Performance evaluation of kinetic code on scalar processors

Takayuki Umeda
Institute for Space-Earth Environmental Research, Nagoya University
Nagoya, Aichi
umeda@isee.nagoya-u.ac.jp

ABSTRACT

The present study deals with the Eulerian kinetic code as a high-performance application, which solves the first-principle kinetic equations known as the Boltzmann equation. A five-dimensional Boltzmann code with two spatial and three velocity dimensions is parallelized with the MPI-OpenMP hybrid parallelism. Strong scaling of the code is measured on various scalar processors.

CCS CONCEPTS

• **Computing methodologies** → **Massively parallel and high-performance simulations**; *Parallel algorithms*; • **Applied computing** → Physics; Astronomy; Earth and atmospheric sciences.

KEYWORDS

High performance computing; kinetic simulation; Eulerian-grid-based method; hybrid parallelism; performance evaluation; Xeon

ACM Reference Format:
Takayuki Umeda. 2019. Performance evaluation of kinetic code on scalar processors. In *ICPP2019: 48th International Conference on Parallel Processing, August 5–8, 2019, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

Manycore scalar processors are one of recent trends of CPUs in high-performance computing, which run at low clock frequency to reduce the power consumption but have a large number of compute cores with processing units for operating multiple data, such as Advanced Vector Extension (AVX) and Single Instruction Multi Data (SIMD) units. It is not easy for users of scientific applications to achieve a high performance (e.g., a computational efficiency of more than 30% to the theoretical peak performance) on recent manycore scalar processors with the multiple data units.

As a high-performance application to scientific computing, the present study deals with a first-principle kinetic simulation based on the Eulerian grid. The first-principle kinetic simulation usually requires enormous computing resources since this solves time development of distribution functions defined in "hyper" dimensions (at most three spatial and three velocity dimensions). In Eulerian-grid-based simulations, such as fluid simulations and the present kinetic simulations, a bottleneck of the computational performance generally exists at the memory bandwidth.

ICPP2019, August 5–8, 2019, Kyoto, Japan

Performance tuning of the Eulerian-grid-based codes on many-core scalar processors is an issue in high-performance computing. In the present study, performance evaluation of the Eulerian-grid-based kinetic code is made on a single compute node with two Xeon Broadwell processors, with a Xeon Phi Knights Landing processor, and with a SPARC64 XIfx processor.

2 OVERVIEW OF NUMERICAL SCHEMES

The present kinetic code solves the first-principle equation, which is known as the Vlasov (collisionless Boltzmann) equation,

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{r}} + \left[\frac{q_s}{m_s} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) + \mathbf{g} \right] \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0, \tag{1}$$

where f represents the distribution function at a given position $\mathbf{r}$, velocity $\mathbf{v}$ and time t, and $\mathbf{E}$, $\mathbf{B}$, $\mathbf{g}$, q, and m represent electric field, magnetic field, gravity, charge, and mass, respectively. The subscript s represents the species of singly charged particles (e.g., $s = i$ and e for positively-charged ions and electrons, respectively). Here, the collisional term in the right hand side of the equation is set to be zero. The self-consistent electromagnetic and gravitational fields are obtained by coupling field equations such as the Maxwell equations and the gravitational field formula.

It is not easy to integrate the "hyper-"dimensional equation numerically in time in terms of both computational accuracy and computational resources. In order to simplify the numerical operation, the Boltzmann equation (1) is separated into two advection and one rotation equations based on operator splitting [1, 2]. The three equations are solved with conservative schemes [3–5].

We adopt the "domain decomposition" with the standard message passing interface (MPI) library as the first-level process parallelism as standard Eulerian-grid-based methods do. However, we decompose the computational domain only in the position dimensions [6]. The velocity dimensions are not decomposed because there arise some additional communications overhead due to a reduction operation in the computation of the density and the momentum. It is well-known that the domain decomposition involves the exchange of halo layers for the distribution function and electromagnetic field data at boundaries of each computational sub-domain. The present non-oscillatory and conservative scheme [3, 4] uses six grid cells for numerical interpolation. Thus, three halo grids are exchanged by using the "`MPI_Sendrecv`" subroutine in MPI for simplicity, portability and stability. As a second-level thread parallelism, we use the "`OMP (PARALLEL) DO`" directive together with the "`COLLAPSE`" clause to parallelize most outer multiple loops with less threading overhead [7].

A hyper-dimensional simulation requires a huge computer resource. For applications to practical scientific computing, however, massively parallel computation with multiple compute nodes is

necessary, since more than 1 TB memory is usually used. Hyper-dimensional Boltzmann simulations are practically in use (but not so widely) in plasma sciences, especially for laser plasma [1], tokamak plasma in thermonuclear fusion devices [8], and collisionless space plasma [9, 10].

3 PERFORMANCE EVALUATION

We use three systems for the performance evaluation. System I has 96 GB of DDR4 shared memory and a single Xeon Phi 7250 (Knights Landing, 1.4 GHz) processor on one compute node. The processor has 16 GB of MCDRAM and 68 compute cores. A total of 272 processes are executable with Hyper Threading (HT) technology enabled. The cluster mode is chosen to be "Quadrant," and the memory mode is chosen to "Cache." System II has 512 GB of DDR4 shared memory and a dual Xeon E5-2697 v4 processor (Broadwell, 2.3 GHz) on one compute node. The processor has 18 compute cores and a total of 36 processes are executable on a compute node. The Intel Parallel Studio XE Cluster Edition Ver.17.0.1.132 is installed in these two systems. The compiler option used in the present performance measurement is "`-ipo -ip -O3 -xMIC-AVX512 -qopenmp`" for System I and "'`-ipo -ip -O3 -xCORE-AVX -qopenmp`" for System II. System III has 32 GB of High Bandwidth Memory (HBM) and a single SPARC64 XIfx processor (2.2 GHz) on one compute node. A total of 32 processes are executable. The compiler option is "`-Kfast,ocl,simd,openmp,parallel`."

The total number of grid cells is $N_x \times N_y \times N_{vx} \times N_{vy} \times N_{vz} = 134 \times 70 \times 40^3$ for ions and electrons, which corresponds to a job size of ~28 GB including temporary work arrays. One process is used for each of the two particle species. The number of processes per compute node is fixed to two, but the number of threads per process is changed in the present strong scaling measurement.

The top panel of Figure 1 shows the strong scaling of our original five-dimensional Eulerian Vlasov code. The vertical axis shows the elapsed time for five time steps. The horizontal axis shows the number of threads per compute node. The diamonds, circles, and "x"-marks correspond to the results on Systems I, II, and III, respectively. The computational speed of the code on one compute core of Systems II and III is almost the same by coincidence, while the scalability on System III is excellent. Although the computational speed of the code on one compute core of Systems I is much slower than the other two systems, the computational speed on one compute node is comparable to the other systems due to the large number of compute cores and HT. We has made a performance tuning of our code by the decomposition of most-inner loops and reduction of temporary work arrays. The code becomes faster by 2 times on System I, by 1.5 times in System II, and by 1.15 times in System III.

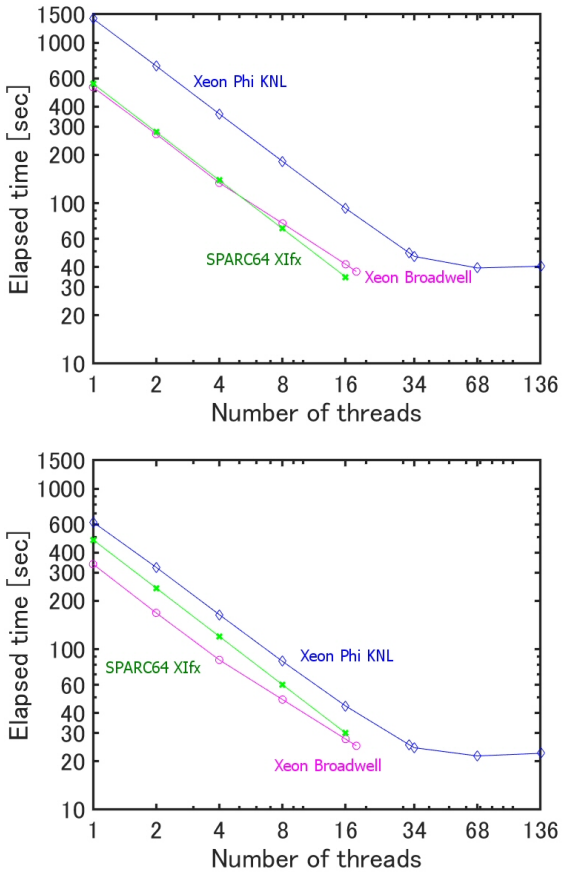

Figure 1: Strong scaling of the five-dimensional Eulerian Vlasov code on Xeon Phi KNL (System I), Xeon Broadwell (System II), and SPARC64 XIfx (System III) processors. (top) The original code and (bottom) tuned code.

ACKNOWLEDGMENTS

This work was supported by MEXT/JSPS KAKENHI No.JP19H01868. The Fujitsu FX100 at ITC in Nagoya University was used as a joint research program of ISEE in Nagoya University and a WG activity of the Society of Scientific Systems (SSKEN).

REFERENCES

[1] A. Ghizzo, F. Huot, and P. Bertrand, A non-periodic 2D semi-Lagrangian Vlasov code for Laser-plasma interaction on parallel computer, *J. Comput. Phys.*, 186(1):47–69, March 2003.

[2] T. Umeda, K. Togano, and T. Ogino, Two-dimensional full-electromagnetic Vlasov code with conservative scheme and its application to magnetic reconnection, *Comput. Phys. Commun.*, 180(3):365–374, March 2009.

[3] T. Umeda, A conservative and non-oscillatory scheme for Vlasov code simulations, *Earth Planets Space*, 60(7):773–779, August 2008.

[4] T. Umeda, Y. Nariyuki, and D. Kariya, A non-oscillatory and conservative semi-Lagrangian scheme with fourth-degree polynomial interpolation for solving the Vlasov equation, *Comput. Phys. Commun.*, 183(5):1094–1100, May 2012.

[5] H. Schmitz and R. Grauer, Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields, *Comput. Phys. Commun.*, 175(2):86–92, March 2006.

[6] T. Umeda, K. Fukazawa, Y. Nariyuki, and T. Ogino, A scalable full electromagnetic Vlasov solver for cross-scale coupling in space plasma, *IEEE Trans. Plasma Sci.*, 40(5):1421–1428, March 2012.

[7] T. Umeda and K. Fukazawa, Hybrid parallelization of hyper-dimensional Vlasov code with OpenMP loop collapse directive, *Adv. Parallel Comput.*, 27:265–274, April 2016.

[8] Y. Idomura, M. Ida, T. Kano, N. Aiba, and S. Tokuda, Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation, *Comput. Phys. Commun.*, 179(6):391–403, September 2008.

[9] S. Von Alfthan, D. Pokhotelov, Y. Kempf, S. Hoilijoki, I. Honkonen, A. Sandroos, and M. Palmroth, Vlasiator: First global hybrid-Vlasov simulations of Earth's foreshock and magnetosheath, *J. Atmos. Sol.-Terr. Phys.*, 120: 24-35, December 2014.

[10] T. Umeda and K. Fukazawa, A high-resolution global Vlasov simulation of a small dielectric body with a weak intrinsic magnetic field on the K computer, *Earth Planets Space*, 67(1):49, April 2015.