Progress in NUCLEAR SCIENCE and TECHNOLOGY, Vol. 2, pp.657-662 (2011)

ARTICLE

Parallelization of Gyrokinetic PIC Code for MHD Simulation

Hiroshi NAITOU [1,*], Hiroki HASHIMOTO [1], Yusuke YAMADA [1],
Shinji TOKUDA [2,4] and Masatoshi YAGI [3,4]

[1] *Graduate School of Science and Engineering, Yamaguchi University, 2-16-1 Tokiwadai, Ube-shi, Yamaguchi-ken, 755-8611, Japan*
[2] *Research Organization for Information Science & Technology, 2-32-3 Kita-shinagawa, Shinagawa-ku, Tokyo-to, 140-0001, Japan*
[3] *Research Institute for Applied Mechanics, Kyushu University, 6-1 Kasuga-kouen, Kasuga-shi, Fukuoka-ken, 816-8580, Japan*
[4] *Naka Fusion Institute, Japan Atomic Energy Agency, 801-1 Mukoyama, Naka-shi, Ibaraki-ken, 311-0193, Japan*

The gyrokinetic PIC (particle-in-cell) code for MHD simulation, Gpic-MHD, was installed on SR16000 ("Plasma Simulator" in NIFS), which is a state-of-the-art scalar SMP (symmetric multiprocessing) cluster system consisting of 8,192 logical cores (128 nodes, each node includes 32 physical cores with SMP architecture, and one physical core is equivalent to two logical cores with multithreading technology). Gpic-MHD assumes a cylindrical coordinate system corresponding to the lowest order tokamak ordering. The hybrid parallel programming model of thread parallel (auto-parallelization) and process parallel (MPI) is used. The total simulation domain (cylinder) is decomposed in one (1d) or two (2d) directions. Replicas of field quantities are used to utilize logical cores larger than the number of decomposed domains (parallelization due to "particle decomposition"). Each process is responsible to one decomposed domain and includes the approximately same number of particles. Gpic-MHD with 1d domain decomposition in an axial direction, demonstrated a good scaling up to 8,192 logical cores. However this scaling will saturate for more than several tens of thousands of logical cores because the communication time between processes will increase as the number of replicas increases. To overcome this deterioration of the scaling, Gpic-MHD with 2d domain decomposition was made, in which the total domain is decomposed in axial and radial directions. The 2d domain decomposed version also showed a good scaling, but the computation time was a little bit longer than the 1d domain decomposed version for the relatively small number of meshes and cores studied in this work; the faster computation time was obtained for the 1d domain decomposed version. However, for the future simulation with much larger meshes and logical cores, it is expected that the 2d domain decomposed version with further optimization will exhibit better parallelization performance.

KEYWORDS: gyrokinetic theory, particle-in-cell code, magnetohydrodynamics, tokamak, symmetric multiprocessing, message passing interface, distributed shared memory, thread parallel, process parallel

I. Introduction

Full understanding of experimentally observed magnetohydrodynamic (MHD) phenomena in high temperature and large tokamaks, inevitably requires kinetic (or extended) MHD theory and simulation. The electromagnetic PIC (Particle-In-Cell) code based on the gyrokinetic theory can simulate these phenomena without the "closure" problem, which is generally encountered in the fluid-type model when deriving moment equations from kinetic equations. Such PIC type code follows a huge number of charged particles (electrons and ions) inside the whole region of a tokamak plasma, and hence requires huge computer resources. The use of a massive-parallel computer is inevitable.

More than a decade ago, based on the standard gyrokinetic model,[1,2] we developed a gyr3d code[3,4] and successfully simulated linear and nonlinear behaviors of the collisionless kinetic internal kink mode. The gyr3d code was programmed for a three dimensional (3d) rectangular coordinate system, hence demanded huge computer resources because the time step size was forced to be quite small to resolve the quite fine time scale of unwanted high frequency shear Alfven waves. For the parallelization model of gyr3d, we employed the combination of one-dimensional (1d) domain decomposition and "particle decomposition", because the number of domain decomposition was less than the number of CPUs. Each CPU has a replica (copy) of a decomposed domain; particles are decomposed to each replica. Some of parallelization algorithms are illustrated in Reference 4. The good parallelization scaling up to 256 CPUs was obtained, although the parameter range was limited to a very small system compared to the realistic experiments and also limited to the low-beta plasma.

Recently we developed a gyrokinetic PIC code for MHD simulation, Gpic-MHD, with cylindrical geometry. Field quantities of the electrostatic potential, ϕ, and longitudinal component of the vector potential, A_z, are represented with Fourier mode expansion both in axial and azimuthal directions; finite difference method is used in the radial direction. Nonuniform mesh can be used in the radial direction so that we can accumulate the mesh around the rational surface. We can increase the time step size by eliminating unwanted shear Alfven modes in Fourier space. Due to the symmetry of the geometry, we have a 2d code with a single helicity in

*Corresponding author, E-mail: naitou@yamaguchi-u.ac.jp addition to a 3d code with multi-helicities.[5)] The 2d version of Gpic-MHD can efficiently simulate the collisionless kinetic internal kink mode. The parameter range was expanded compared to the case of gyr3d, although it was difficult to expand the parameter range further. One of the reasons of this limitation is due to the "cancellation problem" in which the field solver of A_z has inaccuracy. The split-weight scheme can overcome this problem.[6,7)] We proposed another method to use the vortex equation and the generalized Ohm's law along the magnetic field.[8)] We showed that the new Gpic-MHD code could successfully simulate large scale and high-beta collisionless kinetic internal kink mode in tokamaks (although in a limit of cylindrical tokamak).

The standard version of 3d Gpic-MHD has been used to investigate the parallelization performance for various massive-parallel computers. In the previous paper,[5)] the parallelizaion performance up to 1,024 CPUs was examined for Altix3700Bx2 (now replaced to new one) of JAEA (Japan Atomic Energy Agency) computer center. Altix3700Bx2 is based on the rather conservative parallelization architecture; 16 nodes with 128 single-core CPUs. Pure message passing interface (MPI) is used for the parallelization model. The 1d domain decomposed version of Gpic-MHD with replicas showed no saturation up to 1,024 CPUs.

This article shows the parallelization performance of 3d Gpic-MHD for SR16000 ["Plasma Simulator" in NIFS (National Institute for Fusion Science)], which is a state-of-the-art scalar SMP (symmetric multiprocessing) cluster system. Hereafter we use the terminology of core instead of CPU, because the present-day processor usually consists of multi-cores. The total number of logical cores of SR16000 is 8,192. The total system of SR16000 is consisting of 128 nodes and each node includes 32 physical cores with SMP architecture. By using multithreading technology, one physical core is equivalent to two logical cores. In addition to 1d domain decomposition, the performance of the 2d domain decomposed version of Gpic-MHD is tested in order to apply to future large-scale simulation by massive-parallel computers with more than several tens of thousands of cores.

II. Simulation Model and Parallelization Method

Gpic-MHD is the electromagnetic gyrokinetic PIC code with the delta-f method. In the delta-f method, (marker) particles represent only a variation from the equilibrium; the numerical noise caused by particle discreteness, reduces drastically. Gpic-MHD follows a huge number of gyrokinetic particles in the simulation domain like conventional PIC codes. Parallelization algorithms are basically identical to the PIC code. The present version of Gpic-MHD assumes a lowest order tokamak ordering. Plasma fills a cylinder with a radius of a and a height of $2\pi R$. The cylindrical coordinate system is used with periodic boundary condition in the axial direction. The plasma fills the region inside the conducting wall at $r = a$. The basic field quantities are ϕ and A_z, which are obtained from the gyrokinetic Poisson equation and the Ampere' law, respectively. A magnetic field is represented by a sum of a constant toroidal magnetic field and a poloidal magnetic field calculated by A_z. The electrostatic electric field is calculated from ϕ. Although the induced electric field along the magnetic field is equal to $-\partial A_z / \partial t$, it does not appear explicitly in the formulation, because the generalized momentum, $p_z = v_z + (q_s / m_s) A_z$, is used instead of v_z in the formulation, where v_z is a particle velocity along the magnetic field. The Ampere's law for A_z becomes the Poisson-type equation with a nonlinear term including A_z and is solved by the iterative method. The perpendicular components of the electron and ion temperatures are assumed to be zero; the average over the finite gyro-orbits is neglected.

A hybrid parallel programming model of thread parallel and process parallel is employed. Each process consists of the fixed number of threads. Auto-parallelization is utilized for thread parallelization. Hereafter we illustrate methods of process parallelization. The programming model of process parallel is MPI. The parallelization is done by combination of domain decomposition and particle decomposition. The total simulation domain (cylindrical tokamak) is cut into separate N_{DD} local domains (subdomains). Each process treats one subdomain and particles residing inside. Domain decomposition is useful to reduce the computation time if the field solver is well localized to the decomposed domain. However N_{DD} is generally much less than the total number of cores in the present-day and future massive-parallel computers, we are forced to use N_{RP} replicas of field quantities (particle decomposition) in addition to domain decomposition. Total processes are divided into N_{RP} groups corresponding to the number of replicas. Each group includes N_{DD} subdomains making up one total simulation domain. Each process has N_{RP} replicas in the separate groups. The total number of processes is equal to $N_{DD} \times N_{RP}$. N_{DD}=1 and N_{RP}=1 correspond to pure particle decomposition and pure domain decomposition, respectively.

Gpic-MHD consists in two main parts as conventional PIC codes, the particle and the field parts. Field quantities are represented by 3d cylindrical meshes. The particle part pushes electrons and ions following the gyrokinetic equations of motion under the influence of electromagnetic fields using an explicit second-order Runge-Kutta method (modified Euler method). The particle part includes the "charge-assignment" (from particle positions to grid points) and the interpolation of field quantities (from grid points to particle positions). We used the linear-splines for the charge-assignment and the interpolation. The field part calculates the electromagnetic fields from the charge and current densities calculated by charge-assignment.

The consequence of parallelization for the field quantities is as follows. Each process includes only the arrays for the fields of one decomposed subdomain. A surface of the subdomain facing the adjacent subdomain is belonging to both subdomains (guard cells). Hence communication between subdomains in the same group is needed for the gurad cells. We need global sum (all-reduce communication) of the charge and current densities over replicas. For each group we calculate electromagnetic fields from the charge and current densities. This is a redundant calculation; the time for this part of the field calculation is independent of N_{RP}. Because the field quantities are distributed over subdomains inside a group, special care must be done for the field calculation for N_{DD}>1.

The consequence of parallelization for the particle part is as follows. After particle pushing, some particles reside in the adjacent subdomain (assuming the time step size is not so large) of the original subdomain in which particles were located before pushing. Hence we must move these particles to the adjacent subdomains in the same group; this needs communication between processes. There is no communication of particles between different groups. Because the number of particles is not constant in time in each process, we need some extra-margins for the size of the particle arrays.

For the 1d domain decomposed version of Gpic-MHD, the axial direction is uniformly decomposed into N_{DD-z} subdomains. Therefore the load balance between processes is almost uniform. Transfers of particles between different subdomains in the same group need the communication between related processes. There is no transfer of particles between different groups. Fourier (and inverse Fourier) transforms in the axial direction are done after transposing the field data from domain decomposition in z to domain decomposition in θ; the transpose calculation demands the communication between different subdomains in the same group.

It is preferable to increase the number of decomposed domains in order to utilize larger number of cores. Hence, we developed the 2d domain decomposed version of Gpic-MHD. The directions of 2d domain decomposition are selected as axial and radial directions. The load balance between different axial subdomains in the identical radial subdomain is uniform because the axial direction is uniformly decomposed. In contrast, the radial direction is nonuniformly decomposed into N_{DD-r} subdomains in order that the number of particles within different radial subdomains is approximately equal. The total number of subdomains is N_{DD}, which is equal to $N_{DD-r} \times N_{DD-z}$. A load imbalance remains because the number of meshes inside different radial subdomains is uneven. When the ratio of the field calculation to the particle calculation increases, more sophisticated load balance method may be needed, but it is left for the future study. For each group, after particle pushing, transfers of particles between different radial subdomains in the same axial subdomain are done after transfers of particles between different axial subdomains in the same radial subdomain. Fourier (and inverse Fourier) transforms are completely distributed over radial subdomains (except for guard cells in the radial direction); hence the time for the field calculation [a fast Fourier transform (FFT) dominates] reduces drastically as N_{DD-r} increases. The transpose of data for Fourier transforms is basically the same as 1d axial domain decomposition but for the limited area in the radial direction. The gyrokinetic Poisson equation and the Ampere's law are represented by the finite difference method in the radial direction and Fourier mode representations in θ and z directions; both are solved by using the Poisson solver. We use LU decomposition for the Poisson solver, which requires the field data over total radial subdomains in the same axial subdomain of the same group. In addition to the localized field data, each process gathers the global field data over total radial subdomains using allgather communication. Hence communication between different radial subdomains in the same axial subdomain of the same group is needed to gather field data from other distinct radial subdomains. It should be noted here that, compared with 1d domain decomposition, 2d domain decomposition requires much communication between decomposed subdomains (processes). Therefore it is possible that the computation time for the 1d domain decomposed version is faster than the 2d domain decomposed version (if the speed-up of FFT by radial decomposition is cancelled by the increased time for allgather communication), which is the case shown in the next section. However, 2d domain decomposition is necessary for the future simulation with larger spatial scale by larger number of cores. For the large-scale simulation, with the 1d domain decomposition, the computation time for the field calculation will greatly enhance, while the load of field solver will be distributed over different domains with 2d domain decomposition.

III. Parallelization Performance

The parallelization performance of the 1d and 2d domain decomposed versions of Gpic-MHD is investigated for SR16000. The simulation parameters chosen are in the following. Numbers of radial, azimuthal, and axial meshes are $129 \times 128 \times 128$. The uniform radial mesh is used although Gpic-MHD corresponds to a nonuniform radial mesh. We set the spatial distribution of markers uniform. Each logical core includes approximately millions of particles (sum of electrons and ions). Hence, the total program size increases as the number of logical cores increases; we are measuring the so-called "weak scaling" concerning particles. The wall clock time for 1,000 time steps are measured. The best results are always obtained for two-threads (SMP=2). We used N_{RP} replicas of field quantities, because the number of decomposed domains is less than the number of logical cores.

Here, we summarize the results of the 1d domain decomposed version of Gpic-MHD. In the axial direction, the total domain is divided to 64 subdomains including equal number of axial meshes (N_{DD-z}=64). The load balance between different processes is quite well for 1d domain decomposition. The wall clock times depending on the number of logical cores, are shown in **Fig. 1**. The case with 128 logical cores has no replicas because N_{DD-r}=1, N_{DD-z}=64, and SMP=2. The computation time is almost constant for less than 1,024 logical cores but gradually increases as the number of logical cores increases.

Here, to see the performance of Gpic-MHD on SR16000, we used unrealistically large number of particles for the relatively small meshes of $129 \times 128 \times 128$. This kind of calculation is only needed to study a sensitivity analysis concerning a number of particles (sometimes it is important

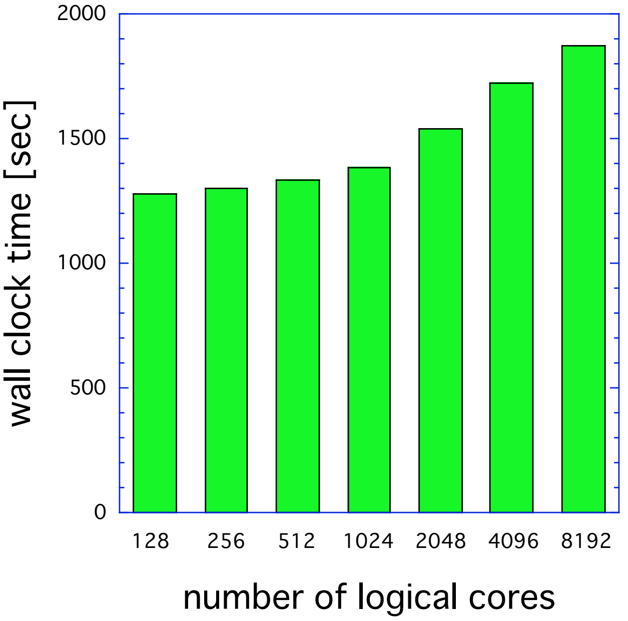

Fig. 1 FLOPS versus number of logical cores for the 1d domain decomposed version of Gpic-MHD. Wall clock time versus number of logical cores for the 1d domain decomposed version of Gpic-MHD. The jobs are executed for approximately one million particles/(logical core). $N_{DD\text{-}z}$=64. $129\times128\times128$ spatial meshes and 1,000 time steps. SMP=2.

to demonstrate the validity of the simulation results). However we are interesting in the large-scale simulation in the future, in which the number of particles per a logical core will not be larger than around millions (assuming performance of each logical core will not increase so much in a near future): therefore we fixed this particle number.

Combination of domain decomposition and particle decomposition (sometimes called "domain cloning") is a standard technique for parallelization and has been presented by different articles.[4,9,10] In the article by C. C. Kim and S. E. Parker, domain cloning was proposed as an alternative to (pure) 2d domain decomposition.[9] The article by R. Hatzky treated an electrostatic gyrokinetic code in cylindrical geometry.[10] Domain cloning usually shows good performance if the number of replicas is not so huge.

The FLOPS (floating point operations per second) versus the number of logical cores are shown in **Fig. 2**. FLOPS increases as the number of logical cores increases. No tendency of saturation was observed. The maximum FLOPS of 2.4 TFLOPS is obtained for the case using 8,192 logical cores.

The wall clock times presented here consist of the calculation and communication times. **Figure 3** shows various calculation times (executed by processors) versus the number of logical cores. In the figure, "PUSH", "SOURCE", and "FFT" mean times spent for particle pushing, charge-assignment, and FFT, respectively. The calculation time of the Poisson solver is dominated by FFT. These calculations are executed independently in the separate processes; hence no significant change is observed as the number of replicas increases.

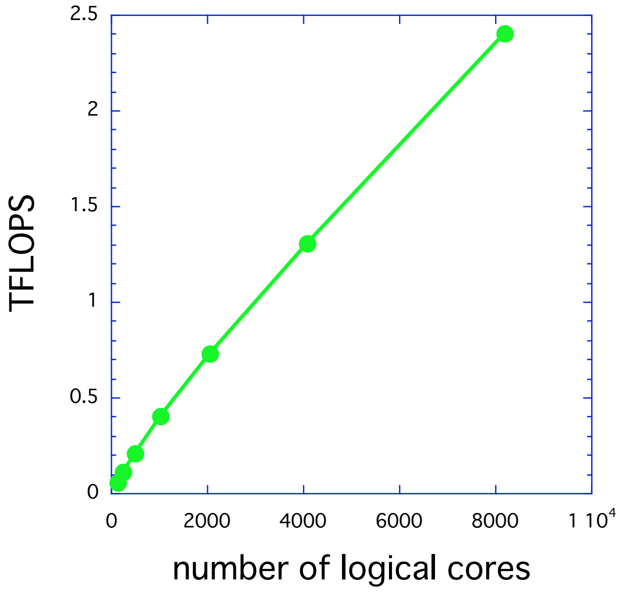

Fig. 2 FLOPS versus number of logical cores for the 1d domain decomposed version of Gpic-MHD

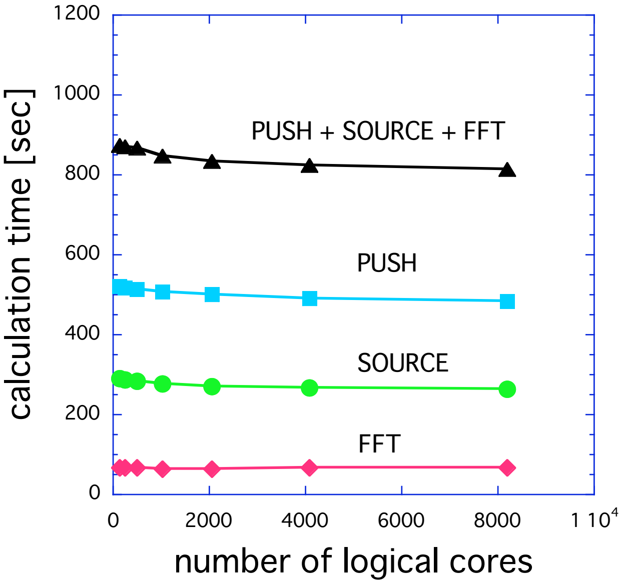

Fig. 3 Various calculation times versus number of logical cores for the 1d domain decomposed version of Gpic-MHD

Figure 4 shows various communication times versus the number of logical cores. The pre- and post-calculations for respective communication are included in the times. In the figure, "move", "transpose", and "sum" mean times spent for particle move between adjacent subdomains, transpose of the field data for FFT, and global sum, respectively. For "move" and "transpose", the amount of relevant data is proportional to N_{RP}. These communications are limited inside a group, but the communication bus is common; hence the communication times increases gradually. For "sum", the amount of relevant data is proportional to $N_{RP}\log_2 N_{RP}$.

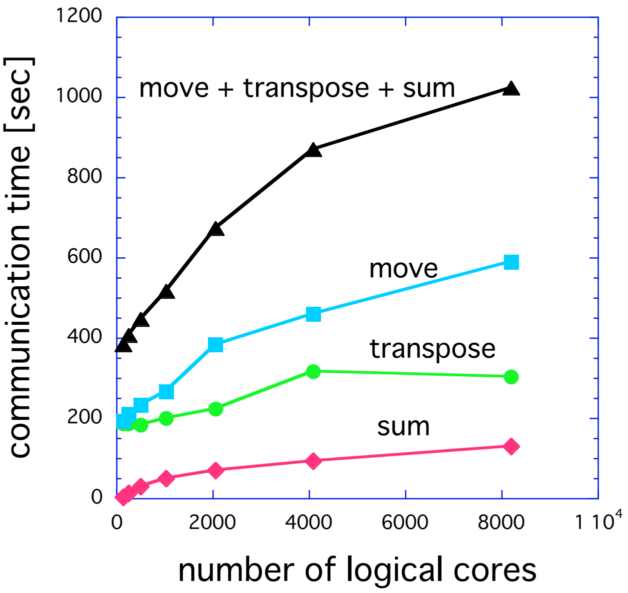

Fig. 4 Various communication times versus number of logical cores for the 1d domain decomposed version of Gpic-MHD

The increase of the communication times illustrates the origin of the increase of the total wall clock time.

The tokamak simulation needs relatively large number of radial meshes because the relevant mode profile is fine in this direction. Hence it is important to decompose a simulation area also in the radial direction in the large-scale simulation. Here, we summarize the parallelization results of the 2d domain decomposed version of Gpic-MHD. The number of axial domain decomposition is 64 and the number of radial domain decomposition is 16 ($N_{DD\text{-}z}$=64, $N_{DD\text{-}r}$=16). The case with 2,048 logical cores has no replicas because the number of decomposed domains is $N_{DD}=N_{DD-r}\times N_{DD-z}=$ 1,024 and SMP=2. The results of scaling on the number of logical cores are obtained by increasing N_{RP}.

The wall clock times depending on the number of logical cores, are shown in **Fig. 5**. The computation time is rather large compared with the 1d domain decomposed version. This is mainly because each radially localized domain has global data corresponding to whole radial domains. The communication time to gather global data to each radially decomposed domain has a dominant effect for this enhancement of computing time. There is a finite increase of wall clock time as the number of logical cores increases.

FLOPS versus the number of logical cores are shown in **Fig. 6**. The maximum value of 2.0 TFLOPS is obtained for 8,192 logical cores. FLOPS increases as the number of logical cores increases. No tendency of saturation was observed.

Here, we show one example showing the superiority of 2d domain decomposed version of Gpic-MHD. We used the larger mesh of 1,029 × 128 ×128. Only the radial mesh size is increased from the previous case of $129\times128\times128$. The case with 1d domain decomposition uses 64 axial domains and 32 replicas ($N_{DD\text{-}z}$=64, $N_{DD\text{-}r}$=1, N_{RP}=32). The case with 2d domain decomposition uses 64 axial subdomains and 32 radial subdomains with no replicas ($N_{DD\text{-}z}$=64, $N_{DD\text{-}r}$=32,

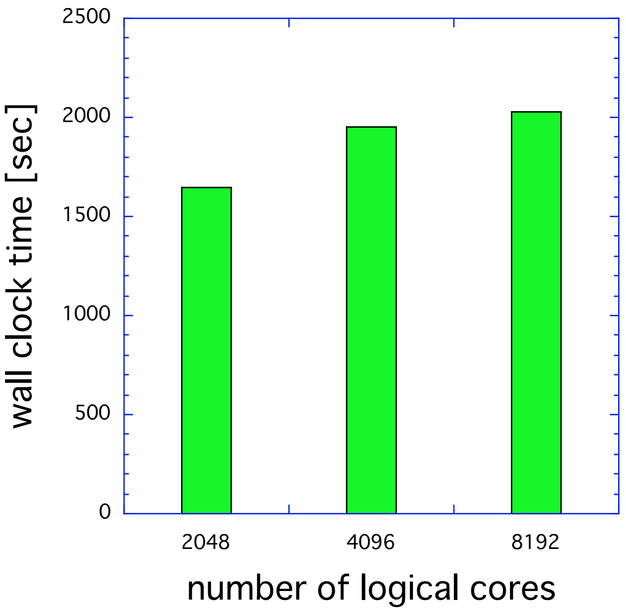

Fig. 5 Wall clock time versus number of logical cores for the 2d domain decomposed version of Gpic-MHD. The jobs are executed for approximately one million particles/(logical core). $N_{DD\text{-}z}$=64 and $N_{DD\text{-}r}$=16. $129\times128\times128$ spatial meshes and 1,000 time steps. SMP=2.

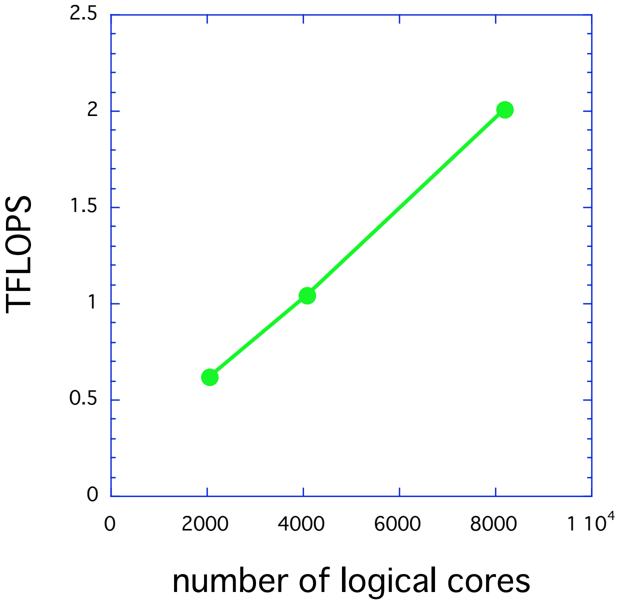

Fig. 6 FLOPS versus number of logical cores for the 2d domain decomposed version of Gpic-MHD

N_{RP}=1). Both cases uses 4,096 logical cores with about one million particles per logical cores and SMP=2. The program size is the same for the both cases. The computation time for 1,000 time steps is 3,242 seconds for the case of the 2d domain decomposition, while it is 6,257 seconds for the 1d domain decomposition. The time related to the Poisson solver including FFTs reduces as $N_{DD\text{-}r}$ increases. Fourier transforms are done independently for the radially decomposed field data; hence calculation time reduces. Communication time for the transpose of the field data also reduces because of the small field data size. The global sum calculation disappears for pure 2d domain decomposition. These time reductions dominate the time increase by all-gather communication. The resultant speed-up accounts for half of the reduced time. The origin of another half of the reduced time is the speed-up of the calculation time for particles (pushing, charge-assignment, etc.). It may be explained as the size of field data becomes small due to radial domain decomposition, the possibility of cash mishitting reduces. The performance of the 2d domain decomposed version of Gpic-MHD for this large system will be presented in detail in a different article.[11)]

IV. Conclusions and Discussion

The parallelization performance of Gpic-MHD is examined for SR16000, which is a state-of-the-art scalar SMP cluster system at NIFS. Total number of logical cores of SR16000 is 8,192. The computing times for the 1d and 2d domain decomposed versions of Gpic-MHD are measured as the number of logical cores increases. The mesh size used for the examinations is $129\times128\times128$. The scaling was measured by increasing the number of replicas as the number of logical cores increases. The highest FLOPS of 2.4 TFLOPS was obtained for the 1d domain decomposed version with 8,192 logical cores. This is because the load balance between different domains is almost equal for 1d domain decomposition. However, the 2d domain decomposed version is necessary for the future large-scale simulation, which will have more than several thousands of radial meshes. The computation time for the field solver will dominate the simulation time if 2d domain decomposition is not employed. It was found that the present version of Gpic-MHD with 2d domain decomposition has less performance compared with the 1d version for the relatively small mesh size presented in this article but still have scaling without saturation. One example, which shows the superiority of 2d domain decomposed version, is presented for the case with the larger number of radial meshes of 1,025 × 128× 128.

The present 2d domain decomposed version of Gpic-MHD is not optimized well because each radially decomposed domain has the data of global radial domains, because the Poisson solver uses global data distributed over radial directions. We are also testing the version with different algorithm, such as BiCGstab-P method,[12)] which will use mainly localized data, but the result will be reported in a different article.

Another possibility of aiming good performance for the large system by using the large number of logical cores is to use thread parallelization effectively. It is expected that the future massive parallel computer will have a node with large number of logical cores with SMP architecture. If we can effectively parallelize the program in each node by using shared memory with thread parallel model, there is no need to gather global data distributed over radial subdomains.

For the moment, Gpic-MHD neglects finite Larmor Radius (FLR) effects of ions, which may have a strong influence on the results. The inclusion of ion FLR effects will enhance the computation time for ions and some extra guard-cells are needed to include the field data on the gyro-orbits in the same subdomain. Also, some special cares must be done for particles near the conducting wall; the fractional part of the gyro-orbit can be outside the boundary even if the guiding center is inside. However, we believe that there will be no essential difficulty in the modification of Gpic-MHD.

Acknowledgment

This research is partially supported by Grant-in-Aid for Scientific Research (C) (20560766), collaboration programs between universities and JAEA (22K450) and universities and NIFS (NIFS09KTBL009).

References

1) W. W. Lee, "Gyrokinetic particle simulation model," *J. Comput. Phys.*, **72**[1], 243-269 (1987).
2) T. S. Hahm, W. W. Lee, A. Brizard, "Nonlinear gyrokinetic theory for finite-beta plasmas," Phys. Fluids, **31**[7], 1940-1948 (1988).
3) H. Naitou, K. Tsuda, W. W. Lee, R. D. Sydora, "Gyrokinetic simulation of internal kink modes," *Phys. Plasmas*, **2**[11], 4257-4268 (1995).
4) H. Naitou, T. Sonoda, S. Tokuda, V. K. Decyk, "Parallelization of gyrokinetic particle code and its application to internal kink mode simulation," *J. Plasma Fusion Res.*, **72**[3], (1996).
5) H. Naitou, H. Hashimoto, Y. Yamada, S. Tokuda, M. Yagi, "Development and parallelization of gyrokinetic PIC code for MHD simulation," *J. Plasma Fusion Res. SERIES*, 1158-1161 (2009).
6) I. Manuilskiy, W. W. Lee, "The split-weight particle simulation scheme for plasmas," *Phys. Plasmas*, **7**[5], 1381-1385 (2000).
7) W. W. Lee, J. L. V. Lewandowski, T. S. Hahm, Z. Lin, "Shear-Alfven waves in gyrokietic plasmas," *Phys. Plasmas*, **8**[10], 4435-4440 (2001).
8) H. Naitou, K. Kobayashi, H. Hashimoto, T. Andachi, W. W. Lee, S. Tokuda, M. Yagi, "Proposal of a brand-new gyrokinetic algorithm for global MHD simulation," *Bull. APS, 51st Annual Meeting of the Division of Plasma Physics*, Nov. 2-6, 2009, Atlanta, Georgia, 35 (2009).
9) C. C. Kim, S. E. Parker, "Massively parallel three-dimensional toroidal gyrokinetic flux-tube turbulence simulation," *J. Comput. Phys.*, **161**[2], 589-604 (2000).
10) R. Hatzky, "Domain cloning for a particle-in-cell (PIC) code on a cluster of symmetric-multiprocessor (SMP) computers," *Parallel Comput.*, **32**[4], 325-330 (2008).
11) H. Naitou, Y. Yamada, S. Tokuda, Y. Ishii, M. Yagi, "HPC parallel programming model for gyrokinetic MHD simulation," *PFR Special Issue on ITC20*, 2010 (accepted for publication).
12) H. A. van del Vorst, "Bi-CGstab: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, **13**[2], 631-644 (1992).