GROMACS: A PARALLEL COMPUTER FOR MOLECULAR DYNAMICS SIMULATIONS

H.Bekker H.J.C.Berendsen E.J.Dijkstra S.Achterop R.v.Drunen
D.v.d.Spoel A.Sijbers H.Keegstra B.Reitsma M.K.R.Renardus
Department of Biophysical Chemistry University of Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands
Department of Computer Science, University of Groningen, Blauwborgje 3, 9747 AC Groningen, The Netherlands

A 32 node, i860 based ring architecture and its Molecular Dynamics software is described.

1. Introduction

The method of Molecular Dynamics (M.D.) has advanced to the point that realistic simulations of complex chemical and biochemical systems are within reach. With accurate interaction potentials and the computational facilities to handle many tens of thousands of particles, thermodynamic equilibrium and dynamic non-equilibrium properties can be probed, predicted, and understood on the basis of simple atomic interactions[1]. The M.D. method has promising potential for the rational design of new materials with specified properties, particularly in biotechnological applications: drugs, vaccines, enzymes. The next decade will see its widespread use in industrial design processes.

Reliable simulations are computationally extremely intensive. For solvated macromolecules a large number of solvent molecules must be included; the time scale on which functional processes occur is often beyond the accessible range. Many relevant properties such as equilibrium constants, binding constants, solubilities, depend on the determination of free energy differences which require a series of simulations over a reversible path.

A typical simulation of a protein in solution requires the computation of the forces on 20,000 particles, due to about 300 non-bonded neighbors every time step of 1 femtosecond. The bottleneck in the computation is the non-bonded force, requiring about 45 flops per interacting pair. For a simulated time span of 1 nanosecond this amounts to 130 Teraflop. On a supercomputer with an average performance of 100 Mflops a single simulation will require 400 hours!

Clearly, parallel computers are a solution for future M.D. simulations. Since the M.D. algorithm is suited for parallelization[2] —the requirements for communication are not extreme, the algorithm is well-tested and stable, and applications to fill the full computer capacity are waiting to be exploited— it is worthwhile to construct and program a parallel computer for M.D. with optimized price-performance ratio.

General-purpose vector or parallel machines will always offer features that are not needed for the special class of problems considered here, but which increase the system costs.

The GROMACS software and machine (GROningen MAchine for Chemical Simulations) aims at bringing M.D. simulations on a parallel architecture down to earth. It uses commercial boards with, but not exceeding, the required functionality. Keywords for GROMACS are: ease of use, wide applicability, high performance, scalability, low cost.

2. Hardware, Architecture, and Portability

The GROMACS architecture consists of i860 based boards connected in a ring. The i860 boards are commercially available (DSM Digital Service GmbH, Munich, Germany) and plug into a normal PC-bus. Each board contains an i860 processor, 8 Mbyte RAM (expandable up to 64 Mbyte), four transputer-link adaptors, and two eight bits wide parallel interfaces (3 Mbyte/sec) to connect the boards in a ring.

The PC-bus is used to load programs and initial data, and to store results. It connects to an i486 based PC which serves as host. The host runs under UNIX in a network environment. The ring of 32 i860 processors is housed in a 19 inch cabinet (See Fig. 1). In this cabinet the PC-bus is split up in eight sections, each containing four i860 boards.

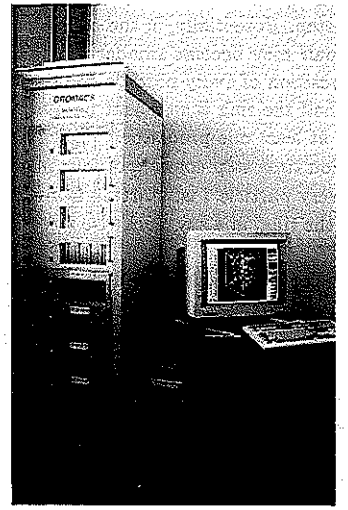

Fig. 1. Photo of GROMACS.

The only hardware specific part of the software is the interboard communication. A machine level procedure of about 100 machine instructions is needed to write

and read the DPRAM which interfaces the boards. All high level communication procedures at different software levels are based on this routine. This eases porting to any message-passing architecture. The algorithm for this procedure is:
Note: DPRAM is the start of the address space of Dual-Port RAM.

```
                out_free = 500; out_avail = 501;
write:          repeat until DPRAM[out_free];
                for i <- 0 to 499 do DPRAM [i] <- x[i];
                DPRAM[out_avail] <- true;

read:           repeat until DPRAM[out_avail];
                for i <- 0 to 499 do y[i] <- DPRAM[i];
                DPRAM[out_free] <- true;
```

3. Parallelization

GROMACS uses a static mapping of particles onto processors. An M.D. system of N atoms, numbered from 1 to N, is mapped on P (in our case $P = 32$) processors by allocating the first N/P atoms on processor 1, the second N/P on processor 2, etc. The processor H_i on which particle i is mapped is called its *home processor*.

Non-Bonded Force (NBF) interactions are evaluated for those pairs of particles for which the relative distance is less than the cut-off radius R_{co}. Because of the diffusive nature of an M.D. system, the set of particles within R_{co} of a given particle may change every time step. Potentially, in the course of a simulation the distance between every particle pair may become smaller than R_{co}. Therefore, at the start of every time step the position of every particle has to be distributed over half the ring. Distribution over half the ring is sufficient because in this way every position pair $\mathbf{r}_i$, $\mathbf{r}_j$ is present on at least one processor.[3]

Every M.D. simulation time step consists of four phases:

1. The position of every particle i is, starting from its home processor H_i, distributed over half the ring, in a say clockwise direction, by $\frac{1}{2}P$ communication pulses.
2. If required, a neighbor list is constructed. Interactions are evaluated.
3. Partial forces on every particle i are communicated, going in the anti-clockwise direction, by $\frac{1}{2}P$ pulses to H_i where they are summed to the net force on particle i.
4. Using these forces, velocities and positions are updated for each particle, on its home processor.

For the communication (during phase 1 and 2) we use an interleaved communication/calculation scheme because the hardware does not allow for efficient simultaneous communication/calculation. This is not a problem because only a small percentage of the total simulation time is spent in communication.

Bonded Force (BF) computations can involve three and four particle interactions, called triplet and quadruplet interactions respectively. For triplets and quadruplets, distribution of particle positions over half the ring may be insufficient because then not all the positions of the triplet or quadruplet are necessarily known on one processor.

During a simulation the set of BF calculations does not change, so the communication characteristics remain the same for a given allocation. This means that a good allocation generated during a preprocessing phase will suffice during the whole simulation.

A good allocation is one in which particles in triplet and quadruplet interactions are assigned close numbers and therefore assigned close processors and thus have little inter-processor communication. An allocation of this type can be found by renumbering particles during preprocessing in such a way that the bandwidth of the interaction matrix becomes small. Minimizing the bandwidth of the interaction matrix can be done with the algorithm REDUCE.[4,5]

4. Software and Functionality

The GROMACS software, written in ANSI C, consists of preprocessing software on the host and M.D. simulation software on the ring. The software is designed to run on a ring of any size. We developed and implemented new and more efficient methods for non-bonded and bonded force calculations,[6] and for neighbor searching.

Neighbor searching is done using a combined grid/list method. First the computational box is divided by a grid. The particles are then distributed over the grid cells. Finally, neighbors of a particle are searched by inspecting neighboring cells. When using a grid size L between $1/2$ and $1/3R_{co}$ we get an optimal neighbor searching speed. In this case approximately one in four inspected particles is in range. A higher in range ratio may be obtained with a finer grid but then the number of visits to empty grid cells will increase, yielding a smaller speed.

The GROMACS software is designed to handle many types of M.D. simulations. Of course we implemented the usual Lennard-Jones, Coulomb and harmonic potentials, many types of bond-angle and dihedral interactions, different types of neighbor searching, and notions like charge groups, exclusions and distance constraints. In the near future we will implement position and distance restraints, coupling to temperature and pressure baths and free energy calculation.

All this would be of little use if we did not offer a comfortable user interface; an M.D. run on GROMACS can be totally prepared and executed using a graphical, X-windows based, user interface. During the actual simulation process the M.D. run can be monitored, interrupted, modified, resumed and stopped. Monitoring involves physical quantities such as temperature, pressure and various types of energy, as well as system behavior such as load balance. After a user interrupt, simulation parameters like time step, cut-off radius, etc. may be changed. The simulation may then be resumed with the modified parameters. Another monitoring feature is a window displaying a user-adjustable view of the instantaneous state of the system.

5. Performance and Scalability

In Table 1 a performance comparison is made. On the CRAY and NEC an M.D. program optimized for these machines was used. The simulations we ran had the following parameters:

first: 1728 water molecules, flexible interactions, Rectangular box with periodic boundary conditions, density 1g/cm^3. Short/long range cut-off of 1.2/1.2nm. Update of the neighbor list either every 10 or (20) time steps.

second: 7776 water molecules, further specification as above.

third: A typical 22-residue peptide in a solvent: 721 water molecules, 85 trifluoroethanol molecules. Rectangular box with periodic boundary conditions.

Table 1. Comparison of machine performance (steps/sec).

processor / simulation	CRAY Y/MP single proc.	NEC SX-3	CONVEX C210	1× i860	32× i860
first[a]	0.52 (0.66)	0.81 (1.04)	n.a.[b]	-	3.33 (3.45)
second[a]	-	0.08 (0.12)	n.a.	-	0.66 (0.7)
third[a]	n.a.	n.a.	0.2	0.6	17.2

[a]See text. [b]not available.

Thus the obtained speed on our 32 processor machine ($300,000) exceeds the speed on the fastest available vector supercomputer in Europe (NEC SX-3) by a factor 3 to 6. Scaling from 1 processor to a ring of 8 processors (tested on 2000 water molecules) gives negligible scaling overhead, but scaling from 8 to 32 processor was only 75% efficient.

Acknowledgements

This research was supported by the Foundation of Technical Research(STW) and administered by the Foundation for Fundamental Research on Matter(FOM) with financial aid from the Netherlands Organization for Scientific Research(NWO).

We thank Dr. Florian Müller-Plathe (ETH Zürich) for running the test simulations with his optimized programs on the CRAY and NEC.

References

1. W.F. van Gunsteren, H.J.C. Berendsen, *Angew. Chem. Int. Ed. Engl.* **29**, (1990) pp. 992–1023.
2. D. Fincham, *Mol. Sim.* 1, (1987) pp. 1–45.
3. A.R.C. Raine, D. Fincham, W. Smith, *Comp. Phys. Comm.* 55, (1989) pp. 13–30.
4. N.E. Gibbs, W.G. Poole, P.K. Stockmeyer, *SIAM J. Numer. Anal.*, 13(2) (April 1976).
5. H. Bekker, E.J. Dijkstra, H.J.C. Berendsen, in *Parallel Computing: From Theory to Sound Practice*, ed. P. T. Wu and E. Milgrom (IOS press, Amsterdam, 1992), pp. 268–279.
6. H. Bekker, H.J.C. Berendsen, E.J. Dijkstra, *et al.*, *Conf. Proc. Physics Computing '92.* (this issue).

GROMACS METHOD OF VIRIAL CALCULATION USING A SINGLE SUM

H.Bekker H.J.C.Berendsen E.J.Dijkstra S.Achterop R.v.Drunen
D.v.d.Spoel A.Sijbers H.Keegstra B.Reitsma M.K.R.Renardus
Department of Biophysical Chemistry University of Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands
Department of Computer Science, University of Groningen, Blauwborgje 3, 9747 AC Groningen, The Netherlands

A single sum method is described to calculate the virial in a molecular dynamics simulation with periodic boundary conditions.

1. Introduction

In most Molecular Dynamics (MD) simulations the pressure of the system is calculated . This is done e.g. to monitor the pressure during the course of a simulation, or to keep the pressure constant by changing the system size. The pressure is calculated as the sum of a kinetic and a configurational part.[1] This last part, called the virial, is the subject of this article.

In the inner loop of the non-bonded force (NBF) calculation every pair interaction is evaluated by calculating the interaction potential, the interaction force, and the contribution to the virial. Because this loop is the most time consuming part of an MD simulation,[2] simplifications in this loop result in a significant reduction in CPU time.

In GROMACS we calculate the total virial of the system by summing over particles instead of the usual summing over interactions. This means that the virial calculations are removed from the inner NBF loop. For simulations of systems in vacuo it is not hard to prove the equivalence of both methods but for systems with periodic boundary conditions it is not as obvious. Therefore, we will concentrate on the proof for systems with periodic boundary conditions.

In section 2 we develop some tools, and use these tools to specify the inner NBF loop. In section 3 we give a algorithm with a double sum virial, and in section 4 we present an algorithm with a single sum virial.

2. Specification of NBF calculations with PBC and virial calculations

An MD system with periodic boundary conditions (PBC) consists of a central box with particles, surrounded by (in principle) an infinite number of translated identical image boxes with particles, stacked in a space filling way. In general the boxes are triclinic, which means that they are defined by six pairwise parallel planes. We define the minimum box size L_{min} as the minimum distance between two of the