High Performance Computing of Meshless Time Domain Method on Multi-GPU Cluster

Soichiro Ikuno1, Susumu Nakata2, Yuta Hirokawa3, and Taku Itoh1
1School of Computer Science, Tokyo University of Technology, Hachioji, Tokyo 192-0982, Japan
2Institute of Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan
3Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
E-mail: ikuno@nal.ikulab.org

Abstract. High performance computing of Meshless Time Domain Method (MTDM) on multi-GPU using the supercomputer HA-PACS (Highly Accelerated Parallel Advanced system for Computational Sciences) at University of Tsukuba is investigated. Generally, the finite difference time domain (FDTD) method is adopted for the numerical simulation of the electromagnetic wave propagation phenomena. However, the numerical domain must be divided into rectangle meshes, and it is difficult to adopt the problem in a complexed domain to the method. On the other hand, MTDM can be easily adept to the problem because MTDM does not requires meshes. In the present study, we implement MTDM on multi-GPU cluster to speedup the method, and numerically investigate the performance of the method on multi-GPU cluster. To reduce the computation time, the communication time between the decomposed domain is hided below the perfect matched layer (PML) calculation procedure. The results of computation show that speedup of MTDM on 128 GPUs is 173 times faster than that of single CPU calculation.

1. Introduction
The finite difference time domain (FDTD) method is often applied for the electromagnetic wave propagation simulation. FDTD has an advantage in terms of parallelization and treatment of problems because the method is based on complete explicit finite difference method (FDM). However, the numerical domain must be divided into rectangle meshes. Therefore, it is difficult to treat the problem in a complex shaped numerical domain.

On the other hand, the meshless approach such as the element-free Galerkin method, the meshless local Petrov-Galerkin method and the radial point interpolation method (RPIM) [1] do not require finite elements or meshes of a geometrical structure. In particular, meshless approaches based on RPIM are applied to time dependent problems, and the method is called meshless time domain method (MTDM) [2].

In recent years, various researches of the general purpose computing on GPU (GPGPU) have been proposed [3, 4]. In previous study, we investigated the speedup of MTDM using a single GPU as well, and the parallelized method on a single GPU achieved about eight times faster than that of CPU with OpenMP [4].

The purpose of the present study is to implement MTDM method for analyzing electromagnetic wave propagation phenomena in a long waveguide on multi-GPU, and to
Table 1. Evaluation environment.

| Parameter                  | Specification                                      |
|----------------------------|----------------------------------------------------|
| CPU                        | Intel E5 2.6 GHz x2 (Sandy-Bridge EP)              |
| CPU memory                 | 128 GB                                             |
| GPU                        | NVIDIA M2090 ×4                                     |
| GPU memory                 | 24 GB                                              |
| Infiniband                 | Mellanox Connect-X3 dual port QDR                  |
| compiler version and options| Intel 14.0.0 (-O3 -xhost -openmp)                  |
| CUDA version and options   | 5.5 (-O3 -use_fast_math -arch=sm_20)               |
| MVAPICH2                   | 2.0 (GPU Optimized, not GDR)                       |
| Intel MPI                  | 4.1.3.049                                          |

investigate the details of the performance of the method. Note that the supercomputer HA-PACS (Highly Accelerated Parallel Advanced system for Computational Sciences) at University of Tsukuba is adopted for evaluations.

2. Meshless Time-Domain Method, MTDM

The basic concept of the MTDM is same as standard FDTD method, and the governing equations are completely same as FDTD as well. The equations are discretized in respect to time using Leap-Flog scheme. On the other hand, physical quantities on scattered node \( x_i \) \((i = 1,2,\ldots,n) \) in analytic domain \( u(x_i) \) (such as a electric field \( E(x_i) \) and a magnetic field \( H(x_i) \)) is discretized in respect to space using the shape function \( \phi_i(x_j) \) which is derived from RPIM as follows.

\[
    u(x_i) = \sum_{j=1}^{n} \phi_i(x_j)u_i = u_i. \tag{1}
\]

Note that the shape functions obtained by RPIM satisfy the Kronecker’s delta function property, i.e.,

\[
    \phi_i(x_j) = \begin{cases} 
    1; & i = j, \\
    0; & i \neq j.
\end{cases} \tag{2}
\]

In the present study, a two dimensional long straight waveguide with width 5 cm is adopted for evaluation. The perfectly matched layer (PML) and the perfect electric conductor (PEC) are used for absorbing boundary condition on edges and sides of the waveguide, respectively. In addition, the number of nodes for width direction is fixed as 51.

3. Parallelization Strategies

As we mentioned above, the supercomputer HA-PACS at University of Tsukuba is adopted for evaluations. HA-PACS is constructed by 332 nodes, and the peak performance of the system is about 1.166 PFLOPS. Dual CPU (Intel E5 2.6 GHz, Sandy-Bridge EP) and four GPUs (NVIDIA M2090) are implemented on a unit node. Additionally, nodes are connected by the Infiniband. The detail of the evaluation environment is shown in Table 1.

The major modification to the present study from the previous study is shown in below [4].

- Parallelized by using multi-GPU, and 32 nodes is used for evaluation (i.e. 128 GPUs).
- Optimize the memory access in the code by using CUDA Coalesced Access.
- Read-Only Cache (ROC) for Kepler generation is adopted for inner product calculation.
Figure 1. The schematic view of the parallelization strategy using MPI on unit node of HA-PACS.

Figure 2. The schematic view of the analytic model of the straight waveguide.

- Texture memory on GPU is adopted for PML calculation.

In Figure 1, we show the schematic view of the parallelization strategy using Message Passing Interface (MPI) on a unit node of HA-PACS. As we mentioned above, two CPUs are implemented on a unit node, and a CPU has four cores. Thus, we divided into four Ranks in unit node as shown in Figure 1. Each Rank is communicated by using MPI even if GPUs are implemented on same node, and CUDA and OpenMP are used for parallelization.

The analytic domain is decomposed into $n$ cells that including overlapped region for task distribution as shown in Figure 2. Each Rank is associated a cell, and data are communicated between neighboring cells using Peer to Peer communication (GPU Direct) to reduce the communication time [5]. Note that PML calculation is achieved at same time of data communication for overlapped region to hide the transmission time of the data.

4. Results and Discussions

Let us investigate the performance of the method using multi-GPU. To this end, the weak scaling and the strong scaling are calculated. In the weak scaling calculation, the length for each Rank is fixed as 10 m, and the node number of the length direction for MTDM is fixed as 10001 on a unit Rank. On the other hand, in the strong scaling calculation, the length of the waveguide is fixed as 100 m, and the node number of the length direction for MTDM is fixed as 100001.

In Figure 3, we show the weak scaling (the performance efficiency) of MTDM using multi-GPU. We can see from this figure that the performance efficiency of MTDM using multi-GPU is about 97%. This result indicates that the communication time between node of HA-PACS becomes suppressed under 3%.

Finally, we calculate the strong scaling of MTDM using multi-GPU, and the values of speedup are shown in Figure 4 and the value of execution time is shown in Table 2. Note that the origin of the speedup (i.e. speedup = 1) is performed on a unit CPU, and the code is parallelized by using OpenMP. We can see from this figure that the value of the speedup increases linearly from a unit node to 16 node. However, the rate of increase slows down at 32 node. This is because associated node number of MTDM for unit Rank becomes about 40000 points, and the calculation cost becomes relatively smaller than the communication time.
**Figure 3.** The performance efficiency and the execution time of MTDM using multi-GPU.

**Figure 4.** The speedup of MTDM using multi-GPU.

**Table 2.** Execution time of MTDM using multi-GPU.

| Number of node | Execution time [sec] |
|----------------|----------------------|
| CPU (origin of speedup) | 7733.5391 |
| 1 | 780.70538 |
| 2 | 402.84217 |
| 4 | 210.99577 |
| 8 | 114.43655 |
| 16 | 67.028203 |
| 32 | 44.689656 |

**Acknowledgment**
This work was supported in part by Japan Society for the Promotion of Science under a Grant-in-Aid for Scientific Research (C) No. 26390135. And the numerical calculations for the present work was carried out under the “Interdisciplinary Computational Science Program” in Center for Computational Sciences, University of Tsukuba.

**References**
[1] J. G. Wang, G. R. Liu, *Int. J. Numer. Meth. Engng*, Vol. 54, No. 11, pp. 1623-1648, 2002.
[2] T. Kaufmann, C. Engstrom, C. Fumeaux, R. Vahldieck, *IEEE Trans. Microwave. Theo. Tech.*, Vol. 58, No. 12, pp. 3399-3408, 2010.
[3] N. Godel, S. Schomann, T. Warburton, and M. Clemens, “GPU Accelerated Adams-Bashforth Multirate Discontinuous Galerkin FEM Simulation of High-Frequency Electromagnetic Fields”, *IEEE Trans. Magn.*, 46, No. 8, (2010) 2735-2738.
[4] S. Ikuno, Y. Fujita, Y. Hirokawa, T. Itoh, S. Nakata, and A. Kamitani “Large-Scale Simulation of Electromagnetic Wave Propagation Using Meshless Time Domain Method With Parallel Processing”, *IEEE Trans. Magn.*, 49, No. 5, (2013) 1613-1616.
[5] S. Ikuno, Y. Kawaguchi, N. Fujita, T. Itoh, S. Nakata, and K. Watanabe “Iterative Solver for Linear System Obtained by Edge Element: Variable Preconditioned Method With Mixed Precision on GPU” *IEEE Trans. Magn.*, 48, No. 2, (2012) 467-470.