

Parallelization of the GROMOS87 Molecular Dynamics Code: An Update

Ken Meacham and Daron Green

Parallel Applications Centre, 2 Venture Road, Southampton, SO16 7NP, U.K.

Tel: +44 1703 760834, Fax: +44 1703 760833

Abstract

In this paper, we describe the performance of the parallel GROMOS87 code, developed under the ESPRIT EUROPORT-2/PACC project, and indicate its potential impact in industry. An outline of the parallel code structure is given, followed by a discussion of the results of some industrially-relevant testcases. Conclusions are drawn as to the overall success of the project, and lessons learned from the porting and benchmarking activities, which show that the parallel code can enable more ambitious use of molecular simulation in industry.

1 Introduction

The GROMOS87 code performs Molecular Dynamics (MD) simulations [1] on a system of atoms or molecules. These calculations involve the numerical integration of Newton's equations of motion. It is a primary concern that these calculations be computed quickly and efficiently, to allow simulations to sample many thousands or millions of configurations. By parallelizing the code, we have significantly reduced the run-time of GROMOS, enabling larger, more industrially significant systems to be simulated. Results are presented in this paper, demonstrating examples of the use of parallel GROMOS87 within industry.

2 Parallel GROMOS Structure

2.1 Overview

A summarised profile of the original sequential code shows the following structure and distribution of computation:

%CPU	Routine	Purpose
100.0	MAIN	Main program
99.5	—RUNMD	MD driver routine
97.3	—FORCE	
6.0	—NBPML	Neighbour list construction
86.9	—NONBML	Non-bonded interactions
1.1	—DIHANG	Dihedral angle forces
0.6	—ANGLE	Bond angle forces
0.0	—DO-LOOP	Update of atom positions
3.7	—SHAKE	Impose bond-length constraints

These cpu-time figures were obtained using a typical system of 4 DNA fragments with 1220 water molecules.

The parallelization scheme used for this code was proposed in [2], and the implementation discussed in detail in [3]. More recent work has focused on parallelizing residual parts of the code (e.g. the dihedral and bond-angle computations), and on testing the performance and industrial capabilities of the parallel GROMOS code.

The main features of parallel GROMOS are described below.

2.2 Data decomposition

GROMOS adopts a Particle Parallelism (PP) approach, whereby work is shared out over the system, for example by dividing it up into small groups of molecules on each processor. This was felt to be preferable to a Geometric Parallelism (GP) approach, where particles are allocated to processors according to their positions in space.

2.3 Neighbour-list construction (NLC)

This is central to the efficiency of GROMOS. A list is created of non-bonded short-ranged atom-atom interactions, which is subsequently used when calculating interaction forces. This list is based on charge groups, and is constructed using a cell-searching algorithm, which allocates charge groups to cells within a compartmentalized system space. By considering only those interactions between charge groups in adjacent cells, the $O(N^2)$ NLC is reduced to an $O(N)$ problem.

2.4 Dynamic load-balancing

In the course of the simulation, it is necessary to update the positions of the atoms synchronously. Consequently, we must manage calculations in such a way that all processors are ready to communicate their data at the same time. In practice, however, movement in the molecular configuration causes fluctuations in the computational load corresponding to each atom, and hence an imbalance in load between processors. We have therefore introduced a scheme which dynamically allocates charge groups to processors, responding to any compute-time imbalance during the non-bonded force calculation (NBFC) phase. This typically reduces the imbalance from 30–40% down to less than 1%, which can result in a significantly reduced overall computation time for the simulation.

2.5 SHAKE Parallelization

The SHAKE routine is an iterative routine used to satisfy bond-constraints, once particle positions have been updated (after the force calculation). For systems of many small molecules, we have split the SHAKE calculation by allocating groups of whole molecules to processors, which can be SHAKEn independently. However, for systems involving, for example, one large protein molecule surrounded by solvent, this decom-

position is inappropriate. To balance out the work-load, it is necessary to split the SHAKE calculation for individual large molecules across two or more processors. This involves communication of atom positions at each iteration, and recent work has revolved around minimizing the communications necessary at each iteration. By ordering the bond lists (before allocating them to processors), it is possible to reduce the number of shared atoms between processors to just one or two, minimising the cost of each communication.

2.6 Data compression

Large amounts of coordinate and velocity data can be output from MD packages, which can seriously reduce the performance of a particular code, particularly a parallel version, as well as leading to excessive disk space requirements. The University of Gröningen has developed a set of portable data compression routines, which have been implemented within parallel GROMOS. These produce coordinate trajectory files around 7 times smaller than the equivalent ASCII files, which are readable across a wide range of architectures. A further advantage resulting from this data compression is an improved performance of the code.

2.7 Communications layer

To ease the process of porting the GROMOS87 code to different architectures, we have developed a library of specific communication constructs required within the code, implemented in terms of different message-passing paradigms (PVM, PARMACS, MPI, etc). This communications layer enables the use of generic calls within the GROMOS code which are independent of the paradigm or machine type. When porting to a particular architecture, it is simply a case of compiling a local version of the layer. This layer has also successfully been used within the MNDO code, as part of the EURO-PORT-2/PACC project.

3 Performance and Capability

3.1 Overview

Assessment of the parallel code has focused on two issues:

- the portability and parallel scalability across various HPC platforms;
- the industrial capability provided by the parallel code.

The first of these was measured by running some medium-sized benchmark cases on a range of different machines. The second was assessed by using the codes to carry out some calculations of industrial interest, provided by Bayer and ENEA.

3.2 Scalability Benchmarks

Scalability benchmarks have been performed on a variety of architectures, which include distributed-memory machines (IBM SP2, Meiko CS2, Intel-i860 Hypercube, DEC Alpha cluster) and shared-memory machines (SGI Power Challenge). Preliminary results have already been reported [4], employing PARMACS and PVM message-passing paradigms. The performance of PVM on the SGI Power Challenge was found to be poor, but recent results obtained using an MPI message-passing layer have been much more encouraging. The code is highly portable, and the conversion from PVM to MPI, taking advantage of the improved message-passing performance, took only 1–2 days.

ENEAs benchmark systems used for GROMOS scalability tests were as follows:

- [E.1] Super-oxide dismutase (SOD) molecule (2682 atoms) without solvent; 2,000 time-steps.
- [E.2] SOD molecule with 4,598 water molecules (total 16,476 atoms); 1,000 time-steps.
- [E.3] SOD molecule with 9,348 water molecules (total 30,726 atoms); 500 time-steps.

processors	Time / s	Speedup	Efficiency / %
1	2,166	1.00	100
2	1,166	1.85	93
4	684	3.16	79
8	452	4.80	60

Table 1. Testcase [E.1] (SOD – 2,682 atoms)

processors	Time / s	Speedup	Efficiency / %
1	11336	1.00	100
2	5852	1.92	96
4	3090	3.64	91
8	1806	6.23	78
16	1094	10.29	64

Table 2. Testcase [E.2] (SOD – 16,476 atoms)

processors	Time / s	Speedup	Efficiency / %
1	11,3367	1.00	100
2	5,826	1.95	98
4	3,074	3.70	93
8	1,722	6.61	83
16	1,115	10.21	64

Table 3. Testcase [E.3] (SOD – 30,726 atoms)

Bayer’s benchmark system used for GROMOS scalability tests was as follows:

[B.1] HIV–1 protease, surrounded by 7892 water molecules
(total of 25,570 atoms); 500 time–steps.

processors	Time / s	Speedup	Efficiency / %
1	9,161	1.00	100
2	4,691	1.95	98
4	2,476	3.70	93
8	1,375	6.66	83
16	886	10.34	65

Table 4. Testcase [B.1] (HIV – 25,570 atoms)

The results from the scalability runs above show good speedups for medium to large sized systems. A speedup of greater than 10 on 16 processors shows a good improvement over previous results [4].

3.3 Industrial Capability Results

Results for two industrially–significant test cases are reported in this paper. These test cases were proposed by industrial end–users of GROMOS (ENEA, Bayer), and scientific results have previously been reported by them at the METECC ’95 conference [4].

ENEA’s testcase for the capability demonstration was:

[E.4] SOD molecule with 9,600 water molecules (total 31,482 atoms);
150,000 time–steps.

SOD is of interest to the medical science community; its function is to catalyse a reaction to eliminate the super-oxide radical from living cells. The SOD molecule has already found a number of clinical applications including:

- reduction of tissue damage in haemolytic diseases (in which red blood cells are destroyed)
- lung cancer from high-energy radiation exposure
- ischaemia (causing reduced blood supply to tissues)
- cosmetic applications

Simulations using parallel GROMOS87 allowed ENEA to investigate of the dynamic behaviour of the SOD molecule in solution, to explore its conformational space.

Bayer's testcase for the capability demonstration was:

[B.1] HIV-1 protease, surrounded by 7892 water molecules
(total of 25,570 atoms); 150,000 time-steps.

Bayer modelled an HIV-1 protease – DMP323 complex, using a combination of all three EUROPORT-2/PACC codes (GROMOS, MNDO and TURBOMOLE). In particular, parallel GROMOS87 was used to study the following areas of interest:

- flexibility of the inhibitor bound in the active site
- flexibility of residues comprising the active site
- displacement of “bridging” water

The ENEA and Bayer systems ([E.4] and [B.1] respectively) were run on a 16-processor SGI PowerChallenge at Cortaillod. The GROMOS simulations were performed in stages ranging from 20,000 steps, for most runs (done during the day), up to 70,000 steps (weekend runs). This segmentation of the simulations provided a means to restart GROMOS, in the event of a system failure. Table 5 shows a summary of the capability demonstration results, with speedups estimated from much shorter runs (since these cases could not be run in full on sequential machines).

End-user	Test case	Actual Number of Time-steps	Run-time / hours	Approx. speedup
ENEA	[E.4] (SOD)	158,530	108.0	10.2
Bayer	[B.1] (HIV)	150,000	80.5	10.3

Table 5. Capability demonstrations on SGI platform

Results were obtained from parallel runs using up to 16 processors, which lasted 108 (ENEA) and 80.5 (Bayer) hours, a reasonable length of time for such computations in an industrial context. To carry out the same computation using a non-parallel code would have taken nearly 2 months, which may be acceptable for academic grand-challenge research, but is too slow in a competitive and responsive industrial R&D context.

4 Conclusions

GROMOS87 has been parallelized successfully and ported to a variety of parallel architectures. This has been achieved without major changes to the basic structure and algorithms within the code. Its use has been clearly demonstrated as a parallel molecular dynamics package, capable of running simulations an order of magnitude faster than before, or providing far superior accuracy in equivalent times.

The parallel GROMOS code makes it possible to simulate much larger systems than ever before, and to achieve overall run-times which allow timely exploitation of results for industrial R&D.

5 References

- [1] M.P.Allen and D.J.Tildesley, "Computer Simulation of Liquids", Clarendon Press, Oxford, 1987.
- [2] D.G.Green, K.E.Meacham and F. van Hoesel, "Parallelization of the molecular dynamics code GROMOS87 for distributed memory parallel architectures", Proceedings of HPCN Europe, Milan, May 1995.
- [3] International EUROPORT Symposium on "New Frontiers in Computational Chemistry: Impact of Parallel Computing on the Chemical and Pharmaceutical Industry", Nov. 1995.
- [4] D.G.Green, K.E.Meacham, M.Surridge, F. van Hoesel and H.J.C.Berendsen, "Parallelization of Molecular Dynamics Code. GROMOS87 parallelization for distributed memory architectures" in Methods and Techniques in Computational Chemistry: ME-TECC-95, Edited E.Clementi and G.Corongiu, STEF, Cagliari, Italy, 1995.