calculate the electrostatic forces to a given degree of precision within a time of the order of $N \ln N$, either by solving the Poison equation [19] or by using a multipole expansion [10, 13]. These methods require complex programs, making them hard to parallelize, and yield significant performance gains only for very large systems that are currently too large to simulate for a sufficiently long time.

It is the purpose of this paper to demonstrate how the calculation of non-bonding interactions can be implemented with a high degree of parallelization and efficiency on the Connection Machine. By taking into account the full range of electrostatic interactions and separating them from the more complex, multi-particle bonding interactions the problem to be handled by the Connection Machine is reduced to an N particle system with two-body interactions only. It turns out that the integration of the trajectory and calculation of bonding interactions take very little time compared to the calculation of pair interactions, making it feasible to implement these tasks on the front end, a conventional serial computer. This allows conventional molecular dynamics codes containing considerable amounts of developed code to be used without having to reimplement and parallelize the code as a whole.

For the parallization of particle simulation several possibilities have been proposed [12]. Two of them are depicted in Figure 1, the *Maximally Parallel Algorithm* (or *Simultaneous Force Calculation*) and the *Systolic Loop Algorithm*.

The Maximally Parallel Algorithm assigns one processor to every pair of atoms, such that each processor calculates exactly one pair interaction per simulation step. This requires $N(N-1)/2$ processors, where N is the number of particles simulated. Because the evaluation of one pair interaction typically involves only a few basic arithmetic operations, communication overhead is large in this method.

The Systolic Loop algorithm, also called "Digital Orrery" algorithm [12], passes the coordinates of all atoms around in a ring of N processors in $N/2$ steps, such that the coordinates of half of the particles pass by every processor exactly once. In addition,

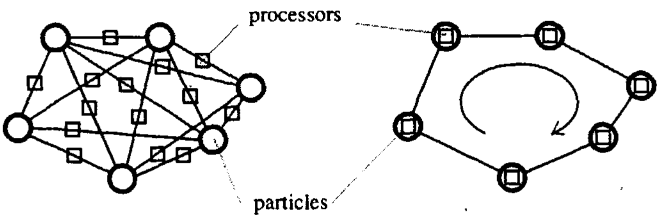

Figure 1 Parallel particle simulation algorithms. Two different algorithms for the simulation of N particles [12] are shown; the *Systolic Loop Algorithm* (right) uses N processors, the *Maximally Parallel Algorithm* (left) uses $N(N-1)/2$ processors, each processor evaluating the interaction of a single pair of particles.

each processor stores the coordinates of one atom. Thus, each processor can evaluate and accumulate the interaction of this atom with half of the other particles. Only half are needed because the interaction is reciprocal, such that the interaction of A with B is equal to that of B with A. This method has very little communication overhead, because coordinates do not have to be distributed over many processors and forces are accumulated locally with no need of communication. However, the method can only make use of N processors, thus severely restricting the problem size for a given machine configuration.

These and more sophisticated methods have been implemented on a variety of parallel processors [9, 2, 1], resulting in good parallelization, however, only for a small set of problem sizes. Some hybrid methods have been proposed and implemented on the Connection Machine [17] as well as on the *Distributed Array Processor* (DAP) [18, 8]. These methods generalize the Maximally Parallel Algorithm to increase the flexibility of system size and should be comparable in efficiency to the method presented here.

The Replicated Systolic Loop algorithm described here is an extension of the Systolic Loop algorithm and can handle any degree of parallization in between the single loop case and the maximally parallel algorithm. For realistic problems in Molecular Dynamics, where the number of atoms N is several thousands, this is an essential feature, as the approximately 32,000 processors of the Connection Machine are too few for the maximally parallel algorithm, but too many for a single systolic loop.

2. THE REPLICATED SYSTOLIC LOOP ALGORITHM

The essence of the replicated loop algorithm is replication of the systolic loop in such a manner that every loop starts with a different pairing of atoms (Figure 2). If there are three loops, for example, the first loop would start out pairing atoms 1 and 2, the second loop pairing atoms 1 and $(N/6) + 1$ and the third loop pairing atoms 1 and $(2N/6) + 1$. After $N/6$ coordinate passing steps, each loop will have calculated one third of all interactions, together yielding the full set of interactions.

Whereas the single systolic loop is being implemented as a one-dimensional ring of processors, the replicated loop algorithm can be mapped onto a two dimensional grid of $M \times N$ processors, where M is the number of loops and N the number of particles. The horizontal axis of length N is called the *loop* axis, while the vertical axis of length M is called the *replication* axis. The flexibility of the algorithm derives from the

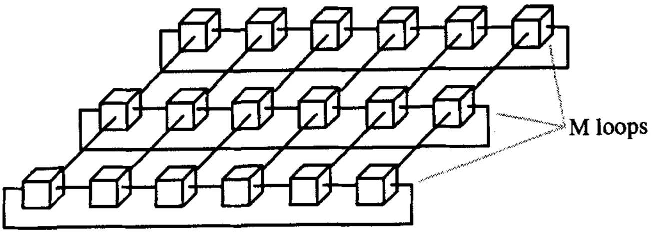

Figure 2 Replicated loop algorithm. The replicated loop algorithm uses an array of M systolic loops, each loop calculating part of the total interaction. The number M can be chosen so as to use any number of processors from N to $N(N - 1)/2$.

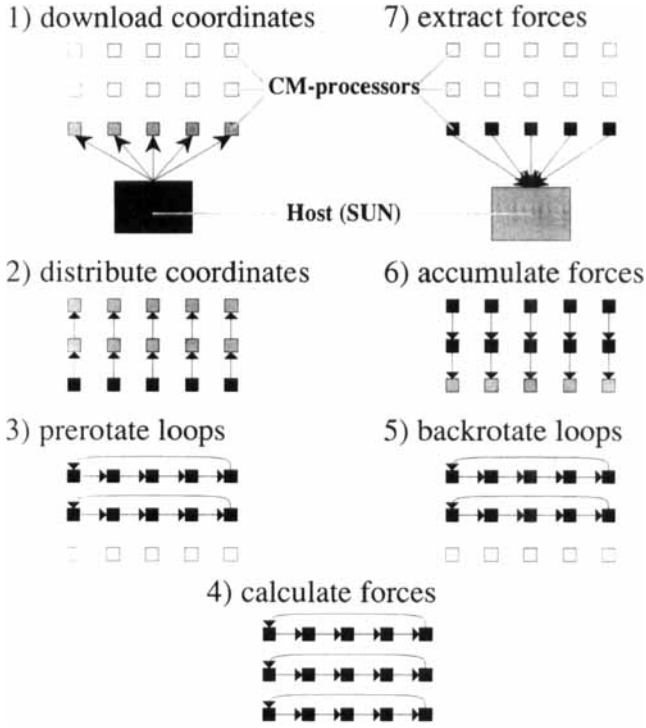

Figure 3.

arbitrariness of M. Theoretically it can be chosen anywhere from 1 to N, in practice some degradation of performance occurs for values of M larger than about $N/10$. This degradation is due to the overhead incurred by the preparation (stage 1–3 below) and collection (stage 5–7 below) phases relative to the calculation itself (stage 4 below), which normally dominates the calculation when M is not too large. A computer of 32,000 processors, thus, would be able to efficiently simulate from 500 up to 32,000 atoms. Exploiting the virtual processor mechanism provided by the Connection Machine, system size can be extended up to one million atoms on the present configuration.

To do one evaluation of the force field, the following stages are performed in sequence (see Figure 3):

1. Transfer of coordinates from the host processor to the first loop of processors.
2. Replication of coordinates along the replication axis.
3. Prerotation of the replicated loops to their starting positions. The m'th loop is rotated by $mN/2M$ positions.
4. Shifting of the coordinates and evaluation of forces. This is the actual calculation stage and is repeated $N/2M$ times.
5. Backrotation of forces along the loop axis. This reverses the shifting effect of stages 3 and 4, but transfers the calculated forces instead of coordinates.

6. Accumulation of forces along the replication axis. This sums the partial forces of each loop to yield the total force acting on each atom.
7. Transfer of forces to the host processor.

Associated with each processor in a loop are the following quantities defining an atom:

- coordinates x, y, z.
- electric charge q.
- type index t.
- lists of van der Waals constants A and B
- forces f_x, f_y, f_z.

The charges, type indices and van der Waals constant lists are initialized at the beginning of the simulation and need not be changed at each integration step. The type index indicates which van der Waals parameters are to be used for a given atom. Each atom is assigned a specific type, usually related to its chemical properties. The van der Waals parameters A and B for a pair of atoms are then given by a two dimensional matrix indexed by the type indices of the two atoms. Because one partner of the pair is always the same in a specific processing element, only one column of the matrix has to be stored per processor. In contrast to the fixed parameters described above, the atom coordinates have to be loaded from the front end on every integration step. This is done in stages 1 and 2. An additional set of quantites is needed to define the partner atoms which are moved around the array during stages 3 and 4:

- coordinates of the partner atom x', y', z'.
- charge of the partner atom q'.
- type index of the partner atom t'.

At each shifting step during stage 4, first the difference vector and distance between the atoms of each processor are calculated according to the scheme

$$r_x = x - x'$$

$$r_y = y - y'$$

$$r_z = z - z'$$

$$r = \sqrt{r_x^2 + r_y^2 + r_z^2}$$

Then the electrostatic and van der Waals potential of the pair are calculated according to

$$E_{\mathrm{elec}} = \frac{qq'}{r}$$

$$E_{\mathrm{vdW}} = \frac{A[t']}{r^{12}} - \frac{B[t']}{r^6}.$$

These energy values are accumulated during stage 4, such that at the end of stage 4 each processor contains the sum of interaction energies with all partner atoms that were shifted through.

The forces on each atom are then calculated by multiplying the derivatives of the Energy values by the normalized difference vector

$$f_x = (E'_{\mathrm{elec}} + E'_{\mathrm{vdW}})\frac{r_x}{r}$$

$$f_y = (E'_{\text{elec}} + E'_{\text{vdW}})\frac{r_y}{r}$$

$$f_z = (E'_{\text{elec}} + E'_{\text{vdW}})\frac{r_z}{r},$$

where the Energy derivatives are

$$E'_{\text{elec}} = \frac{qq'}{e^2}$$

$$E'_{\text{vdW}} = -\frac{12A[t']}{r^{13}} + \frac{6B[t']}{r^7}.$$

At the end of stage 4, the accumulated energy values are globally summed and extracted to yield the total electrostatic and van der Waals energy of the system. Forces are summed along the replication axis at stage 6, yielding the total electrostatic and van der Waals force acting on each atom.

For implementation the following features of the Connection Machine have been extremely useful:

- Array transfer functions allow fast transfer of data between the front end and the Connection Machine. This feature was used for stages 1 and 7.
- Spreading operations distribute values over an array of processors along arbitary axes. This feature was used for stage 2.
- Long distance shift operations cyclically shift values inside an array along an arbitary axis. One command can only shift by a power of two, but combining only a few commands (of the order of $log_2 N$), shifting by any number of positions can be accomplished efficiently. This feature was used for stages 3 and 5.
- Efficient local communications between neighboring processors in an array using the NEWS (North, East, West, South) communication system was used for the shifting operations at stage 4.
- Scan or reduction operations use a binary tree mechanism to combine a large number of operands in parallel (again taking time of the order of $log_2 N$) using various combining rules. Using addition as a combining rule the summations of energy and forces were performed at stages 4 and 6.

The spreading, reduction and long distance shift operations are made possible by the hypercube architecture of the Connection Machine's communication system, which is ideally suited to the kind of binary tree algorithms used. Because the powerful parallel operations described are fully implemted in the PARIS instruction set, their use was very straightforward and resulted in a reasonably small and simple program, in spite of the low level nature of the language.

3. PERFORMANCE COMPARISONS

The algorithm outlined was implemented as a subroutine in a larger, previously existing molecular dynamics program[21] written in the programming language C. This program uses the force field also implemented in the well-known programs

Table 1 Comparison of calculation times for the non-bonding interactions of the complete photosynthetic reaction center (12637 atoms) and a segment thereof (3634 atoms) using different machine configurations. The results show that the full machine power available can be used efficiently regardless of system size. The Cray 2 calculations were done using the program X-PLOR, which uses neighbor lists and thus not accomodate the full reaction center with the available memory.

Machine:	*Connection Machine*			*Cray 2*
Processors:	8,192	16,384	32,768	1
3634 atoms:	6.3 s	3.6 s	1.8 s	4 s
12637 atoms:	106 s	57 s	23s	-

CHARMM[3] and XPLOR[4]. It can read and write the input and output files as well as the force field parameter sets used by these programs.

The parallel algorithm was coded in C/Paris, a subroutine library interfacing the Connection Machine to the Host, a SUN/4, on an assembly language level. Due to the low-level nature of the interface, and because some of the commands are extremely powerful, the implementation is very efficient compared to higher level language implementations.

The flexibility of the implementation was demonstrated by calculating energy and forces for two systems of different sizes. One is the whole photosynthetic reaction centre of *Rhodopseudomonas viridis* [20, 6] with 12,637 atoms, the other is a spherical section of the reaction center including the most important prosthetic groups and containing 3634 atoms. The respective calculations were repeated for machine configurations of 8,192, 16,384 and 32,768 processing elements.

To illustrate the performance in conventional terms a comparison with the Cray 2 was included in the benchmark. The calculations on the Cray 2 were done using the program X-PLOR[4] with an infinite cut-off for the non-bonded forces. This program keeps neighbour lists even in the infinite cut-off case, so that available memory did not allow calculation of all interactions of the photosynthetic reaction centre without modification.

The results shown in Table 1 indicate that there are few limitations on the number of processors used for a given computational task. However, there is an architectural limitation of the Connection Machine that limits processor arrays to dimensions which are powers of two. This means, that a system of 12,637 atoms is not any faster than one with 2^{14} = 16,384 atoms. It is expected that this limitation will be removed in future versions of the Connection Machine.

4. CONCLUSION

When used efficiently, a SIMD computer like the Connection Machine is the most cost effective way to carry out computationally demanding tasks. We have shown how this can be achieved for molecular dynamics simulations even with the complex intra-molecular potentials used for simulating large macromolecules like proteins. By selectively parallelizing the computation-intensive evaluation of non-bonding interactions only, the speed advantage of the massively parallel computer can be realised. At the same time the full capabilities of a conventional molecular dynamics program can be retained for the calculation of all other interactions, eliminating the need for a major reprogramming effort. The Replicated Loop Algorithm in combination with the virtual processor mechanism of the Connection Machine provides a large flexibility with regard to the size of the system simulated and makes the power of an enlarged machine immediately available for the same problems previously calculated on smaller machines. The Connection Machine proves to be ideally suited to the Replicated Loop algorithm because of its efficient grid communication system as well as its powerful spreading, scanning and long distance shift operations made possible by the machine's hypercube architecture.

Acknowledgements

We like to thank the staff of the National Center for the Supercomputing Applications and of Thinking Machines Corporation for their assistance. The development and test computations were carried out on the CM2 of the National Center for Supercomputer Applications supported by the National Science Foundation.

References

[1] D.J. Adams. "The implementation of fluid phase monte carlo on the DAP." *J. Comp. Phys.*, **75**; 138–150, (1988).

[2] L.L. Boyler and G.S. Pawley."Molecular dynamics of clusters of particles interacting with pairwise forces using a massively parallel computer." *J. Comp. Phys.*, **78**; 405–423, (1988).

[3] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus. "CHARMM: A program for macromolecular energy minimization, and dynamics calculations." *J. Comp. Chem.*, **4**; 187, (1983).

[4] A.T. Brünger. "Crystallographic refinement by simulated annealing." In N.W. Isaacs and M.R. Taylor, editors, *Crystallographic computing 4: Techniques and new technologies*, Oxford, 1988. Clarendon Press.

[5] P.G. Debrunner and H. Frauenfelder. "Dynamics of proteins." *Ann. Rev Phys. Chem.*, **3**; 283, (1982).

[6] J. Deisenhofer and H. Michel. "The crystal structure of the photosynthetic reaction centre from rps. viridis." In J. Breton and A. Vermeglio, editors, *The photosynthetic bacterial reaction center: Structure and dyanmics*, pages 1–3, London, 1988. Plenum Press.

[7] R. Elber and M. Karplus. "Multiple conformational states in proteins: A molecular dynamics analysis of myoglobin." *Science*, **235**; 318, (1987).

[8] D. Fincham, N. Quirke, and D.J. Tildesley. "Computer simulation of molecular liquid mixtures. 1. a diatomic Lennard-Jones model mixture for CO_2/C_2H_6." *J. Chem. Phys.*, **84**; 4535–4546, (1986).

[9] David Fincham. "Parallel computers and molecular simulation." *Molecular Simulation*, **1**; 1–45, (1987).

[10] L. Greengard and V. Rohklin. "A fast algorithm for particle simulations."*J. Comp. Phys.*, **73**; 325–348, (1987).

[11] W.D. Hillis. "*The Connection Machine*." MIT Press, Cambridge, Massachusetts, (1985).

[12] W.D. Hills and J. Barnes. "Programming a highly parallel computer." *Nature*, **32**; 27, (1987).

[13] R.W. Hockney and J.W. Eastwood. *Computer Simulation Using Particles*. McGraw-Hill, New York, (1981).

[14] M. Karplus and J.A. McCammon. "Dynamics of proteins: Elements and functions." *Ann. Rev. Biochem*, **53**; 263, (1983).

[15] M. Karplus and J.A. McCammon. "The dynamics of proteins." *Scient. Am.*, **4**; 30, (1986).

[16] J.A. McCammon. "Protein dynamics." *Rep. Prog. Phys.*, **47**; 1, (1984).

[17] B.G.J.P.T. Murray, P.A. Bash, and M. Karplus. " Molecular dynamics on the connection machine system." Technical Report, Thinking Machines Corporation, Cambridge, MA, (1988).

[18] M. Neumann, O. Steinhauser, and G.S. Pawley. "Consistent calculation of the static and frequency-dependent dielectric constant in computer simulations." *Molec. Phys.*, **52**; 97–113, (1984).

[19] C. Niedermeier. "Elektrostatische Kontrolle der Primärprozesse im photosynthetischen Reaktionszentrum von Rhodopseudomonas Viridis. Master's thesis, Technische Universität München, München, (1989).

[20] H. Treutlein, K. Schulten, J. Deisenhofer, H. Michel, A. Brünger, and M. Karplus. "Molecular dynamics simulation of the primary processes of the photosynthetic reaction center of rps. viridis." In J. Breton and A. Vermeglio, editors. *The photosynthetic bacterial reaction center: Structure and dynamics*, pages 139–150, London, 1988. Plenum Press.

[21] A. Windemuth. "Dynamiksimulation von Makromolekülen. "Master's thesis, Technische Universität München, München, (1988).