Scaling of the GROMACS 4.6 molecular dynamics code on SuperMUC

Carsten KUTZNER [a,1], Rossen APOSTOLOV [b,c], Berk HESS [b], and Helmut GRUBMÜLLER [a]

[a] *Max Planck Institute for Biophysical Chemistry, Theoretical and Computational Biophysics Dept., Am Fassberg 11, 37073 Göttingen, Germany.*

[b] *Theoretical and Computational Biophysics, Dept. Theoretical Physics, KTH Royal Institute of Technology, 10691 Stockholm, Sweden*

[c] *PDC Center for High Performance Computing, KTH Royal Institute of Technology, 10044 Stockholm, Sweden*

Abstract. Here we report on the performance of GROMACS 4.6 on the SuperMUC cluster at the Leibniz Rechenzentrum in Garching. We carried out benchmarks with three biomolecular systems consisting of eighty thousand to twelve million atoms in a strong scaling test each. The twelve million atom simulation system reached a performance of 49 nanoseconds per day on 32,768 cores.

Keywords. GROMACS, molecular dynamics, SuperMUC, benchmark, biomolecular simulation

1. Introduction

Proteins and protein complexes are biology's nanomachines, performing the vast majority of all cellular functions. Molecular dynamics (MD) simulations of biomolecular systems are used to study the mechanisms underlying their biological function on the atomic level. Individual proteins and other biomolecules such as DNA, as well as assemblies consisting of multiple molecules and membranes [1], up to whole virus particles [2] and subcellular organelles, are currently accessible. Whereas the current simulation state of the art for typical MD systems lies in the microsecond range, many biological processes happen on timescales of milliseconds or slower [3]. To extend the simulation time spans, MD software steadily evolves to fully exploit the capabilities of the available hardware.

The GROMACS [4] simulation package is developed for optimal absolute performance and is one of the fastest MD engines worldwide. To efficiently distribute the calculations over a parallel machine, each GROMACS process can spawn several OpenMP threads, whereas the processes communicate using the Message Passing Interface (MPI) standard. However, what makes the scaling to a large number of processors particularly challenging is the long-range nature of the electrostatic forces. Nowadays Particle mesh Ewald (PME) [5] is the established method to evaluate these forces. PME separates the Coulomb potential into short-range contributions, which are calculated efficiently in real

[1]Corresponding Author E-mail: ckutzne@gwdg.de

space, and long-range contributions, which are calculated efficiently in reciprocal space. For the reciprocal space part, a Fourier transformation of the charge density is needed; by extrapolating the continuous charge positions on a grid, PME can make use of the fast Fourier transformation (FFT) for that. In parallel, each process has a slab of the charge grid and an all-to-all communication is required for the complete 3D FFT. For large numbers of processors this communication pattern is a scaling bottleneck [6]: N processors have to exchange N^2 messages. GROMACS computes the reciprocal space part of the Coulomb forces on a subset N_{rec} of the available processors, typically $N_{rec} = N/4$, thereby significantly reducing the number of messages, typically to 1/16, sent during all-to-all communication [7].

Parallelization of the direct space interactions is obtained by domain decomposition: the simulation system is divided into $N_{dir} = n_x \times n_y \times n_z$ domains, each assigned to one MPI process. Depending on the size of the cutoffs, some boundary volume has to be communicated between the domains at every time step. To attain optimal parallel performance, GROMACS offers three mechanisms that balance the load between the available processors:

1. The number N_{rec} of PME long-range processes can be chosen with respect to the remaining processes N_{dir}.
2. Computational load can be shifted between the direct and the reciprocal space parts of PME by balancing the Coulomb cutoff against the density of the charge grid.
3. The computational load between the direct space processes can be balanced by adjusting the domain boundaries, which is needed when the interaction density across the MD system is inhomogeneous such as for a membrane + protein + water system.

Based on cutoff and PME grid settings, GROMACS estimates the fraction of long-range processes N_{rec} to be used for the simulation run, but that does not take into account hardware characteristics or the type of interconnect used. To find the optimum of N_{rec} on a specific machine for a specific number of processes N, the `g_tune_pme` tool can be used. This tool tries $N_{rec} : N_{dir}$ values around the estimated value in small benchmark runs and reports the ratio with the highest performance.

GROMACS 4.6 automatically shifts load between PME real and reciprocal parts if needed and also dynamically adjusts the domain boundaries if the domains have uneven load. However, since the balancing mechanisms two and three (see above) need several tens to a few hundred time steps to arrive at the optimal setting, one has to exclude these initial steps from performance measurements.

This report summarizes the results of GROMACS 4.6 benchmarks carried out during the "SuperMUC Extreme Scaling Workshop" at the Leibniz-Rechenzentrum (LRZ) in July 2013. SuperMUC is a cluster of 18,432 Intel Xeon E-2680 "Sandy Bridge" CPUs with eight cores each operating at a clock rate of 2.7 GHz. The cluster is organized in 9,216 nodes, each with two CPUs and 32 GB of memory; 512 nodes each are grouped together to an "island." Within an island the nodes are connected by a fully non-blocking FDR10 Infiniband network, whereas for communication across islands four nodes each share one link to the spine switches [8].

Table 1. Specifications of the benchmark systems. Values marked with an asterisk * are subject to automatic adjustments for better load balance.

MD system	Aquaporin	Ribosome	Peptides
# atoms	81,743	2,136,412	12,495,503
system size / nm	$10.8\times10.2\times9.6$	$31.2\times31.2\times31.2$	$50.0\times50.0\times50.0$
time step / fs	2	4	2
cutoff radii / nm	1.0^*	1.0^*	1.2^*
PME grid spacing / nm	0.120^*	0.135^*	0.160^*
neighbor searching frequency	10	25	40
benchmark steps	10,000	2,000	1,000

2. Benchmarks setup

GROMACS 4.6 was compiled in two alternative configurations: (a) IBM MPI 1.3 was used in combination with the Intel 12.1.6 compiler, and (b) Intel MPI 4.1 was used together with the Intel 13.1.1 compiler. In both cases, GROMACS was compiled in single precision with `-O3 -mavx` compiler flags and linked against the FFTW 3.3.2 library, which was compiled with GCC 4.7.2 and `--enable-sse2`.

We used three "real world" benchmark examples, which were taken from past or ongoing research projects, see Table 1 for their properties. The aquaporin channel [1] is with about 81 k atoms of a typical size for a biomolecular simulation system. It consists of an aquaporin-1 tetramer embedded in a lipid bilayer surrounded by water. The *Escherichia coli* ribosome in water [9] comprises over 2 M atoms. The largest system consists of 250 steric zipper peptides in water and was used for the study of peptide aggregation [10]. The two large systems additionally contain ions (mainly sodium and chloride) at physiological concentration. All systems use periodic boundary conditions, PME electrostatics, and the Verlet cutoff scheme.

The goal of the benchmarks was to find the highest performance in terms of simulated nanoseconds per wallclock time for any number of nodes. We varied the number of MPI processes per node (32, 16, 8, 4, 2, 1) and the number of OpenMP threads (1, 2, 4, 8, 16) per MPI process. For IBM MPI, threads were pinned manually to cores by setting `MP_TASK_AFFINITY=core:$OMP_NUM_THREADS`. For Intel MPI, we set `I_MPI_PIN_DOMAIN=auto` and `I_MPI_PIN_CELL=unit`. For an additional minor performance benefit, we also set `MP_BULK_MIN_MSG_SIZE=32768` for IBM MPI and `I_MPI_DAPL_DIRECT_COPY_THRESHOLD=262114` for Intel MPI. All benchmarks were carried out at a clock rate of 2.7 GHz. `g_tune_pme` was used to derive the optimum number of long-range processes N_{rec}. The number of time steps performed for each individual benchmark run is given in Table 1 depending on the system. The first half of the time steps of each benchmark was excluded from the performance measurements to allow for a proper load balance. Note that in the load balancing process between PME real and reciprocal space, the initial values (marked with * in Table 1) of the Coulomb cutoff and PME grid spacing were automatically slightly adjusted for optimal performance.

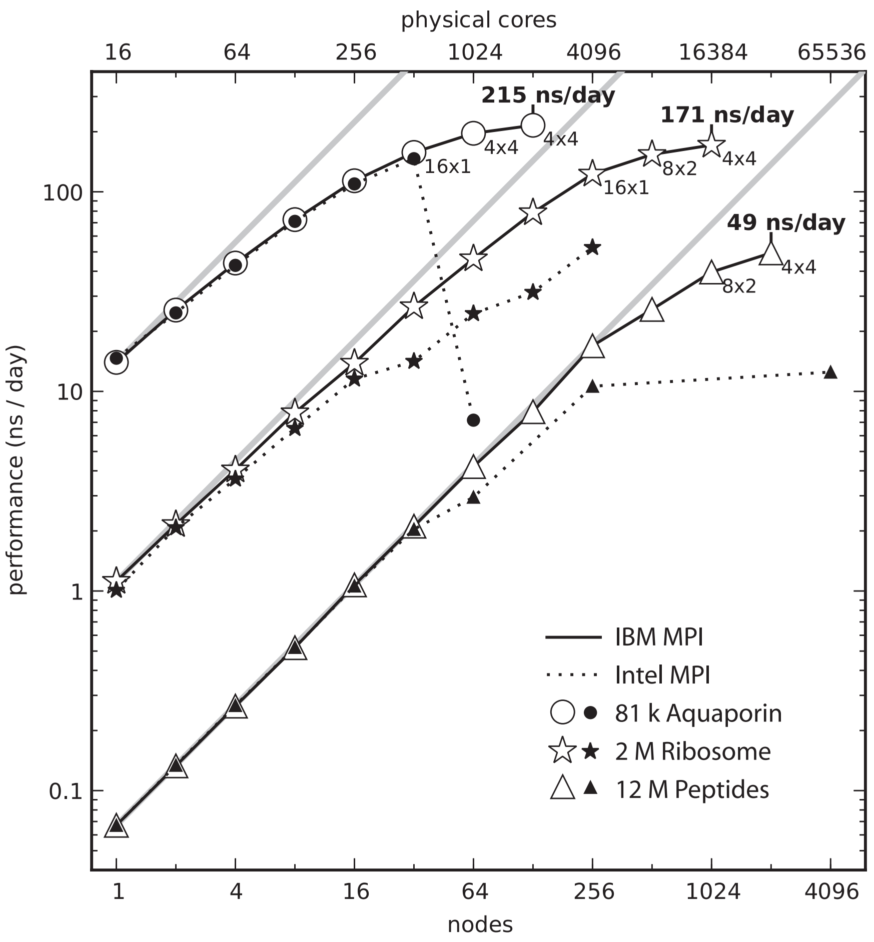

Figure 1. Performance of GROMACS 4.6 on the SuperMUC cluster, using typical benchmark systems of 81 k atoms (circles), 2 M atoms (stars) and 12 M atoms (triangles) in size. Grey lines show perfect scaling. Solid lines with white symbols denote IBM MPI, dotted lines with black symbols Intel MPI. "8×2" indicates that eight MPI processes with two threads each were used per node. The performance degradation of Intel MPI for large numbers of cores could stem from a suboptimal Intel MPI configuration on this cluster and is under investigation.

3. Results

The largest benchmark we ran during the scaling workshop was on 65,536 SuperMUC cores (eight islands) using the 12 M atom MD system. For this MD system, a maximum absolute simulation performance of 49 ns/day was reached on 32,768 cores at 223 TFLOP/s.

Figure 1 summarizes the benchmark results for the three strong scaling tests. Performance (ns/day) is shown for the fastest observed setting, i.e. the most favourable number of long-range processes N_{rec} as well as MPI processes $\times$ OpenMP threads per node. Even though GROMACS estimates N_{rec} and automatically fine-tunes the load between direct and reciprocal space processors, in about a fifth of the cases g_tune_pme could slightly

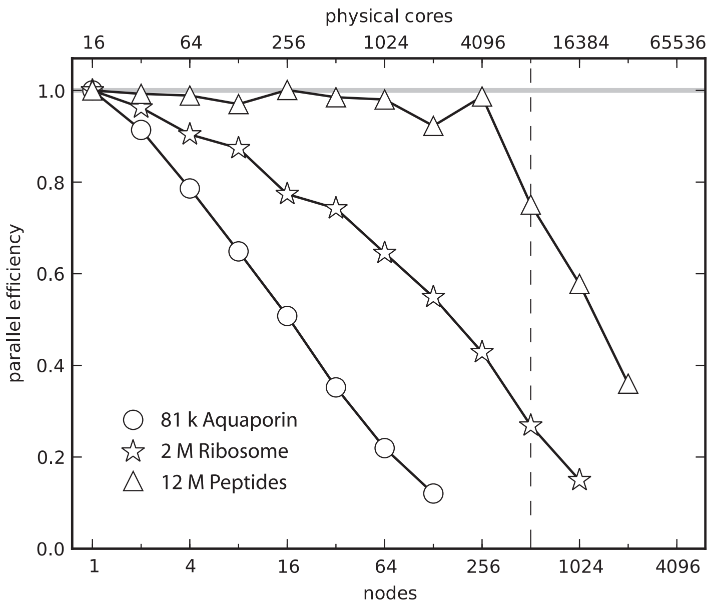

Figure 2. Efficiency of the benchmarks from Figure 1 using IBM's MPI library. Horizontal grey line indicates perfect scaling; left of the dashed vertical line all used nodes were from a single island.

improve on that and found an N_{rec} that yielded about 5–15 % higher performance. Almost always 16 MPI processes per node with two OpenMP threads each were optimal, thereby exploiting the CPU's hyperthreading capabilities. Only at high parallelization other combinations were better, as indicated in the figure. Near the parallelization limit, a smaller number of MPI processes per node but with more threads each is favoured, thereby keeping the total number of MPI processes within a limit.

Note that at the performance maximum of the aquaporin benchmark (on 2,048 cores), each processor core on average has about 40 atoms assigned and more than 1,000 time steps are calculated per second. For the 12 M atom system, more than 250 steps are done per second on 32,768 cores.

For up to 16–32 nodes, performance hardly depends on the employed MPI library, whereas at higher parallelization it is drastically reduced when using Intel MPI. This performance degradation might be overcome in the near future by properly adjusting the cluster's Intel MPI configuration.

Figure 2 shows the parallel efficiency for the IBM MPI cases, calculated as the performance on *n* nodes divided by *n* times the performance on a single node. As expected, the efficiency is higher for larger simulation systems. The 12 M atom system exhibits a nearly perfect scaling on up to 4,096 cores and even across multiple islands we can still observe decent scaling. For the simulations at the parallelization limit (the last two to three points of each curve), the reciprocal processor group needs considerably more time (up to a factor of 2.5) than the direct space processor group. Most of the time is spent in all-to-all communication and thus cannot be reduced by employing more processors.

4. Summary

During the "SuperMUC Extreme Scaling Workshop" at the Leibniz-Rechenzentrum, we carried out a 12 M atom biomolecular simulation using 65,536 cores (eight islands) of SuperMUC with the GROMACS 4.6 MD software package. On four islands a simulation performance of 49 ns/day was reached, i.e. more than 250 integration time steps were executed each second. This MD system showed a nearly optimal parallel efficiency as long as nodes from only a single island were used. One of the main bottlenecks at high parallelization is the all-to-all communication pattern required for the calculation of the long-range Coulomb forces with PME. To efficiently run a biomolecular simulation with realistic parameters on the whole machine (18 islands), an even larger MD system consisting of significantly more than twelve million particles will be required.

Acknowledgments

This study was done in collaboration with IBM and Intel and was supported by the ScalaLife EU FP7 project (contract INFSO-RI-261523), by the DFG priority programme "Software for Exascale Computing" (SPP 1648), and by European Research Council grant nr. 258980. The authors thank F. Merz (IBM) as well as H. Bockhorst and K.-D. Oertel (Intel) for their kind help concerning MPI settings and parallel performance analysis tools. We further thank P. Kellers for reading the manuscript and F. Jamitzky, N. Hammer, M. Brehm and C. Bernau for sharing their experiences with GROMACS on the SuperMUC cluster and for the organization of the scaling workshop at the Leibniz Rechenzentrum.

References

[1] B.L. de Groot and H. Grubmüller. Water permeation across biological membranes: mechanism and dynamics of Aquaporin-1 and GlpF. *Science*, 2001.

[2] G. Zhao, J.R. Perilla, E.L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A.M. Gronenborn, K. Schulten, C. Aiken, and P. Zhang. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. *Nature*, 2013.

[3] R.O. Dror, R.M. Dirks, J.P. Grossman, H. Xu, and D.E. Shaw. Biomolecular Simulation: A computational microscope for molecular biology. *Annu. Rev. Biophys.*, 2012.

[4] S. Pronk, S. Pall, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M.R. Shirts, J.C. Smith, P.M. Kasson, D. van der Spoel, B. Hess, and E. Lindahl. GROMACS 4.5: A high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics*, 2013.

[5] U. Essmann, L. Perera, M. Berkowitz, T. Darden, and H. Lee. A smooth particle mesh Ewald method. *J. Chem. Phys.*, 1995.

[6] C. Kutzner, D. van der Spoel, M. Fechner, E. Lindahl, U.W. Schmitt, B.L. de Groot, and H. Grubmüller. Speeding up parallel GROMACS on high-latency networks. *J. Comput. Chem.*, 2007.

[7] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. GROMACS 4: Algorithms for highly efficient, load-Balanced, and scalable molecular simulation. *J. Chem. Theory Comput.*, 2008.

[8] Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften. SuperMUC system description. *http://www.lrz.de*, accessed 2013-09-17.

[9] L. Bock, C. Blau, G.F. Schroeder, I.I. Davydov, N. Fischer, H. Stark, M.V. Rodnina, A.C. Vaiana, and H. Grubmüller. Energy barriers and driving forces of tRNA translocation through the ribosome. *Nat. Struct. Mol. Biol.*, 2013 (accepted).

[10] D. Matthes and V. Gapsys. Driving Forces and Structural Determinants of Steric Zipper Peptide Oligomer Formation Elucidated by Atomistic Simulations. *J. Mol. Biol.*, 2012.