COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **1**, No. 4, pp. 736-743

Commun. Comput. Phys.
August 2006

The GROMACS and NAMD Software Packages Comparison

Armen H. Poghosyan[1,*], Grigor A. Yeghiazaryan[1], Hrant H. Gharabekyan[1] and Aram A. Shahinyan[1]

[1] *The International Scientific-Educational Center of the National Academy of Sciences of Armenia, Marshall Baghramian Ave. 24d, 375019 Yerevan, Republic of Armenia.*

Received 26 July 2005, Accepted (in revised version) 20 November, 2005

Communicated by Dietrich Stauffer

Abstract. The comparable feature analysis of NAMD and GROMACS molecular dynamics packages has been done. The benchmarks of 72 and 128 Dipalmitoylphosphatidylcholine (DPPC)/water have been constructed using a cluster (3GHz-Xeon processors and Myrinet network) and the comparison has been performed using GROMOS87 and CHARMM27 force fields modified for lipids with GROMACS and NAMD software packages, respectively. The GROMACS has been displayed as faster than NAMD, likely due to united-atom character of GROMACS and good implementation features. The GROMACS reaches saturation and goes to the worst results, the reason of which is that the program spends more time on communications between processors.

Key words: NAMD; GROMACS; molecular dynamics; phospholipid bilayers.

1 Introduction

The usage of Molecular Dynamics (MD) simulation in phospholipid bilayers, the main structural elements of cell membranes, makes it possible to study the physical and chemical processes inside of a bilayer and follow directly the conformational changes, to measure all the parameters and compare with the experimental findings. For this reason, any software package, which in a manner regards to the biophysical problems, namely to phospholipid membranes, are of a great interest. There are a lot of famous MD software packages, such as GROMACS [1,2], NAMD [3], CHARMM [4], AMBER [5], TINKER [6], and a lot of

*Correspondence to: Armen H. Poghosyan, The International Scientific-Educational Center of the National Academy of Sciences of Armenia, Marshall Baghramian Ave. 24d, 375019 Yerevan, Republic of Armenia. Email: sicnas@sci.am

different force fields modified for various systems are developed. It is reasonable to divide existing software packages into 2 major categories: (i) Computational (ii) 2D and 3D Construction. Computational software packages include molecular mechanics using various force fields, as well as *semi-empirical* and *ab initio* quantum mechanical calculations, such as GAMESS [7], MOPAC [8], GAUSSIAN [9], etc. In addition to calculation tools, in many cases the packages include also constructions, visualization and some drawing tools, being multi-purpose packages. Most of them are designed to be run generally under Unix/Linux platform, although, the Windows platform based computational softwares are also available, like HyperChem [Hypercube, *Inc*], etc. As for the visualization software, it should be noted that there are widely-used, free packages, like Rasmol and VMD [10]. These are programs for displaying, animating and analyzing large bio-systems by means of 3D graphics and scripting. The VMD code is an excellent tool, especially for lipid bilayer assemblies, which provides a wide collection of various methods for rendering, and even MD trajectory analyzing tools are already developed.

Thus, a great deal of progress has been made in the past decade, from the software point view. The purpose of the present research is a comparison of features of such known MD software packages, as NAMD and GROMACS, which are aimed at the high performance simulation with parallel support.

2 Results and discussion

Some comparison of the software properties is presented in Table 1. The main difference is surely the implementation type and force fields. There are also great differences in case of parallel running. As GROMACS developers claim "fastest MD" in some manner according to our calculations, GROMACS is certainly faster than NAMD; however, the latter scales well in parallel performances [11]. GROMACS offers a lot of analysis module, whereas NAMD has almost no standard tools for analysis, which indeed creates some additional troubles for users having no programming facilities.

The GROMACS uses the GROMOS force field and their modifications, and NAMD has an ability to work with CHARMM, X-PLOR, AMBER and even GROMACS force fields. As far as the GROMACS and AMBER force fields support is concerned, it has some major difficulties and practically it is impossible to launch NAMD with GROMACS force field, e.g. to deal with lipids. The main problem is the restrictions, i.e. NAMD does not support many specific tools and configuration options, such as GROMACS pairs section, exclusions, all types of bond potentials, which exist in GROMACS, etc., although NAMD developers have pointed out that only GROMACS topology (.top) and coordinate (.gro) files are needed.

The GROMACS performs the *energy minimization* by means of two various methods: *steepest descent and conjugate gradient* (Polak-Ribiere) methods. The NAMD offers only one standard method for energy minimization (efficient conjugate gradient, which is claimed a faster minimizer algorithm based on the conjugate gradient method).

Table 1: Comparison of NAMD and GROMACS properties.

Comparison of features	GROMACS	NAMD
`Supported force fields`	GROMOS and all modifications	CHARMM, AMBER, X-PLOR, GROMACS
`Long-range electrostatics`	Classic Ewald summation, PME, PPPM, User defined and Cut-off	PME, FMA (DPMTA, no longer included in released binaries), Full direct and Cut-off
`van der Waals interaction`	LJ (Cut-off, shift and switch)	LJ (Cut-off, shift and switch)
`Molecular Dynamics Simulation options`	Energy minimization (Steepest Descent, Conjugate Gradient), Shell MD, Constant Energy/ Temperature/Pressure, PBC	Energy minimization (Conjugate Gradient), Constant Energy/ Temperature/Pressure Dynamics, PBC, SMD(Steered MD), IMD(Interactive MD), Free energy calculation
`Analysis toolkits`	About 20 tools	No standard tools
`Constrain algorithms`	LINCS, SHAKE	SHAKE, SETTLE (for waters)
`Input/Output`	.gro .pdb /.trr .trj	.pdb .psf / .dcd
`Temperature control`	Berendsen, Nose-Hoover	Langevin dynamics
`Pressure control`	Berendsen, Parrinello-Rahman	Berendsen, Nose-Hoover Langevin Piston
`Multiple time step`	No	Yes
`Others`	NMR Refinement, Free energy perturbation, Electric Fields, Long range dispersion Corrections for Energy and Pressure, User defined things	Fixed atoms, rigid water, temperature coupling, velocity rescaling and reassignment, harmonic restraints, External Electric Fields
`Scripting`	No	Tcl based scripting Interface
`Running Platforms`	Windows, Linux, Mac OS X, and other UNIX platforms	Windows, Linux, Mac OS X, and other UNIX platforms

In general, the concept of *energy minimization* is rather comprehensive, i. e. the potential function has a *global minimum* and a very large number of *local minima*, where all potential energy derivatives are zero with respect to the coordinates and the second derivatives are nonnegative. So, there is no minimization method implemented in the both software packages that finds the global minimum and that is surely only the *nearest local minima*. If there is really a need to reach the *global minimum*, it is recommended to try the so called "simulated annealing" (SA) (to do MD simulation repeatedly at a high temperature for a certain time and then to cool slowly to the required temperature). Although the SA is not fully certified to get a *global minimum*, however, this algorithm is an efficient technique to find the minimum energy configuration of the system [12]. In both packages, the SA is implemented: in a simple way into GROMACS and using *reassignTemp*, *rescaleTemp* and *langevinTemp* commands in NAMD.

As an MD program GROMACS uses a *Verlet* integration scheme - so-called *leap-frog* (coordinates and velocities are leaping like frogs over each others back) algorithm [13] for the integration of the equation of motion (third-order in r and time-reversible).

The input and output requirements are the following: GROMACS uses ASCII .gro as input coordinate and velocity file and .top as an input topology and structure file. The .gro format has an ASCII character and includes the atom and residue names, coordinates and velocities (almost same as in .pdb format). The force fields have .itp file format. The outputs include the same .gro final configuration and also the velocities and the binary .trr /.xtc trajectory file. According to our calculation (same conditions), the size of .trr dynamics file is small than NAMD .dcd file. The problem is that NAMD takes into account also the hydrogen atoms and writes down the additional information concerning the hydrogen velocities and coordinates.

The input formats of NAMD include the well-known ASCII .pdb (Protein Data Bank) format as a coordinate file and .psf as a structure file. The outputs are .coor (exactly the same as .pdb), a file for final coordinate, and .vel file for final velocity. The output also includes a binary .dcd trajectory file.

There are certainly almost no differences between both input/output formats, and it should be noted that there are modules for the generation of structure file and convert well-known pdb to their own formats. The structure generations can be realized by means of *psfgen* built on a Tcl interpreter (NAMD) and *pdb2gro* (GROMACS) modules.

The basic units describing the system are almost the same in both packages in spite of some units are different (e.g. NAMD uses Angstroms for lengths and GROMACS - nanometer).

The very useful simplistic multiple timestep integration algorithm is implemented only in NAMD, which reduces the cost of computation making it possible to employ an intermediate timestep for short-range non-bonded interactions, while performing only bonded interaction every timestep. The total force acting on each atom is broken into two pieces, a quickly varying local component and a slower long-range component. The local force component consists of all bonded and van der Waals interactions as well as that portion of electrostatic interactions for pairs that are separated by less than the local interaction distance determined by the splitting function. The long-range component consists only of electrostatic interactions outside of the local interaction distance. Since the long-range forces are slowly varying, they are not evaluated every timestep. This algorithm indeed reduces the cost of computation.

The long-range dispersion corrections for energy and pressure are only implemented in GROMACS as a module *DispCorr* (values: no, EnerPres or Ener), which is due to the usage of a cut-off of Lennard Jones interactions. Moreover, it has been stated that the long-range correction leads to the significant differences in the bilayer dimensions and in the some structural parameters such as area per lipid and tails orientational order (unpublished results).

It should be also mentioned that the both packages allow to integrate and test new algorithms and modules very easily, being designed in an object-oriented style. GROMACS started out as a rewrite of the *Fortran77*-based GROMOS in the C programming language. A lot of algorithmic optimizations have been introduced in the code. For instance, the innermost loops are generated automatically at compile time, with optimizations adopted

to target architecture. For x86 processors assembly loops using SSE and 3DNow! instructions are provided, resulting in exceptional performance on PC workstations. For Pentium IV processors there are even SSE2 double precision assembly loops.

NAMD has been designed in an object-oriented with C++ language, using CHARM++ communication library. Both popular packages are distributed free of charge; they include a source code and also are available in both Linux and Windows platforms.

As far the parallelization, GROMACS uses the traditional way of a parallel programming MPI-message passing, which divides the particles over the processors and made possible by means of two pathways: spatial decomposition, which means that the program splits the whole box into slabs and each slab goes to one processor, and particle decomposition - each processor allocates a number of particles. The GROMACS uses mixed particle/domain decomposition. The dynamic components of NAMD are implemented in the Charmm++ parallel language. Charm++ implements an object-based message-driven execution model. In Charm++ applications, users decompose the problem into objects, and since they decide the granularity of the objects, it is easier for them to generate parallelism. NAMD is parallelized using hybrid force/spatial decomposition, where cubes, called *patches*, whose dimensions are slightly larger than the cutoff radius used, and for each pair of neighboring cubes, we assign a non-bonded force computation object, which can be independently mapped to any processor. The load balancer initially assigns patches and during the simulation controls the loads and performs necessary corrections. These combinations of force and decomposition allow large simulation to be efficiently decomposed into hundreds of processors. Of course, these two pathways have their own advantages and disadvantages.

Finally, the GROMACS 3.2.1 and NAMD 2.5 are compiled with gcc 3.2.2 with posix threads on RedHat Linux 9 and the MPI library mpich-gm-1.2.5-12 was used.

The benchmark results presented in Fig. 1, 2, 3 and 4 show that there is a significant difference between parallel behavior of NAMD and GROMACS. In Fig. 1 GROMACS the parallel benchmark of fully hydrated 128 DPPC/water system has been demonstrated. It is obvious, that with the increase of processors, the speed linear increases up to 16 processors (1 node = 2 processors) and the further increase of processors brings to the saturation up to 24 processors. When the number of processors is more than 24, the speed goes down, and in the case of 48 processors, it shows almost the same result as in the case of 4 processors. The problem is that with the increase of processors, each patch takes a small amount of atoms and spends more time on communications between processors. In case of the fully hydrated 128 DPPC /water system with the total number of 17506 atoms, the optimal number of processors is 16-24 and it makes no sense, and even it is not recommended, to increase the number of processors up to 24. The optimal result is 4ns per day, which is surely very good one.

In Fig. 2 NAMD the parallel benchmark of fully hydrated 128 DPPC/water system has been displayed. With the increasing of processors the speed increases linear and no saturation is observed. The number of processors, which corresponds to the maximum value, is 48 (about 1,7ns per day). While comparing with GROMACS results, the value

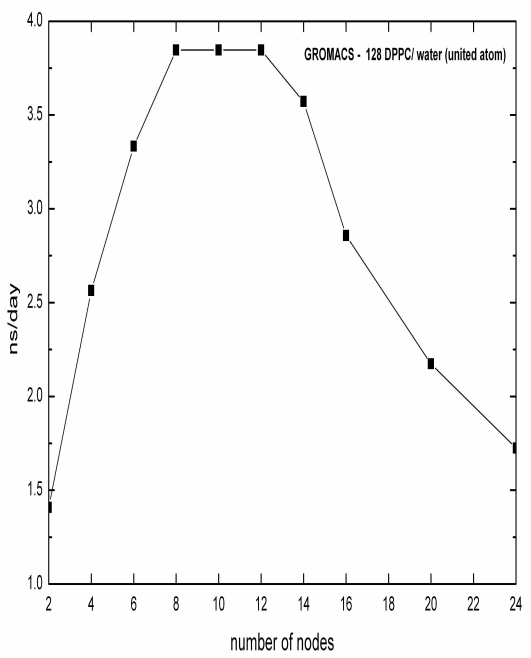

Figure 1: GROMACS parallel benchmarks using fully hydrated 128 DPPC/water system (united model) with total number of 17506 atoms (typical configuration: PME for electrostatics and 1.2 nm for Lennard-Jones interactions, NPT ensemble).

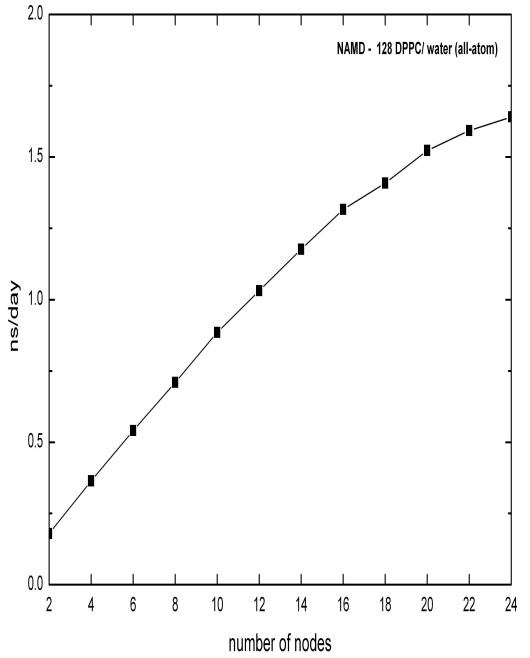

Figure 2: NAMD parallel benchmarks using fully hydrated 128 DPPC/water system (all-atom) with total number of 27746 atoms (typical configuration: PME for electrostatics and 1.2 nm for Lennard-Jones interactions, NPT ensemble).

of GROMACS benchmark, in the case of 16-20 processors, is twice of NAMD value for 48 processors. Of course, the first problem is the united-atom character of GROMACS force field, which reduces number of atoms per lipid from 130 to 50. Actually, GROMACS analyses 17506 atoms, while the same in NAMD is 27546. In this regards we decreased the number of lipid in order to get almost the same number of atoms, and in the case of fully hydrated 72 DPPC/water, the number of atoms is almost the same. In Fig. 3 NAMD parallel benchmark using above mentioned system with a total number of 16539 atoms is shown. The estimated results demonstrate that GROMACS values are more efficient for an equal number of atoms. However, while decreasing the number of lipids the NAMD demonstrates a tendency to speed up the calculation and get the maximum value of about 2,25 ns per day. The equalizing of the number of atoms indeed does not lead to significant changes. The reason is the non-equal number of pairs of atoms and therefore, the next step is the estimation of the benchmark of 128DPPC/water with united atom character. In Fig. 4 NAMD the parallel benchmark of 128DPPC/water system is shown. In the case of 48 processors the value increases up to 4ns per day. Thus, by applying the united atom model, the calculation speeds up by about a factor of two. In addition, the implementation of multiple timestep integration does not give any major changes in our benchmark results, although some small deviation has appeared (data not shown).

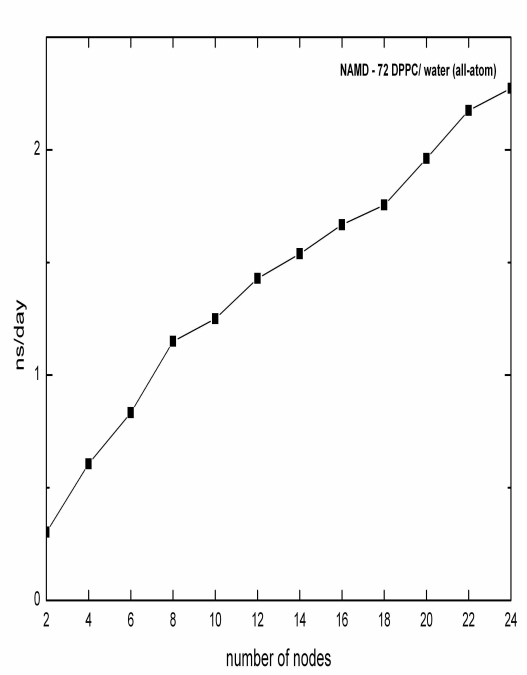

Figure 3: NAMD parallel benchmarks using fully hydrated 72 DPPC/water system (all-atom) with total number of 16539 atoms (typical configuration: PME for electrostatics and 1.2 nm for Lennard-Jones interactions, NPT ensemble).

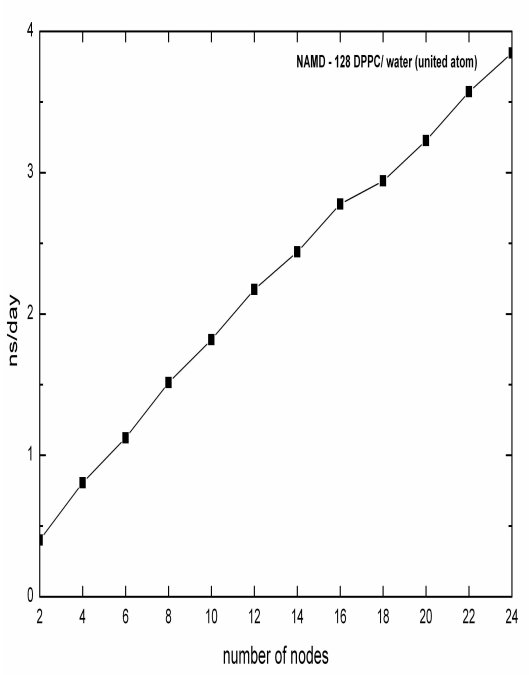

Figure 4: NAMD parallel benchmarks using fully hydrated 128 DPPC/water system (united atom) with total number of 17506 atoms (typical configuration: PME for electrostatics and 1.2 nm for Lennard-Jones interactions, NPT ensemble).

Thus, it is very difficult to give preference to a particular package for the general case, but it is possible to evaluate the packages for specific cases. It can be stated that the GROMACS still remains a packages of choice for long-time simulations of large systems, which is significantly faster than NAMD for large systems due to the united atom character. On the other hand, the NAMD is more suitable for simulations of relatively small systems, using the all-atom model. In case of large clusters the NAMD is very efficient, as it scales linearly even for small systems.

Acknowledgments

This study is supported by the INTAS under YS Fellowship Ref. Number 03-55-699. The authors would like to express their gratitude to Dr. Vladimir Sahakyan for helpful discussion and providing us with the access to the cluster.

References

[1] E. Lindahl, B. Hess and D. van der Spoel, GROMACS 3.0: A package for molecular simulation and trajectory analysis, J. Mol. Model, 7 (2001), 306-317.

[2] D. van der Spoel, A. R. van Buuren, E. Apol, P. J. Meulenhoff, D. P. Tieleman, A. L. T. M. Sijbers, B. Hess, K. A. Feenstra, E. Lindahl, R. van Drunen and H. J. C. Berendsen, (2002). Gromacs User Manual version 3.1.1, Nijenborgh 4, 9747 AG Groningen, The Netherlands. Internet: www.gromacs.org.

[3] L. Kale, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan and K. Schulten, NAMD2: Greater scalability for parallel molecular dynamics, J. Comput. Phys., 151 (1999), 283-312.

[4] 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. Comput. Chem., 4 (1983), 187-217.

[5] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. R. Ross, T. E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel and P. Kollman, AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules, Comput. Phys. Commun., 91 (1995), 1-41.

[6] P. Ren and J. W. Ponder, Polarizable atomic multipole water model for molecular mechanics simulation, J. Phys. Chem. B, 107 (2003), 5933-5947.

[7] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis and J. A. Montgomery, General atomic and molecular electronic structure system, J. Comput. Chem., 14 (1993), 1347-1363.

[8] J. J. P. Stewart, MOSOL, MOPAC for solid-state physics, Quant. Chem. Prog. Exch., 5 (1985), 62-63.

[9] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr. T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V.G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez and J. A. Pople, Gaussian, Inc., Wallingford CT., 2004.

[10] W. Humphrey, A. Dalke and K. Schulten, VMD: Visual molecular dynamics, J. Mol. Graphics, 14 (1996), 33-38.

[11] K. Kadau, T. C. Germann and P. S. Lomdahl, Large-scale molecular-dynamics simulation of 19 billion particles, Int. J. Mod. Phys. C, 15 (2004), 193-201.

[12] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, Optimization by simulated annealing, Science, 220 (1983), 671-680.

[13] R. W. Hockney and S. P. Goel, Quite high resolution computer models of a plasma, J. Comput. Phys., 14 (1974), 148-158.