GROMACS 5.0 benchmarks

Version 0.1, June 2015

About these benchmarks

These benchmarks illustrate the performance of several different GROMACS simulations running in parallel on different hardware types. In each case, performance expected for long runs is reported, in particular resetting the built-in performance counters after any auto-tuning has completed. Both general questions about GROMACS benchmarking, and questions specific to these results should be posted on the `gmx-users` mailing list.

Important caveat

These are "real world" performance numbers, indicative of what GROMACS users might experience on fully-loaded production machines. On many HPC machines, the network is a resource that is shared between multiple running jobs, and like other MD codes, GROMACS performance is limited by the latency of network communication, particularly when running PME simulations. On a quiet machine (or isolated network), it is expected that GROMACS will run noticeably faster and have more regular scaling characteristics than reported here. However, we expect that the numbers here are a more useful guide for users.

It would also be possible to improve upon the numbers reported here with more detailed tuning, for example with the `gmx tune_pme` tool. Users planning long production runs are encouraged to explore the possibilities that this tool can uncover, as well as the capabilities of the built-in tuning.

Simulation systems

Ion channel systems

Several versions of the same 142k particle ion channel simulation with the (all-atom) Amber ff99SB-ILDN force field were used. The GluCl ion channel was embedded in a DOPC membrane and solvated in TIP3P water. Each version of these simulations

- uses the Verlet cut-off scheme defaults, including buffering of short-range interactions,
- has a 3D periodic volume of fixed size,
- uses a VDW cutoff radius 1.0 nm,
- uses a velocity-rescaling thermostat,
- has constraints on all bonds implemented with LINCS and SETTLE, and
- has initial SPME electrostatics settings with a 1.0 nm cutoff, cubic spline interpolation on a Fourier grid whose maximum spacing was 0.12 nm, and `ewald_rtol = 1e-5`.

Some versions of this simulation use a time step of 2.5 fs, which can use the default LINCS parameters for order (4) and number of iterations per MD step (1). At higher parallelism, these defaults restrict the domain decomposition too much, and in some cases higher performance can be obtained from different LINCS parameter combinations, such as order 3 with 2 iterations, or order 2 with 3 iterations. These combinations do not compromise the accuracy of the integration. Such simulations are described with shorthand like "GluCl, LINCS 3-2" for order 3 and 2 iterations.

Other versions of this simulation employ virtual sites in place of many hydrogen atoms, permitting a time step of 5 fs. These require more stringent LINCS parameters, and were run with either order 6 with 1 iteration, order 4 with 2 iterations, or order 3 with 3 iterations. Such simulations are described with shorthand like "GluCl, vsites, LINCS 6-1."

Ethanol-in-water simulations

Several versions of solutions of OPLS all-atom ethanol in SPC/E water are reported, with respectively 54k, 216k and 432k particles. In each case, 2:1 ratio of water to ethanol atoms is present. Such a system is expected to be crudely indicative of performance of protein-in-water simulations, but will likely place lower demands on the dynamic load balancing, and thus have slightly higher performance.

Each version of these simulations

- uses the Verlet cut-off scheme defaults, including buffering of short-range interactions,
- has a 3D periodic volume of fixed size,
- uses a VDW cutoff radius 0.9 nm,
- uses a velocity-rescaling thermostat,
- has constraints on bonds with hydrogen atoms, implemented with LINCS (with default parameters) and SETTLE, and
- has initial SPME electrostatics settings with a 0.9 nm cutoff, cubic spline interpolation on a Fourier grid whose maximum spacing was 0.1125 nm, and `ewald_rtol = 1e-5`.

Machines used in these simulations

We would like to thank the supercomputing centers mentioned below for their cooperation in giving us access to their hardware for GROMACS porting and performance studies.

Beskow

Beskow is a Cray XC40 architecture, with two 16-core sockets per node containing Intel Xeon E5-2698v3 Haswell processors and the Cray Aries network (with Dragonfly topology). It is hosted at the PDC supercomputing center in Stockholm, Sweden.

Piz Daint

Beskow is a Cray XC30 architecture, with one 8-core socket per node containing Intel Xeon E5-2670 Sandy Bridge processors and one NVIDA Tesla K20X GPU, and the Cray Aries network (with Dragonfly topology). It is hosted by the CSCS supercomputing center in Lugano, Switzerland.

Hydra

Hydra is an IBM iDataPlex system, with two 10-core sockets per node containing Intel Xeon E5-2680 Ivy Bridge processors and two NVIDIA K20X GPUs and InfiniBand FDR14 network.

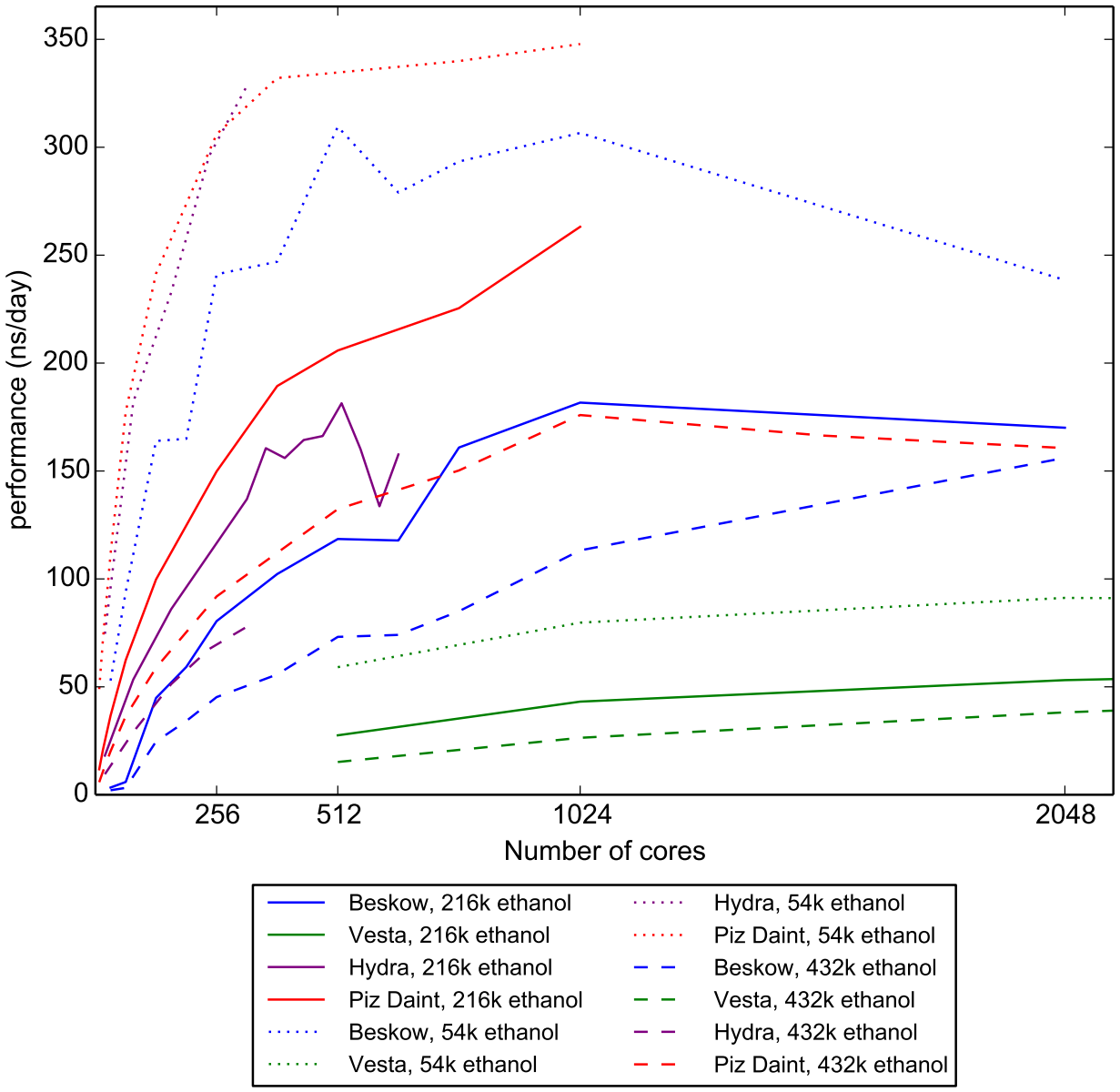
Vesta

Vesta is an IBM BlueGene/Q system, with 16 cores per node containing PowerPC A2 processors and the IBM proprietary 5D torus network. These cores have up to four hardware threads available. Using more than one hardware thread per core is necessary for best performance, because the instruction mix used by GROMACS can benefit from the ability of the core to issue two instructions per cycle, but only to different hardware threads (unlike Intel cores). OpenMP is the most effective way to run more than one hardware thread per core. Generally, 2 or 3 hardware threads per core is optimal.

Benchmark results

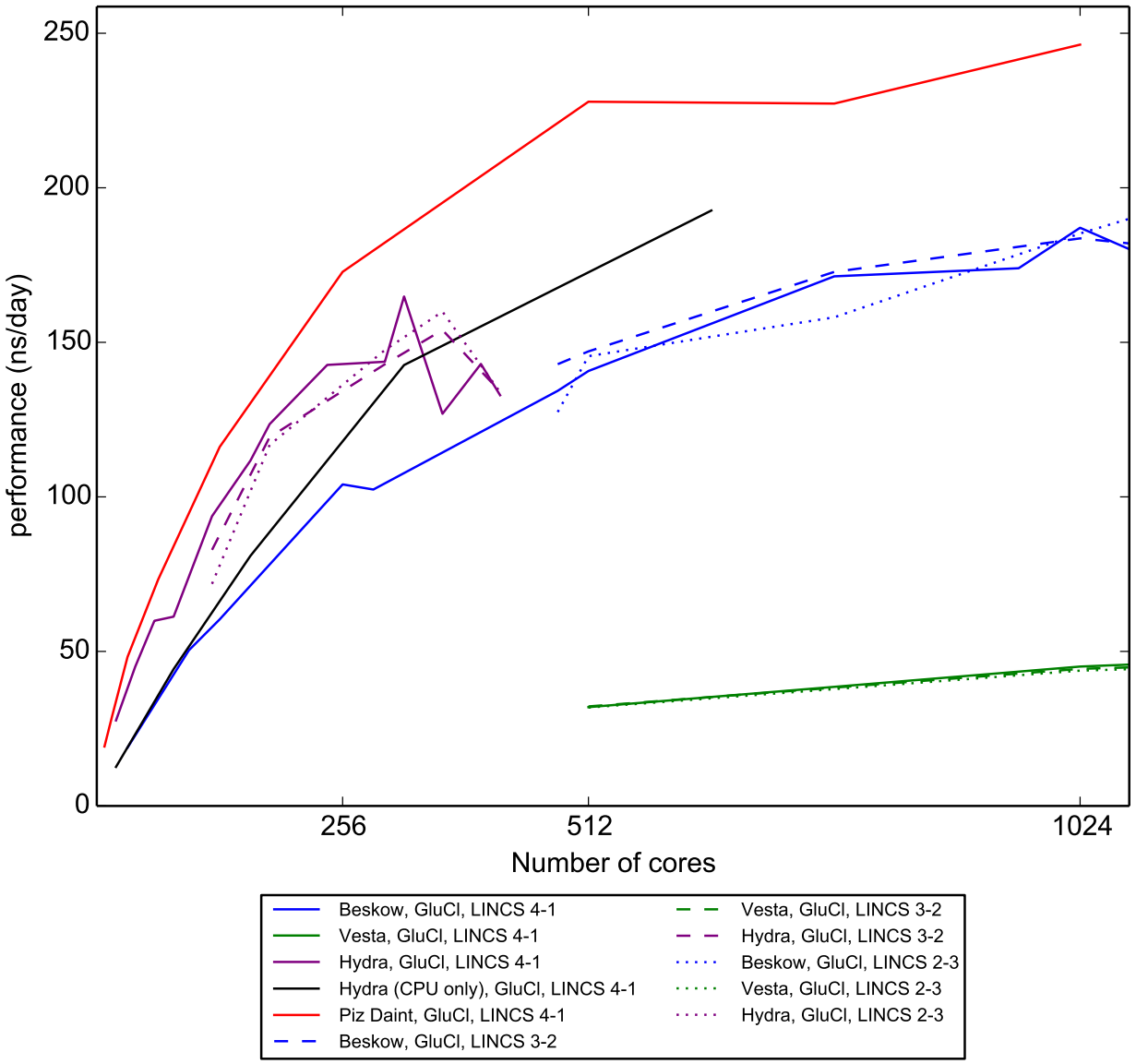
First, combined plots showing the best known performance of each system on each machine for a given number of cores are shown. Later, the same data is shown in more detail, illustrating the variation that can be seen with different numbers of OpenMP threads per MPI rank with total number of cores constant.

All ethanol-in-water on all machines



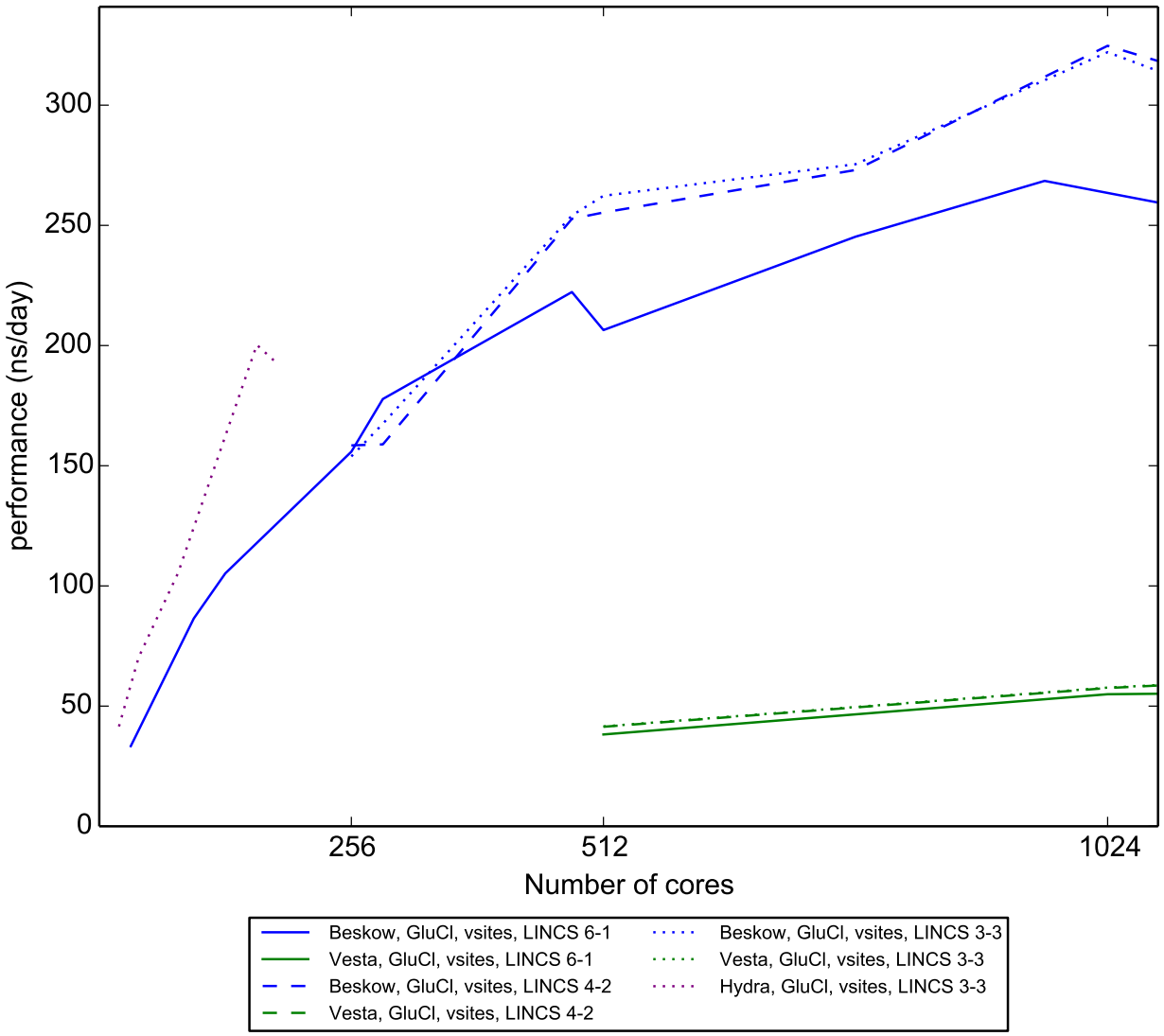
Performance is higher for the smaller systems, as expected. The relative weakness of the Vesta cores is clearly visible. The variation seen for Hydra is likely due to interference from other jobs on the machine.

GluCl without virtual sites on all machines



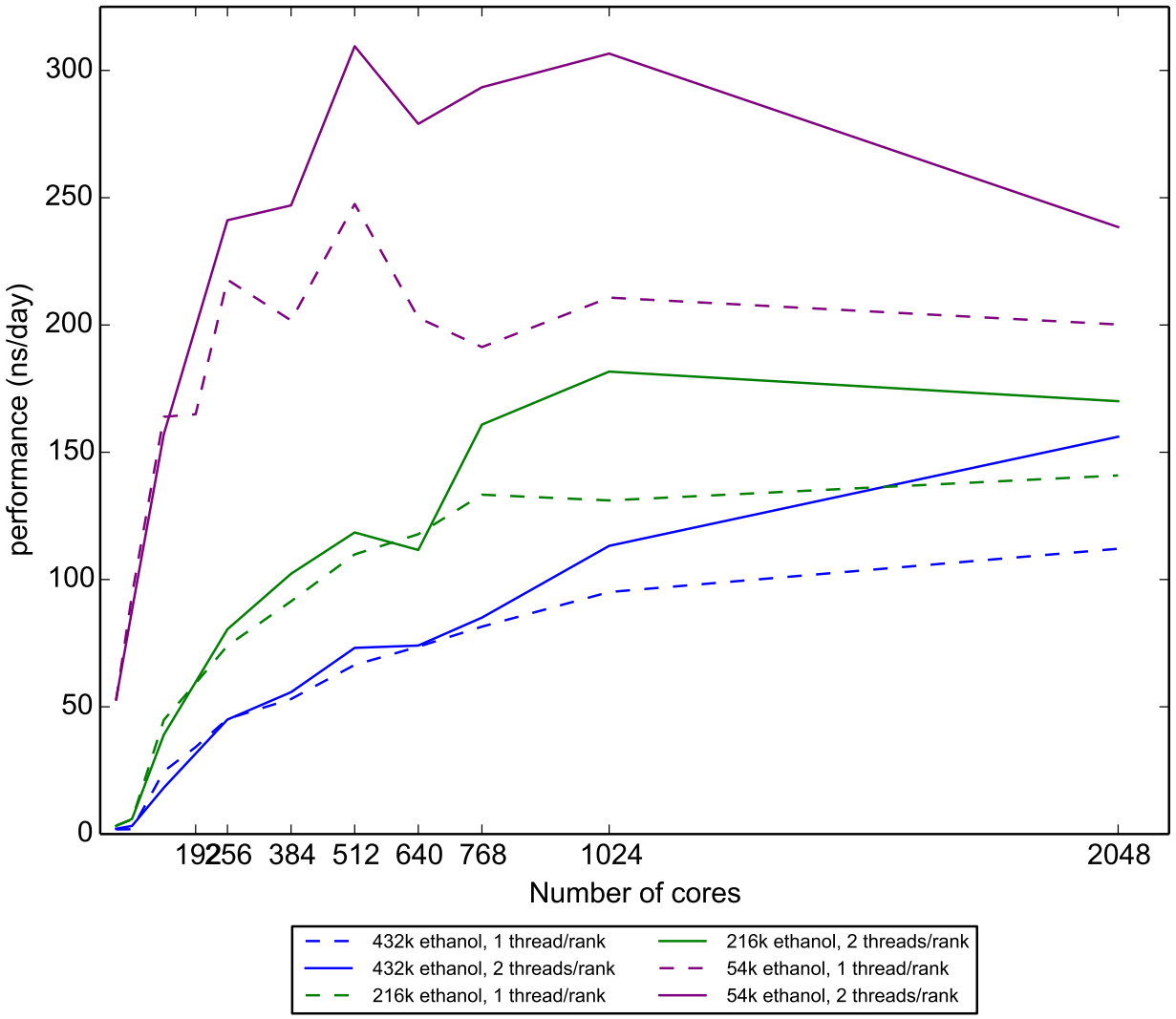
The variation seen for Hydra is likely due to interference from other jobs on the machine. LINCS settings have only minor effects on performance.

GluCl with virtual sites on all machines



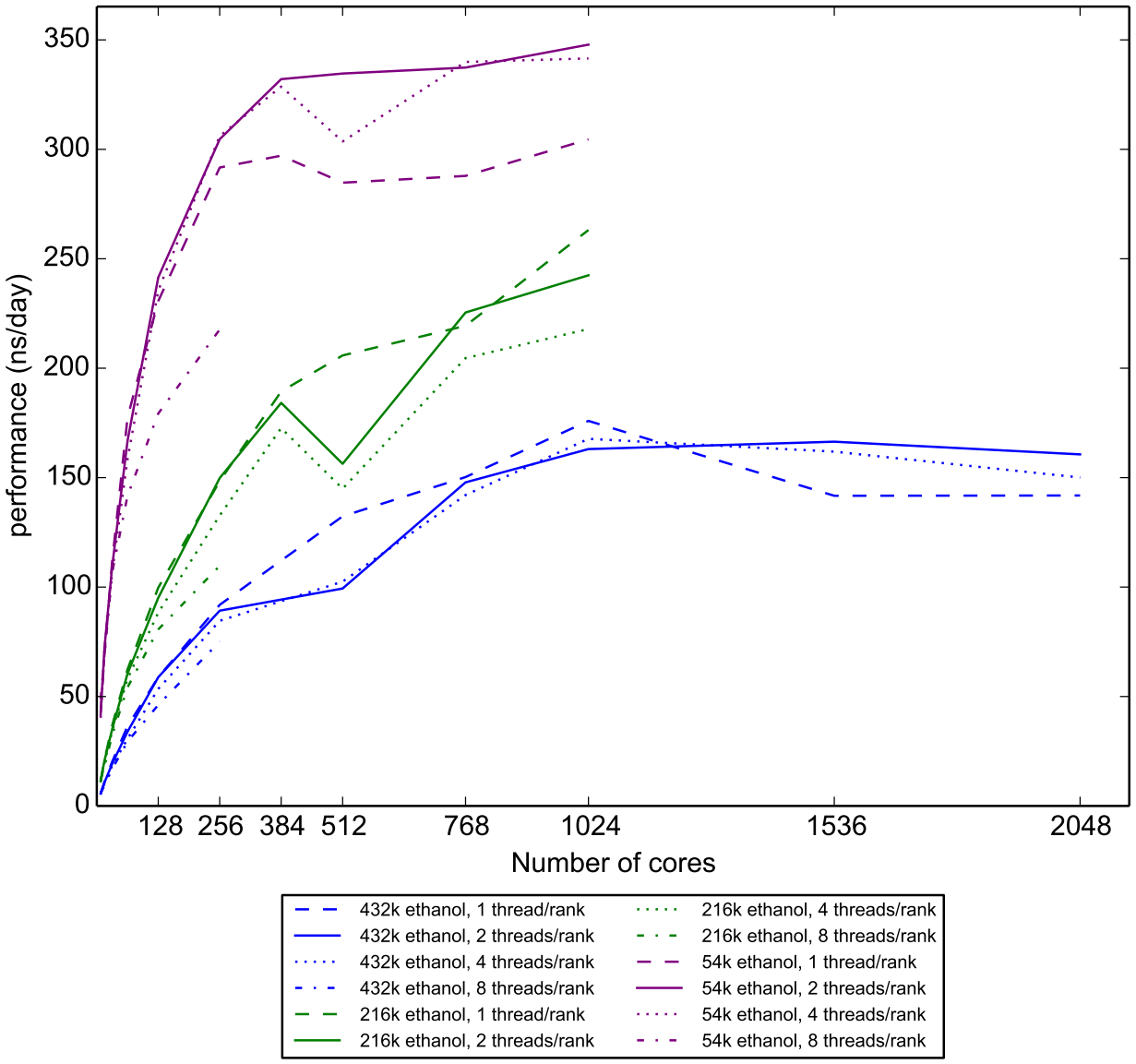
With virtual sites, the improved performance of lower-order LINCS settings can be attributed to the lower minimum size required for domains (see detailed plots below, and the GROMACS Reference Manual). Smaller domains permit the use of more efficient (smaller) numbers of OpenMP threads per MPI rank.

Ethanol-in-water on Beskow



Once the number of particles per domain reaches a threshold, adding more cores improves performance only if more than one OpenMP thread is used per MPI rank. (However, for such CPU-only runs, this generally only works for very small OpenMP thread counts.)

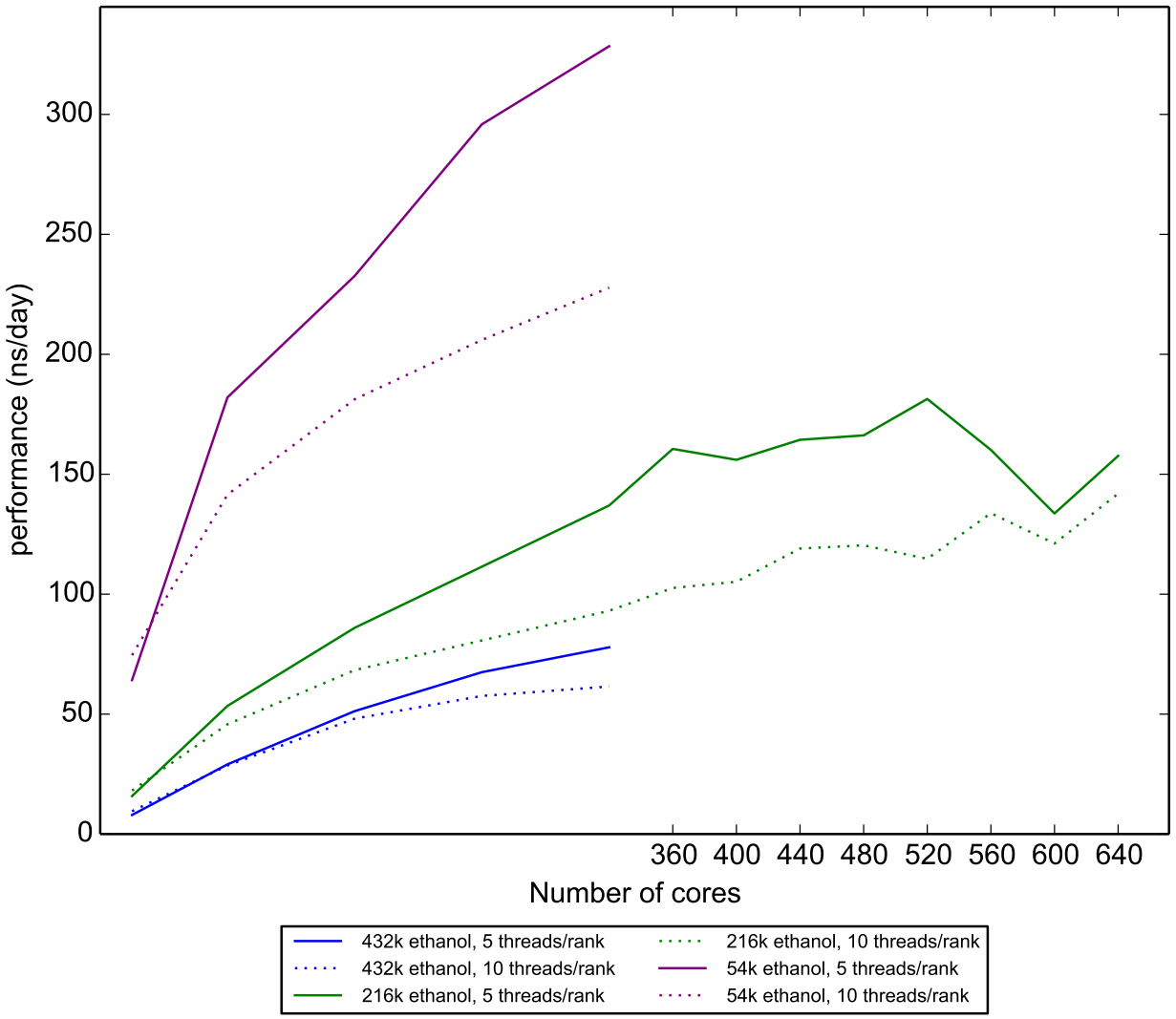
Ethanol-in-water on Piz Daint



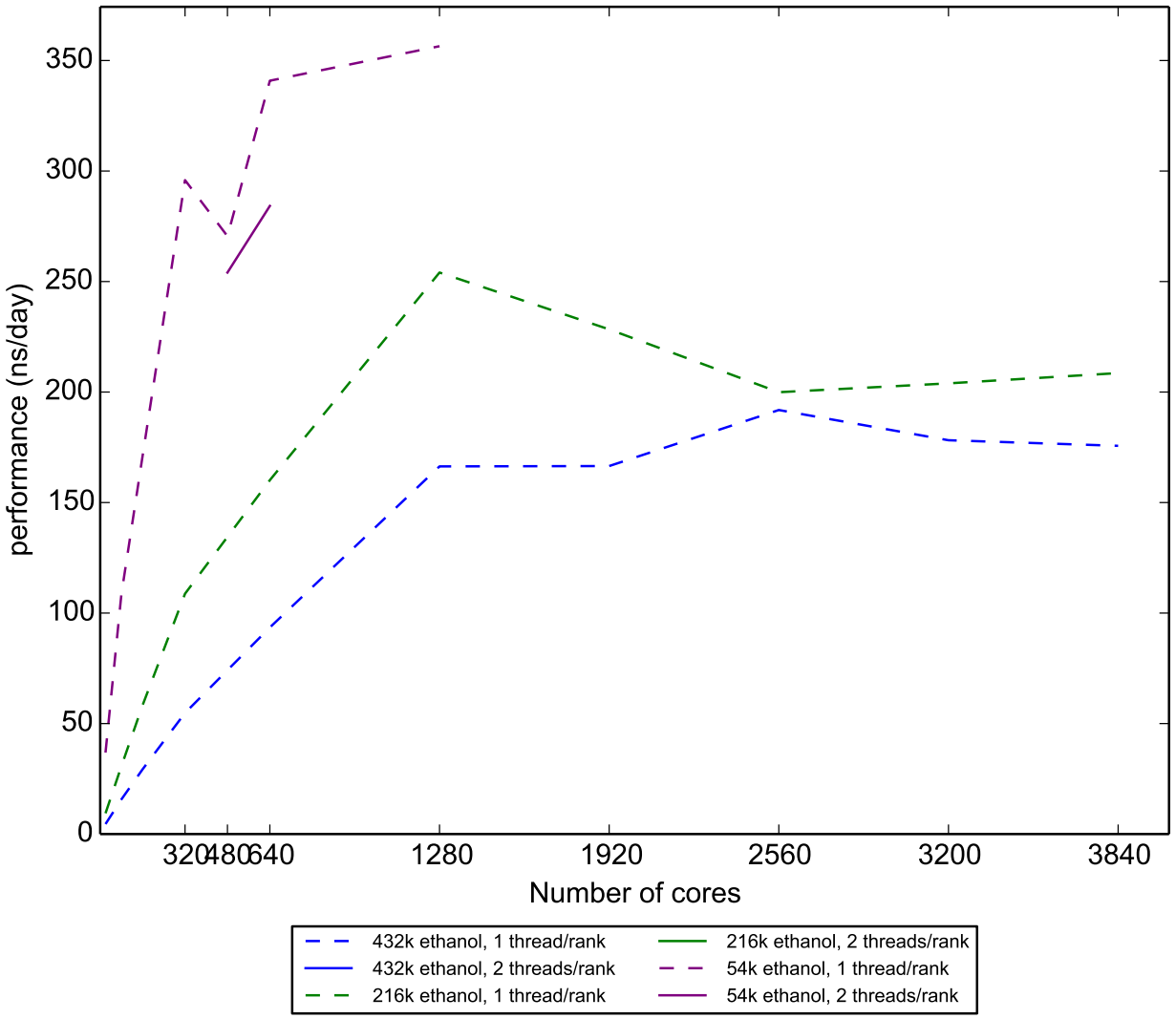
Once the number of particles per domain reaches a threshold, adding more cores improves performance only if more than one OpenMP thread is used per MPI rank.

The dip seen for some configurations at 512 cores is likely due to interference from other jobs on the machine.

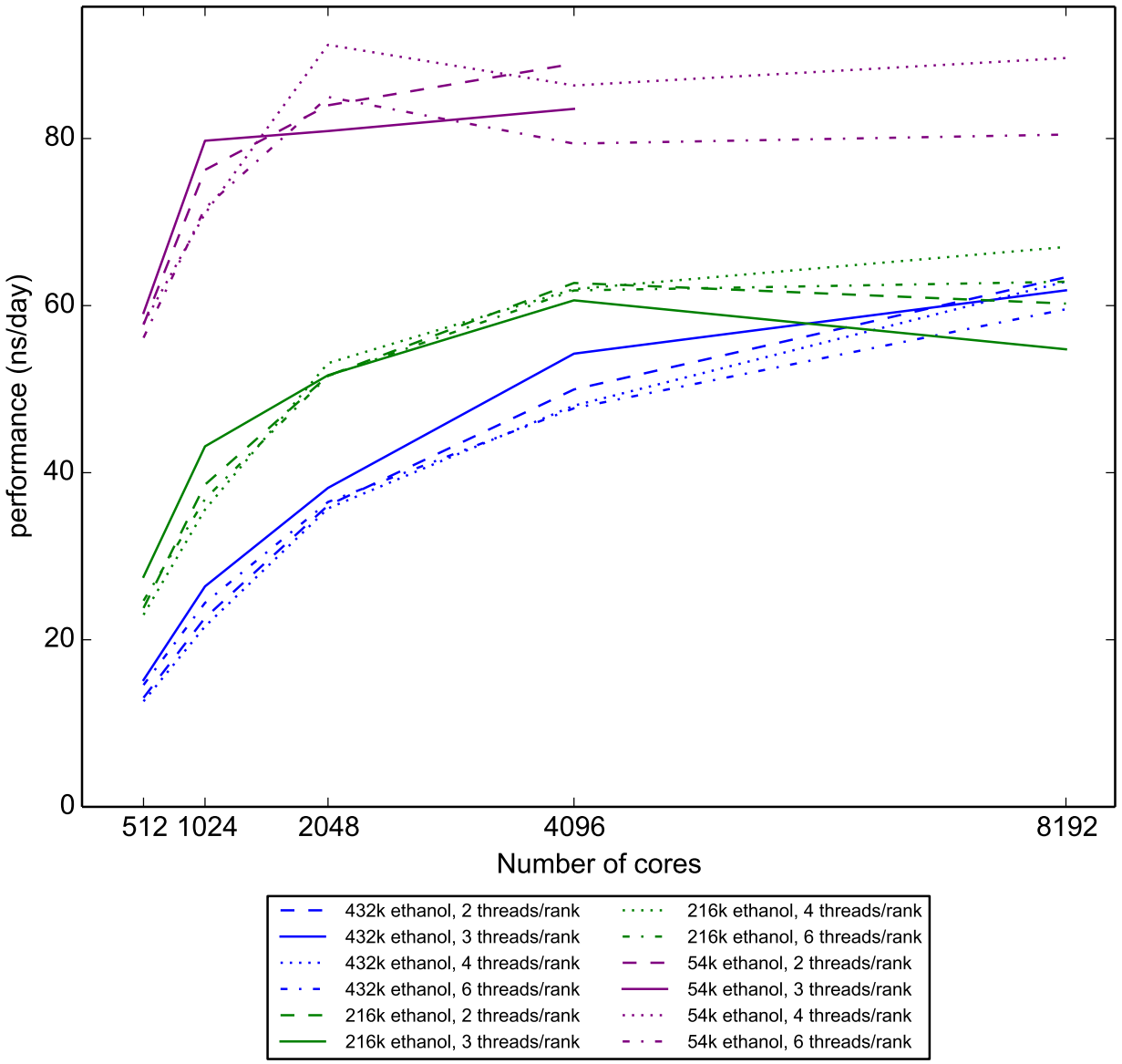
Ethanol-in-water on Hydra



Ethanol-in-water on Hydra (CPU only)

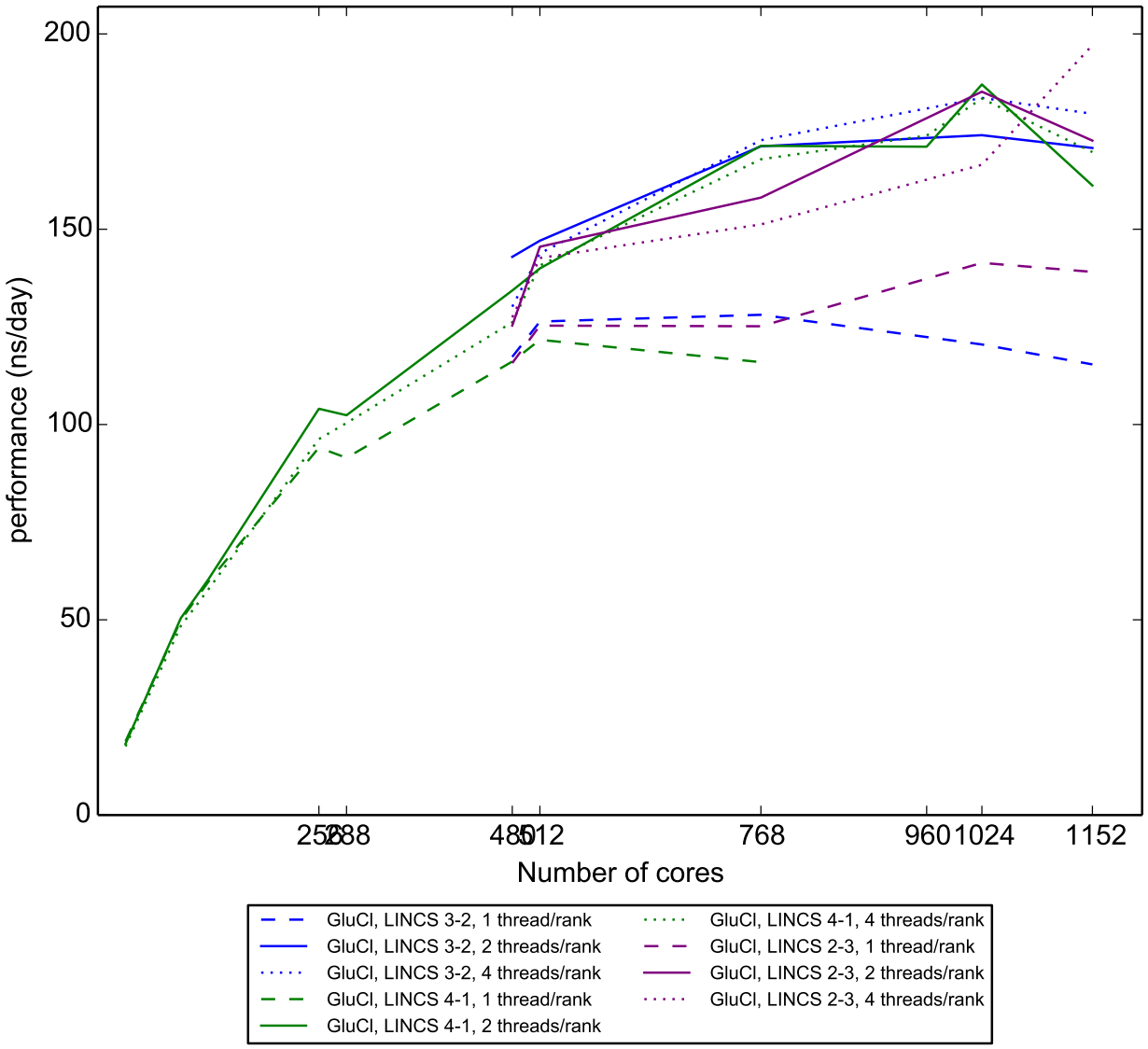


Ethanol-in-water on Vesta



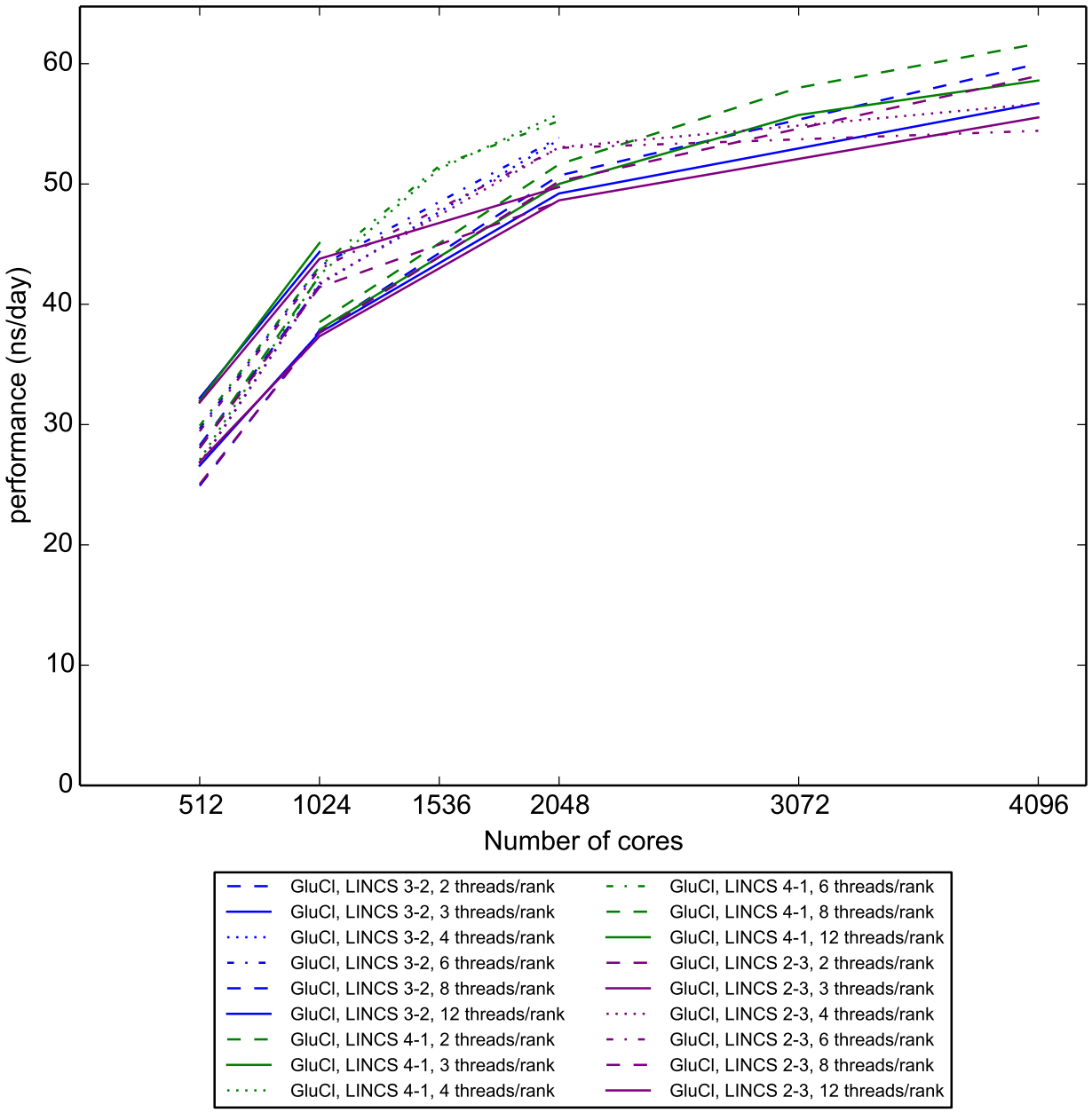
Calculations with 2 or 3 threads per MPI rank were restricted to a single core. Those with 4 or 6 threads per MPI rank distributed those over two cores. Below a certain threshold, the use of 3 threads per MPI rank on a single core seems best, and above that, the use of 4 threads per MPI rank over two cores (ie. 2 per core) seems best.

GluCl without virtual sites on Beskow



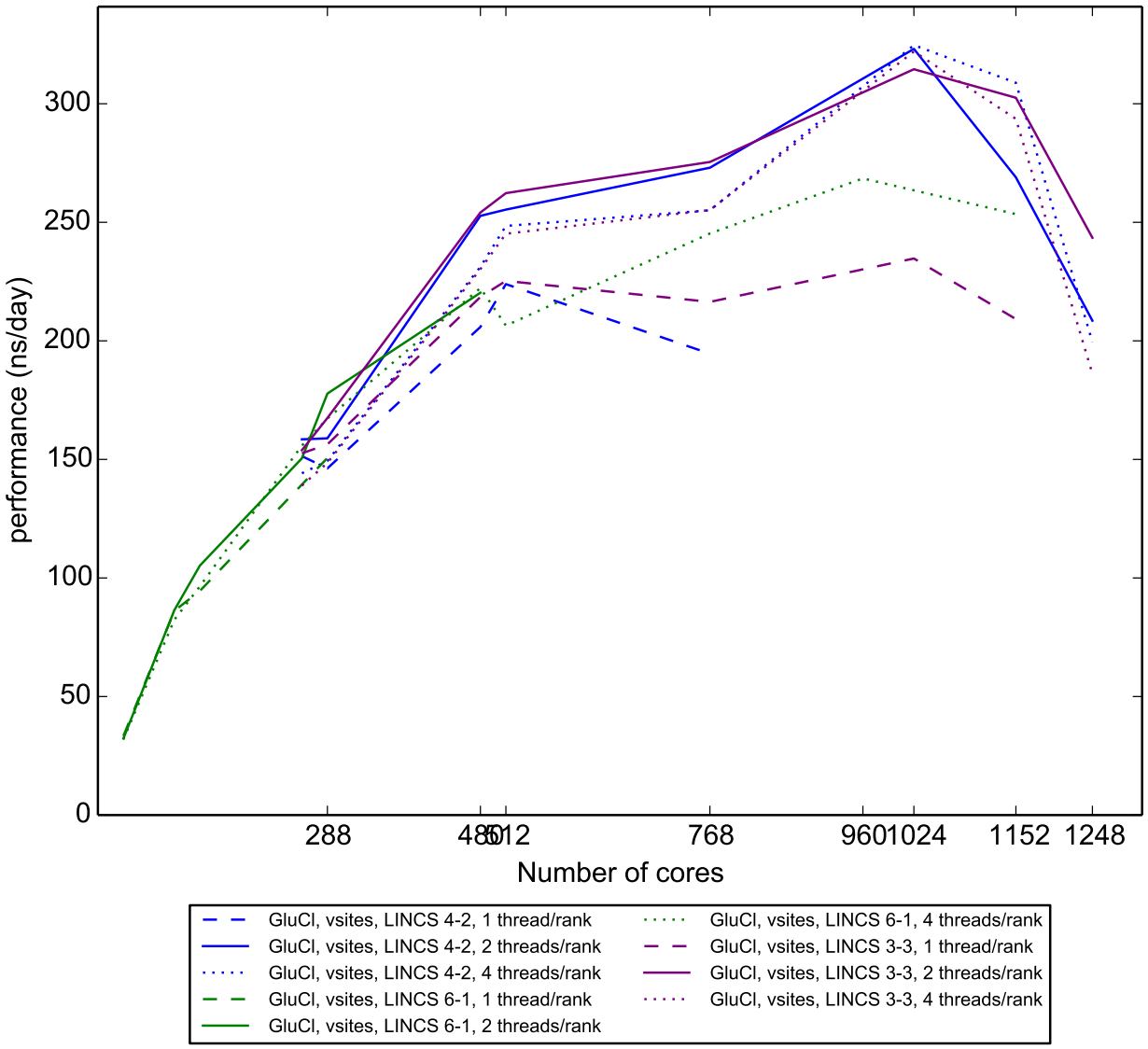
Once the number of particles per domain reaches a threshold, adding more cores improves performance only if more than one OpenMP thread is used per MPI rank. (However, for such CPU-only runs, this generally only works for very small OpenMP thread counts.)

GluCl without virtual sites on Vesta



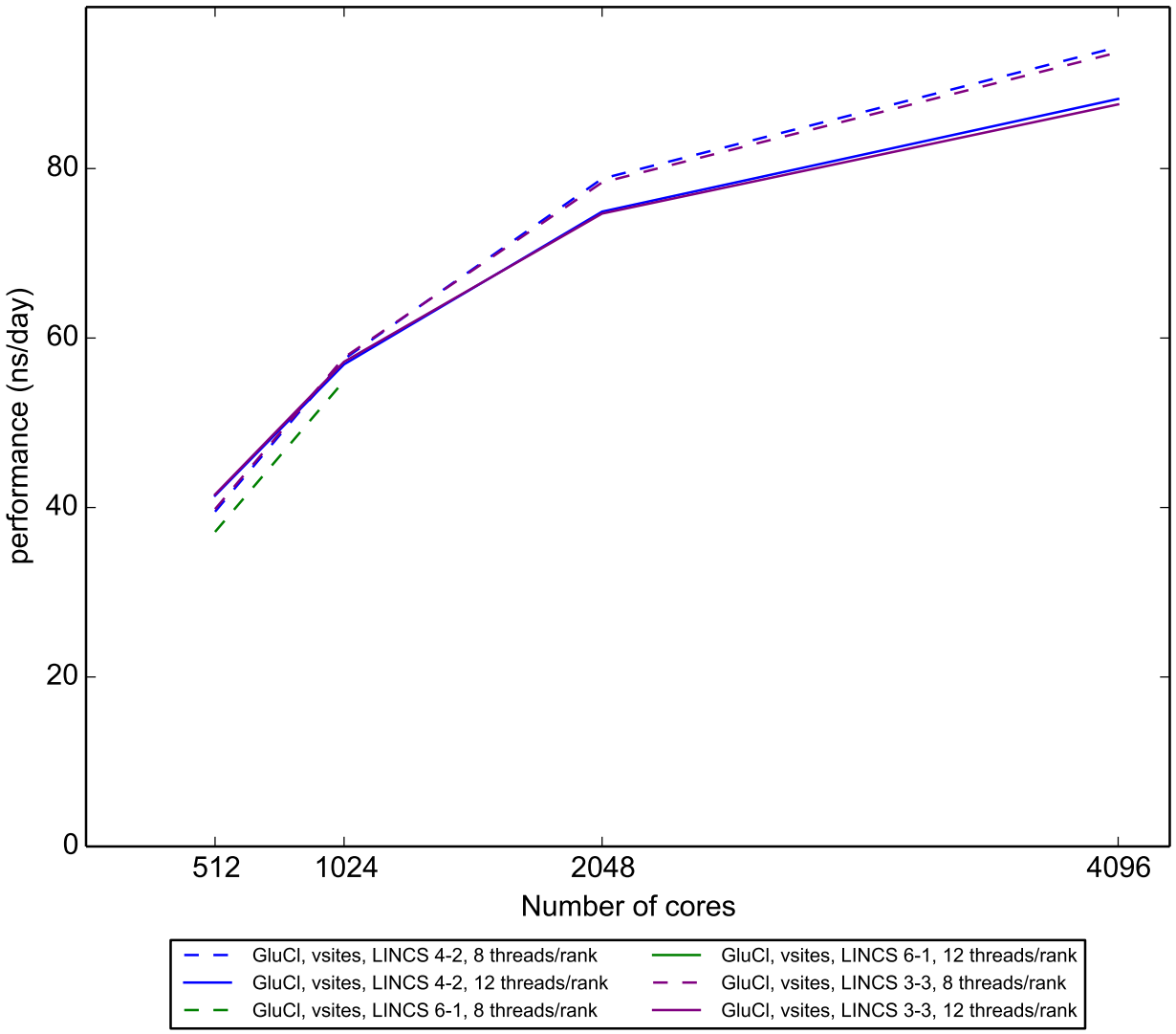
Calculations with 2 or 3 threads per MPI rank were restricted to a single core. Those with 4 or 6 threads per MPI rank distributed those over two cores, and 8 or 12 threads were distributed over 4 cores. Below a certain threshold, the use of 3 threads per MPI rank on a single core seems best, and above that, the use of 2 threads per core and multiple cores in an MPI rank seems best.

GluCl with virtual sites on Beskow



The improved performance of lower-order LINCS settings can be attributed to the lower minimum size required for domains. Smaller domains permit the use of more efficient (smaller) numbers of OpenMP threads per MPI rank.

GluCl with virtual sites on Vesta



The improved performance of lower-order LINCS settings can be attributed to the lower minimum size required for domains. Smaller domains permit the use of more efficient (smaller) numbers of OpenMP threads per MPI rank. MPI ranks with threads distributed over 4 cores were used.