Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

Analysis of symplectic integration algorithms with variable step size for petascale biomolecular simulations

D. Grancharov, E. Lilkova, N. Ilieva[1], P. Petkov, S. Markov and L. Litov

National Centre for Supercomputing Applications, Acad. G. Bonchev Str, Bl. 25-A, 1113 Sofia, Bulgaria

Abstract

Based on the analysis of the performance, scalability, work-load increase and distribution of the MD simulation packages GROMACS and NAMD for very large systems and core numbers, we evaluate the possibilities for overcoming the deterioration of the scalability and performance of the existing MD packages by implementation of symplectic integration algorithms with multiple step sizes.

1. Integration algorithms in the MD simulations

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 numerically solving the Newtonian equations of motion

$$\vec{F} = m\vec{a} = m\frac{d^2\vec{x}}{dt^2} \quad \rightarrow \quad \begin{cases} \vec{v} = d\vec{x}/dt \\ \vec{F} = md\vec{v}/dt \end{cases}$$

$$\vec{F} = -\vec{\nabla}V(x)$$

where *x*, *v* and *m* are particles coordinates, velocities and mass, *V*(*x*) is the potential and *F* – the acting force. 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 taking full advantage 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 processor numbers. Optimization of calculations is essential for reducing the continuance of the simulation.

The Newtonian equations of motions are solved numerically using explicit schemes such as Verlet [1], where the following recursive relations take place

$$p_{n+1/2} = p_n + \frac{\Delta t}{2}F_n \qquad F_{n+1} = F(q_{n+1})$$

$$q_{n+1} = q_n + \Delta t m^{-1} p_{n+1/2} \qquad p_{n+1} = p_{n+1/2} + \frac{\Delta t}{2}F_{n+1}$$

so that

$$(q_n, p_n) \rightarrow (q_n, p_{n+1/2}) \rightarrow (q_{n+1}, p_{n+1/2}) \rightarrow (q_{n+1}, p_{n+1})$$

[1] Corresponding author. *E-mail address*: nilieval@mail.cern.ch

where (q_n ,p_n) are the coordinates and the momenta at the n-th step, F – the corresponding force, and Δt – the integration timestep. Such schemes are simple to implement and also fast, but are known to introduce resonances.

Implicit numerical integrators with high stability have been introduced to molecular dynamics simulations since these integrators usually permit a larger step size than Verlet. However, implicit integrators are computationally more demanding since one has to solve a complicated system of equations which are usually nonlinear in every step. Moreover, most implicit integrators such as the midpoint scheme can only delay energy resonance. When the step size increases and hits specific values, resonance occurs and leads to incorrect phase diagrams.

In all MD simulations the investigated dynamics has a Hamiltonian nature. Hamiltonian systems are described by the Hamilton equations of motion

$$\dot{p} = -\partial H(q,p)/\partial q \qquad \dot{q} = \partial H(q,p)/\partial p \tag{1}$$

where $H(q, p)$ is the Hamiltonian of the system. A theorem due to Poincaré [2] states that the Hamiltonian flow φ of such systems is a symplectomorphism

$$(q(t), p(t)) \xrightarrow{\varphi} (q(t+\Delta t), p(t+\Delta t)) \quad : \quad (\varphi_h')^T \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \varphi_h' = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

Simplecticity implies preservation of the phase-space volume and reversibility with respect to time. It is favorable if these features are adequately incorporated in the algorithms in use as this usually leads to long-time stability of the method [3, 4]. Symplectic integrators like the algorithms of Miller [5], Nose [6] and others fulfill these demands, though most of the usual integrators, e.g. primitive Euler or classical Runge–Kutta methods, are not symplectic ones. Note, that symplectic integrators are known to conserve energy and stability holds at large step sizes for problems in the linear regime [7] and this is not the case for nonlinear systems. Instability, or large energy fluctuation, occurs at timesteps that still satisfy the linear stability condition and these instabilities usually can only be avoided by reducing the timestep [8], so that this sort of nonlinear instability sets a further timestep limitation. Recall in addition, that structure preserving integrators alone cannot guarantee accurate trajectories which can only be obtained with high-order methods and small timesteps [9] but on the other hand, high-order methods may not preserve structural properties such as energy and momentum. Thus, these two actually different concepts have to be carefully weighted when choosing the integrator for every single problem. Particularly, for those systems with complicated, unstable, or chaotic trajectories, such as biomolecules, one should probably concentrate on statistical properties and approach the true solution by preserving as much of the structure as feasible [10].

The default MD integrator in GROMACS [22] is the leap-frog algorithm [11]. This algorithm uses positions r at time t and velocities v at time t–1/2Δt; positions and velocities being updated using the forces $F(t)$ determined by the positions at time t. It produces trajectories that are identical to the Verlet [1] algorithm. The algorithm is of third order in r and is time-reversible[2]. When extremely accurate integration is temperature and/or pressure coupling velocity Verlet integrator [13] may be preferable: it is also implemented in GROMACS, though not yet fully integrated with all sets of options. In velocity Verlet positions r and velocities v at time t is used to integrate the equations of motion; velocities at the previous half step are not required.

A constant timestepping algorithm can only be energy-momentum preserving or symplecticity-momentum preserving, but not both [14]. Another disadvantage of the constant timestepping follows from an important feature of the complex molecular systems – the presence therein of both fast and slow changing degrees of freedom that additionally complicates numerical calculation of their trajectories. On the one hand this determines the typical timescale of the processes in them, but on the other hand imposes severe restrictions on the integration timestep size Δt: it must not exceed the scale of the most rapid vibration mode. This generally limits Δt to be in the femtosecond (10^{-15} s) range – the typical step size recommended in practically all popular MD simulation packages. Since key conformational changes in biomolecules occur on time scales of $10^{-12} - 10^{2}$ s, considerable efforts are focused on techniques that allow for an effective increase and also modulation of the integration timestep. These are based on the development of splitting methods (where the individual components may be numerical flows or certain combinations of exact and numerical flows, as e.g. in [15]), composition methods (which not only increase the algorithm order in r but also provide a tool for step modulation), integrators based on generating functional, or variational integrators in which the action integral is being discretised (see, e.g. [10]).

Development of integration algorithms with variable or adaptable timestep [16] is one promising way to attack this problem. They not only permit effective enlargement of the timestep, but also avoid the occurrence of fake resonances and provide better sampling of phase space, which is another important advantage, at higher computational cost though [17]. This task is by far not trivial as the sympleticity of a given fixed step size method is not automatically preserved when a variable step size is applied,

[2] See [12] for the merits of this algorithm and comparison with other time integration algorithms.

also its accuracy might suffer. On the other hand, the ideas behind the composition methods and the variational integrators seem to allow the specifics of the systems subject to MD simulations being taken into account that offers a constructive way for overcoming these difficulties.

Scientific and technical goals

For large atomic systems (10^6 atoms) the most time-consuming part of the simulation is the calculation of the electrostatic interactions. These are long-range interactions, so non-local, which necessitates the application of special algorithms for their calculation. In the most popular packages for MD simulations, different realizations of the Ewald summation method [18] are used, as it ensures proper description of the long-range electrostatic interactions for spatially bounded systems with periodic boundary conditions. The bottleneck part of these calculations is the FFT part. One possibility to improve the performance of the MD simulations is to employ integration algorithms with variable step size [16] for calculation of full electrostatics. In this approach the total force acting on each atom is broken into two parts, a quickly varying short range component and a slow changing long distance component. Since the long range forces are slowly varying they are not calculated at every timestep, but at some bigger step.

This is better illustrated in the formulation of the Hamilton equations (1) in terms of Liouville operators

$$iL\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} v \\ F(x)/m \end{pmatrix} = \begin{pmatrix} v \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ F/m \end{pmatrix} = iL_v + iL_F \tag{2}$$

where

$$iL_v = \{\ ,H\}, \qquad [L_v, L_F] \neq 0$$

Thus,

$$\begin{aligned} \exp(iL\Delta t) &= \exp\left(\frac{1}{2} iL_F \Delta t\right) \exp(iL_v \Delta t) \exp\left(\frac{1}{2} iL_F \Delta t\right) \\ &= U_F\left(\frac{1}{2}\Delta t\right) U_v(\Delta t) U_F\left(\frac{1}{2}\Delta t\right) \end{aligned} \tag{3}$$

The single-step propagators obtained that way are used to calculate recursively the coordinates *x* and the velocities *v* of the atoms

$$\left.\begin{aligned} U_F(\Delta t)\begin{pmatrix} x \\ v \end{pmatrix} &= \begin{pmatrix} x \\ v + (F(x)/m)\Delta t \end{pmatrix} \\ U_v(\Delta t)\begin{pmatrix} x \\ v \end{pmatrix} &= \begin{pmatrix} x + v\Delta t \\ v \end{pmatrix} \end{aligned}\right\} \qquad \begin{aligned} v_{n+1} &= v_n + \frac{1}{2m} F_n \Delta t + \frac{1}{2m} F_{n+1} \Delta t \\ x_{n+1} &= x_n + \left(v_n + \frac{1}{2m} F_n \Delta t \right) \Delta t \end{aligned}$$

For example, if one splits the simulation volume in subspaces according to some set of rules, e.g.

$$r_1 > r_2 > \ldots \qquad F_i = 0,\ \ r > r_i; \qquad \tau_i :\ \ \tau_i / \tau_{i+1} = M_{i+1}$$

where the *i*-th component of the force field is to be applied with a timestep τ_i, from Eq. (3) follows

$$\exp(\tau_0\{\ ,H\}) \approx \exp(\frac{\tau_0}{2} K_0) \exp[\tau_0 (D + K_1 + K_2 + \ldots)] \exp(\frac{\tau_0}{2} K_0)$$

where

$$D = \{\ ,T\} \qquad K_i = \{V_i, T\}$$

Thus, if an atom is in the periphery of the system, the expansion will terminate already at the second term [19]

$$r_1 > r > r_2 :\ \ \exp\left(\frac{\tau_0}{2} K_0 + \frac{\tau_0}{4} K_1\right) \exp\left(\frac{\tau_0}{2} D\right) \exp\left(\frac{\tau_0}{2} K_1\right) \exp\left(\frac{\tau_0}{2} D\right) \exp\left(\frac{\tau_0}{4} K_1 + \frac{\tau_0}{2} K_0\right)$$

In the general case, with a global timestep τ_0 the iteration will stop at the *i*-th step if holds

$$K_{i+1} + K_{i+2} + ... = 0 .$$

If the potential is in addition decomposed according to the functional role of the molecules that generate it, there is a hope to achieve optimization of the accuracy of the computations and the performance of the method. To this end, a detailed analysis of the work load and communications increase and their distribution among the individual computing cores is indispensable.

Conducted investigations

We have studied the scalability and the work-load increase and distribution among the computing cores in the packages GROMACS (version 4.5.3) and NAMD [24] (CVS from 19.02.2011), on the example of three test systems with increasing size ($5x10^5$, $\sim10^6$ and $\sim2.2x10^6$ atoms respectively), with the profiling tool SCALASCA [20] and also by means of the GROMACS in-built tool g_tune_pme. 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 the smallest test system, while NAMD was studied with all three of them.

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 communications 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) [21, 22].

Particle decomposition is the simplest way to divide the system. At 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.

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. 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 neighbours 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 centre lies.

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 [23] 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 [21].

We have also analysed the stability and scalability of the existing integration algorithms with variable time-step implemented in widely used MD simulations codes like NAMD and GROMACS in order to define the sources of the instabilities (different kinds of resonances).

The test systems were chosen to be substantially different in size and structure – epidermic growth factor, satellite of the tobacco mosaic virus and E.Coli ribosome dissolved in water (Fig. 1). The two packages – GROMACS and NAMD, were those selected for studying and optimization for petaflops architectures within the PRACE initiative. The scalability of the packages was investigated up to 8192 computing cores. Details about the three test systems are given in the Appendix.

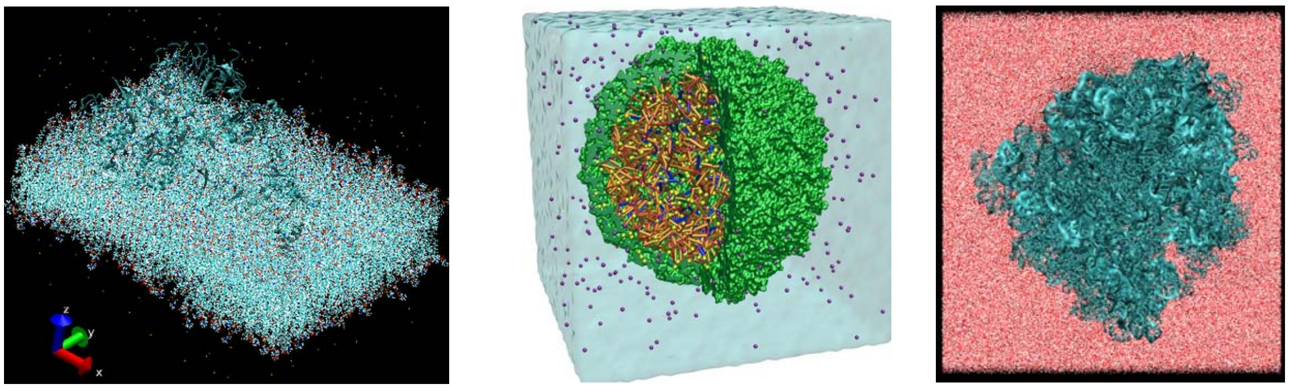

Fig. 1 The test systems: (A) ~5 x 10^5 atoms; (B) ~10^6 atoms; (C) ~2.2 x 10^6 atoms.

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.

These investigations were performed at the IBM BlueGene/P supercomputer of Bulgarian National Center for Supercomputing Applications [25] (8192 computing cores).

Results

Our main results can be summarized as follows:

- 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 only up to that point, even if with a lower overall performance than NAMD. The performance and speedup data for all three test systems on different number of computing cores are given in Table 1. The performance is taken directly from the output information of running the program packages for a certain number of integration steps and represents the simulation time to be obtained for 24 hours work of this package with an integration step of 2 fs. As a reference value in determining the speedup of the simulations was taken the performance at 512 computing cores (Fig. 2). GROMACS performs all calculations in double precision, when NAMD works in single precision. In the same mode GROMACS performance increases by 30%. This partially explains the better results obtained with NAMD;
- By default GROMACS divides the pp and pme computing cores in proportion 3:1 with ordering mode "interleave". In this mode, 1 pme core is placed after every 3 pp cores. 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. 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. Up to 2048 cores, the three domain decomposition modes of GROMACS – interleave, pp_pme and Cartesian – have similar performance, with slight prevalence of the default mode – interleave. However, on 4096 cores this mode has the lowest performance, the other two perform better, with a negligible difference between them (Table 2). We further investigated the dependence of GROMACS performance on the portion of pme cores. The optimal portion was determined by varying the pme cores percentage in the range between 3 and 25 %. The best speedup and performance for set of testing points are plotted on Fig. 3 (left and right respectively) for GROMACS versions 4.5.3 (blue line) and 4.5.5 (red line). As seen on Fig. 3 GROMACS 4.5.3 scales well up to 4096 cores and then its performance declines at 8192 cores. This problem does not occur in the newer version 4.5.5 which demonstrates a significant improvement in scalability with increasing

performance even for 8192 cores also in absolute terms - 11 ns per day. It was found that for GROMACS version 4.5.5 the optimal performance is achieved if the pme cores are between 12 and 14 % of the available computing cores (Fig. 4). Unlike GROMACS 4.5.3, this tendency is stable and hold true even for 8192 cores.

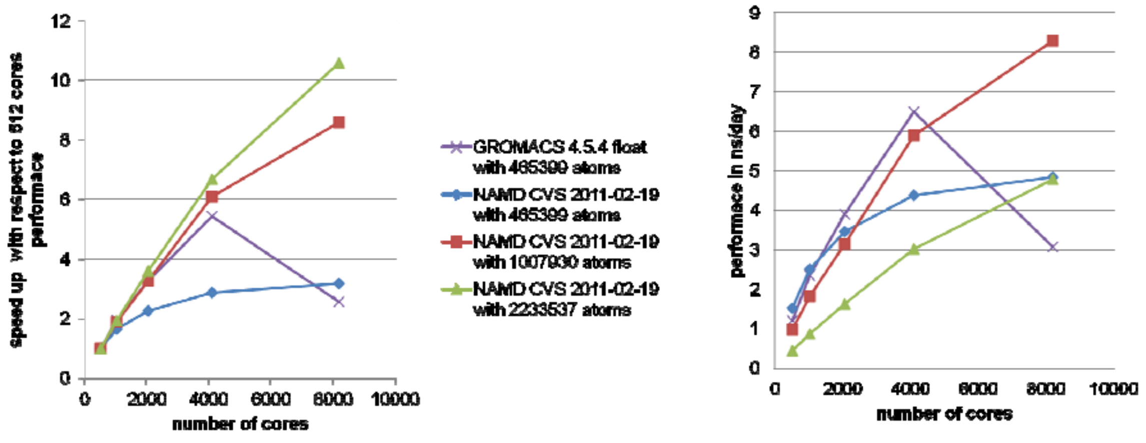

Fig. 2 Performance of GROMACS 4.5.4 and NAMD CVS 2011-02-19 as a function of the number of cores for the three test systems, with 465399, 1007930 and 2233537 atoms resp.

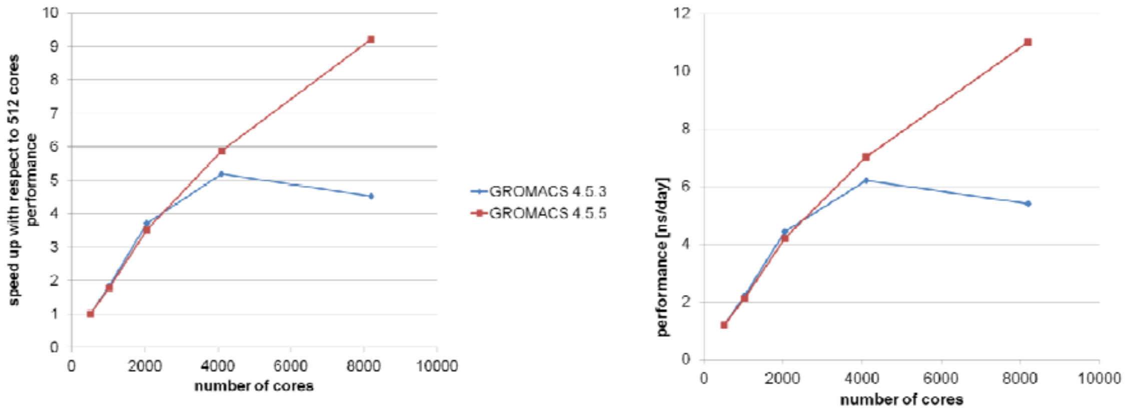

Fig. 3 Performance of GROMACS 4.5.3 and GROMACS 4.5.5 as a function of the number of cores for the system with 465399 atoms.

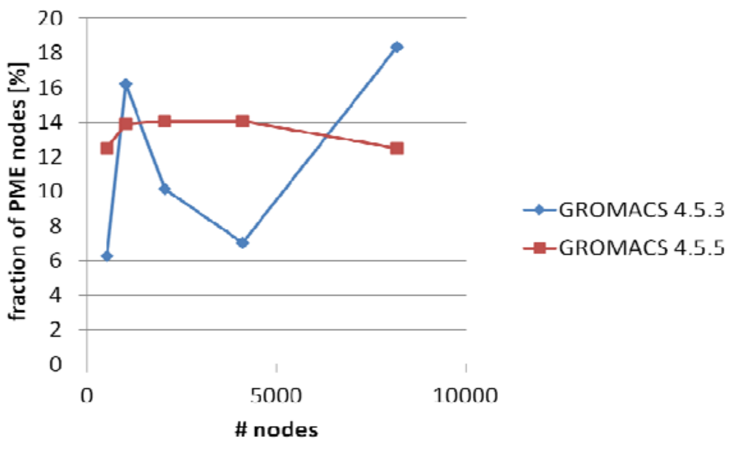

Fig. 4 Number of pme cores / total number of cores ratio vs. total number of cores

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]	Speedup	Performance [ns/day]	Speedup
465 399	8 192	4.84	3.19	3.06	2.56
	4 096	4.38	2.88	6.49	5.43
	2 048	3.45	2.27	3.90	3.26
	1 024	2.50	1.64	2.34	1.96
	512	1.52	1.00	1.19	1.00
1 007 930	8 192	8.30	8.60		
	4 096	5.88	6.10		
	2 048	3.15	3.26		
	1 024	1.81	1.88		
	512	0.97	1.00		
2 233 537	8 192	4.78	10.58		
	4 096	3.02	6.68		
	2 048	1.62	3.60		
	1 024	0.87	1.92		
	512	0.45	1.00		

Table 2. GROMACS performance in the different dd-order modes.

ddorder mode computing cores	Interleave [*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

- The GROMACS 4.5.3 performance analysis with the profiling tool SCALASCA shows that the pme cores which were taken to be ¼ of all computing cores, account for almost 40% of the communications, such a behaviour being common for all three dd-order modes (Fig. 5);

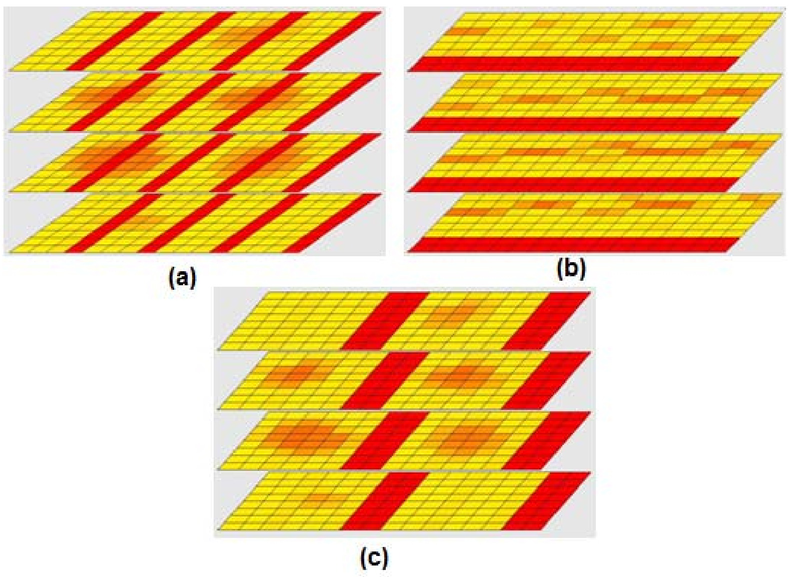

Fig. 5 Distribution of the communications in GROMACS 4.5.3: (a) interleave; (b) pp_pme; (c) Cartesian (red – higher intensity, yellow – lower intensity).

- A study of the work-load and communication distribution over computing cores of the GROMACS executable with the profiling tool SCALASCA at a test system of about 460000 atoms on 512 and on 1024 computing cores (1/8 pme cores) shows that the average time spend by the pp cores is significantly shorter compared to the pme cores, that is consistent with the average amount of communications per core for the pme cores being essentially higher (by a factor of almost 2.5) (Fig. 6). The increase of the pme cores number would accelerate the 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 number the total amount of communications increases, which may lead to a substantial slowdown of the calculation;
- Analysis of short simulations (2 ps) with different fractions of pme cores (1/2, 3/8, 1/4, 1/8 and 1/16) of a test system of 460000 atoms on 512 to 4096 computing cores on the Bulgarian supercomputer IBM BlueGene/P with GROMACS 4.5.4 allows to conclude that the performance increases with the reduction of the number of pme cores up to 4096 cores where saturation is observed (Fig. 7). Nevertheless, with the reduction of the pme only cores scalability drops, because of the increase in communications (Fig. 8);

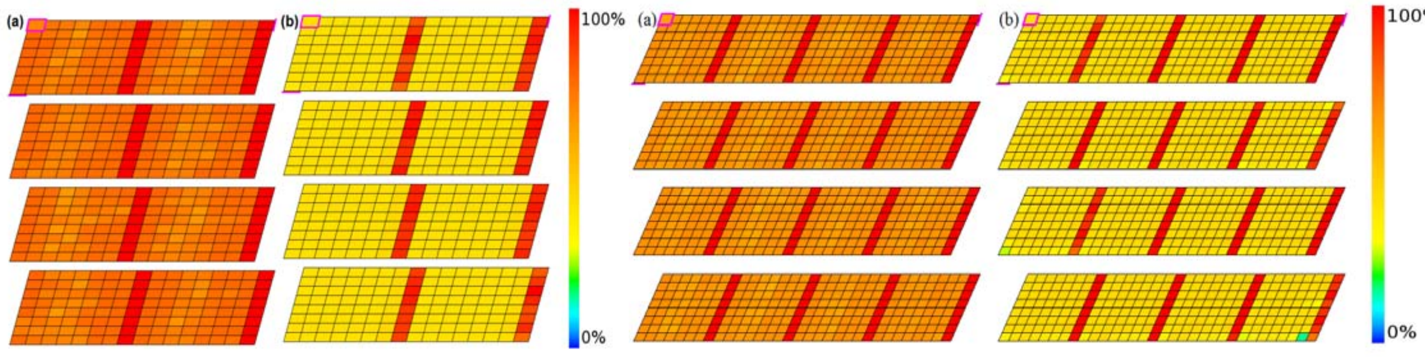

Fig. 6 Distribution over the cores in the interleave regime of the total simulation time and of the communications: (a) for 512 computing cores; (b) for 1024 computing cores.

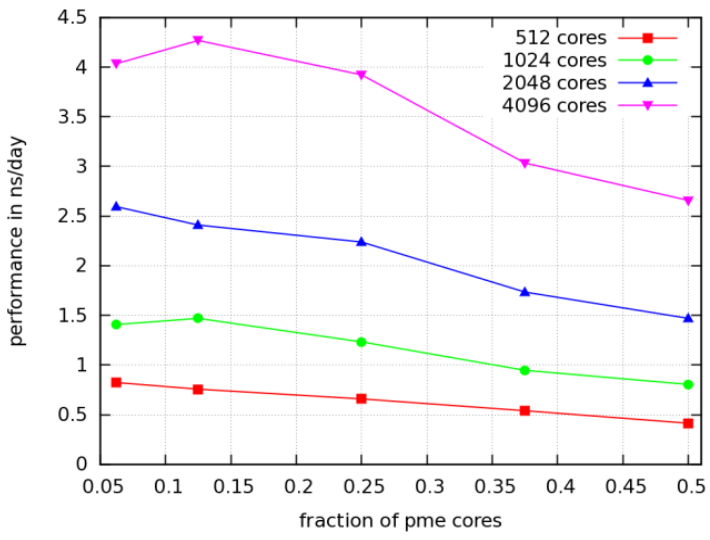

Fig. 7 GROMACS performance as a function of the number of pme only cores (shown as fraction from the total amount of cores).

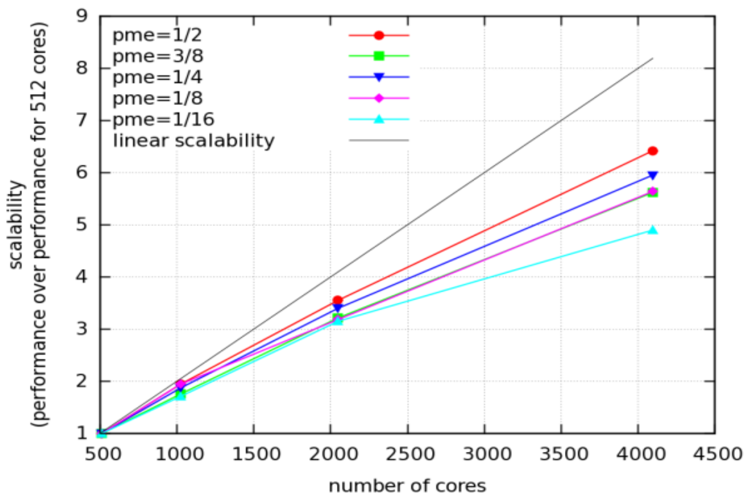

Fig. 8 The scalability of the Gromacs integrator for different number of pme only cores (shown as a fraction from the total amount of cores).

- GROMACS appears to be less suitable for implementation of variable step-size algorithms than NAMD, where an essential improvement of the performance is already achieved that way. Simulations with a system containing approx. 35000 atoms demonstrate that it is possible to speed up the calculations with 47,8 % using the VerletI/r-RESPA multiple timestep algorithm when calculating the short range electrostatic interactions at every timestep and the long-range electrostatic interactions – every 6 timesteps (column 5 on Fig. 9), with perfect energy conservation as shown on Fig. 10. An even greater speedup might be achieved with the parameters from the sixth column on Fig. 8, but at the price of unstable simulation after 34 *ns* with conserved total energy though.

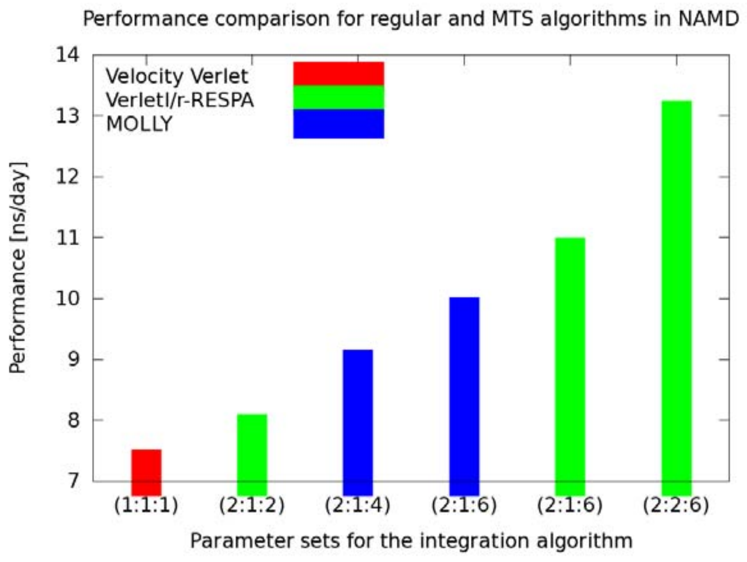

Fig. 9 NAMD performance for different parameter sets (timestep [fs] **:** frequency/ short-range [number of timesteps] **:** frequency/ long-range [number of timesteps]) for different integration algorithms.

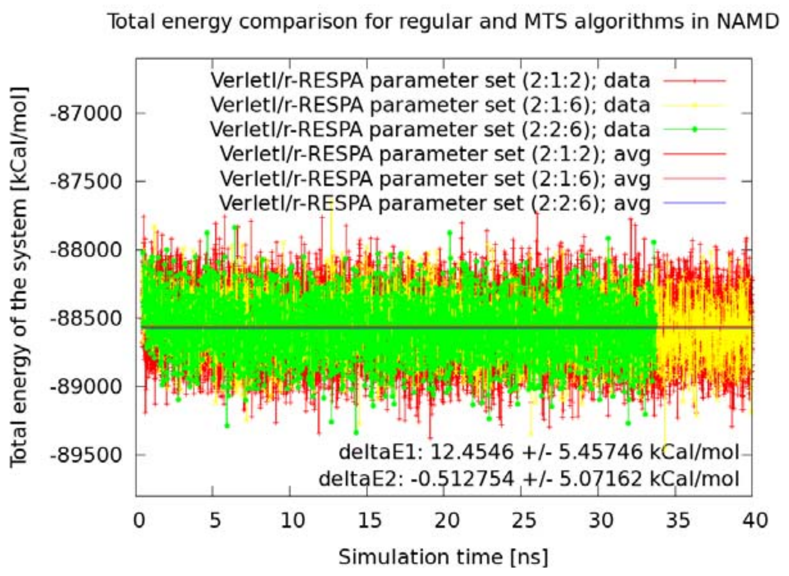

Fig. 10 VerletI/r-RESPA integration algorithm shows perfect energy conservation for different parameter sets. Parameter set (2:2:6) leads to unstable simulation after 34 ns.

Conclusions and summary

The investigations allow identifying the increasing communications as a main reason for the performance deterioration of the MD simulation packages on large numbers of cores – typical for the petascale computations situation. The partial progress with NAMD justifies the idea for overcoming these difficulties by a dedicated integration algorithm with variable step size, specially adapted to the peculiarities of the large biological systems.

Acknowledgements

This work was supported by the PRACE Project funded in part by the EUs 7th Framework Programme (FP7/2007-2013) under grant agreements no. RI-211528 and RI-261557 and by Bulgarian Science Fund under grant no. DOFP7-02/13/2010. The work was performed on the PRACE Research Infrastructure resources – the IBM BlueGene/P of the National Centre for Supercomputing Applications in Sofia [21].

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) [26]. Mutations concerning EGFR expression could lead to various malignant diseases, including lung and colon cancer and multiform glioblastoma [27, 28]. 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 (~ 5 x 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 [29]. 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, accounting to 1007930 (so, roughly 10^6) atoms altogether.

Test system (C). The ribosome 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 metabolic 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 [30]). These structures do not include counter ions, so an specific algorithm for their addition was developed, differing from the one described in [31]. 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 physiological concentration. The whole system accounts 2 233 537 atoms.

References

1. L. Verlet, Phys. Rev. **159** (1967) 98–103.
2. H. Poincaré, *Les M´ethodes Nouvelles de la M´ecanique C´eleste. Tome III*, Gauthiers-Villars, Paris, 1899.
3. V. Arnold, Mathematical Methods of Classical Mechanics. (Springer Science, NY, second ed., 1989).
4. E. Hairer, C. Lubich, and G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. (Springer, Heidelberg, Germany, second ed., 2004).
5. T. F. Miller III, M. Eleftheriou, P. Pattnaik, A. Ndirango, D. Newns, and G. J. Martyna, J. Chem. Phys. **116** (2002) 8649.
6. S. Nosé, J. Phys. Soc. Jpn. **70** (2001) 75.
7. J. Sanz-Serna, M. Calvo, Numerical Hamiltonian Problems. (Chapman and Hall, London, 1994).
8. T. Schlick, M. Mandziuk, R. Skeel, K. Srinivas, J. Comp. Phys. **140** (1998) 1.
9. C. Kane, J. Marsden, M. Ortiz, M. West, Int. J. Numer. Mech. Engng. **49** (2000) 1295.
10. C. Kane, J. Marsden, M. Ortiz, J. Math. Phys. **40** (1999) 3353.
11. R.W. Hockney, S.P. Goel, J. Eastwood, J. Comp. Phys. **14** (1974) 148–158.
12. H.J.C. Berendsen, W.F. van Gunsteren, in *Molecular-Dynamics Simulation of Statistical-Mechanical Systems*, Proc. Int. Sch. Phys. "Enrico Fermi", course 97, G. Ciccotti and W.G. Hoover eds. (North-Holland, Amsterdam 1986) pp. 43-65.
13. W.C. Swope, H.C. Andersen, P.H. Berens, K.R. Wilson, J. Chem. Phys. **76** (1982) 637–649.
14. Z. Ge, J. Marsden, Phys. Lett. **A 133** (1988) 134.
15. D. Janezic and M. Praprotnik, J. Chem. Inf. Comput. Sci. **43** (2003) 1922–1927.
16. R. Skeel, J.J. Biesiadecki, Ann. Num. Math. **1** (1994) 1–9.
17. Wei He and Sanjay Govindjee, *Application of a SEM Preserving Integrator to Molecular Dynamics,* Rep. No. UCB/SEMM-2009/01, Univ. of California, Berkley, 27 pp.
18. T. Darden, D. York, L. Pedersen, J. Chem. Phys. **98** (1993) 10089–10092.
19. M.H. Lee, M.J. Duncan, and H.F. Levison, in *Computational Astrophysics*, Proc. 12th Kingston Meeting, ed. D. A. Clarke and M. J. West (San Francisco: Astronomical Society of the Pacific, 1997) p. 32.
20. M. Geimer, F. Wolf, B.J.N. Wylie, E. Ábrahám, D. Becker, B. Mohr, Concurrency **22** (2010) 702–719.
21. B. Hess 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.
22. http://www.gromacs.org/Documentation/Manual
23. T. Darden, D. York, L. Pedersen, Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems. J. Chem. Phys. **98** (1993) 10089–10092.
24. http://www.ks.uiuc.edu/Research/namd/
25. http://scc.acad.bg/
26. R.S. Herbst, Int. J. Radiat. Oncol. Biol. Phys. **59** (2 Suppl) (2004) 21–6: doi:10.1016/j.ijrobp.2003.11.041. PMID 15142631.
27. H. Zhang, 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.
28. F. Walker, L. Abramowitz, D. Benabderrahmane, X. Duval, V.R. Descatoire, D. Hénin, T.R.S. Lehy, T. Aparicio, Human Pathology **40**/11 (2009) 1517–1527.
    doi:10.1016/j.humpath.2009.05.010. PMID 19716155.
29. http://www.ks.uiuc.edu/Research/STMV/
30. E. Villa et al., Proc. Natl. Acad. Sci. USA **106** (2009) 1063–1068.
31. K. Y. Sanbonmatsu and C.-S. Tung1. Journal of Physics: Conference Series **46** (2006) 334–342.