Large biomolecular simulation on HPC platforms III. AMBER, CHARMM, GROMACS, LAMMPS and NAMD

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

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

Abstract

The runtime performance of the popular biopolymer molecular dynamics simulation packages AMBER, CHARMM, GROMACS, LAMMPS and NAMD have been investigated in an extensive benchmark study on the hardware platforms Cray XE6/Interlagos, IBM iDataPlex and BlueGene/P. Five different protein and protein/membrane systems in the range of 20 000 to 3 million atoms served as the test systems. Performance will be reported as the maximum nanoseconds per day that a simulation package may achieve. We will also present an estimate of the system size (total number of atoms) dependent scalability.

1 Introduction

In this third benchmark report we summarise runtime results of five molecular dynamics (MD) packages popular in the field of biopolymer simulation. Some results on previous versions of AMBER, GROMACS, LAMMPS and NAMD have been reported by us earlier[1,2]. New releases of those codes have become available to us and part of this benchmarks study was to reassess the performance characteristics of the new versions. CHARMM is a new addition to this series.

We have also expanded the number of protein and protein/membrane systems to five ranging from sizes of 20 000 to 3 million atoms. Large multi-million atom

Email address: Hannes.Loeffler@stfc.ac.uk (Hannes H. Loeffler,).

Last updated: 15 March 2012

benchmarks have been published before[3,4]. Here we present a joint-benchmark of well-known MD software applied to biological systems scientifically feasible today.

The hardware platforms chosen were the Daresbury Laboratory's BlueGene/P which has already featured in the first two reports. UK's national flagship supercomputing service HECToR (Cray XE6) is included again as the platform has undergone upgrades to its main processors (now AMD Opteron Interlagos) at the end of 2011. The new-comer is IBM's Intel Xeon Westmere based iDataPlex compute server.

2 Running the tests

2.1 Test systems and force field

We have investigated the runtime behaviour of five differently sized test systems. The smallest system was Crambin (taken from PDB 3NIR) comprising a total of about 20 000 atoms. The glutamine binding protein (GlnBP, PDB 1WDN) was set up to 61 000 atoms. The ectodomain of dimeric and doubly ligated human EGFR (hEGFR, modelled on the basis of PDB 1IVO and 1NQL) with transmembrane helices attached and lying flat on a model lipid bilayer of POPC included 465 000 atoms. Two of those hEGFRs standing proud on POPC were constructed and solvated to create a 1.4 million system. The largest system with two tetrameric hEGFRS lying flat on a POPC bilayer had nearly 3 million atoms. More detailed descriptions regarding the setup procedures have been given in the previous reports[1,2]. Scientific results have been published in 5–7. Table 1 summarises the numbers of atoms in each of the five systems broken down into individual constituent parts.

As all MD simulation packages scrutinised in this report support the CHARMM force field we applied the CHARMM 22 force field for proteins[8] with torsional backbone corrections (CMAP)[9] and ions[10], the CHARMM 27 force field for lipids[11–13], and TIP3P for water[8,14] in all benchmarks (but see discussion in section 3.3). The only exception was LAMMPS which did not support the CMAP terms in the version used in here.

2.2 System setup and simulation parameters

The system setup, in particular the creation of topology/structure files, and the specific runtime parameters used for the benchmark simulations have been

Table 1
The five systems used for benchmarking and their size in number of atoms.

system[a]	protein	lipids	water	ions	total
20 k	642	—	18 963	—	19 605
61 k	3 555	—	57 597	1	61 153
465 k	21 749	134 268	309 087	295	465 399
1.4 M	43 498	235 304	1 123 392	986	1 403 182
3 M	86 996	867 784	2 041 230	1 914	2 997 924

[a] Detailed description in the main text.

described in detail in the previous reports[1,2]. Here we briefly note that structure files (PSF) for CHARMM were created from scratch with the CHARMM executable itself. Amber topology files were converted with the utility program `chamber`[15] using those PSF files.

Input files are listed in Appendix B. We typically ran 50 000 MD steps for the two smaller systems although in some cases this was reduced to 10 000 steps because the runs would not finish within the maximum time the batch system allows. In those cases runtimes were multiplied by five. The larger systems were run for 10 000 steps but a few runs did not finish within the maximum alloted time on low core counts. Final results were then extrapolated from runtimes obtained every 1 000 steps (at least 4 data points) assuming a linear regression. Regression coefficients were larger than 0.99 in all cases.

2.3 Software and hardware

An overview of the MD packages and the hardware they ran on is presented in Table 2. Not all systems sizes (cf. Table 1) were run on all software/hardware combinations. See section 2.4 below for more information.

The hardware of the BlueGene/P and the Cray XE6 (HECToR) systems have been described in some detail previously[1,2]. Meanwhile the Cray XE6 has been upgraded to phase 3 with AMD Opteron 6200 (Interlagos) processors clocked at 2.3 GHz and each with 8 cores/16 threads. Each node hosts 2 processors and thus has 32 logical cores per node. The iDataPlex is an Intel Xeon X5650 (Westmere) processor based system. Each CPU is clocked at 2.66 GHz with 6 cores/12 threads. In general we have run our benchmarks at the maximum available (logical) cores (4 on BlueGene/P, 32 on Cray XE6 and 12 on iDataPlex) but for exceptions see below.

Table 2
The hardware and MD software packages benchmarked in this report. Not all combinations were carried out (see main text).

	BlueGene/P	iDataPlex	HECToR/Cray XE6
Amber 11/pmemd	—	—	+
CHARMM 35b5	—	+	+
CHARMM 36b1	—	+	—
GROMACS 4.5.5	float	float/double	double
LAMMPS 27/10/2011	+	+	+
NAMD 2.8	+	+	+

2.4 Limitations

When benchmarks studies are carried out certain limitations to a general interpretation of results will always be in place. Specific to this study is that benchmark runs for each software/hardware/system size combination was carried out only once. As the goal of our studies is to obtain reliable data from long runs to keep the influence of serial code to a minimum multiple runs are prohibitive in time as well as economically. In tests we generally found that runtimes vary insignificantly although this is not always true when runtimes were very short. We still believe, however, that the results presented here give a very good impression of what performance to expect from each combination.

We also need to point out that all hardware platforms are production systems which means that resources are shared between a potentially large number of users. Necessarily, this will have some impact on runtimes but it is not the goal of this report to obtain ideal runtimes from idle systems. The results presented are therefore representative for systems under "realistic" loads.

It must be understood that studies like this are effectively always benchmarks of a combination of software (the actual MD packages, compilers, operating system, etc.) and hardware (CPUs, amount of memory and data paths to memory, network, I/O devices, etc.). This means that results can not automatically be transferred to other combinations as a change in a single component (typically the user has a choice of compilers) may affect the runtime. In practice, separate benchmark runs may be necessary to assess a different combination in detail but the results here will still be useful as a general guide.

In general, we have used compiler and compiler switches as recommend by the developers. However, this was not always possible as some software pack-

ages on some hardware platforms are not directly supported by the respective build system. In those cases we tried to use compilers and switches in analogy to other compiler/platform combinations. The compilers used may also not always be consistent across platforms (in some cases software precompiled by service administration was used). In one case we found that the recommended way of compilation resulted in a broken executable: GROMACS 4.5.x compiled on BlueGene/P with -O3 or higher lead to extreme values in pressure components (contrary to what was obtained on other platforms). The double precision version was found to be non-functional regardless of optimisation level.

The memory consumption on multi-core systems may pose some problems. NAMD was found to need more memory than available on the BlueGene/P with large systems[16] but we still have managed to run the 3 M system on a single core per node with NAMD 2.6. We now find, however, that the system is too big for versions 2.7 and 2.8. The 1.4 M system had to be run with no more than two cores per node. We also had to reduce the number of cores per node on HECToR from the maximum 32 to 24 with version 2.8. AMBER/pmemd 11 had similar problems with the 1.4 M and 3 M system on HECToR where we were also forced to reduce the cores per node to 24.

AMBER 11 does not support the extended pressure algorithm used for membrane systems available in the other program packages. Those algorithms were designed to compensate for deficiencies in present day lipid force fields. Thus, we simply applied isotropic pressure scaling for benchmarking. Tests performed for this report with NAMD have shown that differences in runtimes with available pressure algorithms (constant area, constant surface tension) is very small. The upcoming release of AMBER 12 (scheduled for early 2012 as of this writing) will include a constant surface tension algorithm.

The compatibility of runtime parameters between programs is a problem that may have serious impact on runtime performance, in particular PME parameters like grid size or maximum error. We have generally followed the guideline of either using a program's default value or use the parameter suggested in the manual (see input files in Appendix B). When comparing the benchmark results below this should be kept in mind.

Finally, we want to point out that here we only benchmark classical MD runs. More sophisticated methods like a particular free energy method or replica exchange method may scale very differently.

3 Results and discussion

3.1 AMBER ff03 vs. CHARMM22 force field

In our first report[1] we benchmarked AMBER 9 with the AMBER ff03 force field. The new version AMBER 11 investigated in this study fully supports CHARMM type force fields. The functional form of the those force fields requires three additional terms: improper torsions, Urey-Bradly cross-term couplings, and torsional backbone corrections (CMAP) which are numerical energies recorded on a point-wise map. For easy comparison we have decided to run all MD packages with the CHARMM22/27 force field. Hence, we tested if there would be any speed differences in the two force fields with AMBER 11.

As Figure 1 shows runtimes were essentially the same with variations between 2-12%. The additional terms in the CHARMM force field obviously do not lead to a very much increased burden on computational resources. The benchmarks were run on the Cray XE6 with the 61 k GlnBP system.

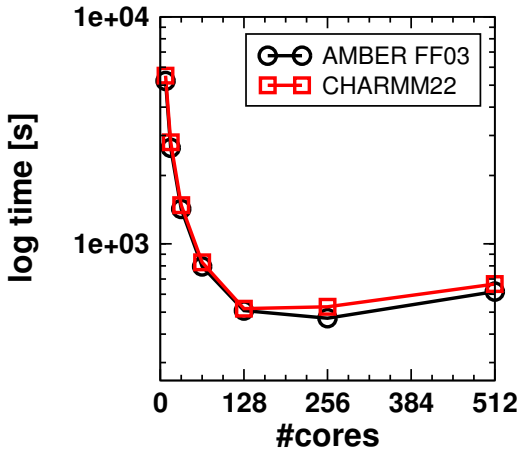

Fig. 1. A comparison of the AMBER ff03 force field with the CHARMM22 force field using AMBER 11/pmemd. The benchmarks were run on the Cray XE6 with the 61 k GlnBP systems.

3.2 CHARMM 35b5 vs. CHARMM 36b1

During 2011 the new version CHARMM 36b1 has been released. This version allows the user to set size parameters at runtime (at the beginning of the script) instead of static sizes at compile time. We have benchmarked both versions on the iDataPlex platform with the 20 k, 61 k and the 465 k systems.

Generally, CHARMM 36b1 is faster than the older version. The 465 k system for example is between 14 and 34% (32 processors) faster with 36b1. The

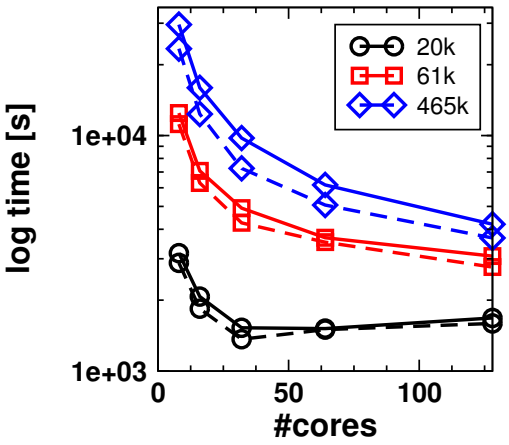

Fig. 2. A comparison of CHARMM 35b5 with the new CHARMM 36b1. The benchmarks were run on the iDataPlex with the 20 k, 61 k and 465 k systems. Solid lines: CHARMM 35b5, broken lines: CHARMM 36b1.

speedup gain in CHARMM 36b1 appears to be stronger in the larger systems while the scaling for each system size is similar. We note that the CHARMM program requires manual settings of PME parameters (see input file in Appendix B). We aimed at a grid resolution of about 1 Å and set κ to 0.34. The spline order was chosen to be 4. Runtimes benefitted from the lookup table feature up to 32 cores. CHARMM requires 2^n cores for PME calculations.

CHARMM is not directly supported on BlueGene/P and the Cray XE6 through the build system. A CHARMM 35 version was compiled for the Cray by the HECToR support team. The 36 version compiled analogously by us was not functional.

3.3 GROMACS — united atom vs. CHARMM 27 force field

In the two previous reports[1,2] we have used an united atom force field with GROMACS as the support for CHARMM type force field was still under development at that time. Here we have applied the CHARMM 22/27 force fields for the first time. We note that we have used the standard TIP3P water model as recommended by the helper program `pdb2gmx` and not CHARMM's modified version with additional Lennard-Jones potentials on the hydrogens. The latter was mentioned to be two times slower than TIP3P. We haven't tested this.

In Figure 3 we see that the CHARMM 27 force field is about three times slower than the united atom force field when comparing to the double precision version. We also include runtimes for the single precision version using the CHARMM 27 force field. This version is about two times faster than the double precision version. The benchmarks were carried out on the iDataPlex

with the 465 k system.

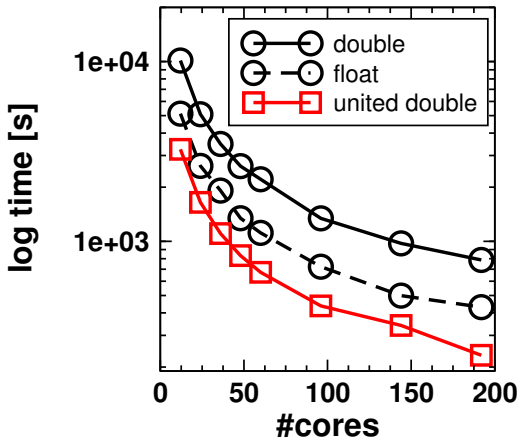

Fig. 3. A comparison of the united atom force field (see 1,2 for details) with the CHARMM 22/27 forcefield in GROMACS 4.5.5. The benchmarks were run on the iDataPlex with the 465 k system.

3.4 NAMD 2.7 vs. NAMD 2.8

The new version NAMD 2.8 has been compared with version 2.7. Figure 4 (left) gives an overview of runtimes obtained on the Cray XE6 with the 465 k systems. NAMD 2.8 was found to perform up to about 30% faster than the predecessor.

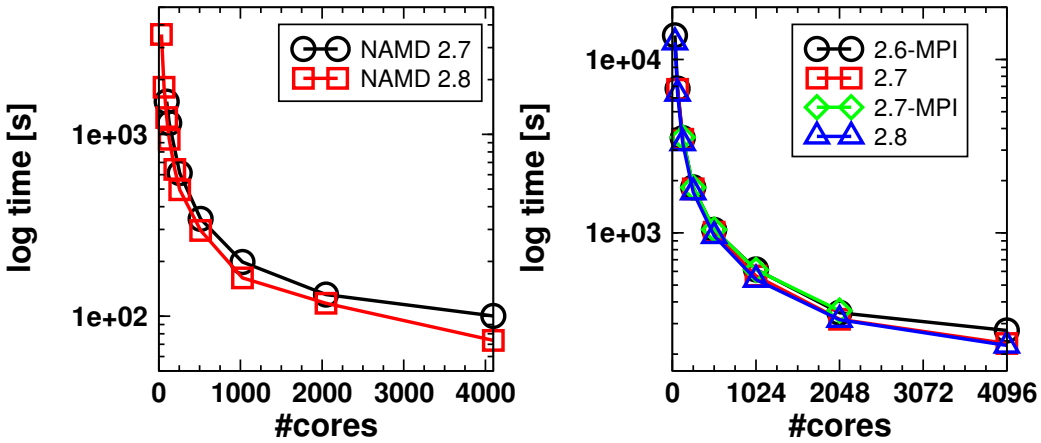

Fig. 4. Left: a comparison of NAMD 2.7 with NAMD 2.8 on the Cray XE6. Right: various versions of NAMD on the BlueGene/P. The benchmarks were run on the Cray XE6 with the 465 k system.

Also in Figure 4 (right) is shown a comparison of several NAMD versions on the BlueGene/P. Version 2.8 is about 5–20% faster than 2.6-MPI but we generally found that the MPI version is slower than the non-MPI version. 2.6-MPI and 2.7-MPI run at approximately the same speed. Versions 2.7 and 2.8

exhibit comparable runtimes too.

3.5 Overall results

All benchmarking results are graphically summarised in the Appendix in Figures A.1– A.3. We will focus here on general trends and outcomes and will only discuss individual results where necessary.

On the BlueGene/P (Figure A.1) we benchmarked GROMACS 4.5.5 float (expected to be about two times faster than the double precision version, see discussion in section 3.3), LAMMPS 27/10/2011 and NAMD 2.8. LAMMPS is slower than the other MD programs with a factor of up to 4.3 relative to GROMACS. This factor is only about 2.5 for the 465 k and 3 M systems and about 3 for the 1.4 M system. GROMACS is also faster than NAMD for the two smaller systems by a maximum factor of 1.6. The 465 k executes faster with NAMD while the 1.4 M systems are equally fast. Please note, that the number of cores per node was reduced from 4 to 2 for the 1.4 M system and the 3 M system failed execution with NAMD due to memory constraints.

For GROMACS we find that the number of cores reserved for PME (`mdrun` flag -npme) should be typically 12.5% for maximum performance. The number of reserved cores is usually found automatically by `mdrun`. For higher core counts this automatic mechanism failed and resulted in terminating the program in many cases because the domain decomposition could not be carried out with the given box dimensions and the minimum cell size. We then manually tried to adjust the number of PME nodes which in many cases was successful (see raw data for details). It is arguable, however, if production runs should be carried out at those high core counts as the increase in performance is typically negligible. In the case of the 21 k system and core counts 8–32 best performance was achieved if no cores were reserved for PME.

We assessed runtime performance on HECToR (Cray XE6) in December 2012 just right after the hardware upgrade (see Figure A.2). All five MD packages were benchmarked on this platform but we were able to run CHARMM 35b5 only for the two smallest system sizes. CHARMM is the slowest program being about 4–12 times slower than GROMACS 4.5.5 with the 21 k system and about 3–20 times slower for the 61 k system. Also, CHARMM does not scale well with increased core counts. AMBER 11/pmemd fares well, especially with the smaller system sizes, and may even outperform or be on par with the other MD programs on lower core counts regardless of system size but pmemd scales less well than GROMACS, LAMMPS or NAMD. LAMMPS and NAMD appear to have problems with smaller system sizes where performance is comparably bad. GROMACS does well with any system size. Scaling behaviour

with GROMACS, LAMMPS or NAMD is very similar and also runtimes on the larger systems are very close to each other with LAMMPS being only less than about 2.5 times slower than GROMACS.

The number of separate PME nodes that should be chosen in GROMACS are zero for smaller core counts. On the larger systems a good ratio was 12.5% for larger core counts. The 465 k system performed best at core counts of 256 and 512 with 6.25% cores reserved for PME. The smaller systems performed best with 25% for the largest core counts. We typically found that load imbalance becomes a problem at the highest core counts.

On the iDataPlex (see Figure A.3) we again found CHARMM 35b5 to be the slowest MD program but less dramatic than on the Cray XE6. CHARMM is slower than GROMACS by only up to 4 times. GROMACS, LAMMPS and NAMD are nearly equally fast with comparable scaling behaviour. LAMMPS is slower than GROMACS by a factor of less than 2. GROMACS was always the fastest except with the 465 k system where it was slower than NAMD and LAMMPS.

For the two smallest system best performance with GROMACS was achieved when no separate PME nodes were reserved. For the three larger systems this was true for 12–48 cores. Larger number of cores required an increasing number of reserved PME nodes from about 8–12%. The automatic assignment of this number works reasonable well on all three hardware platforms but hand tweaking may still boost performance.

3.6 Peak Performance and size dependence

In table 3 we summarise "peak" performance measured in nanoseconds per day (cmp. Figures A.1–A.3). As it was not always possible to benchmark to sufficiently high core counts values had to be taken from the fastest observed runtime. This is especially true in the case of the iDataPlex where only 192 cores were available and with GROMACS where choosing the number of reserved PME nodes failed at high core counts. In those cases, however, the reported nanoseconds per day are still a useful indicator of the performance characteristics.

Table 3 gives some insights into the runtime performance as a function of the system size. It must be noted, however, that the simulation systems are not homogeneous with respect to the force field. The potential function for the water TIP3P model is distinct from the protein/lipid potential function in that there is only one van der Waals centre on the oxygen (unless the CHARMM's TIPS water model is used) and, naturally, there are no dihedrals or similar terms to be considered. Also, TIP3P water is made rigid through appropriate

Table 3
Peak performance in ns/day. Please note that some values are not derived from actually observed peak performance as indicated due to reasons discussed in the main text.

platform	MD program	20 k	61 k	465 k	1.4 M	3 M
BlueGene/P	GROMACS float[a]	44.3	21.6	14.8	8.3	3.9
	LAMMPS	9.2	7.4	5.8[a]	2.7[a]	1.4[a]
	NAMD	24.1	14.3	15.4[a]	4.6[a]	—
Cray XE6	AMBER	37.9	16.7	3.8	1.7	1.0
	CHARMM	3.5	1.5	—	—	—
	GROMACS double	109.2[a]	60.9[a]	45.3[a]	33.6	21.5[a]
	LAMMPS	33.9	20.4	21.4	15.3[a]	10.7[a]
	NAMD	25.3[a]	38.5	47.0[a]	18.6	17.1
iDataPlex[a]	CHARMM	5.7	2.8	0.8	—	—
	GROMACS double	77.1	25.6	4.4	2.9	1.2
	LAMMPS	37.3	16.4	4.2	1.7	0.8
	NAMD	60.7	23.2	6.4	2.5	1.2

[a] Not actual peak performance: value taken from fastest observed runtime.

constraint algorithms. It may be possible to devise specific optimisations for this water model that may not be possible for the biopolymer force field. If there are any implementation differences in the MD simulation package affecting the performance this will necessarily show up.

Figure 5 makes an attempt at demonstrating the runtime dependence as a function of total atom numbers. The water atoms to total atoms ratios are (cmp. table 1) 97% for 20 k, 94% for 61 k, 66% for 465 k, 80% for 1.4 M, and 68% for 3 M. As the inhomogeneity in the force fields as discussed above will have an affect on the scaling behaviour (see also table 3 and Figures A.1–A.3) a true size dependence cannot be established in this way. Changing the water to total atom number ratio will therefore influence runtimes and size scaling.

Acknowledgement. We acknowledge support by HECToR, by the CSED of STFC for access to its IBM Blue Gene/P and iDataPlex systems. We are

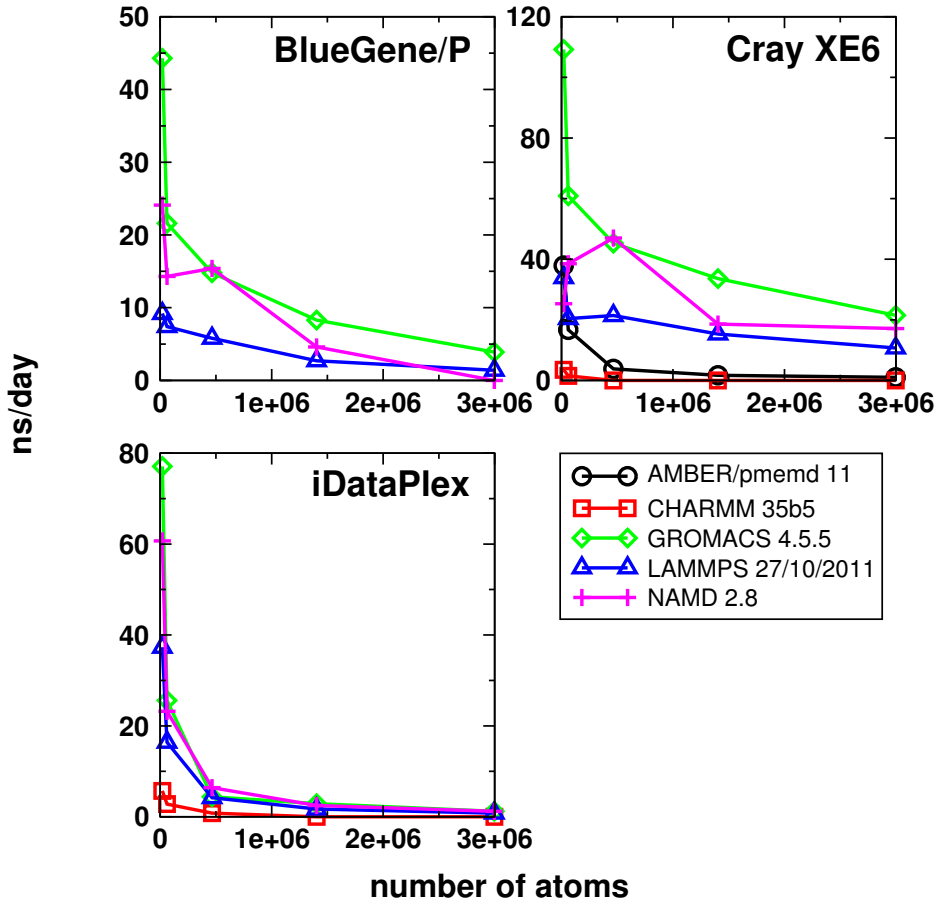

Fig. 5. Runtime dependence on the total number of atoms. Please see main text for critical discussion. Graphs have been created based on "peak" performance in table 3.

especially grateful to the user support teams of these systems.

References

[1] H. H. Loeffler, M. D. Winn, Large biomolecular simulation on HPC Platforms I. Experiences with AMBER, Gromacs and NAMD, Tech. Rep. DL-TR-2009-002, STFC Daresbury Laboratory, Warrington WA4 4AD, UK (2009).
URL http://epubs.stfc.ac.uk/work-details?w=50963

[2] H. H. Loeffler, M. D. Winn, Large biomolecular simulation on HPC platforms
II. Gromacs, LAMMPS and NAMD, Tech. rep., STFC Daresbury Laboratory, Warrington WA4 4AD, UK (2010).
URL http://www.stfc.ac.uk/CSE/randd/cbg/Benchmark/25241.aspx

[3] R. Schulz, B. Lindner, L. Petridis, J. C. Smith, Scaling of multimillion-atom biological molecular dynamics simulation on a petascale supercomputer, Journal of Chemical Theory and Computation 5 (10) (2009) 2798–2808.
URL http://pubs.acs.org/doi/abs/10.1021/ct900292r

[4] C. Mei, Y. Sun, G. Zheng, E. J. Bohm, L. V. Kale, J. C.Phillips, C. Harrison, Enabling and scaling biomolecular simulations of 100 million atoms on petascale machines with a multicore-optimized message-driven runtime, in: Proceedings of the 2011 ACM/IEEE Conference on Supercomputing (SC11), 2011.

[5] H. H. Loeffler, A. Kitao, Collective dynamics of periplasmic glutamine binding protein upon domain closure, Biophys. J. 97 (9) (2009) 2541–2549.
URL http://www.sciencedirect.com/science/article/pii/S0006349509013757

[6] J. Kästner, H. H. Loeffler, S. K. Roberts, M. L. Martin-Fernandez, M. D. Winn, Ectodomain orientation, conformational plasticity and oligomerization of ErbB1 receptors investigated by molecular dynamics, J. Struct. Biol. 167 (2) (2009) 117 – 128.

[7] C. J. Tynan, S. K. Roberts, D. J. Rolfe, D. T. Clarke, H. H. Loeffler, J. Kastner, M. D. Winn, P. J. Parker, M. L. Martin-Fernandez, Human epidermal growth factor receptor (egfr) aligned on the plasma membrane adopts key features of drosophila egfr asymmetry, Mol. Cell. Biol. 31 (11) (2011) 2241–2252.
URL http://mcb.asm.org/cgi/content/abstract/31/11/2241

[8] A. D. MacKerell, D. Bashford, Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, M. Karplus, All-atom empirical potential for molecular modeling and dynamics studies of proteins, J. Phys. Chem. B 102 (18) (1998) 3586–3616.

URL http://pubs.acs.org/doi/abs/10.1021/jp973084f

[9] A. D. Mackerell, M. Feig, C. L. Brooks, Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations, J. Comput. Chem. 25 (11) (2004) 1400–1415.
URL http://dx.doi.org/10.1002/jcc.20065

[10] D. Beglov, B. Roux, Finite representation of an infinite bulk system: Solvent boundary potential for computer simulations, J. Chem. Phys. 100 (12) (1994) 9050–9063.
URL http://link.aip.org/link/JCPSA6/v100/i12/p9050/s1

[11] S. E. Feller, A. D. MacKerell, An improved empirical potential energy function for molecular simulations of phospholipids, J. Phys. Chem. B 104 (31) (2000) 7510–7515.
URL http://pubs.acs.org/doi/abs/10.1021/jp0007843

[12] S. E. Feller, K. Gawrisch, A. D. MacKerell, Polyunsaturated fatty acids in lipid bilayers: Intrinsic and environmental contributions to their unique physical properties, J. Am. Chem. Soc. 124 (2) (2002) 318–326, pMID: 11782184.
URL http://pubs.acs.org/doi/abs/10.1021/ja0118340

[13] M. Schlenkrich, J. Brickmann, A. D. MacKerell, Jr., M. Karplus, An empirical potential energy function for phospholipids: Criteria for parameter optimization and applications, in: K. M. Merz, Jr., B. Roux (Eds.), Biological Membranes: A Molecular Perspective from Computation and Experiment, Birkhäuser, Boston, 1996, pp. 31–81.

[14] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, Comparison of simple potential functions for simulating liquid water, J. Chem. Phys. 79 (2) (1983) 926–935.
URL http://link.aip.org/link/?JCP/79/926/1

[15] M. F. Crowley, M. J. Williamson, R. C. Walker, CHAMBER: Comprehensive support for CHARMM force fields within the AMBER software, Int. J. Quant. Chem. 109 (15) (2009) 3767–3772.
URL http://dx.doi.org/10.1002/qua.22372

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

Appendices

A Graphical Results Summary

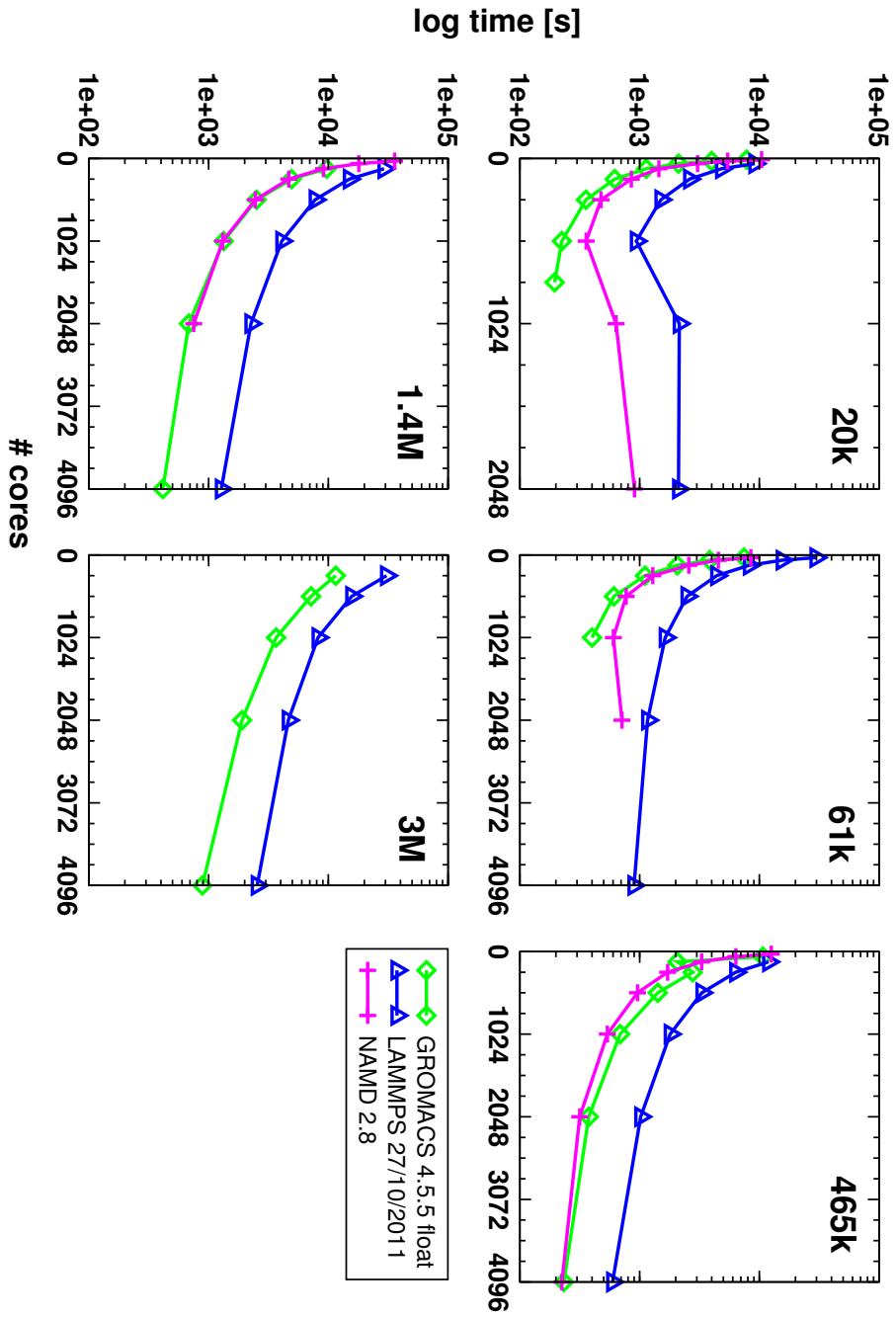

Fig. A.1. All benchmark results for the five system sizes benchmarked with GROMACS float, LAMMPS and NAMD on the BlueGene/P.

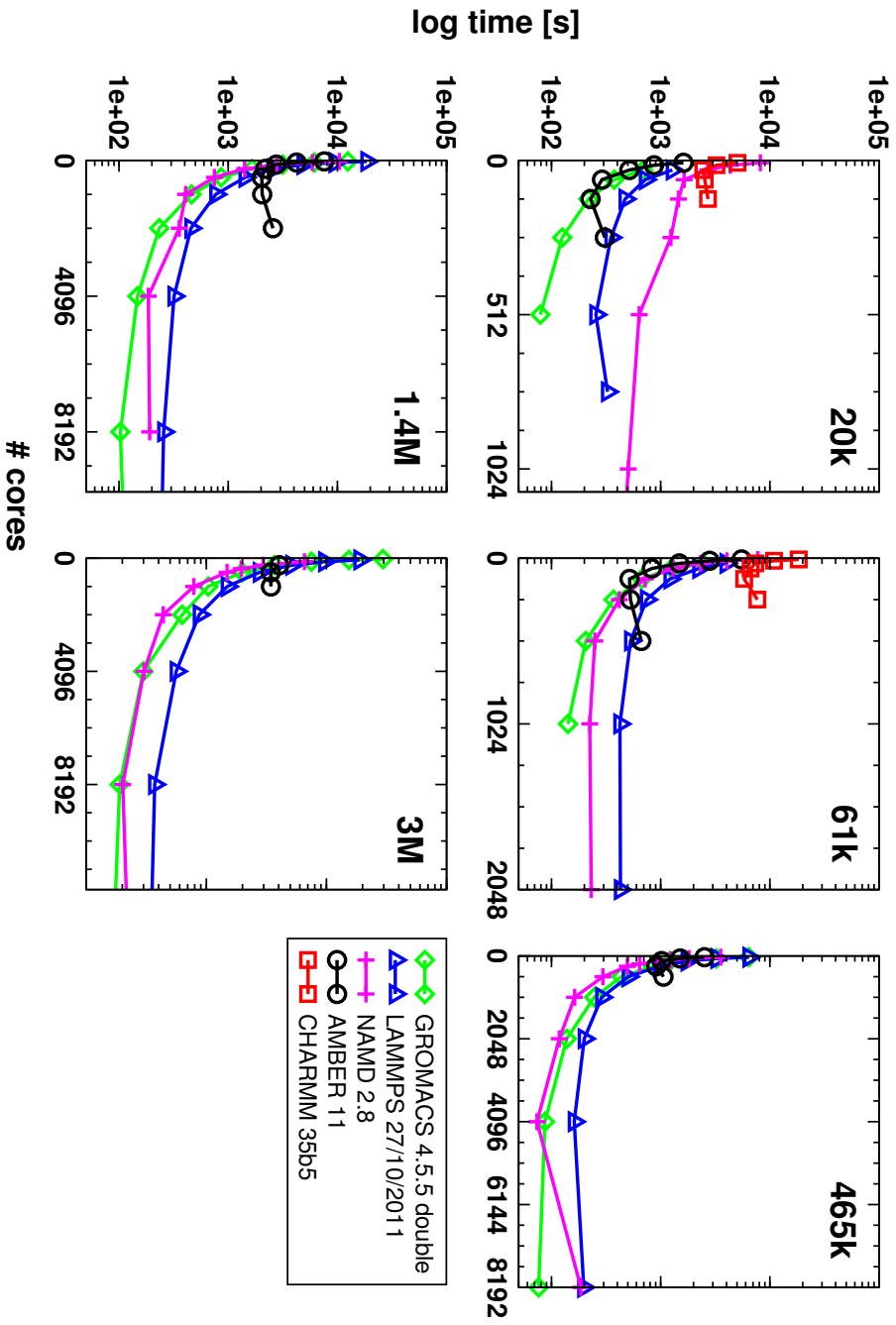

Fig. A.2. All benchmark results for the five system sizes benchmarked with AMBER, CHARMM, GROMACS double, LAMMPS and NAMD on the Cray XE6.

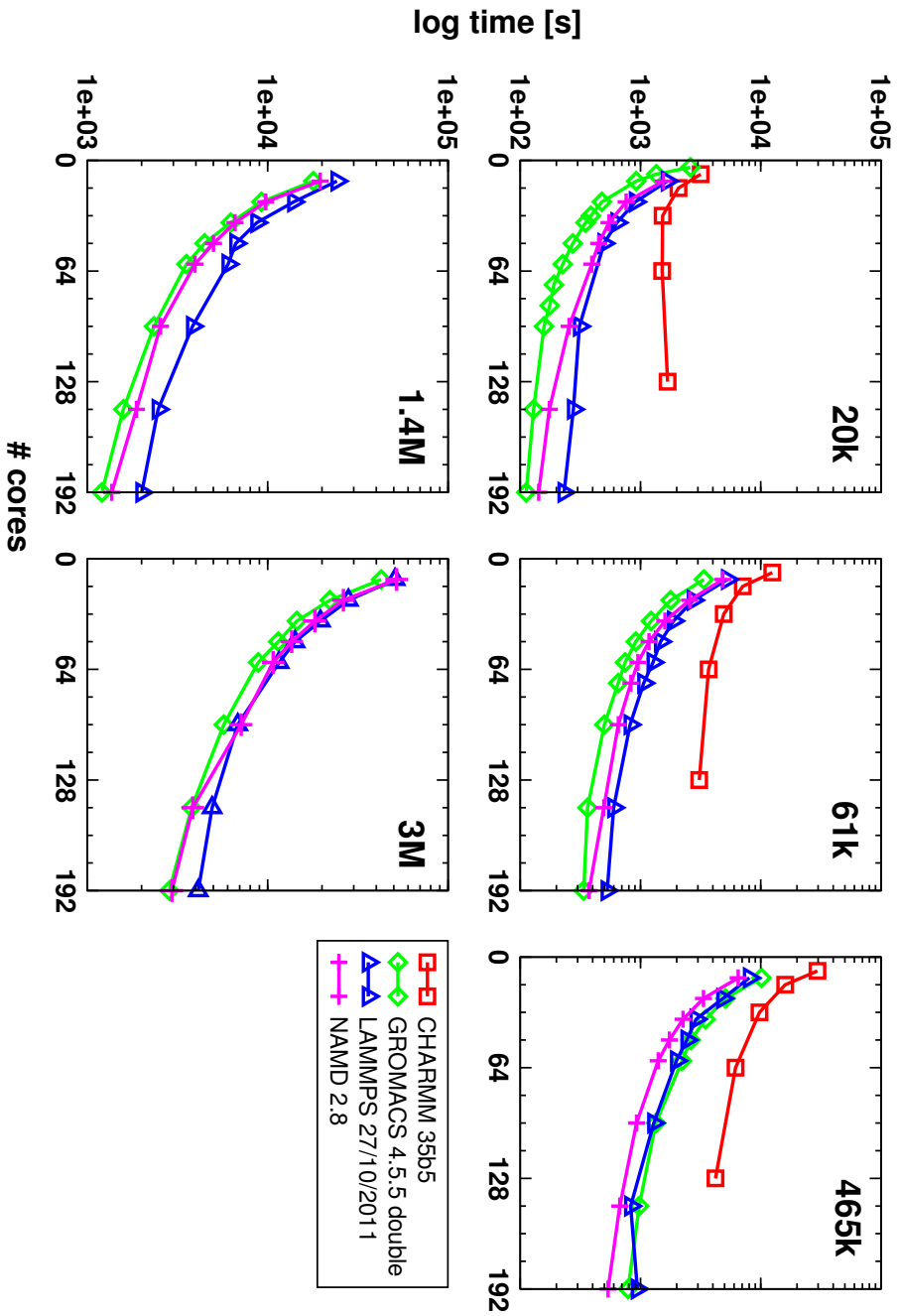

Fig. A.3. All benchmark results for the five system sizes benchmarked with CHARMM, GROMACS double, LAMMPS and NAMD on the iDataPlex.

B Input files

This is a summary of input files for the 3M benchmark studies. All input files including the raw results and the full benchmark suite will be available through `http://www.stfc.ac.uk/CSE/randd/cbg/Benchmark/25241.aspx`.

(1) Amber/pmemd input file

```
3M benchmark

 &cntrl
   imin = 0, irest = 1, ntx = 5,
   nstlim = 10000, dt = 0.002,
   ntt = 1, temp0 = 300.0, tautp = 1.0,
   ntb = 2, ntp = 1, pres0 = 1.0, taup = 2.0,
   ntc = 2, ntf = 2,
   ntpr = 1000, ntwr = 1000, ntwe = 1000, ntwx = 5000,
   ioutfm = 1, iwrap = 0
 /
 &ewald
 /
```

(2) CHARMM 36 input file (will not work with earlier versions!)

```
* bench.inp
* $ charmm job=name < bench.inp > name.log

dimension chsize 4000000 maxt 3000000 maxp 3000000 maximp 200000 -
  maxpad 2000000 maxseg 200 maxgrp 2000000 maxcrt 20000

bomlev -2

read rtf card name "top_all27_prot_lipid.rtf"
read param card name "par_all27_prot_lipid.prm"

read psf card name "bench.psf"
read coor card name "bench.crd"

open unit 20 read form name bench.rst

open unit 10 write form name @job.rst
open unit 11 write unform name @job.dcd
open unit 12 write form name @job.en

crystal define orthorhombic 438.0 452.0 146.3 90.0 90.0 90.0
```

```
crystal build cutoff 14.0 noper 0

image byseg select segid A : P end
image byres select resname POPC end
image byres select resname TIP3 end
image byatoms select segid R end

faster on
shake fast bonh tol 1.0e-6 para

prnlev 2 node 0

energy vdw vshift bycbim -
  ewald pmew kappa 0.34 spline order 4 fftx 400 ffty 432 fftz 144 qc
lookup noenergy interpolation tabincrement 20 select resname TIP3 en

dynamics cpt leap restart nstep 10000 timestep 0.002 -
  tconst treference 300.0 tcouple 5.0 -
  pconst preference 1.0 pmzz 500.0 pmxx 0.0 pmyy 0.0 -
  iunwri 10 iuncrd 11 kunit 12 iunrea 20 -
  nsavc 5000 isvfrq 5000 nprint 1000 iprfrq 10000 ntrfrq 2500 inbfrq
  imgfrq -1 echeck -1.0 -

write coord card name @job.crd

stop
```

(3) GROMACS `.mdp` file

```
integrator              = md
tinit                   = 0.0
dt                      = 0.002
nsteps                  = 10000
nstcomm                 = 1000

nstxout                 = 5000
nstvout                 = 0
nstfout                 = 0
nstlog                  = 1000
nstenergy               = 1000
nstxtcout               = 0

nstlist                 = 10
ns_type                 = grid
pbc                     = xyz
```

```
rlist                    = 1.4

coulombtype              = PME
rcoulomb                 = 1.4
fourierspacing           = 0.12
pme_order                = 4
ewald_rtol               = 1.0E-5
optimize_fft             = no

vdwtype                  = Shift
rvdw_switch              = 1.0
rvdw                     = 1.2
DispCorr                 = no

tcoupl                   = Berendsen
tc_grps                  = System
tau_t                    = 1.0
ref_t                    = 300.0
gen_vel                  = no

pcoupl                   = Berendsen
pcoupltype               = semi-isotropic
tau_p                    = 5.0  1.0
compressibility          = 0.0  4.5e-5
ref_p                    = 0.0  1.0

constraints              = hbonds
constraint_algorithm     = Lincs
continuation             = yes
```

(4) LAMMPS input file

```
variable        T equal 300.0
variable        p equal   1.0
variable        Tdamp equal  100.0
variable        pdamp equal 1000.0

variable        new string "@LAMMPS_JOBNAME@"

units           real
neigh_modify    delay 0 every 1 check yes

atom_style      full
bond_style      harmonic
angle_style     charmm
```

```
dihedral_style   charmm
improper_style   harmonic

pair_style       lj/charmm/coul/long/opt 10 12
pair_modify      mix arithmetic
kspace_style     pppm 1e-4

read_data        bench.data

special_bonds    charmm
fix              1 all shake 1e-6 500 0 m 1.0 b 85 a 184
fix              2 all npt temp $T $T ${Tdamp} z $p $p ${pdamp}

thermo           1000
thermo_style     multi
thermo_modify    flush yes
timestep         2.0

restart          1000 ${new}.rst
dump             1 all dcd 5000 ${new}.dcd
dump_modify      1 unwrap yes

run              10000
```

(5) NAMD config file

```
set p          1.0
set T          300.0
set num_steps  10000

set new        $env(NAMD_JOBNAME)
set previous   relres

structure          ionized.psf
paratypecharmm     on
parameters         par_all27_prot_lipid.prm
coordinates        ionized.pdb
exclude            scaled1-4
1-4scaling         1.0

binvelocities        $previous.vel
bincoordinates       $previous.coor
ExtendedSystem       $previous.xsc
firsttimestep        0
```

```
langevin                on
langevinHydrogen        on
langevinTemp            $T
langevinDamping         1

useGroupPressure        yes
useFlexibleCell         yes
useConstantArea         yes
LangevinPiston          on
LangevinPistonTarget    $p
LangevinPistonPeriod    200
LangevinPistonDecay     100
LangevinPistonTemp      $T

outputname              $new
outputEnergies          1000
outputPressure          1000
outputTiming            10000
binaryoutput            yes

restartname             $new
restartfreq             1000
binaryrestart           yes
DCDfile                 $new.dcd
DCDUnitCell             yes
DCDfreq                 5000
XSTfile                 $new.xst
XSTfreq                 1000

PME                     on
PMEGridSpacing          1.0

nonbondedFreq           1
fullElectFrequency      1

switchdist              10
cutoff                  12
switching               on
pairlistdist            14
pairlistsPerCycle       1
pairlistMinProcs        128
stepspercycle           20

useSettle               on
```

```
rigidBonds          all
rigidIterations     100

wrapAll             on
wrapNearest         on

timestep 2.0
run $num_steps
```