Science & Technology Facilities Council

Technical Report
DL-TR-2009-002

Large Biomolecular Simulation on HPC Platforms I. Experiences with AMBER, Gromacs and NAMD

HH Loeffler, MD Winn

September 2009

Chadwick Library
Science and Technology Facilities Council
Daresbury Laboratory
Daresbury Science and Innovation Campus
Warrington
WA4 4AD

Tel: +44(0)1925 603397
Fax: +44(0)1925 603779
email: library@dl.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk/

ISSN 1362-0207

Large biomolecular simulation on HPC platforms I. Experiences with AMBER, Gromacs and NAMD

Hannes H. Loeffler, [a] Martyn D. Winn [a]

[a] *Computational Science and Engineering Department, STFC Daresbury Laboratory, Daresbury, Warrington WA4 4AD, United Kingdom.*

Abstract

We performed a general assessment of performance of the three molecular dynamics (MD) packages AMBER, Gromacs and NAMD on the four hardware platforms BlueGene/P, HP Cluster Platform 4000, HPCx and the UK's current national flagship HECToR. The membrane-embedded Epidermal Growth Factor Receptor was chosen as a large system of biological interest. Results on a smaller soluble protein are provided to assess scaling with system size. We report the number of nano seconds per day and the relative speedup that can be achieved. The performance is discussed in detail and some general advice for MD simulation of large scale biological systems is given.

1 Introduction

Molecular dynamics (MD) simulations of biologically relevant systems have become reasonably routine now. Since the first attempts at MD of simple systems in the late 1960s, force fields for proteins, nucleotides and lipids have been developed. These force fields have now reached a very high degree of reliability enabling researchers to predict a wide variety of molecular and dynamical properties.

Naturally, with increasing computing power the size of the systems studied has increased over the years. A typical system size that can be handled with current technology consists of several hundred thousands of atoms although systems with a few million atoms have also been simulated (Sanbonmatsu and Tung, 2007; Schulten et al., 2007). Of course, the number of particles is limited not only by hardware and software restraints but also by the physical reliability that is expected from simulation. For this purpose several tens of nano seconds of simulation have to be run to obtain at least a reasonable statistical sample. Depending on the property of interest, longer timescales may be needed to get adequate sampling.

Typical biological applications, where all atoms are treated explicitly, comprise a

single or a small number of proteins, peptides or nucleotides, with associated ligands or cofactors. The inclusion of a membrane into MD simulation is a more recent development and the subject of ongoing research. In particular, the question of which ensemble (most crucially pressure control) should be applied in order to reproduce experimental values is not always clear, especially when membrane proteins are also involved. Furthermore, realistic membrane-protein systems are naturally large scale systems. For these reasons, extensive membrane-protein simulations are less common in the literature and an assessment of the performance is of general interest.

Obviously, real biological systems are much more complex than models containing a few individual molecules. Therefore, attempts have been made to study larger scale systems with so called coarse-grained (CG) models. This term simply means that models are developed with a less detailed description, e.g. several atoms combined into one effective pseudo atom or treating whole molecules as single uniform entities.

In this report, we study the epidermal growth factor receptor (EGFR) dimer on a POPC membrane bilayer with a total of about 465 000 atoms. For comparison, we also provide some data on a smaller protein-only system (glutamine binding protein, GlnBP) of about 61 000 atoms. The simulation software packages that were considered here are AMBER/pmemd in version 9 (Case et al., 2005), NAMD 2.6 (Phillips et al., 2005) and Gromacs 4.0.2 (double precision version) (Hess et al., 2008). These were chosen because of their widespread usage in biomolecular simulation. We also include data from a coarse-grained EGFR system with GROMACS (MARTINI force field) with about 138 000 atoms. The biomolecular systems were run, where possible, on a range of high performance computing platforms, namely the HP Cluster Platform 4000 and the IBM BlueGene/P at the STFC Daresbury Laboratory, the national HPCx service, and the national flagship service HECToR. Our aim is to give an assessment of quantities such as speed, memory requirements, reliability and ease-of-use.

2 Running the tests

2.1 Test systems and force fields

The principal test system is a model of the dimer of EGFR, including the ectodomains and the transmembrane helices but not the cytoplasmic domains, embedded in a POPC lipid membrane. The second system is GlnBP as a small scale model for comparison. Both systems are placed in explicit solvent. Table 1 summarises the number of atoms and residues of the two systems. For the Gromacs simulation a united-atom force field was used resulting in fewer protein and lipid atoms. To make

Table 1
Number of atoms and residues in the two test systems.

	EGFR [a]				GlnBP	
	atoms	residues	CG atoms	CG res	atoms	residues
proteins	21 749/13 996	1425	3112	1426	3555	227
lipids	134 268/52 104	1002	26 169	2013	—	—
ions	295	295	—	—	1	1
water	309 087/398 994	103 029/132 998	108 445	108 445	57 597	19 199
total	465 399/465 389	105 751/135 720	137 726	111 884	61 153	19 427

[a] The second numbers are for the united-atom force field employed with Gromacs. The total number of atoms has been adjusted with additional waters to match the all-atom force fields.

runtimes comparable with the all-atom force fields, we adjusted the total number of atoms by adding additional water molecules.

Most MD simulation software packages support several alternative force fields. NAMD can read both CHARMM and AMBER files. Gromacs is the most versatile of the three, running besides the Gromos force fields also CHARMM and AMBER force fields. Membrane force fields are, however, less well established and therefore force field parameters can be chosen only from a more limited set. For NAMD, we used the CHARMM parameter set for both proteins and membrane, while for Gromacs we used the Gromos force field for the proteins and a mixture of OPLS parameters and the Berger set (Berger et al., 1997) for the lipids. For AMBER, we used the AMBER force field for proteins and the Martinek set (Jojart and Martinek, 2007) for the POPC lipids. The water model was TIP3P for both NAMD and AMBER, while for Gromacs we chose the SPC model. Ion parameters for chloride and sodium ions were taken according to the available protein force fields. In the case of GlnBP we chose the AMBER parameter set as this set is the one supported by all three software packages, although we also checked NAMD with the CHARMM force field. Runtimes for the latter, however, were very similar and thus we will not report these values here.

The CG simulation was carried out with Gromacs. The MARTINI force field (Marrink et al., 2007; Monticelli et al., 2008) was applied which on average maps 4 atoms to 1 coarse-grained bead. Water molecules are represented by single beads. No ions were added as the protein has zero total charge. Effective simulation times are typically obtained by multiplying with a factor of 4 to account for the faster dynamics in the CG model.

2.2 Simulation parameters

The algorithms implemented in the MD packages considered differ in a number of ways, making comparisons between the programs difficult. One example is the time step integrator, although we do not expect much impact on runtimes. Another example is temperature and pressure control. The common denominator of all three programs used in this study is the Berendsen algorithm which we chose for both temperature and pressure.

Computation of long-range electrostatic interactions is potentially very time-consuming. Simple cutoff methods may have serious repercussions on results and are usually avoided. A modern solution to this problem is the Particle Mesh Ewald (PME) method which is implemented in all three programs. We mostly went with the default PME parameters except for the convergence criterion and the grid size which has to be set explicitly in NAMD. For the CG simulation PME was not used but rather a shifted cutoff for both Coulomb and Lennard-Jones interactions in accordance with the original publication (Marrink et al., 2007).

The EGFR simulations have been run with a time step of 2 fs for 10 000 MD steps, and the GlnBP simulation for 50 000 steps with the same time step. The CG simulations have been run with a time step of 20 fs and 100 000 steps. These run times are sufficiently long to keep the time spent in the serial parts of the MD codes to a minimum thereby measuring predominantly the parallel performance. The setup time, i.e. the time spent for reading the input files, allocating memory, etc. was generally smaller than 2% of the CPU time spent in parallel code. The run times are also small enough to allow runs on small processor counts to complete in a reasonable time. This is important because queueing policies imposed on the various hardware platforms only allow certain maximum run times for a single job.

2.3 Software

An overview of the three MD software packages employed in this study is given in table 2. Listed are the version numbers, the force fields used and whether the programs have been compiled by ourselves or by someone else. Executables on Hector and HPCx have been prepared by the respective support teams. NAMD executables for the BlueGene systems have been compiled by the authors of the software.

Table 2
Overview of the three MD simulation packages used for this report.

Program	AMBER/pmemd	Gromacs	NAMD
Version	9	4.0.2	2.6
Force field EGFR	AMBER/Martinek	Gromos/Berger	CHARMM
Force field GlnBP	AMBER	AMBER	AMBER
Compiled HECToR [a]	pre	pre	pre
Compiled HPCx	pre	pre	pre
Compiled BG/P	self	self	pre
Compiled HP/4000	self	self	self

[a] Abbreviations used: pre is pre-compiled and self means compiled by the authors of this report.

Table 3
Overview of the four HPC hardware platforms used for this report.

Machine	Processor	No. processors	Memory
HECToR	AMD 2.8 GHz Opteron	11328	6 GB/dual-core
HPCx	IBM 1.7 GHz POWER5	2560	32 GB/16 procs.
BG/P	IBM 850 MHz PowerPC	4096	512 MB/core
HP/4000	AMD 2.4 GHz Opteron	128	2 GB/core

2.4 Hardware

The test systems were run on four different platforms. HECToR is the current national service, and is thus the standard reference. A brief overview of the hardware is summarised in table 3. The detailed architecture of the systems is more complicated than presented, in particular the combination of cores into nodes and their interconnects. However, the simplified presentation here will suffice to get an overview of performance.

3 Results

3.1 Setup Procedures

Before an MD simulation can be run, various setup tasks have to be performed. All three simulation packages considered here have separate programs assisting in the preparation of starting files. Typical problems such as missing residues may

be resolved with some effort in very simple cases but in the general case external programs must be used.

The AMBER package comes with LeAP in two versions: a terminal only program `tleap` and a graphical program `xleap`. LeAP reads PDB files (also the Tripos MOL2 format typically used for small molecules) and tries to carry out standard conversions. Missing atoms can be added automatically if internal coordinates of the residue in question are known. All implemented force fields come with predetermined internal coordinates. Disulphide bonds have to be defined explicitly as there is no automatic procedure to find them. The POPC lipids are recognised correctly provided that the PDB file follows the correct order and naming convention for the Martinek force field. For this purpose we had to write our own script to convert all lipid entries to the correct format. Occasional problems were found with naming conventions of atoms, for example CD vs. CD1 in ILE or hydrogens labelled with numbers 1 and 2 when LEaP expects them to be 2 and 3. These problems had to be fixed by renaming all offending names externally. Another disadvantage of the program is that it does not keep a record of chains and starts residue numbering consecutively from 1. This is somewhat troublesome if one wishes to compare to the literature with a specific numbering scheme. Here again we wrote little helper scripts to compute residue number offsets automatically.

The GROMACS utility for automatic setup is called `pdb2gmx` and is entirely command line controlled, although the choice of protonation states and disulphide bonds can be done interactively during execution of the program. Input hydrogen atoms can be ignored if required. Disulphide bonds can also be detected automatically. The program can be made to ignore missing atoms but this is strongly discouraged. External tools have to be used to model missing atoms. `Pdb2gmx` deals only with proteins. The tool itself can operate with individual chains (also merging them into one if desired) but renumbers residues in a similar fashion to LeAP. Lipids can be incorporated into the final topology file via topology file fragments which are provided for many cases on Tieleman's website (http://moose.bio.ucalgary.ca/). As in the AMBER case, we had to rearrange the order and names in the PDB file to match the predefined convention.

The NAMD program uses `psfgen` for the creation of CHARMM/X-PLOR formatted input files but can also directly read LeAP generated files. CHARMM input files can of course also be generated by CHARMM itself via the scripting facility of this package. `Psfgen` is also a script driven program but VMD can be used as a convenient graphical front end. `Psfgen` has commands to rename atom and residue names if desired allowing the original PDB file to be left untouched. Disulphide bonds have to be specified as so-called patches but VMD can help in automatic assignment. Lipids are recognised automatically if the PDB file follows naming and order conventions. Missing atoms can be guessed through internal coordinates which are present for all standard residues. The CHARMM format does operate with chains and also keeps a record of the provided order. Internally this is solved

by assigning so-called segments to a chosen number of atoms. The only drawback is that segments cannot contain more than 99 999 atoms which means that larger numbers of waters have to be split into several segments.

A particular problem in all cases is the assignment of protonation states. The assignment itself cannot be done with the available setup programs. External programs or online service can be used (e.g. http://propka.ki.ku.dk/ or http://biophysics.cs.vt.edu/H++/). Once the protonation states are determined, the best choice is usually to rename residues in the PDB files accordingly (e.g. HIE for ε protonated HIS in the AMBER force fields, but HSE for the CHARMM force fields) and allow the program to assign hydrogen atoms based on this information. Another problem in this connection is the compatibility between force fields. Some protonation states may not be available at all, for instance neutral lysine. The option here is to switch force fields or compute missing parameters which, however, usually requires very elaborate procedures.

Conversion between file formats is only possible if the force fields are supported by the target MD software. A force field is not only a set of parameters as generated by the helper programs discussed here but also a energy/force function that must be supported in software. Some utilities are available to aid in conversion but there is still a lot of room for improvement. We have for instance used a tool to convert the AMBER topology file to GROMACS format in the simple case of GlnBP.

3.2 Run Times and Speed

In the following, we summarise the run times of the two simulation systems GlnBP (small system) and EGFR (large system with membrane) on the four hardware platforms BlueGene/P, HP Cluster Platform 4000, HPCx and HECToR. We report performance as the number of nano seconds per day extrapolated from 100 ps (GlnBP) and 20 ps (EGFR) respectively, and as the relative speedup. The speedup is calculated relative to 4 (GlnBP) and 8 (EGFR) processors, rather than the usual definition relative to a single processor. Simulations on a single processor were not available for a number of reasons. First, `pmemd` runs on a minimum of 2 processors. Second, jobs with only 1 or 2 processors could not reach the desired number of timesteps within the maximum allowed wall time, while shorter runs would lower the time spent in parallel code to an unacceptable level. Third, for real production runs lower processor numbers would not be feasible anyway. Finally, some hardware platforms charge the user in multiples of a certain number of processors, typically 16 or 32. As a consequence of our definition, ideal scaling is represented by a line of gradient 1/4 (GlnBP) or 1/8 (EGFR) in the figures.

In figure 1, we show results for the smaller test system GlnBP for the two fastest hardware architectures; HPCx (left-hand side) and HECToR (right-hand side). At

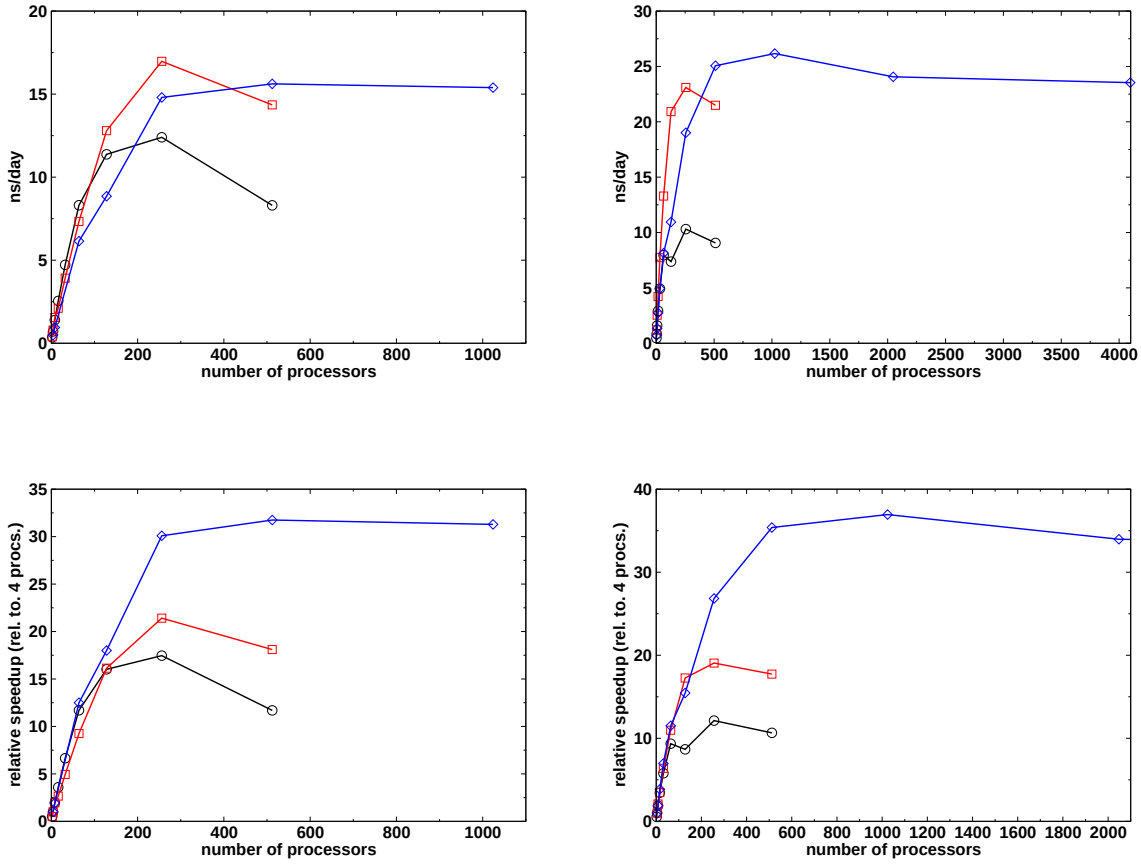

Fig. 1. Nano seconds per day (top) and relative speedups (bottom) for the GlnBP simulation. The left panel is for HPCx and the right panel for HECToR. Black: AMBER, red: GROMACS, blue: NAMD.

low processors counts, all three program packages give comparable run times and speedups. On HPCx, AMBER gives marginally the best performance in terms of ns/day for up to 64 processors, beyond which Gromacs takes over. NAMD however scales better, according to the relative speedup, and produces the most ns/day at 512 processors. GROMACS and NAMD scale in an acceptable way up to 256 processors, while in the case of AMBER this number is 128. The maximum performance with 17 ns/day was achieved with GROMACS on 256 processors.

On HECToR, the picture is different. Here AMBER is the slowest at all processor counts. GROMACS is fastest up to 256 processors. NAMD performs fastest on more than 256 processors and scales up to 1024 processors. The maximum performance with 26 ns/day was achieved with NAMD on 1024 processors.

Figure 2 compares ns/day and speedup for NAMD on the three platforms BlueGene/P, HPCx and HECToR. The software performs most ns/day on HECToR, followed by HPCx and is slowest on the BlueGene/P. The difference in run times is,

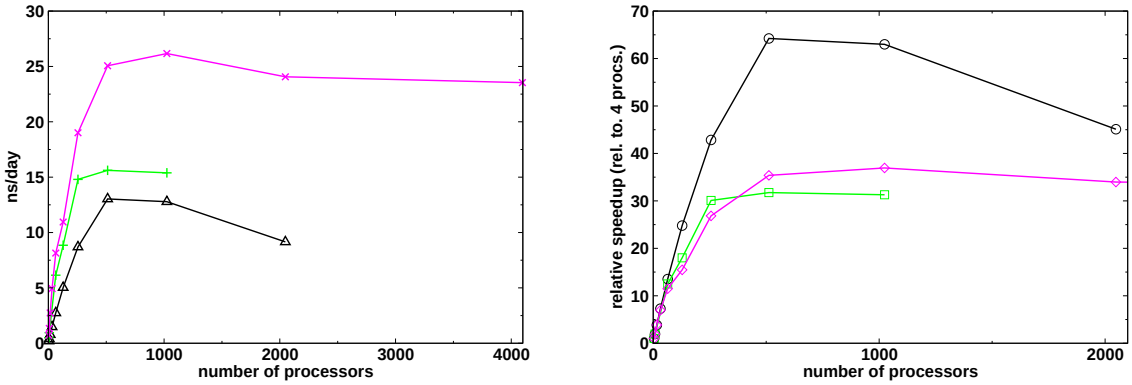

Fig. 2. Nano seconds per day (left) and relative speedups (right) for the GlnBP simulation. All performance data given for runs with NAMD. Black is BlueGene/P, green is HPCx and magenta is HECToR.

however, not so pronounced between HPCx and BlueGene/P as it is between these two architectures and HECToR. The relative speedup is largest on the BlueGene/P while it is quite similar on HPCx and HECToR. On both BlueGene/P and HECToR the performance is still reasonable with 512 processors but on HPCx not much is gained with more than 256 processors.

In figure 3, we plot ns/day for the larger EGFR test system for all four hardware platforms. Up to the maximum of 128 available processors, the performance of the HP Cluster Platform 4000 is very similar to HECToR probably due to the usage of the same processor type although the ones on HECToR run at the higher clock speed of 2.8 GHz vs. 2.4 GHz of the HPCP 4000. Interconnects and local memory do not seem to play a role here.

AMBER is faster on the BlueGene/P than NAMD up to 256 processors but at higher processor counts NAMD is the clear leader. A similar picture emerges on HPCx where AMBER performs better in terms of runtimes up to 256 processors. GROMACS comes in between at lower processor counts but overtakes the other two programs after 128 processors. It is not clear, however, if NAMD would eventually achieve more ns/day than GROMACS at higher processor counts as we observed in the case of the smaller GlnBP system.

On HECToR, we find GROMACS to run fastest for processor counts up to 1024. The most ns/day for the EGFR system with about 8 ns/day can be achieved with this program package on 512 processors. NAMD performs considerably slower but scales quite well up to 2048 processors at which point the software reaches a maximum ns/day similar to GROMACS. AMBER runs as fast as NAMD up to 64 processors but then performance drops considerably and hence scaling is generally bad.

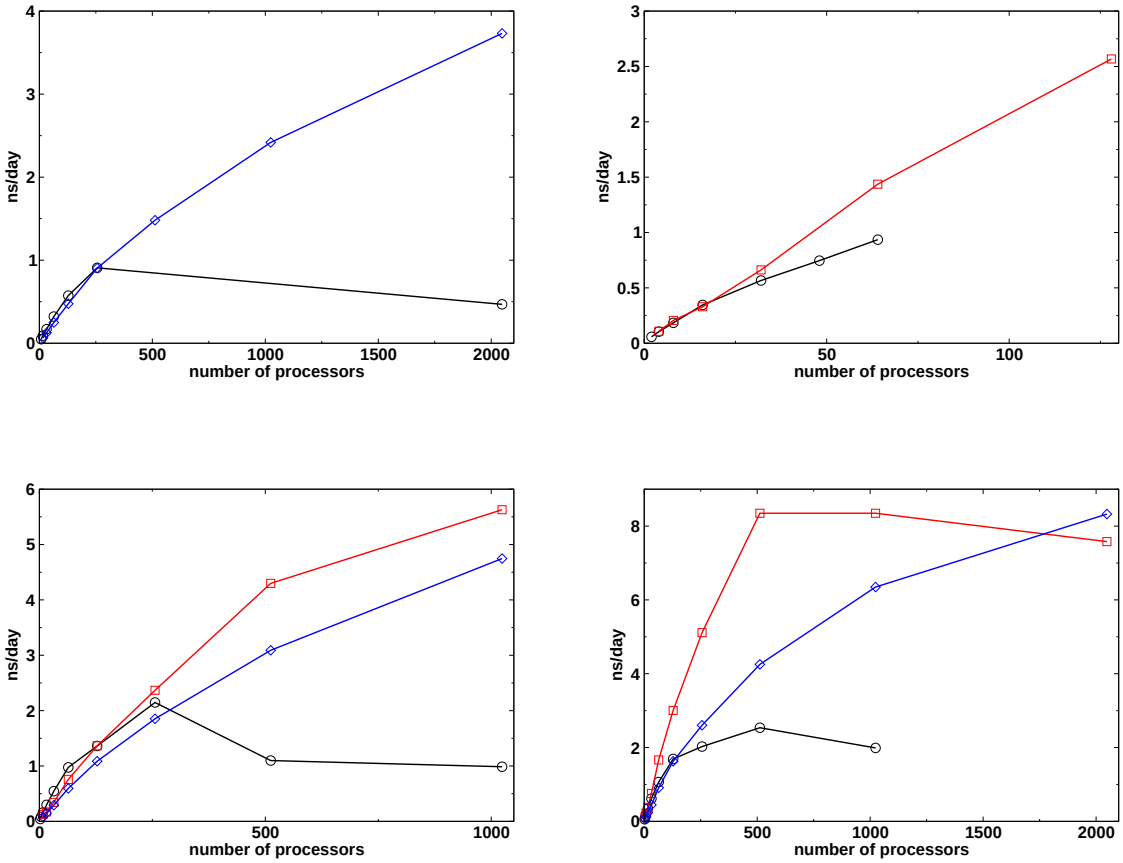

Fig. 3. Nano seconds per day for the EGFR simulation. Top, left: BlueGene/P; top, right: HPCP 4000; bottom, left: HPCx; bottom, right: HECToR. Black: AMBER, red: GROMACS, blue: NAMD.

Figure 4 summarises the speedups relative to eight processors for the four platforms. NAMD scales best on all hardware while GROMACS comes second and AMBER third.

The coarse-gained results for HECToR in Figure 5 show a peak performance of an effective 145 ns/day. This is compared with the peak value of 8.3 ns/day achieved with the atomistic simulation. However, the CG simulation with GROMACS scales only up to 64 processors, compared to 512 processors for the atomistic simulation. It should be noted that the CG run does not make use of the PME method.

Table 4 summarises the maximum nano seconds per day that are achievable on the fastest hardware system used in this test, i.e. HECToR. For our test systems, AMBER clearly performs the least well. GROMACS and NAMD perform equally well in terms of ns/day but NAMD needs considerably more processors to reach its maximum.

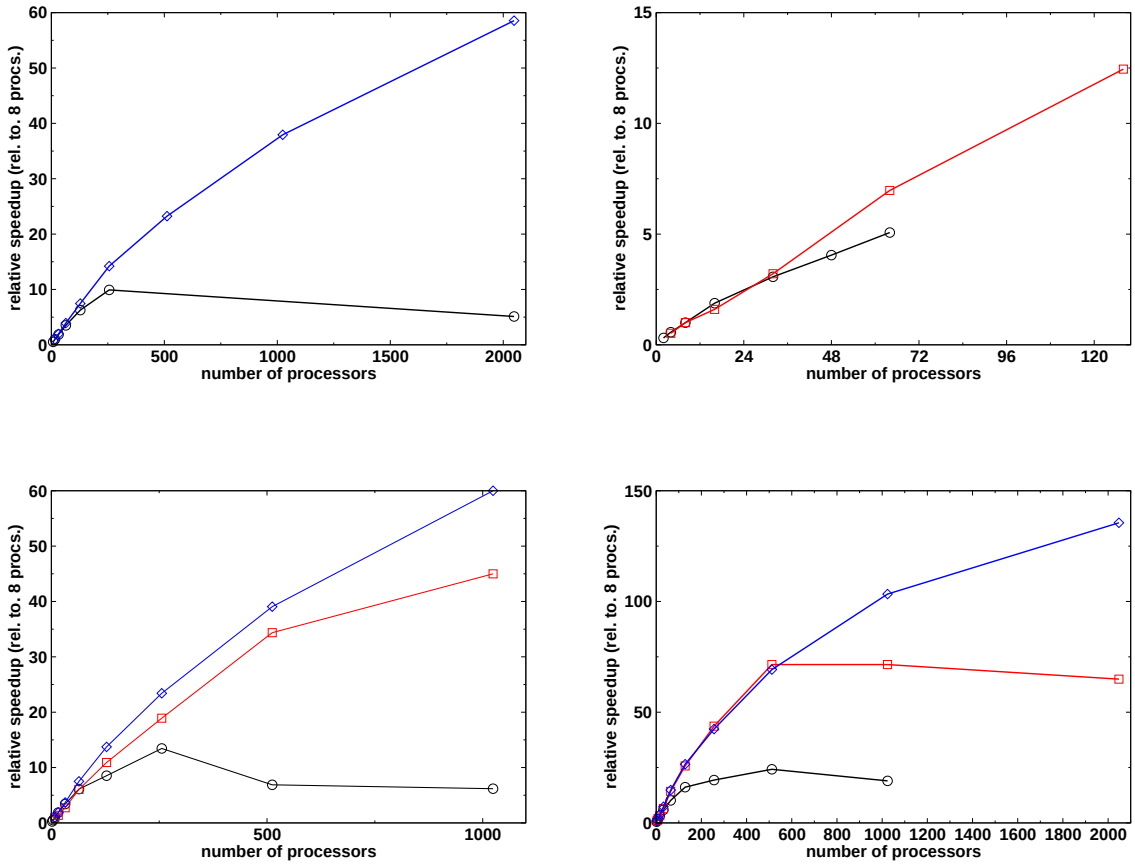

Fig. 4. Relative speedups for EGFR. Top, left: BlueGene/P; top, right: HPCP 4000; bottom, left: HPCx; bottom, right: HECToR. Black: AMBER, red: GROMACS, blue: NAMD.

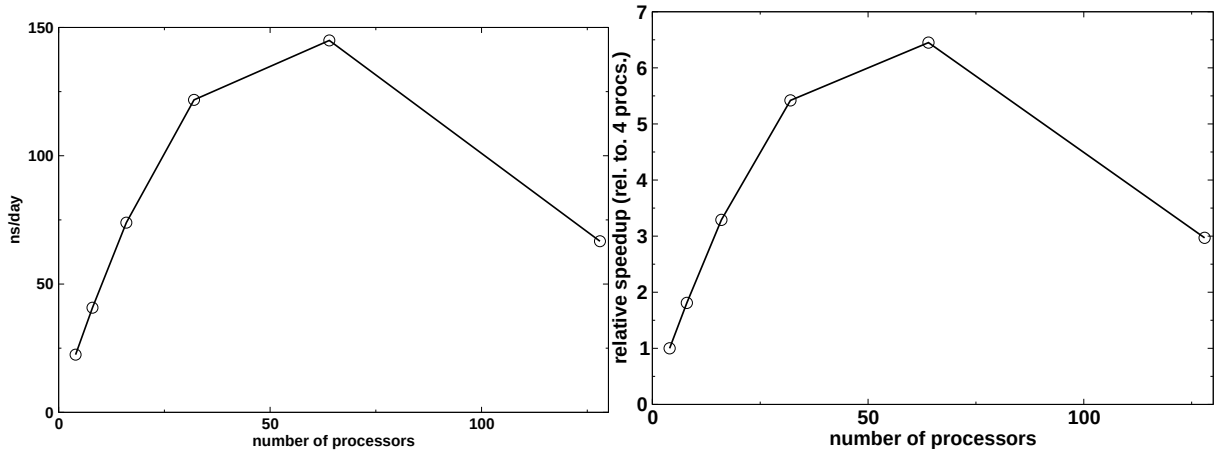

Fig. 5. Nano seconds per day (left) and relative speedups (right) for the EGFR simulation at the coarse-grained level, using Gromacs on HECToR. The ns/day plot is scaled by a factor four to account for the faster dynamics of the CG model.

Table 4
Maximum nano seconds per day and the number of processors for all MD packages achieved on HECToR, the fastest system in this test.

system	software	no. procs.	ns/day
GlnBP	AMBER	256	10
GlnBP	GROMACS	256	23
GlnBP	NAMD	1024	26
EGFR	AMBER	512	2.5
EGFR	GROMACS	512	8.3
EGFR	NAMD	2048	8.3
EGFR	GROMACS/CG	64	145

4 Summary

In the previous section, we have seen that the three MD software packages AMBER, GROMACS and NAMD perform quite differently on the four tested hardware platforms BlueGene/P, HP Cluster Platform 4000, HPCx and HECToR. In general, AMBER's strength appears to be fast run times at low processor counts, although even this is not always the case. Scaling to higher processor counts is poor compared to the other software. GROMACS can be very fast, especially with a small number of processors. However, version 4 of Gromacs also performs well at higher processor counts, and is clearly much improved in this respect over earlier versions. The only exception here seems to be the smaller systems on HPCx. NAMD appears to be rather slow based on a per processor performance but shows the best scaling among the three programs.

Of course, these conclusions are specific to the test systems we have used. The particular combination of AMBER on HECToR has also been benchmarked by Laughton (2007) for 6 biological test systems, ranging from 21,736 to 931,751 atoms. Our two test systems are within this range. Laughton found that the number of ns/day increased up to 256 cores, but that there was little if any gain with 512 cores. This agrees with our experience with GlnBP (figure 1) and EGFR (figure 3). Laughton also found maximum simulation times of about 10 ns/day for system B with 90,906 atoms and about 3 ns/day for system E with 657,585 atoms. These values are similar to those we report in Table 4.

Sanbonmatsu and Tung (2007) investigated the scaling behaviour of NAMD in an all-atom simulation of the ribosome, as well as a number of smaller test systems. The ribosome model included 2.64 million atoms, i.e. almost 6 times larger than the EGFR test system we have used. The scaling simulations were run on up to 1024 processors of the Los Alamos QSC machine. They found excellent scaling behaviour for the larger test systems (more than a million atoms), with 85% ef-

ficiency for the ribosome on 1024 processors, while for smaller systems a peak speedup was observed. That is, for good parallel code the ratio of compute time to communication time is better for larger systems, as we also observed (compare figures 1 and 4). They achieved peak simulation times of 4.5 ns/day (57300 atoms), 3.7 ns/day (328000 atoms) and 1.4 ns/day (2.64 million atoms). We were able to achieve larger simulation rates with NAMD (see figure 4) for comparable system sizes, presumably due to improvements in hardware since their study.

Coarse-grained simulations (here carried out with GROMACS) show a less favourable scaling than the atomistic simulations. It should be noted that Martini uses a shifted electrostatic potential rather than PME, and this is an integral part of the coarse grained parameterisation. The longer cut-off required may lead to a higher level of communication between nodes, and thus poorer scaling. Coarse graining allows the highest ns/day with a peak performance of about 145 ns/day effective simulation time for the EGFR system. This is about 20 times higher than the corresponding atomistic simulation. The choice for the time step of 20 fs is rather conservative, and time steps up to 30 fs or maybe 40 fs are possible in some cases (Periole et al., 2007).

Unfortunately, memory usage is only printed out by NAMD but no problems were encountered. In a different study (Kästner et al., 2009), a system with over 800 000 atoms ran without problems using NAMD on HECToR. Sanbonmatsu and Tung (2007) note that NAMD requires more than 2GB/process for simulations of over 2 million atoms, and that simulations of such large systems with more than 1024 processors terminated with memory problems. The run times used in the current study are too short to compare the physical reliability of the results obtained. Basic checks, however, were carried out to ensure that the model systems stayed in a reasonable state.

A general consideration to be made when running jobs on external hardware is the cost in CPU hours. On HPCx and HECToR, processors have to be allocated in multiples of 16. An “odd” number of, say, 500 doesn’t make much sense economically for obvious reasons. Absolute run times may be an issue but little may be gained from doubling the number of processors, see for example Figure 1 (bottom right) where the relative speedup for 512 is 35.5 and only marginally more (37) for 1024 processors. The user would be charged twice as much for a performance increase of only 4%.

However, the software package cannot always be chosen merely on grounds of maximum performance. One issue may be the features that a particular software package offers, e.g. the number of pressure control algorithms are often limited. Membrane simulation may require algorithms that take care of their special properties. AMBER is not prepared at this moment for such simulations at all. GROMACS supports semi-isotropic scaling (x-y and z directions scaled separately) and constant surface tension. NAMD can do constant surface tension simulations. An-

other point to consider when switching to another code is to make sure that the new program does indeed produce comparable results if backward compatibility is an issue. The major force fields are supported by GROMACS and NAMD. AMBER is most limited in this regards.

The setup programs have been found to be reasonable well suited for the given tasks, although a simple text editor is still a valuable tool. For more complex requirements such as modelling, external programs have to be used. Only AMBER provides a graphical though simple interface, while NAMD relies on VMD and GROMACS is command line only.

Can the three MD software packages perform better than reported here? There are several reasons why the answer may be yes. Using special options at compile may improve runtimes. The development teams of the codes usually take care of this or the support teams of the hardware may optimise the software for this purpose. When the software needs to be compiled by oneself performance increases may be achievable although the resulting executable should be carefully examined for production of reliable results. Another possibility to speed up run times is to modify certain run time parameters in the input file. An obvious choice would be to modify convergence criteria but again reliability checks must be carried out. GROMACS allows the user to adjust the load balancing of a job by assigning the number of PME vs. non-PME nodes. In a quick test a few percent of higher performance could be obtained.

Acknowledgement. We acknowledge support by HECToR and HPCx, and by the CSED of STFC for access to its HP Cluster Platform 4000 and IBM Blue Gene/P systems. We are especially grateful to the user support teams of these systems.

References

Berger, O., Edholm, O., Jähnig, F., 1997. Molecular dynamics simulations of a fluid bilayer of dipalmitoylphosphatidylcholine at full hydration, constant pressure, and constant temperature. Biophys. J. 72, 2002.

Case, D. A., Cheatham III, T. E., Darden, T., Gohlke, H., Luo, R., Merz Jr., K. M., Onufriev, A., Simmerling, C., Wang, B., Woods, R. J., 2005. The Amber biomolecular simulation programs. J. Comput. Chem. 26, 1668.

Hess, B., Kutzner, C., van der Spoel, D., Lindahl, E., 2008. Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. J. Chem. Theory Comput. 4, 435.

Jojart, B., Martinek, T., 2007. Performance of the general amber force field in modeling aqueous popc membrane bilayers. J. Comput. Chem. 28, 2051.

Kästner, J., Loeffler, H. H., Roberts, S. K., Martin-Fernandez, M. L., Winn, M. D., 2009. Ectodomain orientation, conformational plasticity and oligomerization of erbb1 receptors investigated by molecular dynamics. J. Struct. Biol. in press.

Laughton, C., 2007. Benchmarking amber on hector. CCPB Newsletter 6.

Marrink, S.-J., Risselada, H. J., Yefimov, S., Tieleman, D. P., deVries, A. H., 2007. The MARTINI force field: Coarse grained model for biomolecular simulations. J. Phys. Chem. B 111, 7812.

Monticelli, L., Kandasamy, S. K., Periole, X., Larson, R. G., Tieleman, D. P., Marrink, S.-J., 2008. The MARTINI coarse-grained force field: Extension to proteins. J. Chem. Theory Comput. 4, 819.

Periole, X., Huber, T., Marrink, S.-J., Sakmar, T. P., 2007. G protein-coupled receptors self-assemble in dynamics simulations of model bilayers. J. Am. Chem. Soc. 129, 10126.

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R. D., Kalé, L., Schulten, K., 2005. Scalable molecular dynamics with NAMD. J. Comp. Chem. 26, 1781.

Sanbonmatsu, K. Y., Tung, C.-S., 2007. High performance computing in biology: Multimillion atom simulations of nanoscale systems. J. Struct. Biol. 157.

Schulten, K., Phillips, J. C., Kalé, L. V., Bhatele, A., 2007. Biomolecular modeling in the era of petascale computing. In: Petascale Computing: Algorithms and Applications. Chapman & Hall/CRC, p. 165.