Performance tuning of molecular dynamics simulations using PME

Carsten Kutzner[1], Berk Hess[2] and [1]Helmut Grubmüller

[1]Dep. of Theoretical and Computational Biophysics, Max-Planck-Institute for Biophysical Chemistry, Göttingen, Germany
[2]Dep. of Biochemistry and Biophysics, Stockholm Center for Biomembrane Research, Stockholm University, Sweden

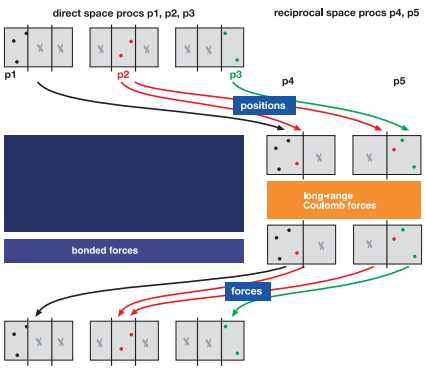
Abstract

Parallel molecular dynamics simulations with PME[1,2] electrostatics enjoy a considerable performance increase in Gromacs 4.5[3] compared to older versions. Part of that increase results from the fact that a subgroup of the processes is assigned exclusively for the reciprocal part of the PME calculation in a multiple-data, multiple-program (MPMD) approach. The reciprocal part requires a Fourier transformation of the charge grid and thus needs all-to-all communication among the participating processors. This communication pattern is a major scaling bottleneck for PME calculations in parallel[4] since on N processors, N*N messages have to be delivered. With separate PME processors, N is typically reduced by a factor of 2–4.

For optimum parallel performance, it is essential that every processor gets assigned an equal amount of work. Most critical for the MPMD approach is that the reciprocal processors complete their time step simultaneously with the real space processors such that waiting time is minimized. This is achieved by carefully balancing the ratio of real to reciprocal space processors and workload.

Gromacs predicts this ratio based upon hard-wired reference numbers. However, it does not know about the underlying network or processor clock rates, facts that influence the optimum ratio. Consequently, we directly seek the actual optimum with the help of short test runs. For a given number of processors, the adjustable parameters are the number of reciprocal processors and also the workload distribution between real and reciprocal space part of the Ewald sum. At high parallelization, the scaling usually benefits from shifting work to real space such that the fraction of reciprocal processors can be kept between a quarter and a half of the total available processors.

The MPMD PME approach



Example benchmark

As a typical example, we chose the DPPC membrane system from the Gromacs benchmark suite. The system contains 1024 DPPC lipids plus 23,552 SPC water molecules and thus altogether 121,856 particles.

For the benchmark, we used cutoffs of 1.0 nm and a reciprocal grid spacing of 0.135 nm yielding a PME grid of 132*140*48 points. For a scaling factor of 1.1 (1.2) for cutoffs and grid spacing the resulting PME grid is 120*125*42 points (110*117*39 points).

The benchmarks were performed on a cluster of 8 processor Intel L5430@2.66 GHz (Harpertown) nodes connected by 4xDDR Infiniband (20 Gbit/s).

Summary

In most cases, performance can be improved notably. For the presented benchmark, the performance could be raised by a factor of 1.2 on average for 16–48 processors.

Outlook

Incorporate the Wang et al. PME error estimate[5] into the g_tune_pme tool so that load can be shifted between real and reciprocal space at a constant absolute error.

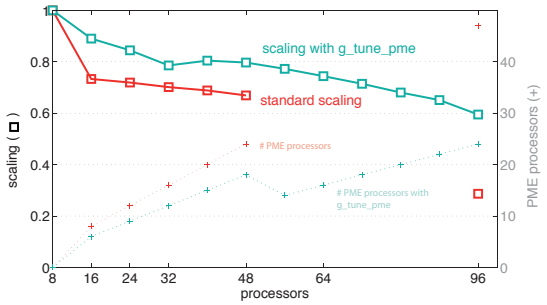


Fig. 2: Squares (left scale) show the scaling that is the execution time on a single node divided by *n* times the execution time on *n* nodes.
Plus signs (right scale) show the number of PME processors. Up to 48 processors the tpr file was left untouched, whereas from 56 processors on a slight shift (factor 1.1) of work from reciprocal to real space came out to be advantageous for the performance. For 88 and 96 processors, the scaling factor 1.2 was applied. Between 48 and 96 processors the required number of PME processors could not be estimated by mdrun.

Performance with and without tuning

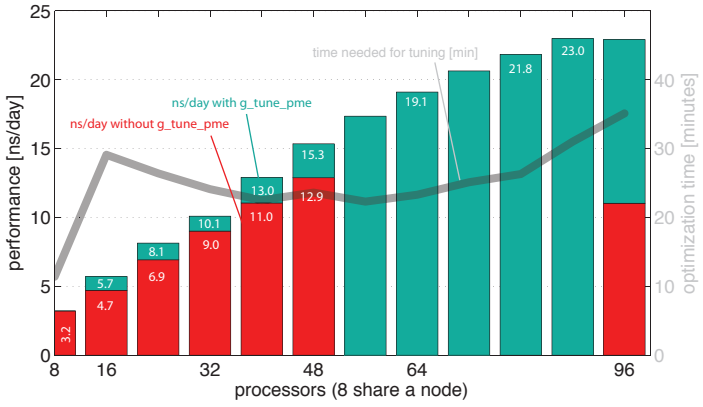


Fig. 1: Red bars (left scale) depict the DPPC performance [ns/day] at a 2 fs time step when the number of PME processors is estimated by mdrun. The green bars show the gain after tuning. The grey line (right scale) shows the wall clock time needed for the tuning.

g_tune_pme

The Gromacs 4.5 tool reads in a simulation tpr file and writes several benchmark tpr files where a) the number of steps is set to the benchmark value – typically 1000 plus 100 equilibration steps – and b) computational load is shifted between real and reciprocal part of the Ewald sum. This is achieved by multiplying both the direct-space cutoff as well as the PME grid spacing by a small factor, typically 1.0, 1.1 and 1.2. The higher the factor, the more work is shifted from the PME to the direct-space processors. This is done because the reciprocal space calculations are most efficient on a small number of processors. Subsequently, the newly created tpr files are benchmarked with variable number of PME processors. The time counters are reset after 100 steps when the dynamic load balancing is equilibrated.

```
export MPIRUN="/usr/bin/mpiexec -hostfile=./hosts"
g_tune_pme -np 128 -s ./protein.tpr -launch
```

References

(1) T. Darden, D. York, L. Pedersen, 1993. Particle mesh Ewald: An N*log(N) method for Ewald sums in large systems, J. Chem. Phys. 98 (12), 10089-10092
(2) U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee, G. Pedersen, 1995. A smooth particle mesh Ewald method, J. Chem. Phys. 103 (19), 8577-8593
(3) B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, 2008. GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. JCTC 4 (3), 435-447
(4) C. Kutzner, D. van der Spoel, M. Fechner, E. Lindahl, U. Schmitt, B. de Groot and H. Grubmüler, 2007. Speeding up parallel GROMACS on high-latency networks. J. Comp. Chem. 28 (12), 2075-2084
(5) H. Wang, F. Dommert and C. Holm, 2010. Optimizing working parameters of the smooth particle mesh Ewald algorithm in terms of accuracy and efficiency. J. Chem. Phys. 133,