Open Problems in High-Performance Molecular-Dynamics Simulations

D. Grancharov, E.Lilkova, N. Ilieva, P. Petkov, L. Litov

Key Words: *Molecular-dynamics simulations; symplectic algorithms; scalability; large atomic systems; work-load distribution; GROMACS; NAMD; SCALASCA.*

Abstract: *Molecular-dynamics simulations provide reliable information about the microscopic behavior of the investigated systems by numerical solution of the equations of motion. The constantly growing computer performance (in particular, the number of computing cores) makes wider the range of research topics – and systems – accessible by these means. However, simulation continuance increases drastically with the size of the systems and the scalability of the most popular simulation packages worsens. We aim at the development of a multiple step-size symplectic integrator adapted to the large biomolecular systems particular features, such as multiple frequency modes, interaction hierarchy and strong inhomogeneity. To this end. we analyze the scalability and the work-load increase and distribution among the computing cores in the packages GROMACS and NAMD on the example of three test systems with increasing size (5xl0^5 ~10^6 and ~2.2x10^6 atoms respectively), by means of the GROMAC S in-built tool g_tune_pme and of the dedicated package SCALASCA.*

1. Introduction

Molecular dynamics is a widely used method for investigation of the time evolution of atomic and molecular systems, with applications in various scientific fields such as the design of new materials, nanotechnologies. drug design, computational chemistry etc. The basic concept of this method implies the parameterization of the interaction potential energy function and calculation of the time evolution of the system (the atomic trajectories) by solving the Newtonian equations of motion numerically. MD simulations allow the microscopic behavior of the investigated system to be followed. It is an extensive calculation which demands high performance computing facilities, as well as proper software packages gaining full advance of the given computational resources. Systems of interest are constantly growing on size and complexity. which necessitates reconsideration of present algorithms not only because of the 'exploding'" calculation volumes, but also due to unsatisfactory scalability with increasing the processors number. Optimization of calculations is essential for reducing the continuance of simulation.

An important feature of all MD simulations is the Hamiltonian nature of the investigated dynamics, which includes preserving the phase volume and reversibility with respect to time. It is favorable if these features are adequately incorporated in the algorithms in use although there are examples, in which this is not the case (the quaternion algorithm of Gear). Sytnplectic integrators (algorithms) like the algorithms of Miller [1], Nose [2] and others fulfill these demands.

A substantial difficulty when numerically calculating the trajectories of complex molecular systems is associated with the fact that the latter consist of both fast and slow changing degrees of freedom. On one hand, tins determines the typical timescale of the processes in them, but on the other hand restricts the time step used in the calculation.

Development of integration algorithms with variable time step [3] is one promising way to attack tins problem This task is far not trivial, arguments from different nature being employed in its elaboration, e.g:

• The effective linearization in case of semi-implicit sympleclic integrators (which gain speed over implicit ones) leads in general to loss of symplecticity and hence to unsatisfactory stability of results from long simulations.

• A constructive idea is to divide the Hamiltonian of the system into high-frequency harmonic part, which is to be analytically investigated and low-frequency part to be numerically attacked (e.g [4]) It should be stressed that the physical motivations of such division, as well as its range of application have not yet been studied enough.

• Symmetries (so, the corresponding constraints) play a substantial role in the behavior and evolution of any system and have to be taken to account in its investigation. Recently a family of symplectic linearly-implicit stable integrators with application in fast MD simulations for systems with holononuc constraints was proposed [5]. The important idea there is the replacement of the holononuc constraints (used for restricting the motion of H-atoms bound to heavy atoms) with hard springs.

It is known that integrators produce artifacts in the behavior of the investigated system like fake resonances. Symplectic algorithms with adjustable time step not only permit effective enlargement of the time step, bur also avoid the occurrence of such resonances and provide better sampling of phase space, which is another important advantage they have, at higher computational cost though [6]. However, the sytnpleticity of a given fixed step size method is not automatically preserved when a variable step size is applied, also its accuracy might suffer. On the other hand, the ideas of the composition methods and splitting methods [7], with the specifics of the large systems subject to MD simulations taken to account, offer some possibilities for overcoming these difficulties.

2. Scalability of the MD-packages GROMACS and NAMD in Simulations of Large Systems Above 2048 Computing Cores

ln all cited papers the considerations are exemplified on small (from the MD simulations point of view) systems – 100 to 1000 atoms. Increasing the size of the investigated system leads to strongly nonlinear increase of the computation volume Gaining insight into the main reasons for the performance deterioration of the MD-simulation packages GROMACS and NAMD when applied to large systems (with more than 1 000 000 atoms) and on more than 4000 computing nodes is a key step in the development of the new/adapted integration algorithm with adjustable time-step for MD simulations of very large systems on computers with Petascale performance and its further implementation.

In what follows we are going to study various aspects of the MD-simulation performance for large systems (scalability, distribution of the computational load and its dependence on the functional assignment to the individual processors) in the case of three substantially different in size and structure systems – epidermic growth factor, satellite of the tobacco mosaic virus and E.Coh nbosome dissolved in water (*figure 1*) and two different packages – GROMACS and NAMD chosen for study and optimization for Petaflops architectures, within the PRACE initiative. The investigation was performed on the IBM BlueGene/P supercomputer of Bulgarian National Center for Supercomputing Applications [8] thus scalability of the packages was investigated up to 8192 computing cores. Details about the three test systems are given in the Appendix.

The study was performed with GROMACS version 4.5.4 and NAMD CVS from 19.02.2011. Both were compiled with the XL compilers of IBM for the architecture of the computing nodes of BlueGene/P. NAMD CVS compilation included the possibility of using compressed input data. This gives the opportunity the memory of the prime MPI process to be used only as an input – output operation buffer.

The performance and speed-up data for all three test systems on different number of computing cores is given in *table 1*. The performance is taken directly from the output information of running the program packages for certain number of integration steps and represents the simulation time to be obtained for 24 hours work of this package with integration step of 2 fs (*figure 2*). As a reference value in determining the speed-up of the simulations was taken the performance at 512 computing cores.

As seen on *figure 3*, the scaling of NAMD improves with the increase of the system size, though with a slower growth beyond 4096 computing cores. For GROMACS, this number of cores appears to be critical, as it scales well up to that point, even if with a lower overall performance than NAMD, but further increase in the number of cores makes no sense.

3. Workload Distribution on the Computing Cores in the MD Simulations

With the increase of the system size and of the number of computing cores, one expects an increase of communication between computing cores and some loss of scalability of the investigated packages. The amount of communication is determined by the particular mechanism for task distribution among computing cores and by the scheme used for calculating long range electrostatics. To determine the type of communication that leads to greatest loss of performance, studies of the scalability, effect of workload distribution schemes and intensity of the commu-

Table 1. Test-run data for GROMACS and NAMD

System size (number of atoms)	Number of cores	NAMD CVS 2011-02-19		GROMACS 4.5.4	
		Performance [ns/day]	Speed up	Performance [ns/day]	Speed up
465 399	8 192	12.21	6.68	2.57	2.74
	4 096	10.24	5.60	5.32	5.68
	2 048	6.08	3.33	3.08	3.29
	1 024	3.14	1.72	1.84	1.96
	512	1.83	1.00	0.94	1.00
1 007 930	8 192	11.37	11.08		
	4 096	7.03	6.85		
	2 048	3.57	3.47		
	1 024	1.97	1.92		
	512	1.03	1.00		
2 233 537	8 192	5.86	12.53		
	4 096	3.44	7.36		
	2 048	1.80	3.84		
	1 024	0.92	1.97		
	512	0.47	1.00		

nications of the GROMACS package were performed with the SCALASCA package.

When running a parallel calculation of a molecular system, an algorithm is needed to divide the system in parts and distribute them among the computing cores. In GROMACS there are two algorithms for system division – particle decomposition (decomposition to individual particles) and domain decomposition (decomposition into appropriately defined spatial areas/domains) [9,10].

The particle decomposition is the simplest way to divide the system. With the beginning of the simulation, certain particles of the system are assigned to each processor. Next, the calculation of the forces that act on the particles is also assigned, so that the workload on the processors is uniformly distributed. This algorithm requires for each processor to have access to the coordinates of at least half the atoms of the system. This means that N computing cores must communicate N x N/2 coordinates. Due to this quadratic dependence on the particle number the particle decomposition shows poor scalability for very large systems and comes therefore in use only in specific cases, when long-range covalent interactions are present in the system.

The domain decomposition takes advantage of the fact that most of the interactions in the system are local. In the general case of triclinic simulation box, space is divided into 1-, 2- or 3-dimensional grid of subareas, called domains. To each processor the algorithm assigns to each processor a certain spatial domain of the system. The processor solves the equations of motion for those particles that are present in its domain at that moment. The neighbors search in GROMACS is based on the idea of charge groups and so is the domain decomposition. Charge groups are assigned to the domain in which their geometrical center lies.

The electrostatic interactions are long-range interactions, so non-local. This necessitates the application of special algorithms for their calculation. Usually in GROMACS the PME algorithm [11] is used that incorporates interaction of every particle with all others and therefore needs global communication. To reduce the effect of this problem, a part of the computing cores are used only for calculating the electrostatic interaction in the PME algorithm (called pme computing cores) and the rest of them (called pp computing cores) to calculate all other interactions [9].

SCALASCA [12] is a software package for profiling parallel codes performance. Its purpose is to guide the optimization process of parallel programs by measuring and analyzing their behavior during the run. Thus, the bottlenecks of the code might be identified, which cause substantial loss of performance, especially those related to uneven distribution of work load, communication and synchronization. The SCALASCA package works in three stages: in the first, the code of interest is instrumented (in this case the MD package GROMACS) by adding functions in it that measure the time spent in every single procedure. In the second, the already instrumented code is started and in the third, the gathered data is analyzed with the aid of the visualization package Cube 3 that allows various characteristics of the calculation to be presented in a clear fashion. Such information is needed for deeper understanding of the algorithms in use and for estimation of their efficiency, speed and parallelization behavior.

The performance of the GROMACS 4.5.3 package with regard to the ordering of the pp/pme computing cores was studied with Scalasca 1.3.1. The test system contains 103079 atoms, the simulated time evolution amounts to 20 ps (10000 MD steps, 2 fs time step). Periodic boundary conditions were applied and temperature was kept constant with the Berendsen thermostat. Holonomic constraints for freezing the vibrational degrees of freedom were not introduced (algorithms LINCS and P-LINCS were not used). For calculating the electrostatic interaction the PME algorithm was used with direct summation cutoff of 1.4 nm. The neighbor lists were updated every 10 time steps.

By default GROMACS divides the pp and pme computing cores in proportion 3:1 with ordering mode –"interleave". In this mode, with proportion of pp:pme cores n:1, 1 pme core is placed after every n pp cores. There are two more modes in GROMACS for ordering the pp and pme cores – pp_pme and Cartesian. In the pp_pme mode the pp cores are placed in the beginning and the pme cores at the end. The Cartesian mode is a mixture of the previous two and is specially designed for architectures that support a real 3-dimensional thoroidal communication system like the IBM BlueGene/P. Profiling with SCALASCA shows that dividing the computing cores in pp and pme only leads to strongly uneven distribution of the communication (*figure 4*).

The results on the performance and speed-up of GROMACS in all three regimes are presented in *table 2* and *figure 5*.

The work-load distribution (the time used for different

Table 2. GROMACS scalability in different regimes

ddorder regime / Number of computing cores	interleave (the default) [ns/day]	pp_pme [ns/day]	Cartesian [ns/day]
512	6.672	6.592	6.600
1024	12.122	11.905	11.973
2048	20.856	20.627	20.426
4096	27.994	31.306	31.544

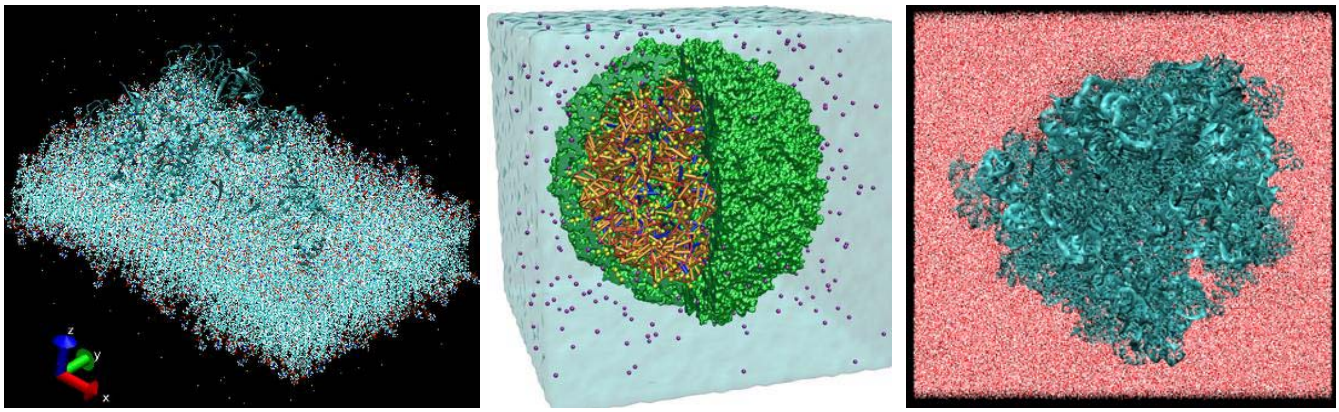

Figure 1. The test systems: (A) ~5 x 10^5 atoms; (B) ~10^6 atoms; (C) ~2.2 x 10^6atoms

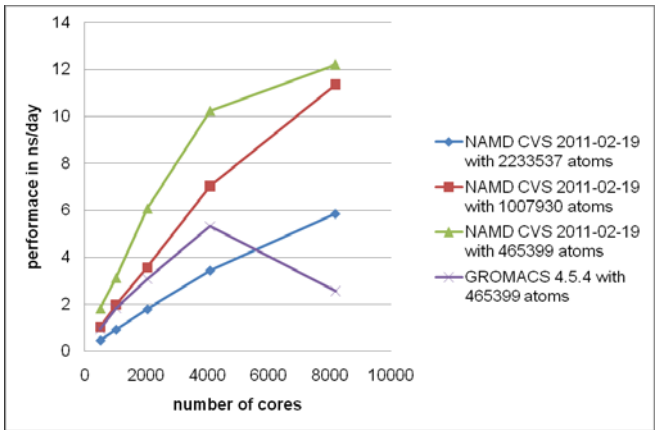

Figure 2. Performance of GROMACS 4.5.4 and NAMD CVS 2011-02-19 as a function of the cores number for the three test systems, with 465399, 1007930 and 2233537 atoms resp

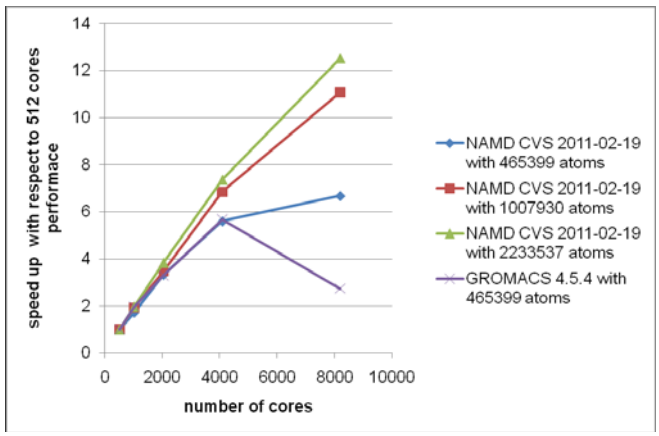

Figure 3. Sped-up of GROMACS 4.5.4 and NAMD CVS 2011-02-19 as a function of the cores number and system size

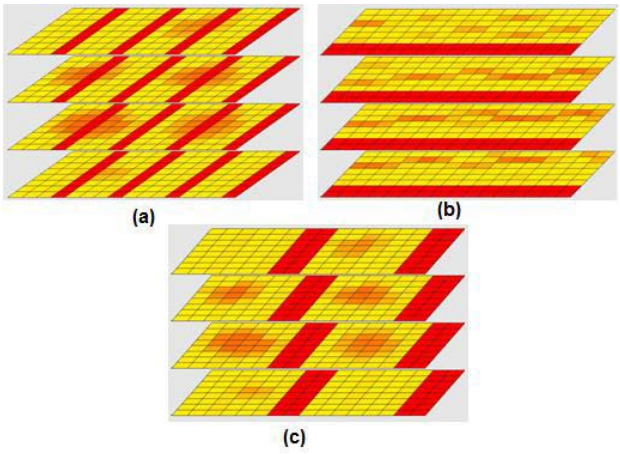

Figure 4. .Distribution of the communications: (à) interleave; (b) pp_pme; (c) Cartesian (red – higher intensity, yellow – lower intensity)

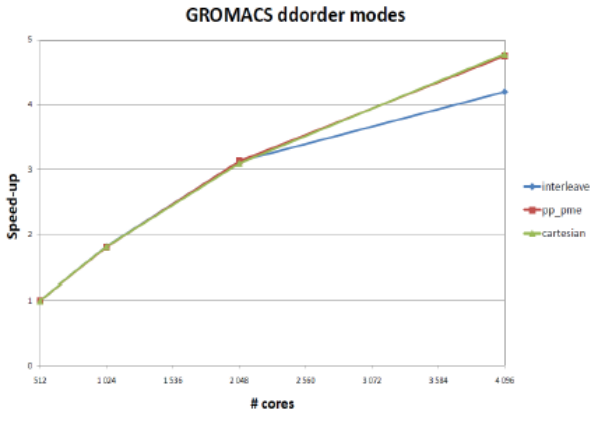

Figure 5. Speed-up for the different regimes

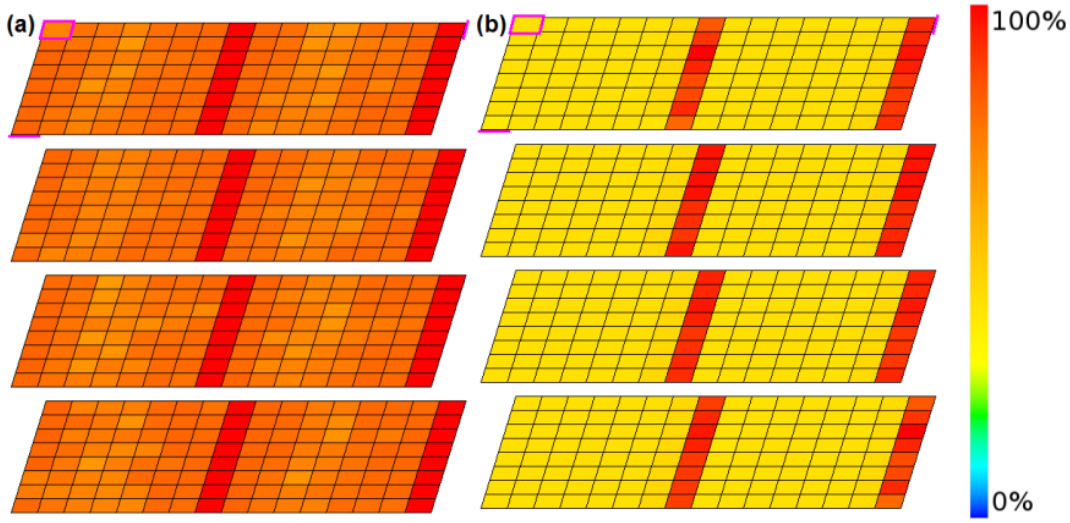

Figure 6. The 512-computing cores run, interleave-regime. Distribution over the cores of: (a) the total simulation time; (b) the communications

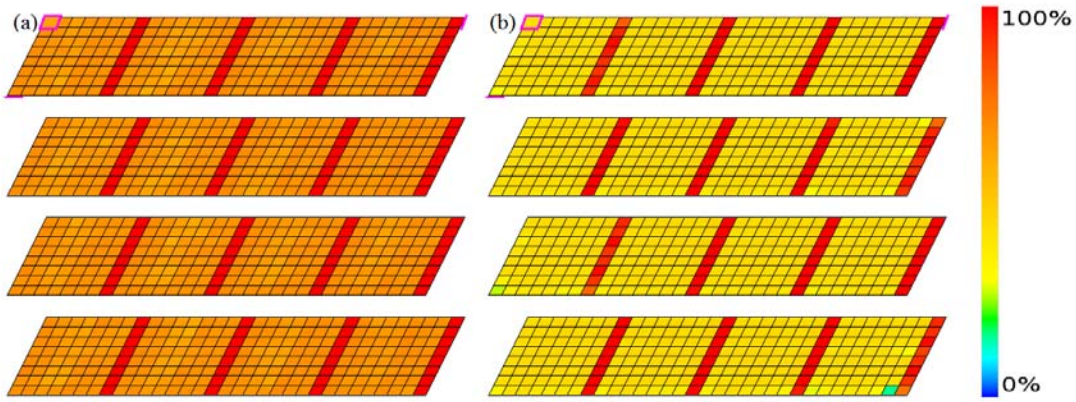

Figure 7. The 1024-computing cores run, interleave-regime. Distribution over the cores of: (a) the total simulation time; (b) the communications

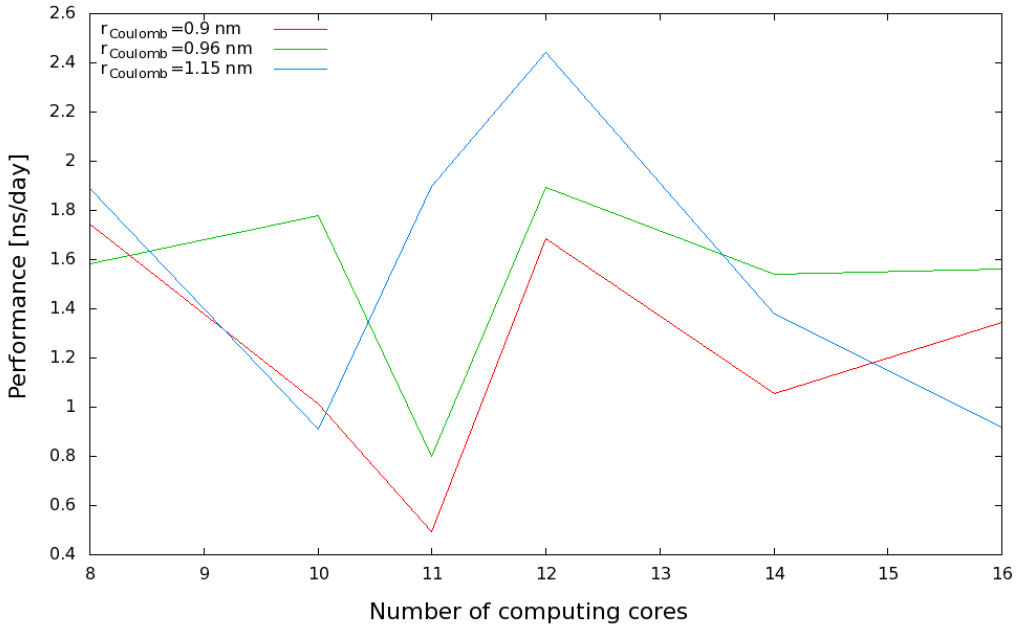

Figure 8. GROMACS 4.5 performance (in ns/day) for pme:pp from 1:1 to 1:3 (16, resp. 8 pme cores out of 32) at different cut-off radii

operations) over the individual computing cores in the MD-simulation package GROMACS was studied on the example of test system (A), with GROMACS 4.5.3 instrumented with the profiling tool SCALASCA 1.3.1. One eighth of all cores were used for calculation of long range electrostatics (pme cores). The investigation was performed on 512 and on 1024 computing cores only, because of the huge size of the output files (already 51 GB on 1024 cores).

The investigation brought up the following data: on 512 computing cores, the total amount of time consumed for the simulation, is $3x10^6$ seconds, $2{,}3x10^6$ of which is pure execution time and the rest is spent on system procedures, needed for instrumentation. The average time spent by each core is about 4668 seconds. The main part of this time is spent on the – "do_md" procedure that performs the actual MD – roughly 70% of the whole time. The remaining 30% are spent mainly for calculating long range electrostatics and for domain decomposition of the system on the computing cores.

On *figure 6a* the distribution (in percentage) of the total time over the cores is shown. The average time differs for pme and pp cores: 5888 and 4494 seconds per core respectively. On *figure 6b* the distribution of the communications over the cores is presented. The communications – $3{,}18x10^6$ in total – are now distributed between the procedure – "do_md" – about 58%, the calculation of the long-range electrostatics – some 21% and (which is new here) the initialization of the parallel environment – about the same amount, 20%. However, the average amount of communications per core for the pme cores is 2.5 times bigger than for the pp cores – 10453 against 4174. The intense communication of the pme cores explains their higher average time consumption compared to the pp ones.

Similar is the picture observed for 1024 computing cores. Again the – "do_md" procedure consumes the largest amount of time – about 66,6%, the rest being essentially distributed between calculation of long-range electro-statics and domain decomposition of the system. The average time spent by each core is 4912 seconds, with 6829 seconds per core for the pme cores, against only 4637 seconds per core for the pp cores (*figure 7a*). On *figure 7b* the distribution of the total amount of communications over the cores is shown: a total of $7{,}1x10^6$ communications, 60% of which spent on the – "do_md" procedure, 22% on calculating long range electrostatics and approximately 18% on the initialization of parallel environment, load the pme cores in average with 13600 counts per core and the pp cores – with only 6100 counts per core.

The increase of the number of computing cores, devoted only to calculation of long range electrostatics would accelerate this calculation, but on the cost of having smaller amount of computing cores, involved in the calculation of procedures, that consume up to 65% of the total simulation time. Also, with the increase of pme cores the total amount of communications increases, which may lead to a substantial slowdown of the calculation, especially if additional synchronizing operations are required. Setting the pp to pme cores ratio is a delicate issue and for its proper treatment additional information is needed.

In this line of considerations, a further investigation of the performance of the GROMACS executable with respect to the number of pme only cores and the size of cut-off radius for the direct summation of the electrostatic forces was performed. The investigation was made with a built-in GROMACS 4.5.3 tool "g_tune_pme" on 32 computing cores of a local Linux cluster with a test system of cyclooxygenase enzyme, dissolved in water, containing approximately 200000 atoms. Molecular dynamics simulations of 2000 steps each with different pp to pme ratios were performed (*figure 8*).

The above investigations allowed to clearly identify the main reasons for the increase of communication between the computing cores and thus for damping down the scalability of the code. The multiple-timestep symplectic integration algorithm we work on aims at resolving this problem.

Acknowledgements

This work was financially supported by the PRACE 2 IP, WP3 (Dissemination), funded in part by the EUs 7th Framework Program (FP7 2007-2013) under grant agreement no. RI-211528 and FP7-261557. The work is achieved using the PRACE Research Infrastructure resources IBM Blue Gene/P computer located in Sofia, Bulgaria.

Appendix

Test system (A). The epidermic growth factor receptor (EGFR; ErbB-1; HER1 by humans) is a membrane receptor for the epidermic growth factor family members (EGF family) [13]. Mutations concerning EGFR expression could lead to various malignant diseases, including lung and colon cancer and multiform glioblastoma [14, 15]. The test system was provided by Dr. Iliyan Todorov of the Computational Science & Engineering Department, CCLRC Daresbury Laboratory (Daresbury, UK). It represents an EGFR dimer on a lipid bilayer, the simulation volume being filled with water molecules, and contains a total of 465399 atoms ($\sim 5 \times 10^5$).

Test system (B) contains a satellite of the tobacco mosaic virus. This is a small virus spread on the icosahedric plants that worsens the symptoms of the tobacco mosaic virus [16]. The structure was taken from PDB (PDB ID 1A34). For the preparation of this test system a special program was written that assembles crystallographic structures and checks for overlapping atoms. The system contains the virus, dissolved in water, acounting to 1007930 (so, roughly 10^6) atoms altogether.

Test system (C). The ribosome (test system three) is a complicated molecular machine that translates the genetic code from its temporary carrier – the informational RNA to proteins, which are the basic material for constructing live cells and catalyze metabollic pathways that provide energy for these cells. The ribosome is one of the most promising targets in the process of designing antibacterial drugs. It

contains two subunits - a small and a big one. The crystallographic structure of both subunits of the E.Coli ribosome is available in the PDB (PDBID: 3FIK and 3FIH [17]). These structures do not include counterions, so an specific algorithm for their addition was developed, differing from the one described in [18]. The neutralized system was placed in a simulation box with dimensions as follows: 309x298x257 $Å^3$, that was filled with water molecules and sodium and chlorine ions with phisyological concentration. The whole system accounts to 2 233 537 atoms.

Remark

Due to a peculiarity in the way that data is loaded into the RAM memory of the computing cores of IBM BlueGene/P, the GROMACS package is able to handle systems of at most 700000 atoms. Therefore the scaling of GROMACS was investigated only with test system (A), while NAMD was studied with all three test systems.

References

1. Miller III, T. F., M. Eleftheriou, P. Pattnaik, A. Ndirango, D. Newns, and G. J. Martyna, J. Chem. – *Phys*, 116**,** 8649, 2002.

2. Nosé, S., – *J. Phys. Soc. Jpn.*, 70**,** 2001, 75.

3. Skeel, R., J. J. Biesiadecki. – *Ann. Num. Math.*, 1**,** 1994, 1-9.

4. Janezic, D. and M. Praprotnik, J. Chem. – *Inf. Comput. Sci.*, 43, 2003, 1922-1927.

5. Tao, M., H. Owhadi, J. E. Marsden. Symplectic, Linearly-implicit and Stable Integrators with Applications to Fast Symplectic Simulatons of Constrained Dynamics. e-Print arXiv: 1103.4645, 2011.

6. He, Wei and Sanjay Govindjee. Application of a SEM Preserving Integrator to Molecular Dynamics. Rep. No. UCB/SEMM-2009/01, Jan 2009, Univ. of California, Berkley, 27.

7. Hairer, E., C. Lubich, and G. Wanner. Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. (Springer, Heidelberg, Germany, second ed., 2004).

8. http://scc.acad.bg/

9. Hess, B. and C. Kutzner and D. van der Spoel and E. Lindahl. GROMACS 4: Algorithms for Highly Efficient, Load-balanced, and Scalable Molecular Simulation. – *J. Chem. Theory Comput*., 4**,** 2008, 435-447.

10. http://www.gromacs.org/Documentation/Manual

11. Darden, T., D. York, L. Pedersen. Particle Mesh Ewald: An Nlog(N) Method for Ewald Sums in Large Systems. – *J. Chem. Phys.*, 98**,** 1993, 10089-10092.

12. Geimer, M., F. Wolf, B. J. N. Wylie, E. Ábrahám, D. Becker, B. Mohr. The Scalasca Performance Toolset Architecture, Concurrency and Computation: Practice and Experience. – *Concurrency*, 22, No. 6, 2010, 702-719.

13. Herbst, R. S. – *Int. J. Radiat. Oncol. Biol. Phys.*, 59 (2 Suppl), 2004, 21-6. doi:10.1016/j.ijrobp.2003.11.041. PMID 15142631.

14. Zhang, H., A. Berezov, Q. Wang, G. Zhang, J. Drebin, R. Murali, M. I. Greene, – *J. Clin. Invest.*, 117/8, 2007, 2051-2058. doi:10.1172/JCI32278.PMC 1934579. PMID 17671639.

15. Walker, F., L. Abramowitz, D. Benabderrahmane, X. Duval, V. R. Descatoire, D. Hénin, T. R. S. Lehy, T. Aparici. – *Human Pathology*, 40/11, 2009, 1517-1527. doi:10.1016/j.humpath.2009.05.010. PMID 19716155.

16. http://www.ks.uiuc.edu/Research/STMV/

17. Villa, E. et al. – *Proc. Natl. Acad. Sci. USA*, 106**,** 2009, 1063-1068.

18. Sanbonmatsu, K. Y. and C. -S. Tung1. – *Journal of Physics: Conference Series*, 46**,** 2006, 334-342.

Manuscript received on 06.11.2012

D. Grancharov
Contacts:
Faculty of Physics
University of Sofia "St. Kl. Ohridski"
Sofia, Bulgaria

E. Lilkova
Contacts:
Faculty of Physics
University of Sofia "St. Kl. Ohridski"
Sofia, Bulgaria

N. Ilieva
Contacts:
Institute for Nuclear Research and Nuclear Energy
BAS
Sofia, Bulgaria

P. Petkov
Contacts:
Faculty of Physics
University of Sofia "St. Kl. Ohridski"
Sofia, Bulgaria

L. Litov
Contacts:
Faculty of Physics
University of Sofia "St. Kl. Ohridski"
Sofia, Bulgaria